

EE4717/IM4717 Web Application Design

Project Report (Design)

Design project group number: F37-DG13

Team members: Wei Haoran (EE4717), Lin Sining (IM4717)

Project Title: *SoundX* - Online Shopping Portal for Audio Device

Summary: This project aims to design and develop a web portal for online purchase of consumer IT product, specifically focusing on audio devices. The scope of the project covers the configuration of application and functional requirements, website hierarchy and interface design framework, and finally the realization of web portal design under the above guideline. The project focuses on stimulating online purchase life cycle for the consumer, from on-boarding, exploring to making purchase and receiving post-purchase *service*. The website, *SoundX*, provides a wide variety of earphone products, serving as a platform for music enthusiasts and audio production professionals to find the earphone that catered to their needs. Once the users sign up and login to the website, they are able to browse through and make the purchase for products from a range of brands, of different types and functionalities. After making the purchase, the website provide follow-up service to help users keep track on their order and delivery status.

Contents

1. Application Requirements and Specifications	4
2. Functional Requirements and Specifications	5
3. Design of the web application.....	7
3.1 Sitemap	7
3.2 Storyboard	7
3.3 Wireframe.....	10
3.4 Database Design.....	16
3.4.1 Relationship Chart	16
3.4.2 Table: <i>products</i>	16
3.4.2 Table: <i>users</i>	17
3.4.3 Table: <i>orders</i>	17
3.4.4 Table: <i>order_items</i>	18
3.4.5 Table: <i>brand</i>	19
3.4.6 Table: <i>status</i>	19
4. Implementation	20
4.1 Implementation of web templates	20
4.1.1 Components.....	20
4.1.2 Layout	22
4.1.3 Font.....	24
4.2 Implementation of the pages.....	24
4.2.1 Signup and login	24
4.2.2 Home Page	26
4.2.3 Product listing	29
4.2.4 Product Details.....	31
4.2.5 Checkout.....	33
4.2.6 Order Details	34
4.2.6 Cart	36
4.2.7 Account and My Order	37
4.3 Implementation of the functions:.....	38
4.3.1 Sign Up and Login:	38
4.3.2 Filters	40
4.3.3 Shopping Cart.....	42

4.3.4 Place an order	44
4.3.5 Orders and Order Status	46
4.3.6 Manage account	47
5. Testing of Web Application	49
5.1 Test Plan.....	49
5.2 Testing Components and Pages	49
5.3 Testing Use Cases	59
6. Conclusion.....	61
Appendices	62
Appendix A: Contributions	62
Appendix B: Source Code	63

1. Application Requirements and Specifications

This project application should allow user to browse different products, make online purchases, sign up and login to account, manage account information and historical orders.

Detailed application requirements and specifications are shown in the table below.

R1	Product browsing
<i>R1.1</i>	Product browsing by product types or brands
<i>R1.2</i>	View product details
R2	Product purchase via online payment
R3	Sign up and log in
R4	Manage account and orders
<i>R4.1</i>	View and modify registered account information
<i>R4.2</i>	View status and details of historical orders

2. Functional Requirements and Specifications

The table below shows necessary functions to achieve the application requirements and specifications listed in the previous section.

R1 Product browsing	
R1.1 Product browsing by product types or brands	
F1	Users are able to browse all products by selecting a product type or a product brand
<i>F1.1</i>	System must display product listing, including product image, name, price, stock, and promotion information, with respect to the category selected
<i>F1.2</i>	System must indicate product availability by prioritizing all available products shown at the front, followed by non-available items
F2	Users are able to apply product property filters to request for respective product listing
<i>F2.1</i>	System must provide filters for price range, product types and brands
<i>F2.2</i>	Users are allowed to choose one or more filters, apply the selection and reset them
<i>F2.3</i>	System should list the only the items match to the filters selected and applied
R1.2 View product details	
F1	Users are able to view product image, name, price, specification and promotion details
R2 Product purchase and payment	
F1	Users are able to add items to the shopping cart
<i>F1.1</i>	System must acknowledge users whether the items are successfully added
<i>F1.2</i>	Users are able to choose the quantities of items to be added
<i>F1.3</i>	System must only allow adding items if the stock is sufficient for the quantity required; otherwise, prompt failure message to the users
F2	Users are able to manage the items in the shopping cart
<i>F2.1</i>	Users are able to check all products added in the cart
<i>F2.2</i>	Users are able to adjust the quantity or remove items from the shopping cart
<i>F2.3</i>	System must prompt warning to users if an item is about to be removed, or quantity is about to go below 1
<i>F2.4</i>	Users are able to check the product name, price, feature selected and quantity of an item
F3	Users are able to make purchase and payment of the items

<i>F3.1</i>	Users are able to make direct purchase on the product page, by clicking on the 'Buy' button
<i>F3.2</i>	Users are able to select multiple items in the shopping cart for purchase
<i>F3.3</i>	System should display order summary, including purchased items and their quantity, amount of payment, after user request for purchase
<i>F3.4</i>	Users should confirm on the order summary, enter delivery address, name and contact number as checkout
<i>F3.5</i>	System should create a unique order ID once user checkout, storing user's order information to the order ID
<i>F3.6</i>	Users are able to select payment method and make online payment for their purchase
<i>F3.7</i>	System must only allow purchase for products that are available and not exceeds current stock
<i>F3.8</i>	System should keep only the orders that are successfully paid, delete order ID and respective information of any unsuccessful transaction
<i>F3.9</i>	System must send confirmation email, including order ID for tracking status, to user's registered email address upon successful purchase
R3 Sign up and login	
F1	Users are able to sign up an account with email address and password
<i>F1.1</i>	System needs to validate email address and password of user input
<i>F1.2</i>	System adds record of new user's information into account database with email as unique ID
F2	Users are able to login with registered email address and password
<i>F2.1</i>	System must validate user's input by querying account database
<i>F2.2</i>	System should prompt user to sign up for if user's email input does not match to existing database record
F3	Users are able to sign out from the current account
R4 Manage account and order	
R4.1 View and modify registered account information	
<i>F1</i>	Users are able to view and edit the account contact number, delivery address; email is not editable once sign up
<i>F2</i>	Database record is updated upon user's modification
R4.2 View status and details of historical orders	
<i>F1</i>	Users are able to check all the historical orders placed
<i>F2</i>	Users are able to check purchase items, date, amount of payment and order status of a specific order

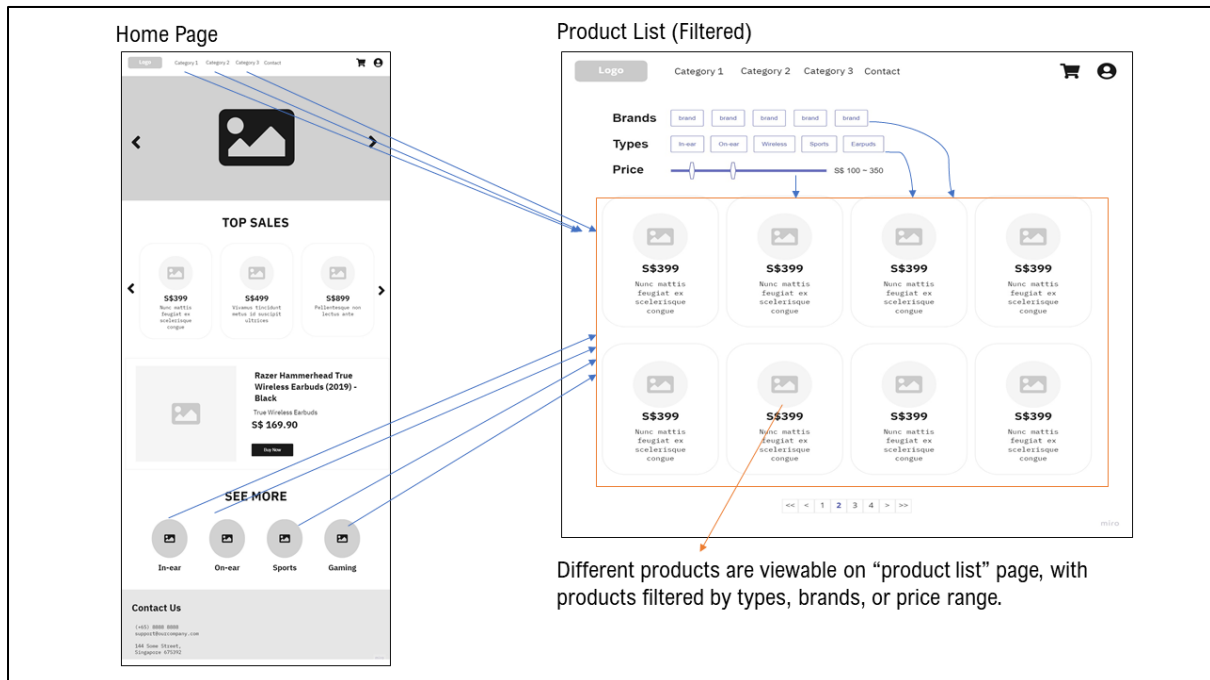
3. Design of the web application

3.1 Sitemap

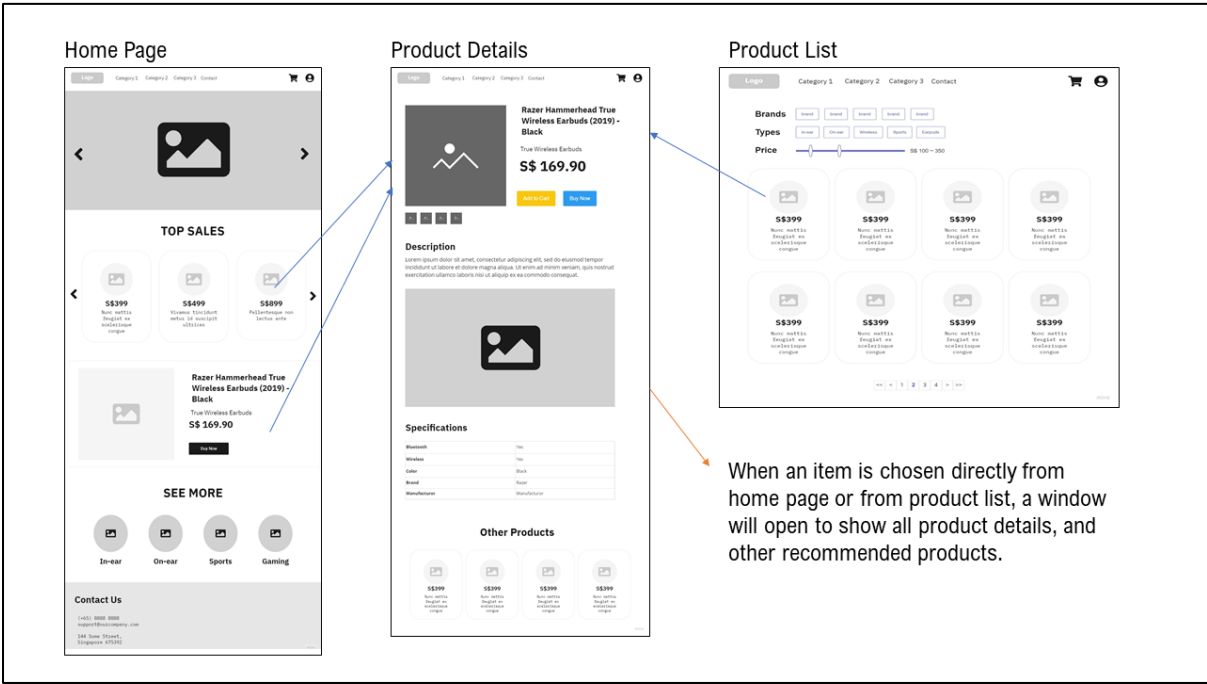


Picture: Sitemap

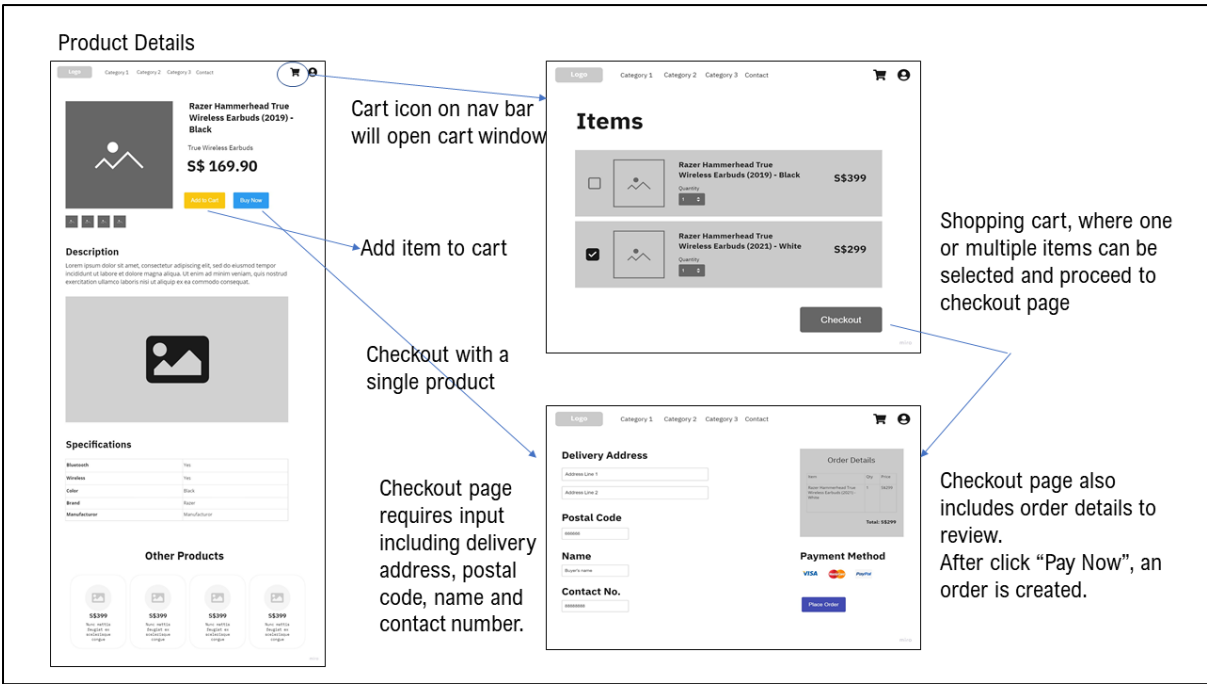
3.2 Storyboard



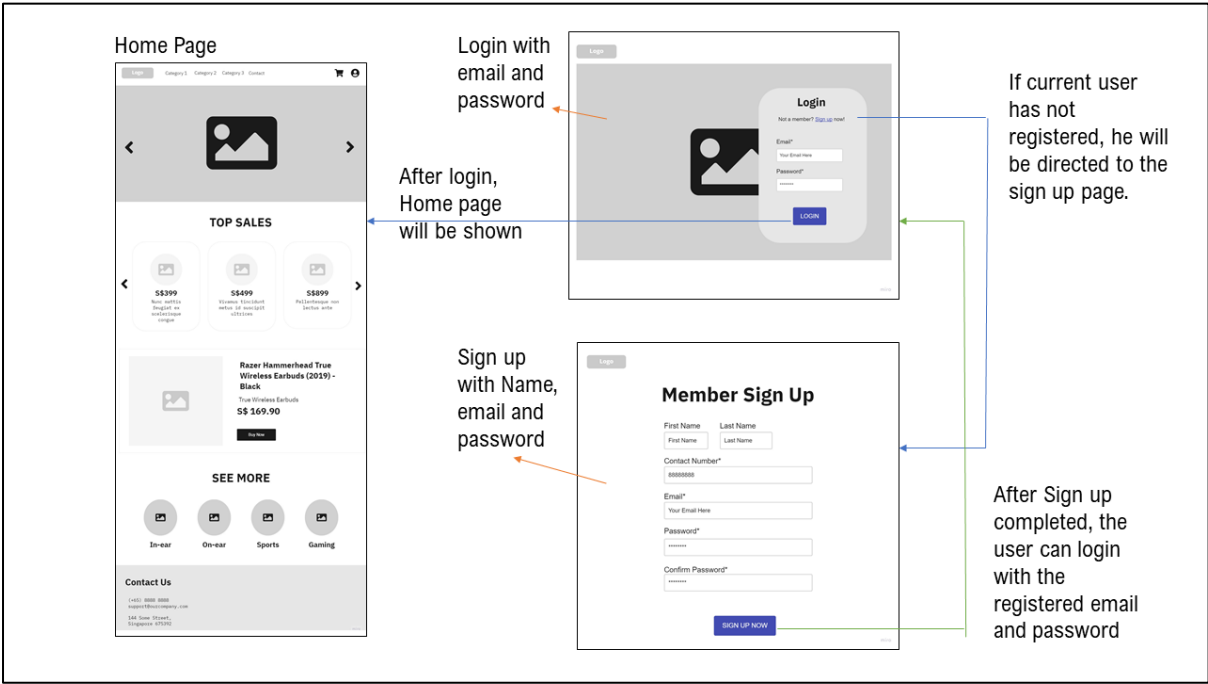
Scenario 1.1: Browsing products by types or brands



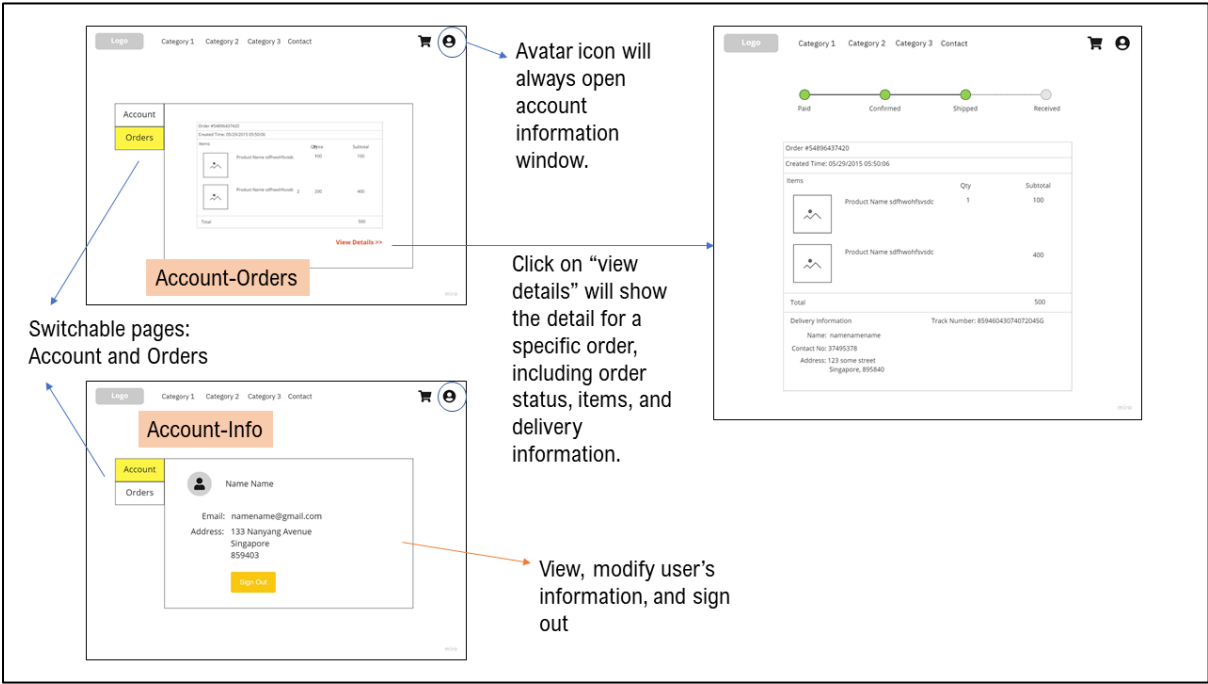
Scenario 1.2: View product details



Scenario 2: Online product purchases

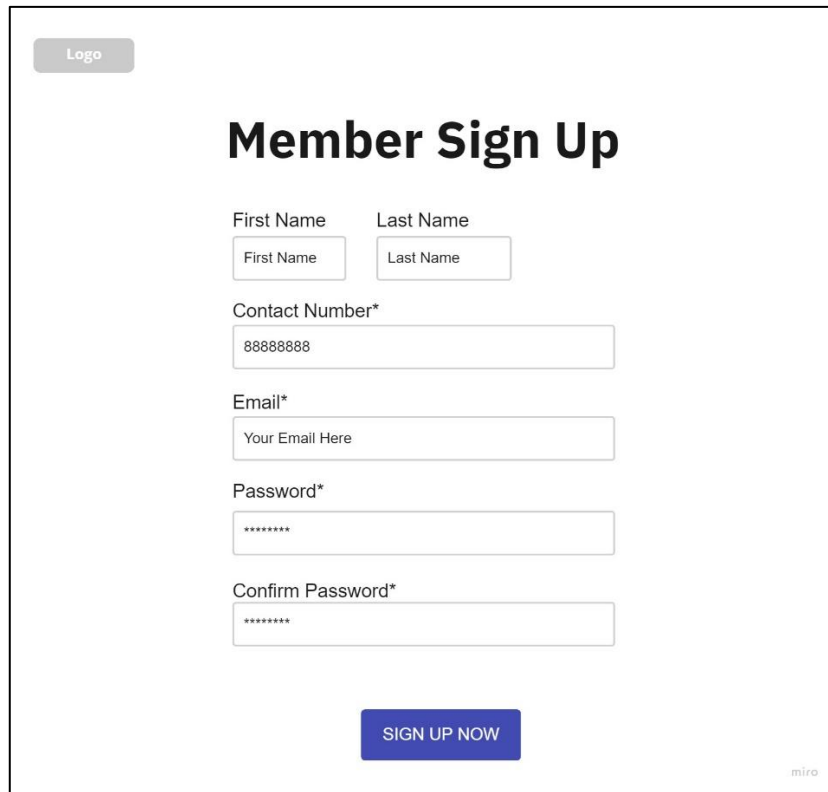


Scenario 3: Sign up and log in



Scenario 4: Manage account and orders

3.3 Wireframe



Logo

Member Sign Up

First Name Last Name

Contact Number*

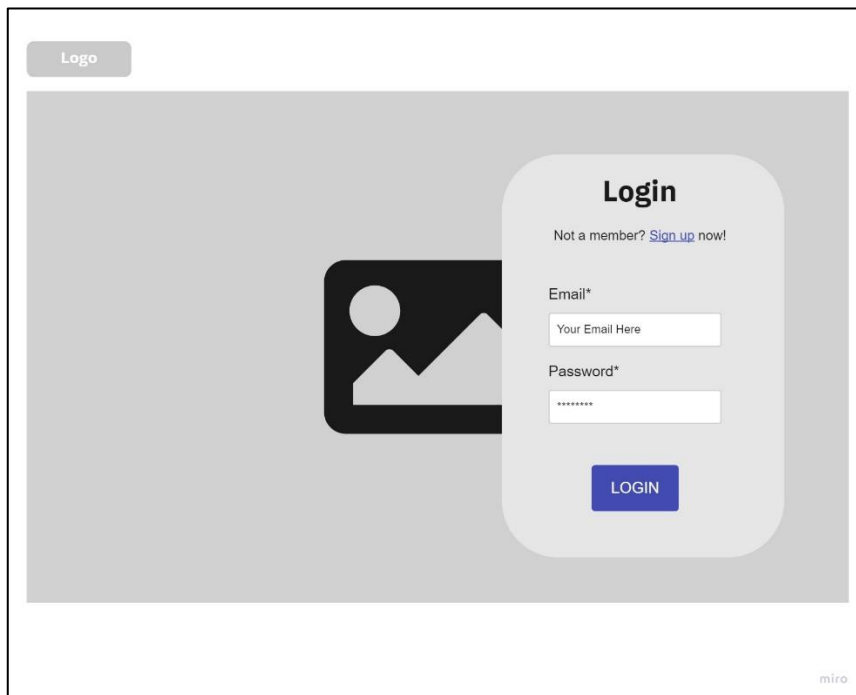
Email*

Password*

Confirm Password*

miro

Page 1: Sign up page



Logo

Login

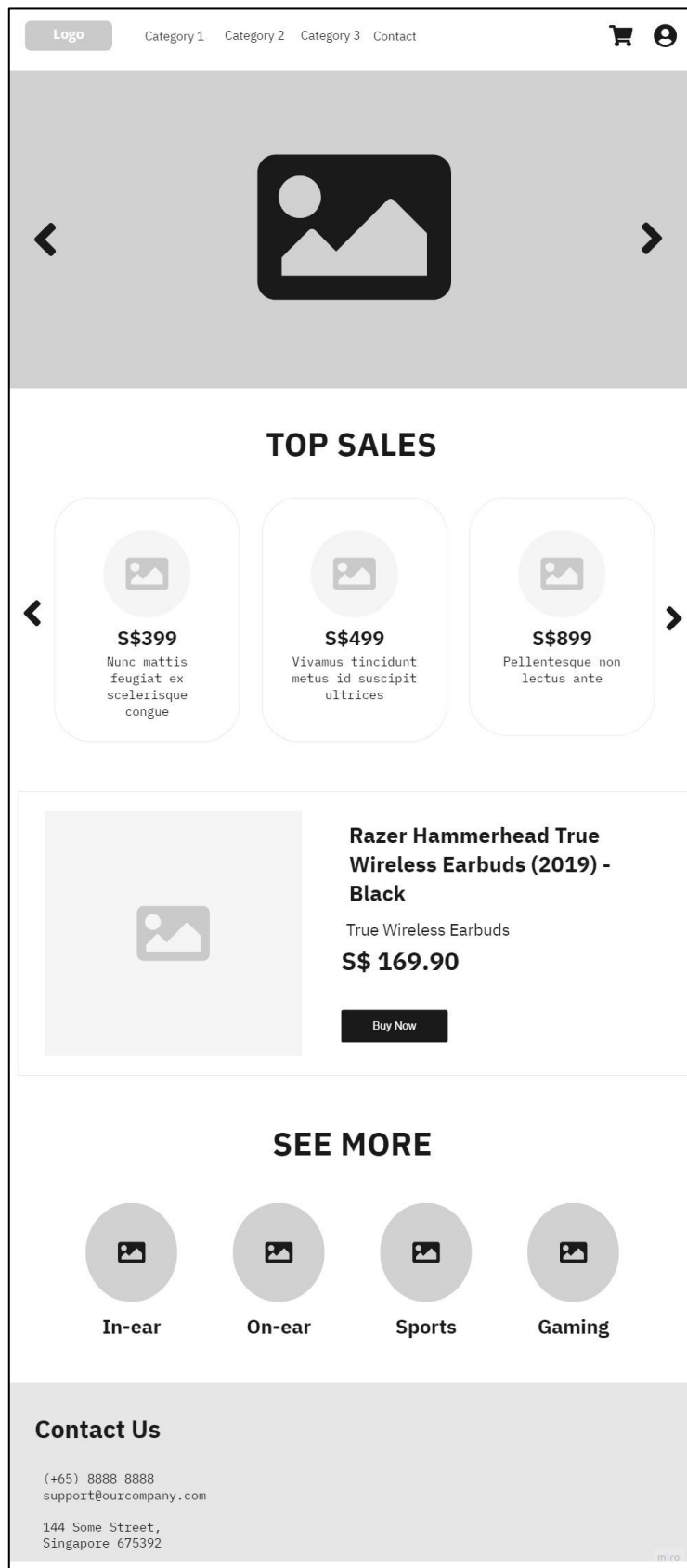
Not a member? [Sign up now!](#)

Email*

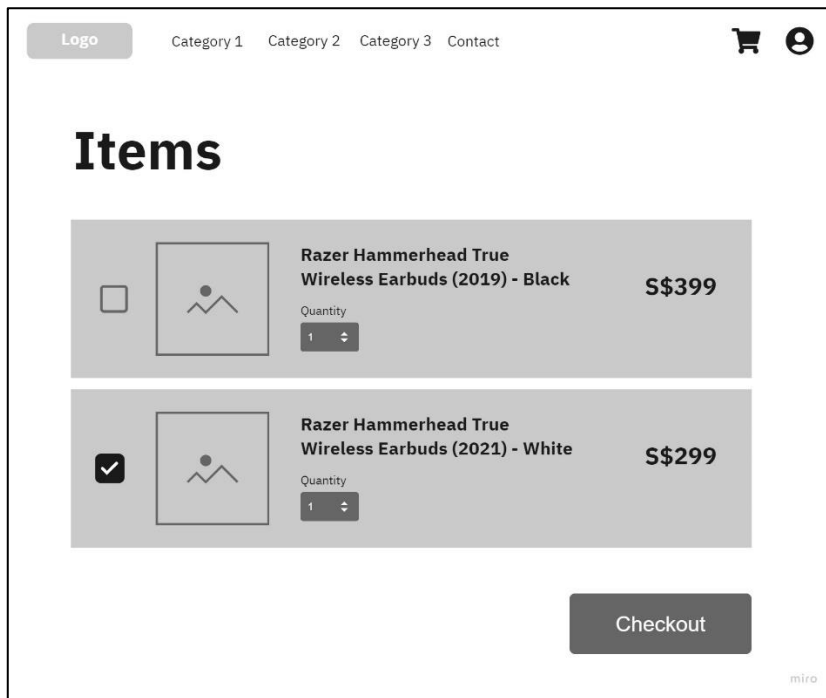
Password*

miro

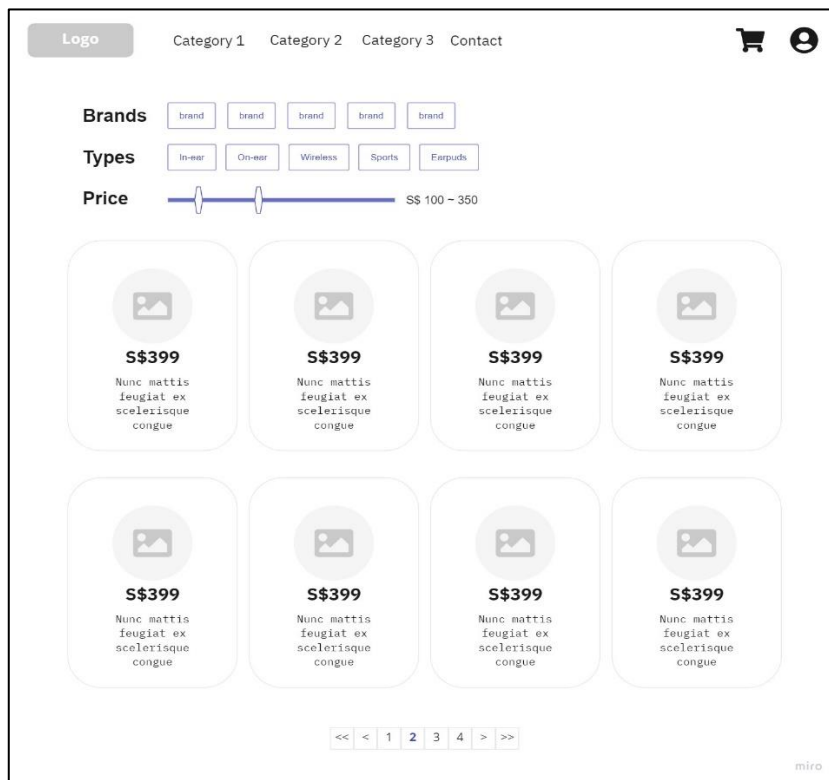
Page 2: Login page



Page 3: Home Page




Page 4: Shopping cart



Page 5: Product List

Logo
Category 1 Category 2 Category 3 Contact
🛒 👤







Razer Hammerhead True Wireless Earbuds (2019) - Black

True Wireless Earbuds


S\$ 169.90

Add to Cart
Buy Now

Description


Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.



Specifications


Bluetooth	Yes
Wireless	Yes
Color	Black
Brand	Razer
Manufacturer	Manufacturer

Other Products




S\$399

Nunc mattis feugiat ex scelerisque congue




S\$399

Nunc mattis feugiat ex scelerisque congue



S\$399

Nunc mattis feugiat ex scelerisque congue



S\$399

Nunc mattis feugiat ex scelerisque congue

miro

Logo
Category 1 Category 2 Category 3 Contact
🛒 👤

Delivery Address

Postal Code

Name

Contact No.

Order Details

Item	Qty	Price
Razer Hammerhead True Wireless Earbuds (2021) - White	1	\$5299

Total: \$5299

Payment Method

Page 7: Checkout

Logo
Category 1 Category 2 Category 3 Contact
🛒 👤

Order #54896437420

Created Time: 05/29/2015 05:50:06

Items	Qty	Subtotal
Product Name sdfhwohfsvsdc	1	100
Product Name sdfhwohfsvsdc		400
Total		500

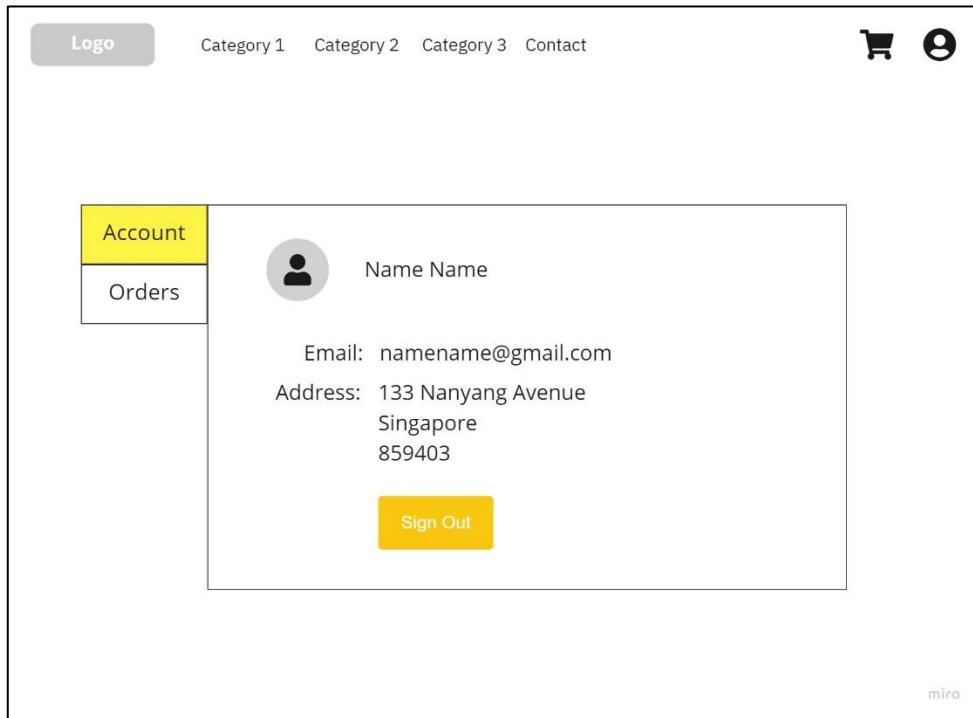
Delivery Information Track Number: 85946043074072045G

Name: namenamename

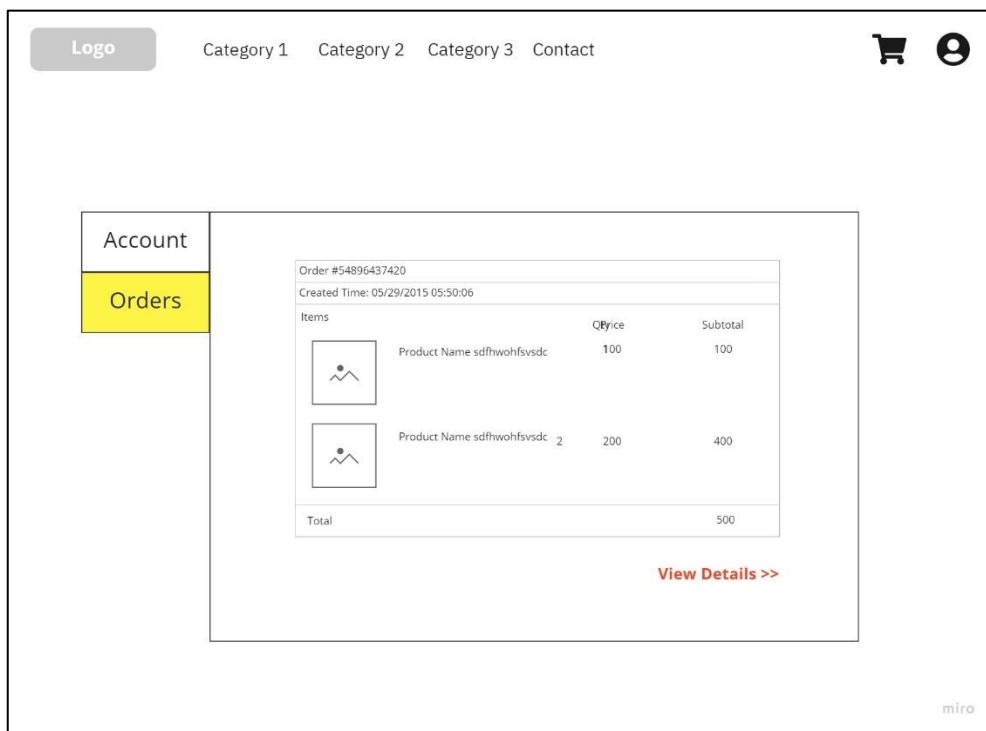
Contact No: 37495378

Address: 123 some street
Singapore, 895840

Page 8: Order details



Page 9: Account Info



Page 10: Orders

3.4 Database Design

3.4.1 Relationship Chart

Database supporting this web application project contains 6 tables: *products*, *users*, *brand*, *status*, *orders* and *order_items*. The relationship between tables is shown in the diagram below.



3.4.2 Table: *products*

Products table contains 26 items (rows) and 15 attributes (columns), including information about all the products. Each records contains information for the respective product.

Column Name(s)	Data Type	Purpose and usage
product_id	Int	Primary key, unique identifier for each product.
product_name	varchar	Indicating the name of product
short_description	varchar	One sentence only to describe the key feature of the product. Displayed on the product details page.
no_pictures	int	Total number of pictures of the product available. Users can toggle around the pictures on the product details page.
brand_id	int	Foreign key, refers to the primary key <i>brand_id</i> in brand table. Used as filters in product list filter section and the navigation bar “brands” tab.

price	double	Price of the product. Displayed in product list and product details pages. Used as range filter in product list filter section.
stock	int	How many stock left for the product.
wireless wired earbuds gaming sport bluetooth	boolean	Boolean values indicates features and types that can be used to filter the products. Used as filters in product list filter section and the navigation bar “types” tab.
description	varchar	Long description of the product, displayed in the “Description” section of product details page.
specification	varchar	Key-value pairs show tech specifications of product. Stored in database as JSON string. Displayed in the “Specification” section of product details page.

3.4.2 Table: *users*

Users table contains information about all users, collected from users when they sign up a new account. Each record contains information one user. When user reaches login page and enters his email and password, basic authentication will be conducted referring to this table.

Column Name(s)	Data Type	Purpose and usage
user_id	Int autoincrement	Primary key, unique identifier for each user. Cannot be changed once created
email	varchar	Email address of user. Used to identify each user. Display on the “account” page.
password	varchar	Password that user set when sign up. If the password does not match, the user cannot login.
first_name	varchar	First name of the user. Display on the “account” page.
last_name	varchar	Last name of the user. Display on the “account” page.
contact	varchar	Contact number of the user. Display on the “account” page.

3.4.3 Table: *orders*

Orders table include all the information about an order that a user placed on the website. Each record contains information about one order.

Column Name(s)	Data Type	Purpose and usage
Order_id	Int (8)	Primary key, unique identifier for each order. Generated from random number when an order is created. Cannot be changed once created. Displayed in “orders” and “order details” page, and can be found in the confirmation email sent to users.
user_id	int	Foreign key, refers to the primary key <i>user_id</i> in users table. Indicates the user who placed the order. Used to get all orders placed by a specific user on “orders” page.
status	Int	Status of the order, refers to the primary key <i>status_id</i> in status table. Displayed on “orders” and “order details” page.
created_time	datetime	Datetime that an order is created.
delivery_address_line_1 delivery_address_line_2 zip_code receiver_name receiver_contact payment_method	varchar	Information that user entered on “checkout” page when an order is created.
track_number	varchar	Tracking number of an order, only available after the status of the order is changed to “shipped”

3.4.4 Table: *order_items*

This table functions as a bridge between table products and table orders. This table contains the items in an order and the quantities of the item ordered. An order may include multiple items and a product can be ordered in multiple orders, therefore, the fields *order_id* and *product_id* is in multiple-to-multiple relationship.

The content of this table is displayed on “orders” and “order details” pages. All the information should not be changed once the order is placed.

Column Name(s)	Data Type	Purpose and usage
order_id	Int (8)	Foreign key, refers to primary key <i>order_id</i> in orders table.
product_id	int	Foreign key refers to primary key <i>product_id</i> in products table.
qty	Int	Quantity of a specific product in an order.

3.4.5 Table: *brand*

This table stores information about the brands which products are available on the SoundX Website. There are totally 5 brands available in this project.

Column Name(s)	Data Type	Purpose and usage
brand_id	Int	Primary Key, unique identifier for a brand. Referred in products table brand_id column
brand_name	varchar	Name of the brand

3.4.6 Table: *status*

After orders are created by users, the status of the order falls in “paid” status. As the order is processed, the status of the order will be updated to “confirmed”, “shipped” or “completed”. This table stores information about the status of an order. There are totally 4 statuses available in this project.

Column Name(s)	Data Type	Purpose and usage
status_id	Int	Primary Key, unique identifier for a status. Referred in orders table status_id column
status_name	varchar	Name of the status

4. Implementation

4.1 Implementation of web templates

The website is implemented with a home page, a navigation bar at the top, a footer at the bottom, and the content for each feature mentioned in section 1.

4.1.1 Components

Global navigation bar: The pages on the website are configured with a global navigation bar. It includes two sections:

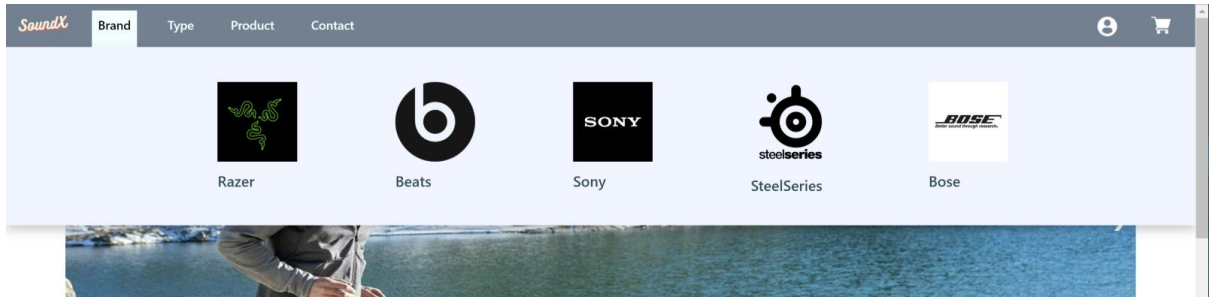
- left-nav section contains functions related to browsing products. This includes 2 sub-navigation components, namely the 'Brand' and 'Type', where the users can view products belong to a specific type or brand from choosing the attributes in the dropdown sub-navigation. It also allows users to access product list ('Product') and the contact information (the 'Contact' as footer).
- right-nav section is user related features, giving access to account and shopping cart.

Each component is set as a link, with text-decoration set to none.

The following is the overall implementation of the navigation bar, with an example of the implementation of the 'Brand' sub-navigation.

```
19 <div class="nav-bar-container">
20 <!-- Left part -->
21 > <a class="logo" href="../home/">...
22 </a>
23 > <div class="brand-container">...
24 </div>
25 > <div class="type-container">...
26 </div>
27 > <div class="product-container">
28 > <a class="left-nav" href="../product_list/">Product</a>
29 </div>
30 > <div class="contact-container">
31 > <a class="left-nav" href="#contact">Contact</a>
32 </div>
33 <!-- Right part -->
34 > <div class="right-nav">
35 > <div class="right-nav-item">
36 > <a href="../account/">
37 > </a>
38 </div>
39 > <div class="right-nav-item">
40 > <a href="../cart/">
41 > </a>
42 </div>
43 </div>
44 </div>
```

(/Element/nav_bar.php)



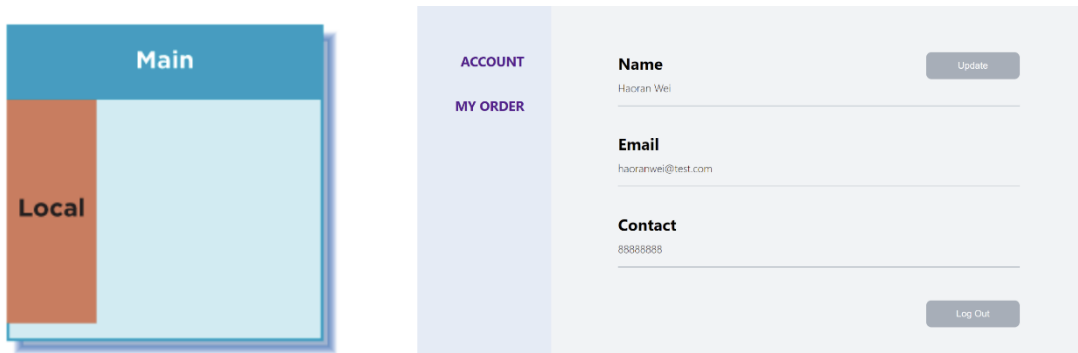
```

19 <div class="nav-bar-container">
20 <!-- Left part -->
21 <a class="logo" href=".." home/">
22 
23 </a>
24 <div class="brand-container">
25 <a class="left-nav brand">Brand</a>
26 <!-- TODO: brand dropdown -->
27 <div class="brand-dropdown-content">
28 <!-- Here is something to display => brand-->
29 <!-- TODO: style the list -->
30 <ul class="nav-dropdown-item">
31 <?php
32 for ($i=0; $i<count($brand); $i++){
33 $brand_row = $brand[$i];
34 $brand_name = $brand_row['brand_name'];
35 $brand_id = $brand_row['brand_id'];
36 echo "<li class='subnav-item'">
37 <div class='item-block'>
38 <a class='item-block-link'
39 href='.."product_list/?brand%5B%5D={$brand_id}'>
40 <div class='item-img'><img src='.."img/brand/{$brand_id}.png" alt='SoundX' width='130px'></div>
41 <div class='font-item'>{$brand_name}</div>
42 </a>
43 </div>
44 </li>;
45 }
46 >
47 </ul>
48 </div>
49 </div>

```

(/Element/nav_bar.php, line 19-49)

Local navigation bar: There is a local navigation bar inside the 'Account' page, where users can switch between account and order information page. The reference is shown on the left and the actual page layout is shown on the right, where the local navigation section is in the left panel. The template implementation code, including html and css, is shown below.



```

33     <div class="left-nav-column">
34         <div id="account">
35             <a href="index.php">
36                 <span>ACCOUNT</span>
37             </a>
38         </div>
39         <div id="order">
40             <a href=" ../orders/index.php">
41                 <span>MY ORDER</span>
42             </a>
43         </div>
44     </div>

```

```

.right-nav-column, .right-nav-column-order{
    display: block;
    width: 800px;
    padding-top: 50px;
    background-color: #rgb(241, 243, 245);
}

.main-content-order, .right-nav-column-order{
    height: fit-content;
}

.right-nav-column-order{
    display: flex;
    justify-content: center;
}

```

Footer: The footer is the last section of each page, which includes contact number, email (clickable) and address.

```

1  <!-- Contact -->
2  <div id="contact">
3      <div id="contact-text">
4          <h2>Contact Us</h2>
5          <p>
6              <!-- (+65) 8888 - 8888 -->
7              &#40;&#43;65&#41; 8888 &#45; 8888 <br />
8              <!-- support@soundx.com -->
9              <a href="mailto:support@soundx.com" style="color:azure; text-decoration: none;">
10                 support&#64;soundx.com</a> <br /><br />
11                 156 Nanyang Cres<br />
12                 Singapore, 636866
13             </p>
14         </div>
15     </div>

```

4.1.2 Layout

Overall layout: The overall layout for the home page and the main content is aligned at the center, using the either the flex display with justify-content = center, or the auto margin. The main content is set with the min-width of 1300px.

For the product listing content, which uses a grid template with 4 columns, as shown in the below image. The grid view is implemented in the css stylesheet.

These are some sample code of centered display implementations:

```

1  body{
2    min-width: 1300px;
3    display: block;
4    margin: auto;
5  }
6
7  .content{
8    margin:auto;
9    min-width: 1300px;
10   padding: 40px 100px;
11 }

```

(/css/product-list-style.css, line 1-11)

```

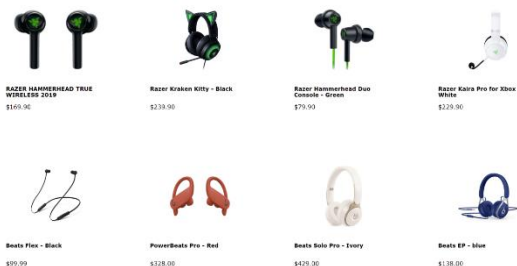
1  body{
2    min-width: 1300px;
3    display: block;
4    margin: auto;
5  }
6
7  .main-content, .main-content-order{
8    display:flex;
9    justify-content: center;
10   min-width: 1300px;
11   margin: 100px 0;
12 }

```

(/css/account-order.css, line 1-12)

The below shows grid layout:

RESULTS



```

131 .product-result {
132   display: grid;
133   grid-template-columns: auto auto auto auto;
134   grid-column-gap: 110px;
135   grid-row-gap: 100px;
136   margin: auto;
137 }

```

(/css/product-list-style.css, line 131-137)

Product list item layout: Each product listing item contains a product image, product title and price. it has a fixed width of 250px, to fit the vertical alignment of the grid view (Figure x.x (a)), and the product image is with a fixed height of 150px and width is set to fit-content (Figure x.x (vb)) to ensure the horizontal alignment in the layout.

```

152   echo '<div class="filter-result-container">';
153
154   echo '<div class = "product-result">';
155   for ($i=0; $i<count($result_arr); $i++){
156     $row = $result_arr[$i];
157     $product_id = $row['product_id'];
158     $product_name = $row['product_name'];
159     $price = $row['price'];
160
161     echo '<a class = "product-card-link" href=" ../product_details/index.php?product_id='.$product_id.'">';
162     echo '<div class = "product-card" id="product_id_'.$product_id.'">';
163     echo '<div>';
164     echo '</div>';
165     echo '<div class = "product-list-info">';
166     echo '<div class="product-list-info-title"><p>'.$product_name.'</p></div>';
167     echo '<div class="product-list-info-price"><p>'.$price.'</p></div>';
168     //echo '<hr id="filter-separation">';
169     echo '</div></div></a>';
170   }
171   echo '</div></div>';
172 }

```

(/product-list/index.php, line 152 – 172)

```

139  .product-card {
140      width: 250px;
141      height: auto;
142      display: block;
143  }
144
145  .product-small-pic {
146      padding-top: 30px;
147      display: block;
148      height: 150px;
149      width: fit-content;
150      overflow: hidden;
151      margin: auto;
152  }

```

(/css/product-list-styl.css/line 139-152)

Section structure: Each section in the page starts with a section header, for example, as shown in the above product listing result page, the header ‘Results’ indicates the below content is product listing results. The text-align in each division (<div>) is left-align.

4.1.3 Font

Heading: Segoe UI",Arial,sans-serif; weight: 700; font size: 28px

Content: Gotham, Verdana, sans-serif; weight: 400; font size: 18px

See sample css code for product listing page:

```

27  .filter-title,
28  .product-result-title {
29      font-size: 28px;
30      font-weight: 700;
31      letter-spacing: 2px;
32      width: 120px;
33      left: 0;
34  }

```

(Heading; /css/product-list-style.css, line 27-34)

```

120  label {
121      font-size: 18px;
122      font-weight: 200;
123      bottom: 5px;
124      vertical-align: middle;
125  }

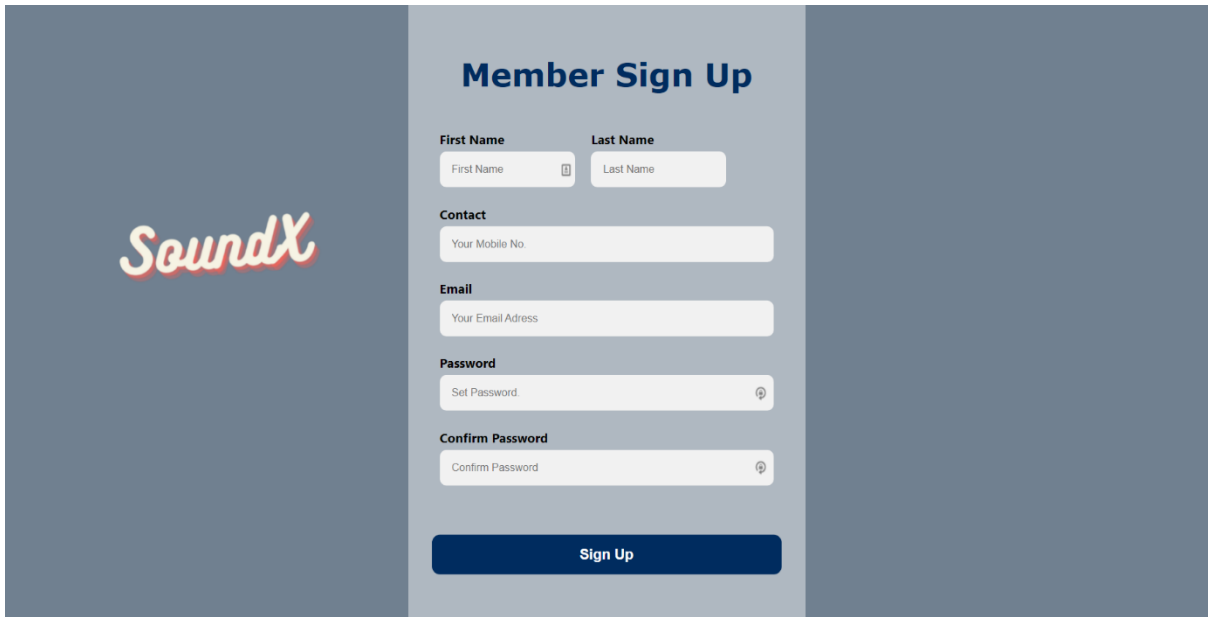
```

(Content (the label for the filters); /css/product-list-style.css, line 120 – 125)

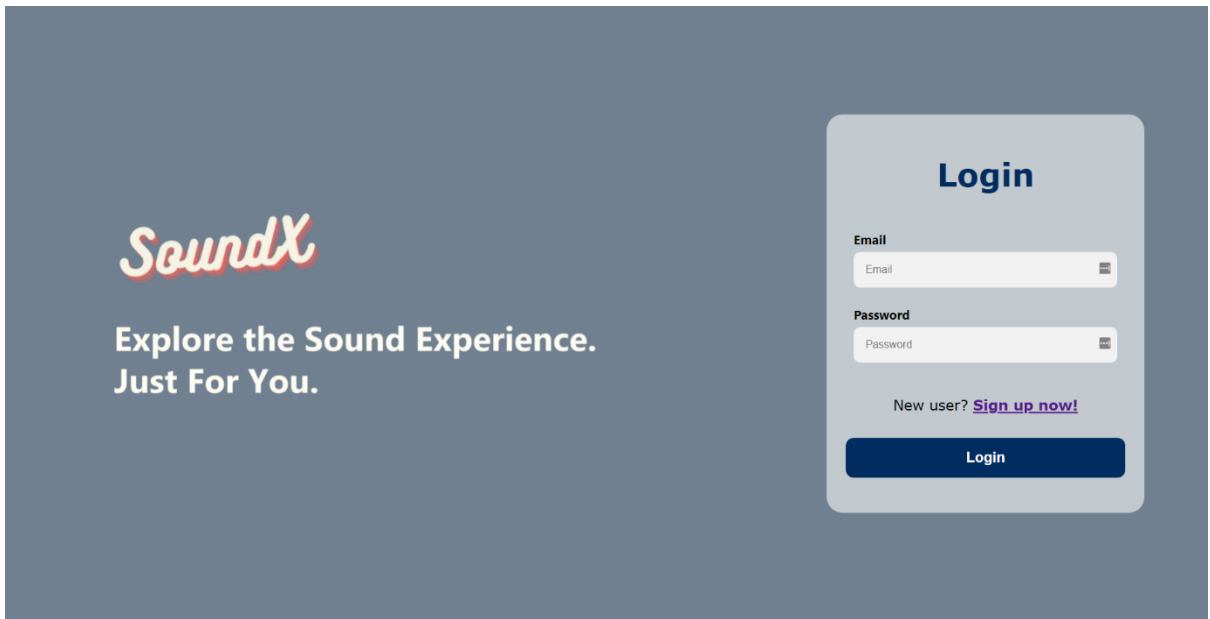
4.2 Implementation of the pages

4.2.1 Signup and login

The main content for signup and login is implemented as a form, with several input fields required for sign up or log in information, as well as a submit button to post the form.



Screenshot 1: Sign up page



Screenshot 2: Login page

```

8 <head>
9 <title>SoundX - Sign Up</title>
10 <meta charset="utf-8" />
11 <link rel="stylesheet" href="../css/login-style.css" />
12 <link rel="stylesheet" href="../css/elements.css" />
13 </head>
14
15 <body>
16 <div class="signup-bg">
17 <form action="" method="post" class="signup-container">
18 <div class="sign-up-input-block">
19 <div class="signup-title">
20 <h1>Member Sign Up</h1>
21 </div>
22 <div class="input-field">
23 <div id="name-input-row">
24 <div id="name-input">
25 <label for="FirstName"><b>First Name</b></label>
26 <input type="text" placeholder="First Name" name="firstName" id="FirstName" required>
27 </div>
28 <div id="name-input">
29 <label for="LastName"><b>Last Name</b></label>
30 <input type="text" placeholder="Last Name" name="lastName" id="LastName" required>
31 </div>
32 </div>
33 </div>
34
35 <div id="invalidNameAlert"></div>
36 <div>
37 <label for="Contact"><b>Contact</b></label>
38 <input type="text" placeholder="Your Mobile No." name="contact" id="Contact" required>
39
40 <div id="invalidContactAlert"></div>
41
42 <label for="Email"><b>Email</b></label>
43 <input type="email" placeholder="Your Email Address" name="email" id="Email" required>
44
45 <div id="invalidEmailAlert"></div>

```

(/signup/index.php)

```

21 <body>
22 <div class="login-bg">
23 <form action="" method="post" class="login-container">
24 <div>
25 <div class="login-title">
26 <h1>Login</h1>
27 </div>
28 <div class="input-field">
29
30 <label for="Email"><b>Email</b></label>
31 <input type="text" placeholder="Email" name="email" id="Email" required>
32
33 <label for="Password"><b>Password</b></label>
34 <input type="password" placeholder="Password" name="password" id="Password" required>
35 </div>
36 <div id="sign-up-request">
37 <p>New user? <a href="../sign_up">Sign up now!</a></p>
38 </div>
39 <div>
40 <button type="submit" class="login-button">Login</button>

```

(/login/index.php)

4.2.2 Home Page

Home page includes 3 major sections in the main content: slideshow banner, top sales & see more, as well as a promotion banner.



TOP SALES



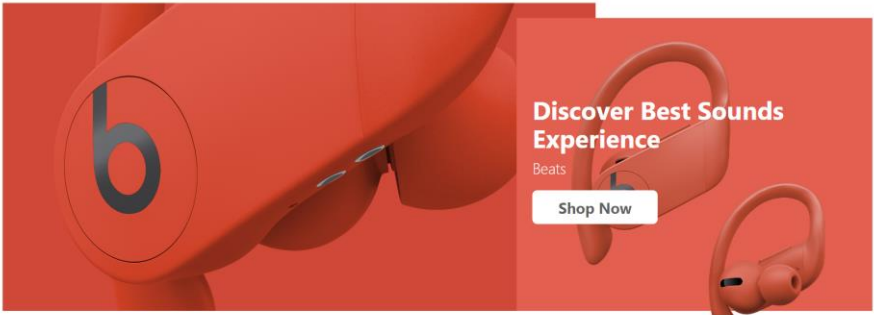
WF-1000XM4 Wireless Noise Cancelling Headphones
Noise Cancelling
\$379.00



RAZER HAMMERHEAD TRUE WIRELESS 2019
True Wireless Earbuds
\$169.90



Beats EP blue
Wired Headphone
\$138.00



SEE MORE



WIRELESS



GAMING



WIRED



BLUETOOTH

Contact Us

(+65) 8888 - 8888
support@soundx.com

156 Nanyang Cres
Singapore, 636866

Screenshot 3: Home page

```

21 <body>
22 <!-- the navigation bar -->
23 <?php include '../Elements/nav_bar.php';?>
24
25 <!-- Big Banner container -->
26 <div class="top-banner-container">
27 <!-- Full-width images -->
28 <div class="banner-page transition">
29 <a href="../product_details/?product_id=21">
30 
31 </a>
32 <div class="caption-text">Bose Sport Open Earbuds</div>
33 </div>
34
35 <div class="banner-page transition">
36 <a href="../product_details/?product_id=13">
37 
38 </a>
39 <div class="caption-text">IER-H500A h.ear in 2 In-ear Headphones</div>
40 </div>
41
42 <div class="banner-page transition">
43 <a href="../product_details/?product_id=3">
44 
45 </a>
46 <div class="caption-text">Razer Hammerhead Duo Console</div>
47 </div>
48
49 <!-- Next and previous buttons -->
50 <a class="prev" onclick="next(-1)"&#10094;</a>
51 <a class="next" onclick="next(1)"&#10095;</a>
52 </div>

```

(slideshow banner)

```

55 <!-- Top Sales -->
56 <div class="top-sales">
57 <div class="title">
58 <h1>TOP SALES</h1>
59 </div>
60 <div class="top-sales-column">
61 <div class="product-list">
62 <div class="product-list-flex">
63 <div class="product-list-item-block">
64 <div class="product-list-item-img">
65 <a href="../product_details/?product_id=10" class="item-img-link">
66 
67 </a>
68 </div>
69 <div class="product-list-item-info">
70 <a class="item-info-link">
71 <div class="item-info-body">
72 <p class="item-info-body-title">WF-1000XM4 Wireless Noise Cancelling Headphones</p>
73 <p class="item-info-body-type">Noise Cancelling</p>
74 <p class="item-info-body-price">$379.00</p>
75 </div>
76 </a>
77 </div>
78 </div>
79 > <div class="product-list-item-block">...
94 </div>
95 > <div class="product-list-item-block">...
110 </div>
111 </div>
112 </div>
113 </div>

```

(top sale section. The implementation of ‘see more’ section is similar to top sale, except that instead of listing a product as the item, the see more display the product category as for each item)

```

114 <!-- Full-width banner promotion -->
115 <div class="topSale-banner-container">
116 <div class="topSale-banner-img">
117 
118 </div>
119 <div class="topSale-banner-text-container">
120 <div class="topSale-banner-text">
121 <div id="topSale-title">
122 <p style="font-size:40px; font-weight:700; line-height:45px; margin:0px 0 10px 0; color:white">
123 Discover Best Sounds Experience</p>
124 <p style="font-size:24px; font-weight:300; margin:10px 0; color:white">Beats</p>
125 </div>
126 <div>
127 <!-- top sales promotion product shop now-->
128 <a href="../product_details/?product_id=6" class="product-button">
129 <span class="product-button-text">
130 Shop Now
131 </span>
132 </a>
133 </div>
134 </div>
135 </div>
136 </div>
137 </div>

```

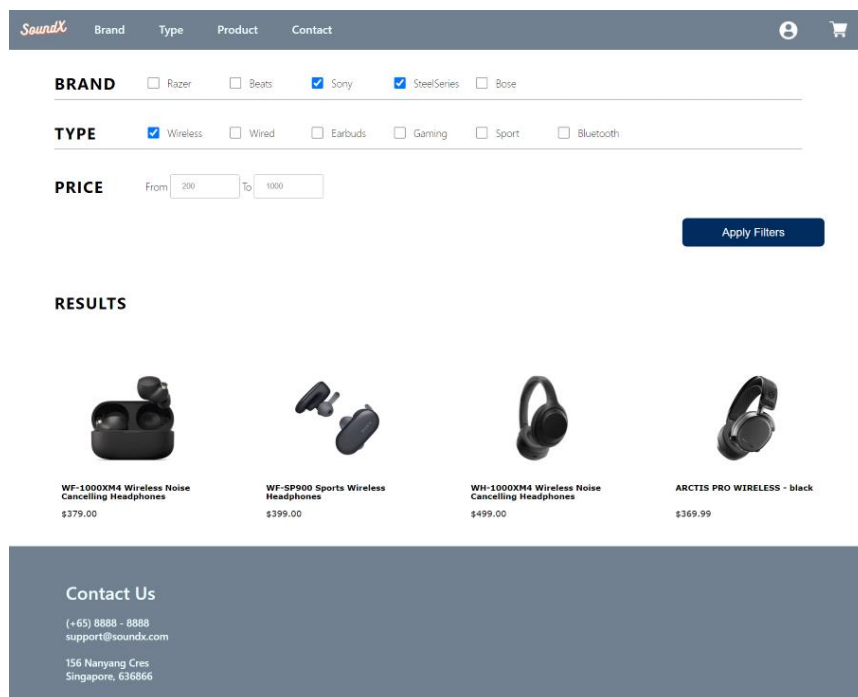
(top sale banner)
(/home/index.php)

4.2.3 Product listing

The page contains a filter section and filter result section.

In the filter section, 3 filters are available: brand, type and price range. For each item in brand and type filter, it is implemented as a checkbox type input. Price range is text input.

The result section all results belong to the selected filter attributes (by default, it displays all products if no filter applied).



Screenshot 4: products page

```

13 <?php include '../Elements/nav_bar.php';?>
14 <div class="filters-container">
15     <div class="filters">
16         <form id="all_filters" action="" method="get">
17             <?php
18                 if (isset($_GET)){
19                     $get_variable = json_encode($_GET);
20                     echo "<script>var filter_get = {$get_variable};</script>";
21                 }?>
22                 <div class="brand-filters">
23                     <div class="filter-title">BRAND</div>
24                     <div class="filter-checkbox-list">
25                         <div class="checkbox-item">
26                             <input type="checkbox" name="brand[]" id="brand1" value="1">
27                             <label for="brand1">Razer</label>
28                         </div>
29 > <div class="checkbox-item">...
32 </div>
33 > <div class="checkbox-item">...
36 </div>
37 > <div class="checkbox-item">...
40 </div>
41 > <div class="checkbox-item">...
44 </div>
45     </div>
46 </div>
47 <hr id="filter-separation">
48
49     <div class="filters">
50         <div class="type-filters">
51             <div class="filter-title">TYPE</div>
52             <div class="filter-checkbox-list">
53                 <div class="checkbox-item">
54                     <input type="checkbox" name="Wireless" id="Wireless" value="true">
55                     <label for="type1">Wireless</label>
56                 </div>
57 > <div class="checkbox-item">...
60 </div>

```

(type and brand filters)

```

82     <div class="price-filter">
83         <div class="filter-title">PRICE</div>
84         <div class="filter-price-range">
85             <label for="min_price">From</label>
86             <input type="text" name="min_price" onClick="this.select();" id="min_price" size="2" value="0"
87                 onchange="validateNumberInput(this);">
88             <label for="max_price">To</label>
89             <input type="text" name="max_price" onClick="this.select();" id="max_price" size="2" value="1000"
90                 onchange="validateNumberInput(this);">
91         </div>
92         <div id="alert-msg" style="margin-right: 400px;"></div>
93     </div>
94     <input id="filter-button" type="submit" value="Apply Filters">
95 </form>

```

(price range filter)

```

152     echo '<div class="filter-result-container">';
153
154     echo '<div class = "product-result">';
155     for ($i=0; $i<count($result_arr); $i++){
156         $row = $result_arr[$i];
157         $product_id = $row['product_id'];
158         $product_name = $row['product_name'];
159         $price = $row['price'];
160
161         echo '<a class = "product-card-link" href="../../product_details/index.php?product_id='.$product_id.'">';
162         echo '<div class = "product-card" id="product_id_'.$row['product_id'].'">';
163         echo '<div>';
164         echo '</div>';
165         echo '<div class = "product-list-info">';
166         echo '<div class="product-list-info-title"><p>'.$product_name.'</p></div>';
167         echo '<div class="product-list-info-price"><p>'.$price.'</p></div>';
168         //echo '<hr id="filter-separation">';
169         echo '</div></div></a>';
170     }
171     echo '</div></div>';
172 }

```

(result section)
(/product-list/index.php)

4.2.4 Product Details

The product details page includes 2 sections, one is to showcase product images, basic information and action buttons (I.e. ‘add to cart’ and ‘buy now’), followed by a section to describe the product specifications.

```

27 <div class="product-show">
28 <div class="image-and-selector">
29 <div class="main-img">
30 <img src=<?php echo '../../img/product/{product_id}-1.jpg';?> id="product_big_picture" />
31 </div>
32 <div class="image-selector-small">
33 <?php $num_of_pictures = (int) $row["no_pictures"];
34 for ($pic = 1; $pic <= $num_of_pictures; $pic++){
35 $pic_source = "../../img/product/{product_id}-{pic}.jpg";
36 echo '<div class="image-selector-small-item">';
37 echo '<img class="pic-thumbnail" src='.$pic_source.' id="product_pic_'.$pic.'" width="60px" height="60px" onclick="switchImage(\'product_pic_'.$pic.'\');"/>';
38 }>
39 </div>
40 </div>
41 <div class="text-title-part">
42 <h1><?php echo $row["product_name"]; ?></h1>
43 <h2><?php echo $row["short_description"]; ?></h2>
44 <p class="price"><strong><?php echo $row["price"]; ?></strong></p>
45 <p class="stock">Stock Left: <?php echo $row["stock"]; ?></p>
46 <div class="button-group">
47 <div id="add-to-cart-button">
48 <form action="add_to_cart.php" method="post">
49 <input type="text" hidden name="product_id" value="<?php echo $product_id; ?>">
50 <button id="cart-button" onclick="this.form.submit();">
51 <?php
52 if ($added_in_cart){
53 echo "View In Cart";
54 } else {echo "Add to Cart";}
55 <?php
56 </button>
57 </form>
58 </div>
59 <div id="buy-now-button">
60 <form id="buyNow" action="../../checkout/index.php" method="post">
61 <input type="hidden" id="buy-now-select-this" name="selected[]"
62 value="<?php echo $product_id; ?>">
63 <input type="hidden" id="buy-now-qty-one" name="items[<?php echo $product_id; ?>]"
64 value="1">
65 <button id="buy-button" onclick="buyNow.submit();">Buy Now</button>
66 </form>
67 </div>

```

(product image, basic information and buttons)

```

73 <div class="product-detail-info">
74   <div class="detail-info-container">
75     <h1>Description</h1>
76     <hr class="solid">
77     <p><?php echo $row["description"]; ?></p>
78   </div>
79
80   <div class="detail-info-container">
81     <h1>Specification</h1>
82     <hr class="solid">
83     <table id="spec-table">
84       <?php $json_content = json_decode($row["specification"], true);
85       foreach($json_content as $key => $value){
86         echo "<tr class='spec-table'>";
87         echo "<td class='spec-table-key'>".$key."</td>";
88         echo "<td class='spec-table-value'>".$value."</td>";
89         echo "</tr>";
90       }
91     >
92   </table>
93 </div>
94 </div>

```

(product description and specification)
(/product-details/index.php)

Razer Hammerhead Duo Console - Green
Dual Driver In-Ears
\$79.90
Stock Left: 22
Add to Cart Buy Now

Description
Dual Driver Technology, Three Silicone Tip Sizes, Aluminum Frame & Braided Cables

Specification

HEADPHONE FREQUENCY RESPONSE	20 Hz-20 KHz
HEADPHONE IMPEDANCE	32 Ohm ± 15%
HEADPHONE SENSITIVITY	112 ± 3 dB (Max SPL)
HEADPHONE INPUT POWER	10 mW/20 mW
HEADPHONE DRIVERS	Dynamic + Balanced Armature
HEADPHONE CONNECTOR	3.5 mm angled jack
CABLE LENGTH	1.2 m
WEIGHT	17 g
MICROPHONE FREQUENCY RESPONSE	100 Hz -10 kHz
MICROPHONE PICK UP PATTERN	Omni-directional
MICROPHONE SIGNAL-TO-NOISE RATIO	58 dB
MICROPHONE SENSITIVITY (@1KHZ)	-40 ± 3 dB
COMPATIBILITY	Devices with 3.5 mm audio jack, Devices with 3.5 mm audio + microphone combined jack

Screenshot 5: product details page

4.2.5 Checkout

Checkout is divided into left and right column. The left column contains a form for buyer to fill up delivery information. In the right column is the order summary and payment method selection.

```
40 <div class="order_details_form">
41 <div style = "display:block">
42 <form name="checkout" action="create_order.php" method="post" class="form_content">
43 <div class="order_details_left_col">
44 <h1>Delivery Information</h1>
45 <div class="details_address">
46 <h2>Delivery Address</h2>
47 <input type="text" id="delivery_address_line_1" name="delivery_address_line_1"
48 placeholder="Address Line 1" required>
49 <input type="text" id="delivery_address_line_2" name="delivery_address_line_2"
50 placeholder="Address Line 2">
51 </div>
52 <div class="details_code">...
56 </div>
57 <div class="details_personal">...
63 </div>
64 </div>
65 <div class="order_details_right_col">
66 <h1>Order Details</h1>
67 <div class="order_summary">
68 <?php ...
114 ?>
115 </div>
116 <div class="payment">
117 <h1>Payment Method</h1>
118 <div class="payment_button">
119 <label>
120 <input type="radio" id="payment_visa" name="payment_method" value="visa" required>
121 
122 </label>
123 <label>...
126 </label>
127 <label>...
130 </label>
131 </div>
132 </div>
133 <div id="noPaymentAlert" style="margin-bottom: 30px"></div>
134 <div id="submit_button">
135 <input id="checkout-button" type="submit" value="Place Order">
```

(/checkout/index.php)

Back

Delivery Information

Delivery Address

Address Line 1 Address Line 2

Postal Code

Name

Contact No.

Order Details

Items	Qty	Price	Subtotal
Beats Solo Pro - Ivory	1	429.00	429.00
Total			\$ 429.00

Payment Method



Please Select Payment Method.

Place Order

Screenshot 6: checkout page

4.2.6 Order Details

The order details page will be generated after the order is created. At the top, a progress bar shows the order status. The main content is a table to show the ordered item, followed a section to summarize the delivery information.

```
174 <div class="status-bar-container">
175 <?php
176 $curr_status = (int) $order_details['status'];
177 $all_status = select_all_from_table($db, 'status');
178 for ($node = 0; $node < count($all_status); $node++){
179     echo "<div class='phase'>";
180     if ($node < $curr_status){
181         echo "<div class='status-dot done'></div>";
182         if ($node != 0){
183             echo "<div class='status-line solid'></div>";
184         }
185     } else {
186         echo "<div class='status-dot'></div>";
187         echo "<div class='status-line'></div>";
188     }
189     echo "<div class='status-text'>{$all_status[$node]['status_name']}</div></div>";
190 }
191 ?>
```

(Progress bar for order status)

```
193 <div class="order-details-container">
194 <h1>Order Details</h1>
195 <!-- order items table -->
196 <?php include 'item_in_order.php';
197 echo "<div class='index-order-detail-container'>";
198 order_item_table($order_id, $order_details, $order_items);
199 echo "</div>";
200 ?>
201 </div>
202 <div class="delivery-info-container">
203 <!-- delivery information row -->
204 <h1>Delivery Information</h1>
205 <?php
206 echo "<div class='delivery-info'><div id='personal'><p class='title'>Name</p><p>{$order_details['receiver_name']}</p>";
207 echo "<p class='title'>Contact No. </p><p>{$order_details['receiver_contact']}</p></div>";
208 echo "<div id='address'><p class='title'>Address</p>";
209 echo "{$order_details['delivery_address_line_1']}<br>";
210 if (strlen($order_details['delivery_address_line_2'] > 0)){
211     //if delivery address line 2 is set
212     echo "{$order_details['delivery_address_line_2']}<br>";
213 }
214 echo "{$order_details['zip_code']}</div>";
215 $track_number = (strlen($order_details['track_number'])>0)? $order_details['track_number'] : "Not Available Now";
216 if ($order_details['track_number'] == NULL){
217     echo "<div id='track'><p class='title'>Track Number</p><p>{$track_number}</p></div></div>";
218 }
219 ?>
220 </table>
221 </div>
```


(Main content)

(/order_details/index.php)

Table for ordered item:

Order: #60437442

Created Time: 2021-11-05 14:56:43

Item	Quantity	Price	Subtotal
 WF-1000XM4 Wireless Noise Cancelling Headphones	1	\$379.00	\$379.00
Total			\$379.00

```

2 function order_item_table($order_id, $order_details, $order_items){
3     $order_id = str_pad($order_id, 8, '0', STR_PAD_LEFT);
4     echo "<div class='order-table-main'>
5     <div id='order-id-row'>
6         <p><strong>Order:</strong> #{$order_id}</p>
7     </div>
8     <div id='create-time-row'><!-- create time row -->
9         <p><strong>Created Time:</strong> {$order_details['create_time']}</p>
10    </div>";
11
12    echo "<div class='item-row-container'><!-- items row, contains items table -->
13    <table>
14        <tr>
15            <td colspan='2' id='table-head-item'>Item</td>
16            <td id='table-head-qty'>Quantity</td>
17            <td id='table-head-price'>Price</td>
18            <td id='table-head-subtotal'>Subtotal</td>
19        </tr>";
20    $total = 0;
21    foreach($order_items as $item){
22        echo "<tr class='item-row'>";
23        echo "<td id='img-container'><a href='../product_details/?product_id={$item['product_id']}'><img src='../img/product-snapshot/{
24        <td id='product-name-container'><p>{$item['product_name']}</p></td>"; //product name
25        echo "<td id='qty-container'><p>{$item['qty']}</p></td>"; //Qty
26        $price = sprintf('%0.2f', $item['price']);
27        echo "<td id='price-container'><p>\${$price}</p></td>"; //Price
28        $subtotal = sprintf('%0.2f', $item['qty'] * $item['price']);
29        $total += $item['qty'] * $item['price'];
30        echo "<td id='subtotal-container'><p>\${$subtotal}</p></td>"; //Subtotal
31        echo "</tr>";
32    }
33    echo "<tr class='total-row'><td colspan='4' id='total'><p>Total</p></td>";
34    echo "<td id='total-amount'><p>\$";
35    echo sprintf('%0.2f', $total);
36    echo "</p></td></tr></table></div>";
37 }

```

(The function of showing the ordered item information in a table is defined in ‘/order_details/item_in_order.php’)


SoundX Brand Type Product Contact

Back

paid confirmed shipped received

Order Details

Order: #24053638
Created Time: 2021-10-01 15:49:06

Item	Quantity	Price	Subtotal
 Razer Kraken Kitty - Black	1	\$239.90	\$239.90
Total			\$239.90

Delivery Information

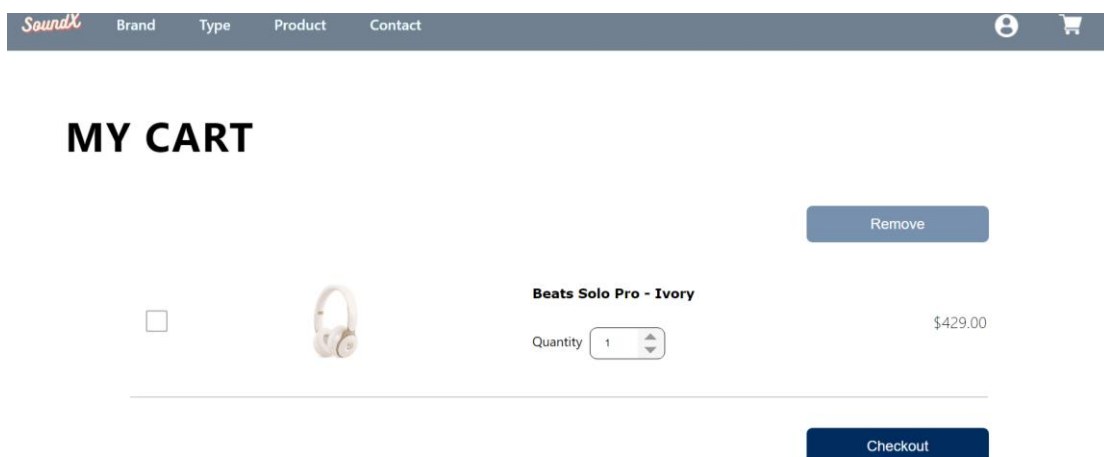
Name	Address	Track Number
Haoran Wei	156 Nanyang Cres 636866	Not Available Now
Contact No. 88888888		

Screenshot 7: order details page

4.2.6 Cart

The page shows the items added to the cart. The main content is displayed in a table. Each item contains an image snapshot, title and price, a quantity field (as a 'number' type input). As users are allowed to delete or buy the items, each item in the table also comes with a checkbox. The actions are done by selecting the products and clicking on the 'Remove' or 'Buy' button.

```
17 <div class="item-cart-title">
18 <h1>MY CART</h1>
19 </div>
20 <?php
21 if (count($_SESSION['cart'])==0){
22     echo '<div class = "cart-container">';
23     echo '<div id="empty_cart"><p>No Items In cart, Please keep shopping.</p></div>';
24     echo '</div>';
25 } else {
26     echo '<div class = "cart-container">';
27     echo '<form id="cart-form" action="../checkout/" method="post">';
28     echo "<input type='hidden' name='remove' value='false' id='removed-placeholder'>";
29     echo '<button id="remove-button" onclick="removeItemFromCart();">Remove</button>';
30     echo '<table class="cart-table">';
31     foreach ($_SESSION['cart'] as $product){
32         $info = get_details_by_id($db, $product);
33         echo "<tr><td>";
34         echo '<div class="cart-row">';
35         echo "<input type='checkbox' name='selected[]' value='{$_product}' id='cart-checkbox'>";
36         echo '<div class="cart-img-container">';
37         echo "<a href='../product_details/?product_id=$_product'>";
38         echo "<img src='../img/product_snapshot/{$_product}.png' id='cart-img'></a></div>";
39         echo '<div id="product_name_quan">';
40         echo "<p>.$info['product_name'].</p>";
41         echo '<div id="quan_field">';
42         echo '<label for="brand2">Quantity</label>';
43         echo "<input type='number' name='items[{$_product}]' min='1' value='1' max='{$_info['stock']}'>";
44         echo '</div></div>';
45         echo '<div id="price-field">';
46         echo "<p>.$info['price'].</p>";
47         echo '</div></div>';
48         echo "</td></tr>";
49     }
50     echo "</table>";
51     echo '<input type="submit" value="Checkout">';
52     echo '</form>';
53     echo '</div>';
54 }
55 ?>
```



Screenshot 8: my cart page

4.2.7 Account and My Order

Account page contains user information and allows users to update it. By using the local navigation, the user can switch to view all the historical orders, displayed in a table view (same as the one mentioned in 4.2.5) for each order.

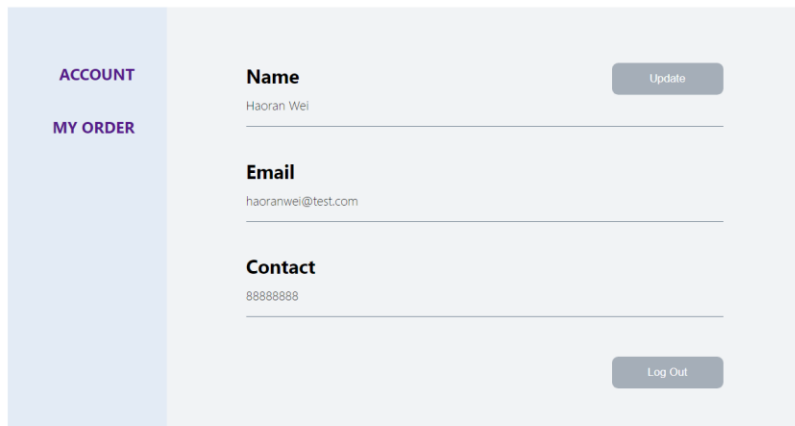
The implementation of the local navigation bar at the left panel is shown in 4.1.1. The following is the implementation of the account information and order history at the right panel.

```
45 <div class="right-nav-column">
46 <div id="user-info-text">
47 <button id="update-button" onclick="displayUpdateContent();">Update</button>
48 <form id="account-update" action="update_account_info.php" method="post">
49 <h2>Name</h2>
50 <p id="name-display">
51 <?php echo $name; ?>
52 </p>
53 <input required type="text" hidden id="first-name-update" name="first_name"
54 value="<?php echo $firstname;?>" onClick="this.select();">
55 <input required type="text" hidden id="last-name-update" name="last_name"
56 value="<?php echo $lastname;?>" onClick="this.select();">
57 <hr>
58 <h2>Email</h2>
59 <p id="email-display">...
60 </p>
61 <hr>
62 <h2>Contact</h2>
63 <p id="contact-display">...
64 </p>
65 <input required type="text" hidden id="contact-update" name="contact" value="<?php echo $contact;?>" ...
66 </p>
67 <hr>
68 <div id="invalidContactAlert"></div>
69 <input id="confirm-update-button" type='hidden' value='Update'>
70 </form>
71 <form action='logout.php'>
72 <input id="logout-button" type='submit' value='Log Out'>
73 </form>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
```

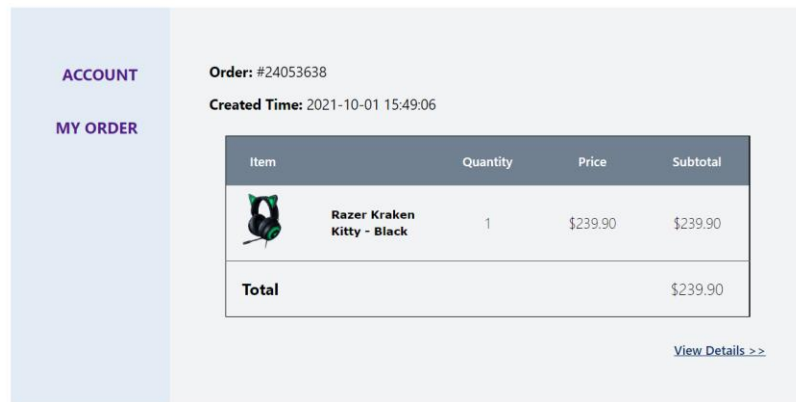
(/Account/index.php)

```
165 <?php
166 $curr_user = $_SESSION['user_id'];
167 $query_orders = "SELECT * FROM orders WHERE user_id = {$curr_user} ORDER BY create_time DESC";
168 @$order_result = $db -> query($query_orders);
169 echo '<div id="order-item-table">';
170 $num_row = $order_result -> num_rows;
171 //echo $num_row;
172 if($num_row==0){
173 echo "<div id='empty_order'><p>You do not have any order at this time.</p></div>";
174 }else{
175 while($row = $order_result -> fetch_assoc()){
176 $order_id = $row['order_id'];
177 $order_items = get_order_items_with_product_info_by_order_id($db, $order_id);
178 order_item_table($order_id, $row, $order_items);
179 echo "<div id='order-view-details'>";
180 echo "<a href='../order_details/?id={$order_id}'>View Details >></a></div>";
181 }
182 }
183 echo '</div>';
184 ?>
```

(the implementation of showing the table for ordered item is the same as the one in 4.2.5)
(/Orders/index.php)



Screenshot 9: my account page



Screenshot 10: my orders page

4.3 Implementation of the functions:

4.3.1 Sign Up and Login:

Sign up

A user who first visits the *SoundX* website is required to sign up for a new account. By clicking on the link “New user? Sign up now!”, user will be directed to the "Sign Up”

page, where the user needs to input the name, contact number, email address, and create a new password.

After the user fills up the information above and submits, all the inputs will be sent by POST method. The server will first check if the email entered is already in the *users* table.

- If it is already in the table, the page will throw a warning message “The email you entered is already registered! Login here”, directing the user to the "Login" page if the user click on the “Login here”.
- If it is not registered in users table, a new set of values will be inserted in the table (line 70), with an auto-incremented *user_id*. A message will inform user that the sign up is successful and redirect user to the login page after 3 seconds.

```
58 <?php
59 //first check whether the entered email is already in database
60 if (isset($_POST['email']) && isset($_POST['password'])){
61     $try_select = "SELECT email FROM users WHERE email='{$_POST['email']}'";
62     $result = $db -> query($try_select);
63     if ($result -> num_rows > 0){
64         //the email has already been registered
65         echo '<div class="alert">';
66         echo "The email you entered is already registered! ";
67         echo '<a href="..">login here</a>';
68         echo '</div>';
69     } else {
70         $insert_query = "INSERT INTO users (email, password, first_name, last_name, contact)";
71         $insert_query .= "VALUES('{$_POST['email']}', '{$_POST['password']}', '{$_POST['firstName']}', '{$_POST['lastName']}';";
72         $insert_res = $db -> query($insert_query);
73         echo "<div class='success'>Sign up successfully! Redirecting to login in 3 seconds...</div>";
74         header('Refresh: 3; url=../login/');
75     }
76 }
```

(/signup/index.php, line 58-78)

Log in

On the login page, the users enter the email and password registered. Upon submission, the fields will be sent by POST method and the server will verify whether the email and password are in the table *users*:

```
SELECT user_id, email, password FROM users WHERE email='{$_POST['email']}'
AND password='{$_POST['password']}'
```

If there's no matching user email-password pair in the table, the page will refresh and a warning message will be shown, “Input email or password is not correct.”

If there is matched pair of user email and password, the user can successfully log in to the home page. From here, SESSION starts and three SESSION variables are created: *'user_id'* which is the *user_id* of current user queried from the *users* table; *'email'*, the email of the user, and *'cart'*, an empty array which will be stored with the items that user adds in cart.

```

41 <?php
42 //check whether the user is already in database
43 if (isset($_POST['email']) && isset($_POST['password'])){
44 //user has key in email and password, verify in database
45 $query = "SELECT user_id, email, password FROM users WHERE ";
46 $query.= "email='{$_POST['email']}' 'and password='{$_POST['password']}'";
47 $result = $db->query($query);
48 if ($result->num_rows > 0) {
49 $row = $result -> fetch_row();
50 $_SESSION['user_id'] = $row[0];
51 $_SESSION['email'] = $_POST["email"];
52 $_SESSION['cart'] = array();
53 echo "<script>alert({$_SESSION['user_id']});</script>";
54 header("Location: http://192.168.56.2/f32ee/EE4717-SoundX/");
55 } else {
56 echo "<div class='alert'>&#10069 &nbsp;Input email or password is not correct.</div>";
57 }
58 $_POST = array();
59 }?>
60 </div>

```

(/login/index.php, line 41-60)

4.3.2 Filters

General:

On the product listing page, user can choose one and more product characteristics, including product brands, types and price range (From *min_price* to *max_price*), to filter the desired products. The outcome of filtering will be all the items stored in the database having the matching attributes.

Using *\$filters = \$_GET* to obtain all filter inputs, which are passed into *\$constrant_set* array to query the products with corresponding attributes values

```

101 $query = "SELECT * FROM products ";
102 $constrant_set = array();
103 $filters = $_GET;
104 if (isset($filters["min_price"])){
105     if ($filters["min_price"] != ""){
106         $constrant_set[] = " price >= {$filters["min_price"]} ";
107     }
108     unset($filters["min_price"]);
109 }
110
111 if (isset($filters["max_price"])){
112     if ($filters["max_price"] != ""){
113         $constrant_set[] = " price <= {$filters["max_price"]} ";
114     }
115     unset($filters["max_price"]);
116 }
117
118 if (isset($filters["brand"])){
119     $brands = $filters["brand"];
120     $comma_saparate = implode(",", $brands);
121     $constrant_set[] = " brand_id IN ({$comma_saparate})";
122     unset($filters["brand"]);
123 }
124
125 $type_constraint = array();
126 foreach ($filters as $key => $value){
127     $type_constraint[] = $key."=".$value;
128 }
129
130 if (count($type_constraint) > 0){
131     $or_saparate = implode(" or ", $type_constraint);
132     $constrant_set[] = " (".$or_saparate.")";
133 }
134 if (count($constrant_set) > 0){
135     $query .= " WHERE ".implode(" and ", $constrant_set);
136 }
137
138 $result_arr = get_product_list_by_query($db, $query);

```


(/product_list/index. php, line 101-138)

```
12 function get_product_list_by_query($db, $query){
13     $result = $db -> query($query);
14     $result_arr = array();
15     while ($row = $result -> fetch_assoc()){
16         array_push($result_arr, $row);
17     }
18     return $result_arr;
19 }
```

(query_utils. Php, line 12-19)

Restriction:

The inputs of price should be a positive number

```
1 function validateNumberInput(new_input){
2
3     if (new_input !== null){
4         if (new_input.value == ""){
5             new_input.value = (new_input.id == "min_price")? 0 : 1000;
6         }
7     }
8     var min_price_ele = document.getElementById("min_price");
9     var max_price_ele = document.getElementById("max_price");
10    var pattern = /^[+-]?([0-9]+\.\.?[0-9]*|\.[0-9]+)$/;
11    if (min_price_ele.value.search(pattern) == 0 && max_price_ele.value.search(pattern) == 0){
12        clearAlertMessage();
13        if (parseFloat(min_price_ele.value)<0){
14            min_price_ele.value = 0;
15        }
16        if (parseFloat(max_price_ele.value)<0){
17            max_price_ele.value = 1000;
18        }
19    }
```

(/product_list/product_list.js, line 1-18)

Exception:

If the inputs for price range is not a positive number, when user clicks on the ‘Apply Filter’ button, the price range will be automatically reset to the default (I.e. from 0 to 1000); if the inputs are non-digit, after clicking on the ‘Apply Filter’ button, an alert message ‘Not a valid input, please try again’ will be displayed and the inputs are reset

```
19     } else if (min_price_ele.value.search(pattern) !== 0){
20         showAlertMessage();
21         min_price_ele.value = 0;
22     } else if (max_price_ele.value.search(pattern) !== 0){
23         showAlertMessage();
24         max_price_ele.value = 1000;
25     }
26 }
27
28 validateNumberInput(null);
29
30 function showAlertMessage(){
31     document.getElementById("alert-msg").innerHTML = "Not a valid number input, please try again! ";
32     document.getElementById("alert-msg").className = "alert";
33 }
```

(/product_list/product_list.js, line 19-33)

When none of the product in database can be matched to the choosing attributes, the webpage will return ‘No Result Found’.

```

136     if (count($constrant_set) > 0){
137         $query .= " WHERE ".implode(" and ", $constrant_set);
138     }
139     //var_dump($query);
140
141     $result_arr = get_product_list_by_query($db, $query);
142
143     echo '<div class="product-result-title">RESULTS</div>';
144
145     //if no result is available
146     if (empty($result_arr)){
147         echo '<div id="no_result" style="display: flex; justify-content: center;">
148     }
149

```

(/product_list/index.php, line 136-148)

4.3.3 Shopping Cart

General:

Shopping Cart is created with a session using `session_start()`, initialized as an empty array after login.

On the product details page, users can click on the “Add to Cart” button to add one item to the shopping cart. After that, the “Add to Cart” button will be changed to “View in Cart” which allows the user to check the added item in the shopping cart.

```

48     <div id="add-to-cart-button">
49         <form action="add_to_cart.php" method="post">
50             <input type="text" hidden name="product_id" value="<?php echo $product_id; ?>">
51             <button id="cart-button" onclick="this.form.submit();">
52                 <?php
53                 if ($added_in_cart){
54                     echo "View In Cart";
55                 } else {echo "Add to Cart";}
56

```

(/product_details/index.php, line 48-56)

```

1  <?php include_once '../db_conn.php';
2  session_start();
3  $product_id = $_POST["product_id"];
4  //product_id (str) => quantity (int)
5  if (!in_array($product_id, $_SESSION["cart"])){
6      $_SESSION["cart"][] = $product_id;
7      header("Location: http://192.168.56.2/f32ee/EE4717-SoundX/product_details/index.php?product_id={$product_id}");
8  } else {
9      header("Location: http://192.168.56.2/f32ee/EE4717-SoundX/cart/");
10 }
11 exit();
12

```

(/product_details/add_to_cart.php, line 1-12)

In the Shopping Cart, user can view the items added with the respective thumbnail image, name, quantity and prices by using function `get_details_by_id` to query for the product row and fetch the attribute values (e.g. `$Info['price']` is with respect to the price of the item)

```

4  function get_details_by_id($db, $product_id){
5      $query = 'SELECT * FROM products
6          WHERE product_id = '.$product_id;
7      $result = $db -> query($query);
8      $row = $result -> fetch_assoc();
9      return $row;
10 }

```

(query_utils. php, line 4-10)

```

31  foreach ($SESSION['cart'] as $product){
32      $info = get_details_by_id($db, $product);
33      echo "<tr><td>";
34      echo '<div class="cart-row">';
35      echo '<input type="checkbox" name="selected[]" value="'.$product.'" id="cart-checkbox">';
36      echo '<div class="cart-img-container">';
37      echo '</div>';
38      echo '<div id="product_name_quan">';
39      echo '<p>'.$info['product_name'].'</p>';
40      echo '<div id="quan_field">';
41      echo '<label for="brand2">Quantity</label>';
42      echo '<input type="number" name="items['.$product.'])" min="1" value="1" max="'.$info['stock'].'">';
43      echo '</div></div>';
44      echo '<div id="price-field">';
45      echo '<p>'.$info['price'].'</p>';
46      echo '</div></div>';
47      echo "</td></tr>";
48  }

```

(/carts/index.php, line 31-48)

In the Shopping Cart, user can adjust the quantity of the added item using the input element with *type = 'number'*, or remove the selected items: *removeItemFromCart()*

```

1  function removeItemFromCart(){
2      var hiddenInput = document.getElementById("removed-placeholder");
3      hiddenInput.value = "true";
4      var form = document.getElementById("cart-form");
5      form.submit();
6  }

```

(/carts/remove_cart.js, line 1-6)

Restriction:

In the Shopping Cart, the adjustment to the quantity of the item should be within the range from 1 to the number of available stocks

Exception:

When product stock is 0, “Add to Cart” button will be disabled

```

10  function noStock(){
11      var cartButton = document.getElementById('cart-button');
12      var buyNowButton = document.getElementById('buy-button');
13      cartButton.disabled = true;
14      cartButton.style.backgroundColor = "grey";
15      buyNowButton.disabled = true;
16      buyNowButton.style.backgroundColor = "grey";
17  }
18

```

(/product_details/productdetails.js, line 10-17)

When the quantity range is invalid, either about to go below 1 or exceed the number of available stocks, the count field will stop the user from decreasing or increasing the number respectively

```
41     echo '<label for="brand2">Quantity</label>';
42     echo "<input type='number' name='items[{"$product}]" min='1' value='1' max='{$_info['stock']}'>";
```

(/carts/index.php, line 41-42)

4.3.4 Place an order

Proceed to Checkout:

User can select one or more items in the Shopping cart, adjust the quantities and click on the “Checkout” button to proceed to checkout. Alternatively, on the product details page, user can click on the “Buy Now” button and be directed to the checkout page to purchase one such item.

```
61     <form id="buyNow" action="../checkout/index.php" method="post">
62         <input type="hidden" id="buy-now-select-this" name="selected[]"
63             value="<?php echo $product_id; ?>">
64         <input type="hidden" id="buy-now-qty-one" name="items[<?php echo $product_id; ?>]"
65             value="1">
66         <button id="buy-button" onclick="buyNow.submit();">Buy Now</button>
67     </form>
```

(/product_details/index.php, line 61-67)

Checkout from shopping cart:

On loading the shopping cart page, the server will query from products table to get the stocks of items in the cart. Users can only choose quantities to checkout with minimum 1 and maximum the stock left.

There are two types of inputs in the form element on the cart page, checkbox for the item selection, and number for ordered quantity of products. Correspondingly, there are two session variables, `$_SESSION["items"]` for number inputs, which is an array stores `product_id => quantity pairs`, and `$_SESSION["selected"]` which captures checkbox inputs, as an array of `product_id`. Only the items that user selected will be proceed to checkout and generate order summary. If no item is selected, the user will be direct back to cart.

Check out directly from product details page:

`$_SESSION["items"]` stores only this `product_id => quantity as 1`.

`$_SESSION["selected"]` only includes this current item.

Checking out:

User is required to fill in the required delivery information, contacts and to select a payment method in order to place the order.

The items that are selected in the cart by the users will be displayed in a table summary view. The table is generated by iterate through all in `$_SESSION['selected']` and fetch the quantity from `$_SESSION['items']`.

By clicking on the “Place Order” button, the order is created, with an email sent to the user’s mailbox. When creating order, a new record is added to the orders table, with `order_id` as a random number up to 8 digits. New records of ordered items: `product_id`, `order_id` and quantity are added to `order_items` table. The quantities of ordered items are deducted from the corresponding records in products table. Finally, a confirmation email is sent and the user will be redirected to the order status page, with a message indicating a successful order placed.

```
10 //Step 1: add order, items into order_items table
11 foreach($_SESSION["order_item"] as $product_id => $qty){
12     $order_item_query = "INSERT INTO order_items (order_id, product_id, qty) VALUES ({$SESSION['order_id']}, {$product_id}, {$qty})";
13     $order_item_table_res = $db->query($order_item_query);
14 }
15
16 //Step 2: add order details into orders table
17 $order_query = "INSERT INTO orders (order_id, user_id, status, delivery_address_line_1, ";
18 $order_query.= "delivery_address_line_2, zip_code, receiver_name, receiver_contact, payment_method) ";
19 $order_query.= "VALUES ({$SESSION['order_id']}, {$SESSION['user_id']}, ";
20 $order_query.= "1, '{$POST['delivery_address_line_1']}', '{$POST['delivery_address_line_2']}', ";
21 $order_query.= "'{$POST['zip_code']}', '{$POST['receiver_name']}', '{$POST['receiver_contact']}', ";
22 $order_query.= "'{$POST['payment_method']}')";
23
24 $orders_table_res = $db -> query($order_query);
25
26 //Step 3: reduce current stock
27 foreach($_SESSION["order_item"] as $product_id => $qty){
28     $current_stock_query = "SELECT stock FROM products WHERE product_id={$product_id}";
29     $current_stock_res = $db->query($current_stock_query);
30     $current_stock_row = $current_stock_res->fetch_row();
31     $new_stock = $current_stock_row[0] - $qty;
32     $update_stock_query = "UPDATE products SET stock={$new_stock} WHERE product_id={$product_id}";
33     $db->query($update_stock_query);
34 }
```

(/Checkout/index.php, line 10-34)

Restriction: all the input fields should be filled up, and a payment method must be selected. Contact number should be in 8-digit, and postal code should be in 6-digit.

```

8  ✓ function validateContactNumber(){
9      let contact = contactField.value;
10     let alert = document.getElementById("invalidContactAlert");
11     ✓ if (contact.search(/^\d{8}$/) !== 0){
12         alert.className = "alert";
13         alert.innerHTML = "A valid phone number contains 8 digits.";
14         checkoutButton.disabled = true;
15     ✓ } else {
16         alert.className = "";
17         alert.innerHTML = "";
18         checkoutButton.disabled = false;
19     }
20 }
21
22 ✓ function validateZipCode(){
23     let zip_code = zipCodeField.value;
24     let alert = document.getElementById("invalidZipCodeAlert");
25     ✓ if (zip_code.search(/^\d{6}$/) !== 0){
26         alert.className = "alert";
27         alert.innerHTML = "A valid Zip Code contains 6 digits.";
28         checkoutButton.disabled = true;
29     ✓ } else {
30         alert.className = "";
31         alert.innerHTML = "";
32         checkoutButton.disabled = false;
33     }
34 }
35
36 checkPaymentMethodSelected();
37 let radios = document.forms["checkout"].elements["payment_method"];
38 ✓ for (var radio of radios){
39     radio.addEventListener("change", checkPaymentMethodSelected);
40 }
41
42
43 ✓ function checkPaymentMethodSelected(){
44     let radios = document.forms["checkout"].elements["payment_method"];
45     let selectedPayment = false
46     ✓ for (var radio of radios){
47         selectedPayment ||= radio.checked;
48     }
49     let alert = document.getElementById("noPaymentAlert");
50     ✓ if (selectedPayment){
51         alert.className = "";
52         alert.innerHTML = "";
53     ✓ } else {
54         alert.className = "alert";
55         alert.innerHTML = "Please Select Payment Method.";

```

(/Checkout/checkout-validate.js, line 8-55)

4.3.5 Orders and Order Status

User can check historical orders with the respective summary on the "My Order" page. The page can be accessed from the navigation bar -> account icon -> left panel "orders" link.

A list of historical orders is generated by query:

```
SELECT * FROM orders WHERE user_id = {$curr_user} ORDER BY create_time
DESC
```

And the product details and items in orders are generated by query:

```
SELECT order_items.order_id, order_items.product_id, order_items.qty,
products.product_name, products.price
FROM order_items LEFT JOIN products
ON order_items.product_id = products.product_id
WHERE order_items.order_id = {$order_id}
```

For each order, if the user click on the “View Order Details” link, the order status and details order information will be displayed. The status is shown as a status bar and can be updated once the “status” attribute of the “orders” table is changed (the values of “status” attribute are from 1 to 4, with respect to paid, confirmed, shipped and received states on the status bar).

```
174 <div class="status-bar-container">
175 <?php
176 $curr_status = (int) $order_details['status'];
177 $all_status = select_all_from_table($db, 'status');
178 for ($node = 0; $node < count($all_status); $node++){
179     echo "<div class='phase'>";
180     if ($node < $curr_status){
181         echo "<div class='status-dot done'></div>";
182         if ($node != 0){
183             echo "<div class='status-line solid'></div>";
184         }
185     } else {
186         echo "<div class='status-dot'></div>";
187         echo "<div class='status-line'></div>";
188     }
189     echo "<div class='status-text'>{$all_status[$node]['status_name']}</div></div>";
190 }
191 ?>
192 </div>
```

(/order_details/index.php, line 175-192)

4.3.6 Manage account

General:

On the “Account” page, user’s account information is displayed. Users can logout by clicking on the “Log out” button, and update the account information using the “Update” button.

Update Information allows users to modify the name and contact, but not the email address since it is unique to individual account. Clicking on the “Save” button to update.

Restriction:

Any of the fields in the Update form should not be empty.

The contact field should be an 8-digit input.

Exception:

If any of the fields are left blank, or contact number is invalid, 'Save' button is disabled.

```
13 var contactField = document.getElementById("contact-update");
14 var confirmButton = document.getElementById("confirm-update-button");
15 contactField.addEventListener("change", validateContactNumber);
16
17 function validateContactNumber(){
18     let contact = contactField.value;
19     let alert = document.getElementById("invalidContactAlert");
20     if (contact.search(/^\d{8}$/) !== 0){
21         alert.className = "alert";
22         alert.innerHTML = "A valid phone number contains 8 digits.";
23         confirmButton.disabled = true;
24     } else {
25         alert.className = "";
26         alert.innerHTML = "";
27         confirmButton.disabled = false;
28     }
29 }
```

(/account/account_update.js, line 13-29)

5. Testing of Web Application

5.1 Test Plan

Testing of web application SoundX includes 2 main categories of tests:

- Components and pages test. A list of test cases is generated, with steps and respective expected outcome. Next, the test will be conducted against each case by following the step instruction. The actual test results will be logged to compare against the expected outcome. The test would pass if the actual results matched the expectation; otherwise, rectify the code and retest.
- Use case simulation test is designed corresponding to the requirements of the application, which is defined in section 2. After conducting the test, pass/fail is indicated.

5.2 Testing Components and Pages

Test Suite 1: Sign Up				
#	Description	Steps	Expected Result	Actual Result
1.1	Password and Confirm Password mismatch	Key in different values for “password” and “confirm password”	A warning message should appear to indicate the mismatch. The signup button is not able to click.	A warning message appears above the signup button. The button is unclickable until the user enters matching password
1.2	Missing Required fields	Leave any field empty and click sign up button	An alert message will require the user to put in the required values. Signup button is disabled. The cursor will be focused on the missing fields.	A warning message appears above the signup button. The button is unclickable until the user enters the matching password. The cursor is focused on the first missing fields.
1.3	Invalid ‘First Name’, ‘Last Name’ fields	Enter non-alphabetical characters	A message will require user to key in alphabetical characters. Signup button is disabled.	Message shown up.
1.4	Invalid “email” field	Enter an email without “@” symbol	A message will require user to key in correct email. Sign up button is not able to click.	Alert message prompts when user click on other fields or on the signup button: “Please enter a valid email.”

1.5	Invalid "contact" field	Enter a contact number that is not 8-digit numbers	A message will require user to key in correct contact number. Sign up button is not able to click.	Alert message prompts when user click on other fields or on the signup button: "A valid phone number contains 8 digits."
1.6	Successful Sign Up Action	Fill in all the fields, with "password" and "confirm password" the same value. Click Sign Up Button.	Client Side: A message will show that sign up is successful. User will be redirected to login page. Server Side: new row is inserted into users table with the values just entered.	A message is showed at the bottom of signup field, indicating that sign up is successful. User is redirected to login page after 3 seconds.

Test Suite 2: Login

#	Description	Steps	Expected Result	Actual Result
2.1	The email provided is not matched to any in the database.	Enter an email that is not registered before (I.e. not an entry in the <i>users</i> table in the database.) and a password.	A warning message should appear. The login button is not able to click.	A warning message is shown below the login button: "Input email or password is not correct."
2.2	Wrong password	Enter an email that exists in users table, but together with a mismatched password.	A warning message should appear. The login button is not able to click.	A warning message is shown below the login button: "Input email or password is not correct."
2.3	Missing Required fields	Leave any field empty and click login button	A pop-out alert will request the user to put in required values	A pop-out alert appears at the missing field to prompt the user to enter values in the required fields. The cursor automatically focuses on the first missing field.
2.4	Successful Login	Enter an email and password pair existing in users table. Click login button.	The user will be directed to the <i>Home</i> page.	User is directed to the <i>Home</i> page.

Test Suite 3: Account

#	Description	Steps	Expected Result	Actual Result
3.1	Display correct user information	After login, click the account icon on the top right corner at the navigation bar.	User will be directed to the account page. The information on the account page is matching with <i>Users</i> table.	The account information, including Name, Email and Contact, is matched to user's signup input.
3.2	Update user information – valid name and contact inputs	After login, click the account icon on navigation bar. Then click update button. Update the information with valid name (non-empty) and valid contact (8 digits). Click 'save' button.	<u>Client side:</u> User will be able to update information except email address. After updating, user click on the "update" button to synchronize the input data to the <i>Users</i> table. <u>Server side:</u> <i>Users</i> table is updated corresponding to the changed fields	User can modify first and last name, as well as the contact No. Email field un-editable. Users table is subsequently updated.
3.3	Update user information – invalid name or contact inputs	After login, click the account icon on navigation bar. Then click update button. Update the information with invalid name (empty) or invalid contact (not 8-digit number). Click 'save' button.	A message will be displayed to indicate the correct form of input. Save button is unclickable.	Warning message are prompted
3.4	Update user information – empty inputs	After login, click the account icon on navigation bar. Then click 'update' button. Remove the values of any entry in the update form. Click on the 'save' button	A message will prompt the user that no empty field is allowed.	Warning message are prompted
3.5	Log out	After login, click the account icon on navigation bar.	User will be redirected to <i>Login</i> page.	User is redirected to the <i>Login</i> page

		Then click the log out button.		
--	--	--------------------------------	--	--

Test Suite 4: Product details				
#	Description	Steps	Expected Result	Actual Result
4.1	<i>Product Details</i> page loaded from the banner in the <i>Home</i> page	After login, click on the banner on the home page.	User will be directed to the respective product details page. Content of the product details is loaded successfully.	User access the respective product details. Content of the product including images, description and specification are loaded successfully.
4.2	Toggle product pictures	On product details page, click the image snapshot on the left.	The large product picture will display the thumbnail image selected.	The large picture display the thumbnail image selected. The width of the large image is fixed.
4.3.1	Add to cart	On product details page, click the “Add to cart” button.	“Add to cart” button text will change to “View in cart”	Once the “Add to cart” button is clicked, the text will be changed to “View in cart”
4.3.2	View in cart	After steps in testcase 4.3.1, Click the “view in cart” button.	User will be directed to <i>shopping cart</i> page. The product added just now will be displayed as the first item in the <i>Cart</i> page.	User is directed to <i>shopping cart</i> and can see the product added just now shown as the first item in the cart.
4.4	Stock count	On product details page, check the stock left.	The stock left should be the same as “stock” column in products table.	The current stock is shown.
4.5	No Stock	On product details page, check the stock left. Update the “stock” column value for this product to 0 in products table. Refresh the page.	After update the stock to 0 and refresh, the page will shows 0 stock left. Both add to cart and buy now button are disable and displayed in grey color.	When the stock is 0, both ‘add to cart’ and ‘buy now’ buttons are disabled and in grey color

4.6	Buy Now	On product details page, click buy now button.	User will be sent to <i>Checkout</i> page. The product will be displayed in order summary, with quantity of 1.	User is directed to <i>Checkout</i> page, with the selected product displayed in the order summary with quantity 1.
4.7	Return to previous page	Click on the 'Back' button	User will return to the previous page	User is able to return the previous page they accessed

Test Suite 5: Product List

#	Description	Steps	Expected Result	Actual Result
5.1	Product list loaded	After login, click the 'Product' button on the navigation bar.	User will be directed to the <i>Product List</i> page. A list of all products is loaded successfully.	User can see a list of products displayed
5.2	Product List Filter 1: one brand selected	Select one brand and Click 'Apply Filters'.	The page is refreshed. Products in selected brand are displayed.	The page is refreshed. Products from one single selected brand are displayed.
5.3	Product List Filter 1: multiple brands selected	Select multiple brands and Click 'Apply Filters'.	The page is refreshed. Products in selected brand are displayed.	The page is refreshed. Products from multiple selected brands are displayed.
5.4	Product List Filter 2: one type selected	Select one type and Click 'Apply Filters'.	The page is refreshed. Products in selected type are displayed.	The page is refreshed. Products from one single selected type are displayed.
5.5	Product List Filter 2: multiple types selected	Select multiple types and Click 'Apply Filters'.	The page is refreshed. Products in selected types are displayed.	The page is refreshed. Products from multiple selected types are displayed.
5.6	Product List Filter 3: valid price range	Enter a valid price range and click 'Apply Filters'.	The page is refreshed. Products in selected price range are displayed.	The page is refreshed. All products from the indicated price range are displayed.
5.7	Product List Filter 3: invalid price range	Enter a price range that is not a number. Click on the 'Apply Filters' button.	Values of price range will be reset to default: min 0 , max 1000.	When click on the screen, an alert message is displayed:" Not a valid number

				input, please try again!”
5.8	Product List Filters: Mix and Match	Apply both brand and type filters. Apply both brand and price filters. Apply both type and price filters. Apply all the three filters.	The page is refreshed. Products filtered are displayed successfully. Manually check with SQL for products table, the filtered result is the same as query result.	The page is refreshed. The selected filters and filtered products are displayed successfully.
5.9	Clear all the filters	Uncheck all the filters and click apply filters.	The page is refreshed. A list of all products is loaded successfully.	The page is refreshed. All products are loaded successfully.
5.10	No Matching Product available	Apply both brand and type filters. Apply both brand and price filters. Apply both type and price filters. Apply all the three filters.	The page is refreshed. The applied filters are displayed as requested. Since there is no matching attribute values in the product table, the page will display a line to inform user there is no available product for the filtering.	The page is refreshed. The applied filters are displayed as requested. An image of ‘No result found’ is displayed.

Test Suite 6: Shopping Cart

#	Description	Steps	Expected Result	Actual Result
6.1	Empty Shopping cart	After login, click the cart icon on the navigation bar.	User will be redirected to shopping cart page. No item instead of a message indicating no item in cart displayed. Click on the link “Browse Products” in message will redirect user to product list page.	Showing message” No Items In cart. Please keep shopping.”. No itmes in the shopping cart

6.2	Non-empty shopping cart	Add several products into cart with step in testcase 4.3.1. Click the cart icon on the navigation bar.	User will be redirected to shopping cart page. Items added will be displayed in the page.	All the products added are shown in chronological order
6.3	Delete one Item from shopping cart	On shopping cart page, click on the checkbox before the item that wished to remove. Then click the 'Remove' Button.	The page is refreshed. Item selected is removed	The page is refreshed. Item selected is removed
6.4	Delete multiple Item from shopping cart	On shopping cart page, select more than one items and click on the 'Remove' Button.	The page is refreshed. Items selected are removed	The page is refreshed. Items selected are removed
6.5	Delete without selecting any items	On the shopping cart page, click Remove Button.	The page is refreshed. Nothing changed.	The page is refreshed. Nothing changed.
6.6	Checkout without selecting any items	On shopping cart page, click Checkout Button.	The page is refreshed. Nothing changed.	The page is refreshed. Nothing changed.
6.7	Checkout one item	On shopping cart page, select one item, click Checkout Button.	User will be redirected to checkout page. Item selected will be displayed in the order summary.	User will be directed to <i>Checkout</i> page. Item selected is displayed in the order summary.
6.8	Checkout multiple items	On shopping cart page, select one item, click Checkout Button.	User will be redirected to checkout page. Items added will be displayed in the order summary.	User will be directed to <i>Checkout</i> page. All the item selected are displayed in the order summary.
6.9	Stock check – maximum	On shopping cart page, increase the quantity to the number of stocks in products table.	The quantity should not be able to increase any more.	The quantity cannot be increased any more when the quantity reaches the stock limit.
6.10	Stock check – minimum	On shopping cart page, decrease the quantity 1	The quantity should not be able to decrease once the quantity is 1	The quantity cannot decrease any more once the quantity reaches 1

6.11	Modify product quantity	For each item in the shopping cart, click on the quantity field to increase or decrease the quantity.	The quantity is able to be changed within the available range, I.e. between 1 and the available stock.	The quantity is able to be modified within the permissible range
6.12	Sold out after product added to cart	Add products to cart. Update database products table, set stock column of product in cart to 0. Select item with 0 stock and Click checkout.		

Test Suite 7: Checkout				
#	Description	Steps	Expected Result	Actual Result
7.1	Missing inputs for required input field	On checkout Page, leave any field empty except address line 2. Click 'Place Order' Button.	Users will not be able to place order if any fields missing. A pop-out will be shown at the first missing field. The "Place Order" button is disabled.	User cannot click on the place order. A pop-out is shown at the first missing field. The "Place Order" button is disabled.
7.2	Invalid contact number	On checkout Page, enter a contact number that is not 8-digit number.	A warning message will be displayed indicating invalid number.	A warning message is displayed: "A valid phone number contains 8 digits."
7.3	Invalid postal code	On checkout Page, enter a postal code that is not 6-digit number. (Singapore Standard)	A warning message will be displayed indicating invalid postal code.	A warning message is displayed: "A valid postal code contains 6 digits."
7.4	Missing payment methods	On checkout Page, filling up all the required field but do not select any payment method. Click 'Place Order' Button.	A warning message will be displayed to ask user to select one payment method. The "Place Order" button is disabled.	Alert message is always shown. The "Place Order" button is disabled.
7.5	Successful Checkout	On checkout page, fill all the fields with contact number 8-digit and postal code 6-digit.	<u>Client side:</u> User will be redirected to order details page showing the order just created.	Once the order is placed, the page is directed to <i>order details</i> page, showing the order details

		Select one payment method. Click on the 'Checkout' button.	A message shows that the order is successfully created. User should receive an email informing the order is placed <u>Server side:</u> new order entry added in <i>orders</i> table in the database. New entries added in <i>order_items</i> table.	including order status, order items summary and delivery information. A message shows that the order is successfully created. And user can receive an email informing the order is placed successfully. Server side: tables updated accordingly
--	--	---	--	--

Test Suite 8: Orders

#	Description	Steps	Expected Result	Actual Result
8.1	All orders and the respective details including order ID, create date, ordered items summary and amount of payment displayed in descending date order	After login, click the account icon on the navigation bar, then click orders on the left panel.	A list of historical orders and respective information will be displayed. Content of order list is the same as database records.	A list of historical orders and respective information are displayed in the descending orders.
8.2	Switch to account	On orders page, click "account" on the left panel.	User will be redirected to account panel.	User is redirected to account panel.
8.3	View order details	On orders page, click "view details" link.	User will be redirected to order details page.	User is redirected to order details page.
8.4	No establishing order	Once user who does not have any orders accesses the order page	On order page, the right panel is empty with a message informing user there is no existing order	Showing message: "You do not have any order at this time."

Test Suite 9: Order details

#	Description	Steps	Expected Result	Actual Result
9.1	Order details displayed correctly, including order id, create date, ordered items summary and delivery information	Repeat steps in testcase 8.3.	User will be redirected to order details page. Information displayed on the page is the same as database records.	The details order information including order ID, create date, item summary and delivery information are displayed.
9.2	Order status update	Go to the order details page of one order. In database orders table, update the status column of this order by changing the value within 1-4. Refresh the page.	Status of the order will be updated and shown in the top status bar. For the status is done, the status dot will be in green colour; for the pending status, the colour will be in grey.	Status bar is updated. For the status is done, the status nodes are in green colour; for the pending status, the colour are grey.
9.3	Return to order page	Click on the return button on the <i>Order Details</i> page	User is directed back to the <i>Order</i> page	User is able to go back to the <i>Order</i> page

Test Suite 10: Navigation Bar and Home Page

#	Description	Steps	Expected Result	Actual Result
10.1	Access different product brands from the navigation bar	Hover on the 'Brand' section on the navigation bar and click on the brand icon	Hover on the 'Brand' section will show a dropdown section, containing a list of brand icons. By clicking on one of the icon, users will see all product listings for the selected brand.	Hover on the 'Brand' and click on the brand icon can view all product of the selected brand.
10.2	Access different product types from the navigation bar	Hover on the 'Type' section on the navigation bar and click on the type icon	Hover on the 'Brand' section will show a dropdown section, containing a list of type icons. By clicking on one of the icon, users will see all	Hover on the 'Type' and click on the brand icon can view all product of the selected type.

			product listings for the selected type.	
10.3	Access product list from navigation bar	Click the 'Product' section on the navigation bar	View all products offered.	Product filters and all product are displayed
	Jump to contact footer	Click the 'Contact' section on the navigation bar	Directed to the footer section	Directed to the footer at the bottom.
	Access 'Account' from navigation bar	Click the 'account' icon on the navigation bar	Directed to the account page	Access the account page
	Access 'Cart' from navigation bar	Click the 'cart' icon on the navigation bar	Directed to the shopping cart	Access the cart. The cart is empty since no item has been added in
	Return to home page	Click the 'SoundX' logo	Refresh the home page	Refresh and go to the top of the home page
	Browse through and go to corresponding product detail page through the slideshow banner	Click the left and right arrows on the banner browse slideshow; click on the banner to access the product page	Switch to next/previous slides. After clicking on the banner, users will be directed to the respective product detail page	View the next/previous slides. After clicking on the banner, access the respective product details page
	Access respective product details page from the 'Top Sale' and banner	Click on the product	users will be directed to the respective product detail page	Access the respective product details page
	Access different product type from the 'See More'	Click on the type icon	users will be directed to the product list page where shows all products of the selected type	Display all products of the selected type

5.3 Testing Use Cases

Test Suite 11: Product Browsing		
#	Description	Pass/Fail
11.1	Browsing all the products	Pass
11.2.1	Browsing products by brands (access from product page)	Pass
11.2.2	Browsing products by brands (access from navigation bar)	Pass
11.3.1	Browsing products by types (access from product page)	Pass
11.3.2	Browsing products by types (access from navigation bar)	Pass
11.3.3	Browsing products by types (access from home page)	Pass

11.4	Browsing products by price range (access from product page)	Pass
11.5	Product filters mix and match	Pass
11.6.1	View product details (access from product page)	Pass
11.6.2	View product details (access from home page)	Pass
11.6.3	View product details (access from account/orders page)	Pass
11.6.4	View product details (access from order details page)	Pass

Test Suite 12: Product Purchase		
#	Description	Pass/Fail
12.1	Add product to the cart	Pass
12.2	Delete items from the cart	Pass
12.3	View the cart (access from product detail page)	Pass
12.4	Check out from the cart (one product)	Pass
12.5	Check out from the cart (multiple products)	Pass
12.6	View order created (after checkout; order details page)	Pass
12.7	Confirmation email received	Pass

Test Suite 13: Sign up and login		
#	Description	Pass/Fail
13.1	Sign up an account	Pass
13.2	Login with previously created account	Pass
13.3	Sign up a new account and login with this account	Pass
13.4	Sign out from account page	Pass

Test Suite 14: Manage account and order		
#	Description	Pass/Fail
14.1	Update account information	Pass
14.2	View all historical orders	Pass
14.3	View individual order details	Pass

6. Conclusion

In the project, a functional online shopping portal is developed. All the designed functional requirements and specification are implemented, where users can freely explore products, make purchases, and check personal and order information. Both client side (PHP) and server side (JavaScript) functions, as well as the database are well-integrated, enabling accurate validation and handling of events, extraction and update of data. Session is created properly for each user's account and login, enabling the users to check their history order.

The design follows the general design guidelines to achieve high usability. The webpage is well-structured with scalable layout, with clear definition of each section and navigation flow. CSS stylesheet is applied for each page, ensuring the consistency in styles and visual identity.

There is no deferral in progress at the end of the project. If to further stimulate a holistic online shopping experience for the users, some other features can be considered in the future: comparison of products, search functions, email subscriptions, etc.

Appendices

Appendix A: Contributions

Task	Subtask	Assignee
Requirements Definition	Application Requirements	Lin Sining
	Functional Requirements	Lin Sining
Application Design	Sitemap	Lin Sining, Wei Haoran
	Wireframe	Wei Haoran
	Story board	Wei Haoran
	Testing Plan	Lin Sining
Implementation	Database design	Wei Haoran
	Product data collection	Lin Sining (product selection, pictures) Wei Haoran (description and specification)
	Page templates, layout (HTML)	Lin Sining
	Styling (CSS)	Lin Sining
	Server-side functions (PHP)	Wei Haoran
	Client-side functions (JavaScript)	Wei Haoran
Testing	Testcase design	Wei Haoran, Lin Sining
	Manual test	Lin Sining
Report writing	Summary & Conclusion	Lin Sining
	Code implementation	Wei Haoran, Lin Sining
	Proofreading	Lin Sining
	Report Styling	Wei Haoran

Appendix B: Source Code

Structure of this project folder:

```
/
|
| db_conn.php
| index.php
| query_utils.php
|
|-----account
|     account_update.js
|     index.php
|     logout.php
|     update_account_info.php
|
|-----cart
|     index.php
|     remove_cart.js
|
|-----checkout
|     checkout_validate.js
|     create_order.php
|     index.php
|
|-----css
|     account-order.css
|     cart.css
|     check-out.css
|     elements.css
|     home-style.css
|     login-style.css
|     nav-bar-style.css
|     order-details.css
|     order-item.css
|     product-list-style.css
|     product_detail.css
|
|-----Elements
|     footer.php
|     nav_bar.php
|
|-----home
|     index.php
|     script.js
```

```

|—img
|
|—login
|   index.php
|
|—orders
|   index.php
|
|—order_details
|   index.php
|   item_in_order.php
|   send_email.php
|
|—product_details
|   add_to_cart.php
|   index.php
|   productdetails.js
|
|—product_list
|   index.php
|   product_list.js
|
|—sign_up
|   index.php
|   signup.js
|
|—sql
|   tables_import.sql

```

/db_conn.php

```

<?php
    $server_name = 'localhost';
    $username = 'f32ee';
    $password = 'f32ee';
    $database = 'f32ee';
    $db = new mysqli($server_name, $username, $password, $database);
    if ($db->connect_errno) {
        echo 'Error: Could not connect to database. Please try again later.';
        exit;
    }
?>

```

/index.php

```

<?php
    header("Location: http://192.168.56.2/f32ee/EE4717-SoundX/home/");

```



```
    exit();  
?>
```

/query_utils.php

```
<?php
```

```
function get_details_by_id($db, $product_id){  
    $query = 'SELECT * FROM products  
            WHERE product_id = '.$product_id;  
    $result = $db -> query($query);  
    $row = $result -> fetch_assoc();  
    return $row;  
}
```

```
function get_product_list_by_query($db, $query){  
    $result = $db -> query($query);  
    $result_arr = array();  
    while ($row = $result -> fetch_assoc()){  
        array_push($result_arr, $row);  
    }  
    return $result_arr;  
}
```

```
function get_order_details_by_order_id($db, $order_id){  
    $query = 'SELECT * FROM orders  
            WHERE order_id = '.$order_id;  
    $result = $db -> query($query);  
    $row = $result -> fetch_assoc();  
    return $row;  
}
```

```
function get_order_items_by_order_id($db, $order_id){  
    $query = 'SELECT * FROM order_items  
            WHERE order_id = '.$order_id;  
    $result = $db -> query($query);  
    $items = array();  
    while ($row = $result -> fetch_assoc()){  
        $items[] = $row;  
    }  
    return $items;  
}
```

```
function get_order_items_with_product_info_by_order_id($db, $order_id){
```

```

    $query = "SELECT order_items.order_id, order_items.product_id,
order_items.qty, products.product_name, products.price";
    $query.= " FROM order_items LEFT JOIN products ON order_items.product_id =
products.product_id";
    $query.= " WHERE order_items.order_id ={$order_id}";
    $result = $db -> query($query);
    while ($row = $result -> fetch_assoc()){
        $items[] = $row;
    }
    return $items;
}

```

```

function select_all_from_table($db, $table_name){
    $query = "SELECT * FROM {$table_name} WHERE 1";
    $result = $db -> query($query);
    $rows = array();
    while ($row = $result -> fetch_assoc()){
        $rows[] = $row;
    }
    return $rows;
}

```

?>

/account/account_update.js

```

function displayUpdateContent(){
    document.getElementById("update-button").hidden = true;
    document.getElementById("name-display").hidden = true;
    document.getElementById("first-name-update").hidden = false;
    document.getElementById("last-name-update").hidden = false;
    document.getElementById("contact-display").hidden = true;
    document.getElementById("contact-update").hidden = false;
    document.getElementById("confirm-update-button").type = "submit";
    document.getElementById("confirm-update-button").value = "Save";
    document.getElementById("logout-button").type = "hidden";
}

```

```

var contactField = document.getElementById("contact-update");
var confirmButton = document.getElementById("confirm-update-button");
contactField.addEventListener("change", validateContactNumber);

```

```

function validateContactNumber(){
    let contact = contactField.value;
}

```

```

let alert = document.getElementById("invalidContactAlert");
if (contact.search(/^\\d{8}$/) !== 0){
    alert.className = "alert";
    alert.innerHTML = "A valid phone number contains 8 digits.";
    confirmButton.disabled = true;
} else {
    alert.className = "";
    alert.innerHTML = "";
    confirmButton.disabled = false;
}
}

/account/index.php
<?php include_once '../db_conn.php';
    session_start();

    if (!isset($_SESSION['user_id'])){
        //If not login, head in to login page
        header("Location: http://192.168.56.2/f32ee/EE4717-SoundX/login/");
        exit();
    }

    $account_query = "SELECT * FROM users WHERE
user_id={$_SESSION['user_id']}";
    $user_result = $db -> query($account_query);
    $user_info = $user_result -> fetch_assoc();

    $name = ucwords($user_info['first_name']).'
'.ucwords($user_info['last_name']);
    $lastname = ucwords($user_info['last_name']);
    $firstname = ucwords($user_info['first_name']);
    $email = strtolower($user_info['email']);
    $contact = strtolower($user_info['contact']);
?>
<!DOCTYPE html>
<html lang="en">

<head>
    <title>Account</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="../css/account-order.css" />
    <link rel="stylesheet" href="../css/elements.css" />
</head>

```

```

<body>
  <?php include '../Elements/nav_bar.php';?>
  <div class="main-content">
    <div class="left-nav-column">
      <div id="account">
        <a href="index.php">
          <span>ACCOUNT</span>
        </a>
      </div>
      <div id="order">
        <a href=" ../orders/index.php">
          <span>MY ORDER</span>
        </a>
      </div>
    </div>
    <div class="right-nav-column">
      <div id="user-info-text">
        <button id="update-button"
onlick="displayUpdateContent();">Update</button>
        <form id="account-update" action="update_account_info.php"
method="post">
          <h2>Name</h2>
          <p id="name-display">
            <?php echo $name; ?>
          </p>
          <input required type="text" hidden id="first-name-update"
name="first_name"
          value="<?php echo $firstname;?>"
onClicK="this.select();">
          <input required type="text" hidden id="last-name-update"
name="last_name"
          value="<?php echo $lastname;?>"
onClicK="this.select();">
          <hr>
          <h2>Email</h2>
          <p id="email-display">
            <?php echo $email; ?>
          </p>
          <hr>
          <h2>Contact</h2>
          <p id="contact-display">
            <?php echo $contact; ?>
          </p>

```

```

        <input required type="text" hidden id="contact-update"
name="contact" value="<?php echo $contact;?>"
        onClick="this.select();">

        <hr>
        <div id="invalidContactAlert"></div>
        <input id="confirm-update-button" type='hidden'
value='Update'>
    </form>
    <form action='logout.php'>
        <input id="logout-button" type='submit' value='Log Out'>
    </form>
</div>
</div>
<?php include '../Elements/footer.php';?>
<script src="account_update.js"></script>
</body>

</html>

```

```
/account/logout.php
```

```

<?php
    session_start();
    // remove all session variables
    session_unset();

    // destroy the session
    session_destroy();

    header("Location: http://192.168.56.2/f32ee/EE4717-SoundX/login/");
?>

```

```
/account/update_account_info.php
```

```

<?php
    include_once '../db_conn.php';
    session_start();

    if (!isset($_POST["contact"])){
        header("Location: http://192.168.56.2/f32ee/EE4717-SoundX/account/");
    }

```

```

$update_account_info_query = "UPDATE users SET ";
foreach($_POST as $column => $new_value){
    $update_account_info_query.= $column."='".$new_value."'";
}

//remove the last comma
$update_account_info_query = substr($update_account_info_query, 0, -1);
$update_account_info_query.= " WHERE user_id = {$_SESSION['user_id']}";

$db->query($update_account_info_query);

header("Location: http://192.168.56.2/f32ee/EE4717-SoundX/account/");
?>

/cart/index.php
<?php
    include_once '../db_connect.php';
    include_once '../query_utils.php';
    session_start();
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Cart</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="../css/cart.css"/>
    <link rel="stylesheet" href="../css/elements.css"/>
</head>

<body>
    <?php include '../Elements/nav_bar.php';?>
    <div class="item-cart-title">
        <h1>MY CART</h1>
    </div>
    <?php
    if (count($_SESSION['cart'])==0){
        echo '<div class = "cart-container">';
        echo '<div id="empty_cart"><p>No Items In cart, Please keep
shopping.</p></div>';
        echo '</div>';
    } else {
        echo '<div class = "cart-container">';
        echo '<form id="cart-form" action="../checkout/" method="post">';

```

```

        echo "<input type='hidden' name='remove' value='false' id='removed-
placeholder'>";
        echo '<button id="remove-button"
onclick="removeItemFromCart();">Remove</button>';
        echo '<table class="cart-table">';
        foreach ($_SESSION['cart'] as $product){
            $info = get_details_by_id($db, $product);
            echo "<tr><td>";
            echo '<div class="cart-row">';
            echo "<input type='checkbox' name='selected[]' value='{$product}'
id='cart-checkbox'>";
            echo "<div class='cart-img-container'>";
            echo "<a href='../product_details/?product_id=$product'>";
            echo "<img src='../img/product-snapshot/{$product}.png' id='cart-
img'></a></div>";
            echo '<div id="product_name_quan">';
            echo '<p>'.$info['product_name'].'</p>';
            echo '<div id="quan_field">';
            echo '<label for="brand2">Quantity</label>';
            echo "<input type='number' name='items[{$product}]' min='1'
value='1' max='{$info['stock']}'>";
            echo '</div></div>';
            echo '<div id="price_field">';
            echo '<p>'.$info['price'].'</p>';
            echo '</div></div>';
            echo "</td></tr>";
        }
        echo "</table>";
        echo '<input type="submit" value="Checkout">';
        echo '</form>';
        echo '</div>';
    }
    ?>
    <?php include '../Elements/footer.php';?>
    <script src="remove_cart.js"></script>
</body>

</html>

```

```

/cart/remove_cart.js

```

```

function removeItemFromCart(){
    var hiddenInput = document.getElementById("removed-placeholder");

```

```

        hiddenInput.value = "true";
        var form = document.getElementById("cart-form");
        form.submit();
    }

/checkout/checkout_validate.js

var checkoutButton = document.getElementById("checkout-button");
var zipCodeField = document.getElementById("zip_code");
var contactField = document.getElementById("receiver_contact");

contactField.addEventListener("change", validateContactNumber);
zipCodeField.addEventListener("change", validateZipCode);

function validateContactNumber(){
    let contact = contactField.value;
    let alert = document.getElementById("invalidContactAlert");
    if (contact.search(/^\d{8}$/) !== 0){
        alert.className = "alert";
        alert.innerHTML = "A valid phone number contains 8 digits.";
        checkoutButton.disabled = true;
    } else {
        alert.className = "";
        alert.innerHTML = "";
        checkoutButton.disabled = false;
    }
}

function validateZipCode(){
    let zip_code = zipCodeField.value;
    let alert = document.getElementById("invalidZipCodeAlert");
    if (zip_code.search(/^\d{6}$/) !== 0){
        alert.className = "alert";
        alert.innerHTML = "A valid Zip Code contains 6 digits.";
        checkoutButton.disabled = true;
    } else {
        alert.className = "";
        alert.innerHTML = "";
        checkoutButton.disabled = false;
    }
}

checkPaymentMethodSelected();
let radios = document.forms["checkout"].elements["payment_method"];

```



```

for (var radio of radios){
    radio.addEventListener("change", checkPaymentMethodSelected);
}

function checkPaymentMethodSelected(){
    let radios = document.forms["checkout"].elements["payment_method"];
    let selectedPayment = false
    for (var radio of radios){
        selectedPayment ||= radio.checked;
    }
    let alert = document.getElementById("noPaymentAlert");
    if (selectedPayment){
        alert.className = "";
        alert.innerHTML= "";
    } else {
        alert.className = "alert";
        alert.innerHTML = "Please Select Payment Method.";
    }
}
}

```

/checkout/create_order.php

```

<?php

    session_start();
    include_once '../db_conn.php';
    include_once '../query_utils.php';

    //POST contains [delivery_address_line_1, delivery_address_line_2,
zip_code, receiver_name, receiver_contact, payment_method]
    //SESSION contains [order_id, order_item_query, user_id]

    //Step 1: add order, items into order_items table
    foreach($_SESSION["order_item"] as $product_id => $qty){
        $order_item_query = "INSERT INTO order_items (order_id, product_id,
qty) VALUES ({"$_SESSION['order_id']"}, {"$product_id"}, {"$qty})";
        $order_item_table_res = $db->query($order_item_query);
    }

    //Step 2: add order details into orders table
    $order_query = "INSERT INTO orders (order_id, user_id, status,
delivery_address_line_1, ";

```

```

$order_query.= "delivery_address_line_2, zip_code, receiver_name,
receiver_contact, payment_method) ";
$order_query.= "VALUES ({$_SESSION['order_id']}, {$_SESSION['user_id']},
";
$order_query.= "1, '{$_POST['delivery_address_line_1']}',
'${$_POST['delivery_address_line_2']}' , ";
$order_query.= "'${$_POST['zip_code']}', '${$_POST['receiver_name']}',
'${$_POST['receiver_contact']}' ,";
$order_query.= "'${$_POST['payment_method']}' )";

$orders_table_res = $db -> query($order_query);

//Step 3: reduce current stock
foreach($_SESSION["order_item"] as $product_id => $qty){
    $current_stock_query = "SELECT stock FROM products WHERE
product_id={$product_id}";
    $current_stock_res = $db->query($current_stock_query);
    $current_stock_row = $current_stock_res->fetch_row();
    $new_stock = $current_stock_row[0] - $qty;
    $update_stock_query = "UPDATE products SET stock={$new_stock} WHERE
product_id={$product_id}";
    $db->query($update_stock_query);
}

$order_id = $_SESSION['order_id'];
$new_cart = array();
foreach($_SESSION["cart"] as $product_id){
    if (!array_key_exists($product_id, $_SESSION['order_item'])){
        $new_cart[] = $product_id;
    }
}

$_SESSION['cart'] = $new_cart;
unset($_SESSION['order_id']);
unset($_SESSION['order_item']);

header("Location: http://192.168.56.2/f32ee/EE4717-
SoundX/order_details/?id={$order_id}&from_checkout=true");
?>

```

/checkout/index.php

```

<?php
    session_start();
    include_once '../db_conn.php';
    include_once '../query_utils.php';

    if (!isset($_SESSION['user_id'])){
        //If already login, jump to home page directly
        header("Location: http://192.168.56.2/f32ee/EE4717-SoundX/login/");
        exit();
    }
    //generate a random 8 digit order id nubmer
    function generate_random_order_id($db){
        do {
            $rand_number = rand(1, 99999999);
            $check_repeat = "SELECT * FROM orders WHERE
order_id=".$rand_number;
            $res = $db->query($check_repeat);
            $num_row = $res->num_rows;
        } while ($num_row > 0);
        $order_id = str_pad($rand_number, 8, '0', STR_PAD_LEFT);
        return $order_id;
    }
?>
<!DOCTYPE html>
<html lang="en">

<head>
    <title>Checkout</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="../css/check-out.css" />
    <link rel="stylesheet" href="../css/elements.css" />
</head>

<body>
<div class="content">
<div id="return_button" style="margin-top;margin-left:150px">
    <form>
        <input type="button" value="Back" onclick="history.back()">
    </form>
</div>
<div class="order_details_form">
    <div style = "display:block">
        <form name="checkout" action="create_order.php" method="post"
class="form_content">

```

```

        <div class="order_details_left_col">
            <h1>Delivery Information</h1>
            <div class="details_address">
                <h2>Delivery Address</h2>
                <input type="text" id="delivery_address_line_1"
name="delivery_address_line_1"
                placeholder="Address Line 1" required>
                <input type="text" id="delivery_address_line_2"
name="delivery_address_line_2"
                placeholder="Address Line 2">
            </div>
            <div class="details_code">
                <h2>Postal Code</h2>
                <input type="text" id="zip_code" name="zip_code"
style="margin-bottom: 20px;" required>
                <div id="invalidZipCodeAlert" style="margin-bottom:
20px;"></div>
            </div>
            <div class="details_personal">
                <h2>Name</h2>
                <input type="text" id="receiver_name"
name="receiver_name" required>
                <h2>Contact No.</h2>
                <input type="tel" id="receiver_contact"
name="receiver_contact" required>
                <div style="margin-top: 25px;"
id="invalidContactAlert"></div>
            </div>
        </div>
        <div class="order_details_right_col">
            <h1>Order Details</h1>
            <div class="order_summary">
                <?php
                    if (!isset($_POST['selected'])) {
                        //no items selected in cart
                        header("Location:
http://192.168.56.2/f32ee/EE4717-SoundX/cart/");
                    } else {
                        //the action is remove item from cart, instead
of checkout

                        if ($_POST['remove'] === "true") {
                            foreach ($_POST['selected'] as $product) {
                                if (($key = array_search($product,
$_SESSION['cart'])) !== false) {

```

```

        unset($_SESSION['cart'][$key]);
    }
}
header("Location:
http://192.168.56.2/f32ee/EE4717-SoundX/cart/");
}
$order_id = generate_random_order_id($db);
$total = 0;
$items = array();
echo '<div class="summary_table">';
echo '<table>';
echo '<tr><th
style="width:200px;">Items</th><th style="width:40px;">Qty</th><th
style="width:80px;">Price</th><th width:80px;">Subtotal</th></tr>';
foreach($_POST['selected'] as
$selected_product){
    echo '<tr>';
    $qty = $_POST['items'][$selected_product];
    $items[$selected_product] = $qty;
    $info = get_details_by_id($db,
$selected_product);
    $product_name = $info['product_name'];
    echo "<td
style='width=50%'>{$product_name}</td>";
    echo "<td style='width=20%'>{$qty}</td>";
    $price = $info['price'];
    echo "<td
style='width=20%'>{$price}</td>";
    $subtotal = (float) $price * (float) $qty;
    $subtotal_format =
number_format($subtotal, 2);
    echo "<td>{$subtotal_format}</td>";
    echo '</tr>';
    $total += $subtotal;
}
$_SESSION["order_id"] = $order_id;
$_SESSION["order_item"] = $items;
$total_format = number_format($total, 2);
echo '<tr>';
echo '<td colspan="3"
id="total_title">Total</td>';
echo "<td
id='total_amount'>$ {$total_format}</td>";
echo '</tr>';

```

```

        echo '</table>';
        echo '</div>';
    }
    ?>
</div>
<div class="payment">
    <h1>Payment Method</h1>
    <div class="payment_button">
        <label>
            <input type="radio" id="payment_visa"
name="payment_method" value="visa" required>
            
        </label>
        <label>
            <input type="radio" id="payment_mastercard"
name="payment_method" value="mastercard">
            
        </label>
        <label>
            <input type="radio" id="payment_paypal"
name="payment_method" value="paypal">
            
        </label>
    </div>
</div>
<div id="noPaymentAlert" style="margin-bottom:
30px"></div>
<div id="submit_button">
    <input id="checkout-button" type="submit" value="Place
Order">
</div>
</div>
</form>
</div>
</div>
</div>
<script src="checkout_validate.js"></script>
</body>

</html>

/css/account-order.css
body{
    min-width: 1300px;

```

```

    display: block;
    margin: auto;
}

.main-content, .main-content-order{
    display: flex;
    justify-content: center;
    min-width: 1300px;
    margin: 100px 0;
}

.left-nav-column , .left-nav-column-order{
    display: block;
    width: 200px;
    padding-top: 50px;
    background-color: rgb(227, 235, 245);
}

.right-nav-column, .right-nav-column-order{
    display: block;
    width: 800px;
    padding-top: 50px;
    background-color: rgb(241, 243, 245);
}

.main-content-order, .right-nav-column-order{
    height: fit-content;
}

.right-nav-column-order{
    display: flex;
    justify-content: center;
}

#order-item-table{
    display: block;
    margin: 0;
    padding: 0;
}

#update-button{
    width: 140px;
    height: 40px;
    float: right;
}

```

```

    background-color: rgb(165, 174, 184);
    color: white;
    border:none;
    border-radius: 8px;
    padding: auto;
    cursor: pointer;
    font-size: 14px;
    font-weight: 500;
}

#user-info-text{
    margin: 0 100px;
}

#user-info-text h2{
    margin-bottom: 10px;
}

#user-info-text p{
    font-size: 16px;
    font-weight: 300;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    margin-top:0;
}

#account, #order{
    display: block;
    margin: 20px 40px 40px 0;
    text-align: right;
    font-size: 20px;
    font-weight: 700;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

#account a:link, #order a:link{
    text-decoration: none;
    color: rgb(36, 46, 66);
}

#account a:hover, #order a:hover{
    text-decoration: none;
    color: rgba(85, 101, 134, 80);
}

```



```

hr{
  margin:10px 0 40px 0;
  background-color:slategray;
  height: 0.4px;
  border-width:0;
}

input[type=submit] {
  width: 140px;
  height: 40px;
  margin-top:10px;
  margin-bottom: 50px;
  float: right;
  background-color: rgb(165, 174, 184);
  color: white;
  border:none;
  border-radius: 8px;
  padding: auto;
  cursor: pointer;
  font-size: 14px;
  font-weight: 500;
}

#order-view-details{
  margin-bottom: 100px;
}

#order-view-details a{
  margin-top: 20px;
  float: right;
  color: rgb(0, 44, 95);
  cursor: pointer;
  font-size: 16px;
  font-weight: 500;
}

#order-view-details a:hover {
  color:rgb(118, 144, 173);
}

input[type=text]{
  width: 90%;
  padding: 15px;
}

```

```

    border-radius:8px;
    margin: 5px 0;
    border: none;
    background: #ffffff;
}

#empty_order p{
    font-size: 24px;
    font-weight: 600;
    margin: 50px auto 100px auto;
    color:rgb(63, 79, 95);
}

/css/cart.css
body{
    min-width: 1300px;
}

.item-cart-title{
    display: block;
    padding: 40px 80px;
    font-size: 28px;
    font-weight: 700;
    letter-spacing: 2px;
    left:0;
}

.item-cart-title h1{
    margin-bottom: 0;
}

.cart-container{
    display:flex;
    justify-content: center;
    /*min-width: 1300px;*/
}

.cart-table{
    display: block;
}

.cart-table td{
    border-bottom: 1px solid #999;
}

```

```

.cart-row{
  display: flex;
  margin: 50px 0;
  align-items: center;
  padding: 0;
}

.cart-img-container{
  margin: 0 60px;
  display: flex;
  width: 300px;
  justify-content: center;
}

#product_name_quan{
  display: block;
  margin-left: 60px;
}

#product_name_quan p{
  display: block;
  margin: 0px;
  width: 550px;
  height: 60px;
  line-height: 1.5;
  font-family: Gotham, Verdana, sans-serif;
  font-style: normal;
  font-size: 18px;
  font-weight: 800;
}

#cart-checkbox{
  display: block;
}

#cart-img{
  display: block;
  height: 100px;
  width: fit-content;
  overflow: hidden;
}

```

```
#quan_field{
  display: block;
  font-size: 18px;
  font-weight: 400;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

#price-field{
  display: block;
  font-size: 22px;
  font-weight: 300;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

#empty_cart{
  display: block;
  margin:auto;
}

#empty_cart p{
  font-size: 28px;
  font-weight: 600;
  margin: 50px auto 150px auto;
  color:rgb(63, 79, 95)
}

#remove-button{
  float:right;
  width: 250px;
  height: 50px;
  float:right;
  margin:20px 0 4px;
  background-color: rgb(118, 144, 173);
  color: white;
  border:none;
  border-radius: 8px;
  padding: auto;
  cursor: pointer;
  font-size: 20px;
  font-weight: 500;
}

input[type=checkbox]:checked, input[type=checkbox]{
  cursor: pointer;
```

```

        width: 30px;
        height: 30px;
        margin: 20px;
    }

input[type=checkbox]{
    background-color:darkgrey;
}

input[type=checkbox]:checked {
    background-color:#1E80EF;
}

input[type=number]{
    height: 40px;
    width:80px;
    margin-left: 10px;
    border: 0.8px solid;
    border-radius: 10px;
    padding-left: 20px;
    font-size: 14px;
}

input[type=number]::-webkit-inner-spin-button {
    opacity: 0.6;
    height:40px;
    width: 10px;
    padding: 0 10px;
}

input[type=submit] {
    width: 250px;
    height: 50px;
    float:right;
    margin:40px 0 80px;
    background-color: rgb(0, 44, 95);
    color: white;
    border:none;
    border-radius: 8px;
    padding: auto;
    cursor: pointer;
    font-size: 20px;
    font-weight: 500;
}

```

```
input[type=submit]:hover {  
    background-color: rgb(118, 144, 173);  
}
```

```
/css/check-out.css
```

```
body{  
    min-width: 1300px;  
    display: block;  
    margin: auto;  
}  
  
.content{  
    margin:auto;  
    min-width: 1300px;  
    padding: 40px 100px;  
}  
  
.order_details_form{  
    display: flex;  
    padding: 100px 0;  
    margin-left:150px;  
}  
  
.order_details_left_col{  
    display: inline-block;  
    width: 550px;  
    min-width: 550px;  
}  
  
.order_details_right_col{  
    display: inline-block;  
    width: 400px;  
    min-width: 400px;  
    margin-left: 150px;  
}  
  
.form_content{  
    display: flex;  
    align-items: flex-start;  
}  
  
.order_summary{  
    display: block;
```

```

    height: 300px;
    overflow: auto;
    margin-bottom:50px;
}

.summary_table table{
    display: block;
    border-collapse: collapse;
    border-spacing:0;
}

.payment{
    display: block;
    margin-bottom: 0px;
}

.payment_button{
    height: 120px;
}

th {
    font-size: 18px;
    font-weight: 700;
    text-align: left;
    padding: 15px 10px 15px 10px;
    background-color:rgb(185, 191, 197);
}

td{
    font-size: 14px;
    font-weight: 500;
    text-align: left;
    padding: 15px 10px 15px 10px;
    border-bottom: solid 1pt rgb(54, 93, 133);
}

h1{
    margin-bottom: 25px;
}

h2{
    font-size: 20px;
    font-weight: 400;
    margin-bottom:5px;
}

```

```

}

img{
    height: auto;
    width: 90px;
    margin-right: 20px;
}

[type=radio] {
    position: absolute;
    width: 0;
    height: 0;
    opacity: 0;
}

[type=radio] + img {
    cursor: pointer;
}

[type=radio]:checked + img {
    border: 5px solid rgb(216, 224, 247);
}

input[type="text"], input[type="tel"] {
    cursor: text;
    height: 20px;
    width: 200px;
    padding: 12px 15px;
    border-width: 1px;
    border-color:rgb(133, 133, 133);
    border-radius: 10px;
}

#total_title, #total_amount{
    background-color:rgb(229, 234, 238);
    border: none;
}

input[type=submit] {
    width: 180px;
    height: 60px;
    float: right;
    margin-right: 20px;
}

```



```

    background-color: rgb(0, 44, 95);
    color: white;
    border:none;
    border-radius: 8px;
    padding: auto;
    cursor: pointer;
    font-size: 20px;
    font-weight: 600;
}

input[type=submit]:hover{
    background-color: rgb(118, 144, 173);
}

input[type="button"]{
    background-color: rgb(165, 174, 184);
    border: none;
    color: white;
    text-decoration: none;
    cursor: pointer;
    font-size: 18px;
    font-weight: 600;
    text-align: center;
}

/css/elements.css
body {
    padding: 0;
    margin: 0;
    font-family: "Segoe UI",Arial,sans-serif;
    font-weight: 400;
    min-width: 1300px;
}

/* navigation bar design */
.nav-bar-container {
    height: 50px;
    min-width: 1300px;
    padding: 10px 40px 10px 0;
    border-width: 0px;
    font-size: 20px;
    align-content: center;
}

```

```

    position: sticky;
    top: 0;
    right: 0;
    z-index: 20;
    background-color: SlateGray;
    color: Azure;
}

.brand-container a, .type-container a, .product-container a, .contact-
container a{
    font-weight: 600;
    margin: auto 20px;
}

/*nav dropdown list item*/
.nav-dropdown-item {
    list-style-type: none;
    display: flex;
    justify-content: center;
    flex-grow: 1;
    min-width: 1300px;
}

.subnav-item{
    display:flex;
    justify-content: center;
    align-items: center;
    padding:30px 60px;
}

.item-block{
    display: flex;
    justify-content: flex-start;
}

.font-item{
    display: block;
    margin-top:10px;
    font-size: 24px;
    color:darkslategray;
}

.item-img{
    display: block;

```

```

        max-width: 130px;
        display: block;
    }

    a.item-block-link{
        text-decoration:none;
    }

    /* Top left website logo */
    .logo{
        float: left;
        padding: 0 10px 0 10px;
    }

    /* nav part on left */
    .left-nav{
        float: left;
        padding: 10px 10px 0 10px;
        min-width:40px;
        text-decoration: none;
        color: Azure;
        height: 50px;
    }

    /* change nav text color to */
    .left-nav:hover{
        background-color: azure;
        color: black;
    }

    /* nav part on right */
    .right-nav{
        margin: 10px 0px;
        float: right;
        text-decoration: none;
        color: Azure;
    }

    .right-nav-item{
        margin:auto 0 auto 50px;
        display:inline-block;
    }

    .brand-dropdown-content, .type-dropdown-content {

```

```

    color: black;
    top: 70px;
    display: none;
    position: absolute;
    background-color:rgb(240, 245, 255);
    width: 100%;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    z-index: 1;
    margin:auto;
}

.brand-container:hover .brand-dropdown-content{
    display: block;
}

.type-container:hover .type-dropdown-content{
    display: block;
}

/* footer */
/* contact part */
#contact {
    display:block;
    background-color: SlateGray;
    min-width:1300px;
    position:relative;
}

#contact-text{
    display:block;
    position:relative;
    text-align: left;
    color: Azure;
    margin: auto 0;
    padding: 40px 100px;
}

/*contact heading*/
#contact-text h2{
    font-size: 32px;
    font-weight: 600;
    padding: 20px 0;
    margin: 0;
}

```

```

}

#contact-text p{
  line-height: 1.3;
  font-size: 18px;
  font-weight: 500;
}

/*alert*/
.alert{
  padding: 10px;
  background-color: #fcaca64d;
  border: 3px solid rgb(212, 92, 92);
  border-radius: 20px;
  color: rgb(212, 92, 92);
  margin-bottom: 15px;
  text-align: left;
  font-weight: 600;
}

/*success*/
.success{
  padding: 10px;
  background-color: #a6fdc34d;
  border: 3px solid rgb(92, 212, 166);
  border-radius: 20px;
  color: rgb(64, 185, 111);
  margin-bottom: 15px;
  text-align: left;
  font-weight: 600;
}

#return_button{
  display: block;
  width: 100px;
  height: 30px;
  margin-top: 50px;
  margin-bottom: 10px;
  background-color: rgb(165, 174, 184);
  border: none;
  border-radius: 8px;
  cursor: pointer;
  font-size: 18px;
  font-weight: 00;
}

```

```

    text-align: center;
    padding-top: 5px;
}

#return_button a{
    color: white;
    text-decoration: none;
}

/css/home-style.css
body {
    padding: 0;
    margin: 0;
    font-family: "Segoe UI", Arial, sans-serif;
    font-weight: 400;
}

.title{
    margin-top: 80px;
    margin-bottom: 50px;
}

/* product list*/

.product-list-flex{
    display: flex;
    justify-content: center;
    margin: 0 auto;
}

.product-list-item-block{
    display: block;
    width: 250px;
    min-height: 250px;
    margin: 0px 90px 100px 90px;
}

.product-list-item-img{
    display: block;
}

.topSale-item-img{
    padding-top: 30px;
}

```

```

    display: block;
    height: 200px;
    width: auto;
    overflow: hidden;
}

.product-list-item-info{
    display: block;
    width: 100%;
    text-decoration: none
}

.item-info-body-title{
    padding: 10px 0 5px;
    margin: 0px;
    font-family: Gotham, Verdana, sans-serif;
    font-style: normal;
    font-size: 15px;
    line-height: 15px;
    font-weight: 800;
}

.item-info-body-type{
    padding: 5px 0 5px;
    margin: 0px;
    font-family: Gotham, Verdana, sans-serif;
    font-style: normal;
    font-size: 15px;
    line-height: 15px;
    font-weight: 400;
}

.topSale-banner-container{
    height: 500px;
    min-height: 500px;
    width: 100%;
    min-width: 1300px;
    position: relative;
    margin: auto;
}

.topSale-banner-img{
    display: block;
    /*z-index: 0;*/

```

```

    width: 100%;
    height: 500px;
    min-width: 1300px;
    background-image: url("../img/banner/banner3_product6.jpg");
    background-position: left;
    background-repeat: no-repoeat;
    background-size: cover;
}

.topSale-banner-text-container{
    margin-left: 60%;
    text-align: left;
    position: absolute;
    /*z-index: 10;*/
    width: 30%;
    top: 50%;
    transform: translateY(-50%);
}

#topSale-title {
    text-align: left;
}

.product-button{
    display: flex;
    justify-content: center;
    background-color: #ffffff;
    text-decoration: none;
    padding: 10px 25px;
    margin: 20px 0 0;
    background-clip: padding-box;
    border-radius: 8px;
    cursor: pointer;
    width: 150px;
}

.product-button-text{
    padding: 1px 2px;
    color: #5f5f5f;
    font-size: 24px;
    font-weight: 700;
}

```



```

/*see more*/
.type-list-flex{
    display:flex;
    justify-content:center;
    margin: 80px 30px;
}
.type-list-item-block{
    display:block;
    margin: 0px 60px;
}

#type-item-img{
    height: 250px;
}

.type-img{
    display:block;
    border-radius: 10%;
}

.type-item-title{
    display:flex;
    justify-content:center;
    font-weight:800;
    font-size:20px;
}

}

/* banner container */
.top-banner-container{
    max-width: 90%;
    min-width:1300px;
    position: relative;
    margin: auto;
    padding-top: 20px;
}

/* Hide the images by default */
.banner-page {
    display: block;
}

/* Next & previous buttons */

```

```

.prev, .next {
  cursor: pointer;
  position: absolute;
  top: 50%;
  width: auto;
  margin-top: -22px;
  padding: 16px;
  color: azure;
  font-weight: bold;
  font-size: 40px;
  transition: 0.6s ease;
  border-radius: 0 3px 3px 0;
  user-select: none;
}

/* Position the "next button" to the right */
.next {
  right: 0;
  border-radius: 3px 0 0 3px;
}

/* On hover, add a black background color with a little bit see-through */
.prev:hover, .next:hover {
  background-color: rgba(0,0,0,0.8);
}

/* Caption text */
.caption-text {
  color: azure;
  font-size: 25px;
  padding: 16px 24px;
  position: absolute;
  bottom: 20px;
  width: 100%;
  text-align: center;
}

.transition {
  -webkit-animation-name: fade;
  -webkit-animation-duration: 1.5s;
  animation-name: fade;
  animation-duration: 1.5s;
}

```

```

@-webkit-keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

@keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

/* heading 1 style */
h1 {
  text-align: center;
  font-family: Calibri, sans-serif;
  letter-spacing: 1px;
  font-size: 60px;
  margin:0;
}

/css/login-style.css
* {box-sizing:border-box}

body, html {
  height: 100%;
  font-family: Arial, Helvetica, sans-serif;
}

.login-bg,.signup-bg {
  background-image: url("../img/login-bg-blue-small.png");
  height:100%;
  max-width:100%;
  min-width: 400px;
  min-height:500px;
  overflow:hidden;
  background-repeat: no-repeat;
  background-size: cover;
  position: relative;
}

.signup-bg{
  display:flex;
  justify-content:center;

```

```
    background-image: url("../img/signup-bg.png");
}
```

```
.signup-container{
  position: absolute;
  display: block;
  background-color: #ffffff70;
  top: 50%;
  transform: translateY(-50%);
  max-width: 500px;
  min-width: 400px;
  min-height: 800px;
  overflow: hidden;
  padding: 50px 30px;
}
```

```
.login-container {
  position: absolute;
  display: block;
  right: 0;
  margin-right: 80px;
  margin-left: 80px;
  top: 50%;
  border-radius: 20px;
  transform: translateY(-50%);
  max-width: 400px;
  min-width: 200px;
  height: fit-content;
  padding: 24px;
  background-color: #ffffff90;
}
```

```
.login-title, .signup-title{
  text-align: center;
  font-size: 20px;
  font-weight: 800;
  font-family: Gotham, Verdana, sans-serif;
  color: rgb(0, 44, 95);
}
```

```
.input-field{
  padding: 20px 10px;
}
```

```

.login-button, .signup-button{
  background-color: rgb(0, 44, 95);
  color: white;
  margin:20px 0;
  padding: 10px 20px;
  border-radius:10px;
  border: none;
  cursor: pointer;
  width: 100%;
  height:50px;
  font-size: 18px;
  font-weight: 600;
}

.login-button:hover, .signup-button:hover {
  background-color: rgb(118, 144, 173);
}

input[type=text], input[type=password], input[type=email] {
  width: 100%;
  padding: 15px;
  border-radius:8px;
  margin: 5px 0 22px 0;
  border: none;
  background: #f1f1f1;
}

input[type=text]:focus, input[type=password]:focus, input[type=email]:focus {
  background-color: #ddd;
  outline: none;
}

#sign-up-request{
  text-align:center;
}

#sign-up-request p{
  margin:0;
  padding-top:0;
  padding-bottom:10px;
  font-size: 18px;
  font-family: Gotham,Verdana,sans-serif;
}

```

```

#sign-up-request a:link{
    font-weight:680;
}

#sign-up-request a:hover{
    color:rgb(118, 144, 173);
}

#name-input-row{
    display:flex;
    justify-content: flex-start;
    flex-grow: 1;
    padding-right: 40px;
}

#name-input{
    margin-right:20px;
}

/css/nav-bar-style.css
.nav-bar-container {
    min-height: 50px;
    border-width: 0px;
    font-size: 20px;
    display: block;
    position: sticky;
    top: 0;
    z-index: 20;
    background-color: SlateGray;
    text-align: center;
    color: Azure;
}

/*nav dropdown list item*/
.nav-dropdown-item {
    list-style-type: none;
    display: flex;
    justify-content: center;
    flex-grow: 1;
    padding-right: 40px;
}

.subnav-item{

```

```

        display:flex;
        flex:0 0 20%;
        justify-content: center;
        align-items: center;
        padding:30px 20px;
    }

    a.item-block-link{
        text-decoration:none;
    }

    /* Top left website logo */
    .logo{
        float: left;
        padding: 0 10px 0 10px;
    }

    /* nav part on left */
    .left-nav{
        float: left;
        padding: 10px 10px 0 10px;
        min-width:40px;
        text-decoration: none;
        color: Azure;
        height: 50px;
    }

    /* change nav text color to */
    .left-nav:hover{
        background-color: azure;
        color: black;
    }

    /* nav part on right */
    .right-nav{
        margin-top: auto;
        margin-bottom: auto;
        padding: 5px 10px;
        float: right;
        text-decoration: none;
        color: Azure;
    }

    .brand-dropdown-content, .type-dropdown-content {

```

```

    color: black;
    top: 50px;
    display: none;
    position: absolute;
    background-color: Ivory;
    width: 100%;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    z-index: 1;
    margin: auto;
}

.brand-container:hover .brand-dropdown-content{
    display: block;
}

.type-container:hover .type-dropdown-content{
    display: block;
}

/css/order-details.css
body{
    min-width: 1300px;
    display: block;
    margin: auto;
}

.content{
    margin: auto;
    min-width: 1300px;
    padding: 40px 100px;
}

.status-bar-container{
    top: 50px;
    display: flex;
    align-items: flex-start;
    margin: auto;
    align-items: center;
    position: relative;
    flex: 1;
}

.phase{

```



```

    display: flex;
    flex-direction: column;
    align-items: center;
    position: relative;
    padding-bottom: 20px;
    flex: 1;
}

.status-dot{
    width: 40px;
    height: 40px;
    border-width: 2px;
    border-style: solid;
    border-color: rgb(170, 170, 170);
    border-radius: 100%;
    margin: auto;
    background-color: #f5f9f8;
    position: relative;
    z-index: 1;
}

.status-dot.done{
    background-color: #62B230;
    border:none;
}

.status-line{
    width: 100%;
    height: 4px;
    position: absolute;
    background: rgb(170, 170, 170);
    right: calc(50%);
    top: 20px;
    z-index: 0;
}

.status-line.solid{
    background-color:#62B230;
}

.status-text{
    margin-top:10px;
    font-size:18px;
}

```

```
.order-details-container{
  margin-top: 100px;
}

h1{
  font-size: 28px;
  font-weight: 700;
  margin-bottom: 50px;
  margin-top: 50px;
  left:0;
}

.delivery-info{
  display: flex;
  justify-content: center;
  column-gap: 100px;
}

p{
  font-size: 18px;
  margin-top: 5px;
  margin-bottom: 20px;
}

.title{
  font-size: 24px;
  font-weight: 700;
  margin-bottom:5px;
}

#return_button{
  display: block;
  width: 100px;
  height: 30px;
  margin-top:50px;
  margin-bottom: 10px;
  background-color: rgb(165, 174, 184);
  border:none;
  border-radius: 8px;
  cursor: pointer;
  font-size: 18px;
  font-weight: 500;
  text-align: center;
}
```

```

        padding-top:5px;
    }

#return_button a{

    color: white;
    text-decoration: none;
}

/css/order-item.css
.status-bar-container{
    display: block;
    min-width: 1300px;
}

.index-order-detail-container{
    display: block;
    min-width: 1300px;
}

.order-table-main{
    display: block;
    border-collapse: collapse;
    width: 100%;
    margin: auto;
}

.item-row-container{
    display: block;
    position: relative;
    padding: 10px 20px;
}

.item-row-container th{
    display: flex;
    height: 40px;
}

.item-row-container td{
    display: block;
}

.item-row{
    display: flex;

```

```

        margin: 20px 0;
        align-items: center;
    }
#img-container, #product-name-container, #qty-container, #price-container,
#subtotal-container{
    display: block;
}

#img-container{
    width: 50px;
    display: flex;
    justify-content: center;
}

#order-img{
    display: block;
    height: 50px;
    width: fit-content;
    overflow: hidden;
}

#product-name-container{
    display: block;
    margin-left: 50px;
    width: 500px;
    height: fit-content;
    text-align: left;
}

#product-name-container p{
    line-height: 1.5;
    font-family: Gotham, Verdana, sans-serif;
    font-style: normal;
    font-size: 14px;
    font-weight: 800;
}

#qty-container, #price-container, #subtotal-container{
    display: block;
    text-align: center;
    font-size: 18px;
    font-weight: 300;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

```

```

#qty-container{
    width:10%;
    min-width:130px;
}
#price-container, #subtotal-container{
    width:20%;
    min-width: 260px;
}

#table-head-item,#table-head-qty,#table-head-price,#table-head-subtotal,
#total-amount{
    font-size: 14px;
    font-weight: 600;
    display: block;
}

#table-head-item{
    text-align: left;
    padding-left: 30px;
    width: 50%;
    min-width:630px;
}

#table-head-qty{
    text-align: center;
    width: 10%;
    min-width: 130px;
}

#table-head-price,#table-head-subtotal{
    text-align: center;
    width: 20%;
    min-width: 260px;
}

#order-id-row{
    display: block;
    padding: 10px 40px;
    font-size: 14px;
}

#total{
    display: block;
}

```

```

    text-align: left;
}

tr {
    padding: 20px 50px;
    display: block;
}

td, th{
    border-bottom: 1px solid rgb(90, 90, 90);
}

```

```

/css/product-list-style.css
.filter-result-container {
    display: flex;
    justify-content: left;
    padding: 40px 0px;
    min-width: 1150px;
}

.filters-container {
    display: block;
    padding: 40px 80px;
    min-width: 1150px;
}

.brand-filters,
.type-filters,
.price-filter {
    display: flex;
    justify-content: flex-start;
    min-width: 1150px;
    width: 1150px;
    align-items: center;
}

.filter-title,
.product-result-title {
    font-size: 28px;
}

```

```

    font-weight: 700;
    letter-spacing: 2px;
    width: 120px;
    left: 0;
}

.product-result-title {
    display: block;
    margin: 120px 0 40px 80px;
}

.filter-checkbox-list,
.filter-price-range {
    display: block;
    position: relative;
    margin: auto;
    margin-left: 40px;
}

.checkbox-item {
    display: inline-block;
    position: relative;
    margin: auto;
    margin-right: 20px;
    margin-left: 0px;
    width: 120px;
}

#filter-separation {
    margin: 10px 40px 40px 0;
    background-color: slategray;
    height: 0.6px;
    border-width: 0;
}

input[type=checkbox]:checked,
input[type=checkbox] {
    cursor: pointer;
    width: 20px;
    height: 20px;
    margin-right: 10px;
    vertical-align: middle;
}

```

```
input[type=checkbox] {
    background-color: darkgrey;
}

input[type=checkbox]:checked {
    background-color: #1E80EF;
}

input[type=text] {
    padding: 12px 20px;
    margin: 5px 0;
    display: inline-block;
    border: 1.2px solid;
    border-radius: 3px;
    border-color: darkgrey;
    height: 15px;
    width: 80px;
    vertical-align: middle;
    color: grey;
}

input:focus {
    background-color: #ddd;
}

input[type=submit] {
    width: 250px;
    height: 50px;
    float: right;
    margin-top: 30px;
    background-color: rgb(0, 44, 95);
    color: white;
    border: none;
    border-radius: 8px;
    padding: auto;
    cursor: pointer;
    font-size: 20px;
    font-weight: 500;
}

input[type=submit]:hover {
    background-color: rgb(118, 144, 173);
}
```



```

label {
  font-size: 18px;
  font-weight: 200;
  bottom: 5px;
  vertical-align: middle;
}

/*display product*/

.product-result {
  display: grid;
  grid-template-columns: auto auto auto auto;
  grid-column-gap: 110px;
  grid-row-gap: 100px;
  margin: auto;
}

.product-card {
  width: 250px;
  height: auto;
  display: block;
}

.product-small-pic {
  padding-top: 30px;
  display: block;
  height: 150px;
  width: fit-content;
  overflow: hidden;
  margin: auto;
}

.product-list-info {
  display: block;
  padding: 10px 0 5px;
  margin: 0px;
  font-family: Gotham, Verdana, sans-serif;
  font-style: normal;
  font-size: 15px;
  line-height: 15px;
  font-weight: 800;
}

```

```
.product-list-info-title {  
    padding: 30px 0 0 0;  
    margin: 0px;  
    height: 40px;  
    font-family: Gotham, Verdana, sans-serif;  
    font-style: normal;  
    font-size: 14px;  
    font-weight: 800;  
}
```

```
p {  
    margin: 0;  
}
```

```
.product-list-info-price {  
    padding: 5px 0;  
    margin: 0px;  
    font-family: Gotham, Verdana, sans-serif;  
    font-style: normal;  
    font-size: 15px;  
    line-height: 15px;  
    font-weight: 400;  
}
```

```
.product-card-link {  
    text-decoration: none;  
    color: black;  
}
```

/css/product_detail.css

```
body {  
    font-family: "Segoe UI",Arial,sans-serif;  
    font-weight: 400;  
    min-width: 1300px;  
}
```

```
.maint-content{  
    display: block;  
    min-width: 1300px;  
}
```

```
.product-show{
```

```

display: flex;
flex-grow: 1;
padding: 60px 150px;
min-height: fit-content;
min-width: 1300px;
align-content: center;
}

.image-and-selector {
display: inline-block;
margin: 0 100px 0 0;
}

.main-img{
display: block;
padding: auto;
overflow:hidden;
}

.image-selector-small{
display: flex;
justify-self: flex-start;
flex-grow: 1;
}

.pic-thumbnail {
display: block;
padding-right: 10px;
padding-top: 15px;
cursor: pointer;
}

#product_big_picture {
width:600px;
min-width: 600px;
height:fit-content;
vertical-align: middle;
overflow:hidden;
margin: auto;
}

.text-title-part {
display: inline-block;
}

```

```

.text-title-part h1{
  margin: 30px 0 20px 0;
  font-weight: 800;
  font-size: 32x;
  overflow: hidden;
}

.text-title-part h2{
  margin: 0 ;
  font-weight: 300;
}

.price {
  font-size: 40px;
  margin-top: 30px;
}

.stock {
  font-size: 20px;
  margin-bottom: 30px;
}

.button-group{
  display: flex;
  flex-grow: 1;
  align-content: center;
}

#add-to-cart-button, #buy-now-button{
  display: block;
}

#cart-button, #buy-button {
  font-size: 20px;
  font-weight: 600;
  border-radius: 25px;
  height: 60px;
  width: 200px;
  border:0;
  color: white;
  padding: 15px 35px;
  margin-right: 20px;
  text-align: center;
}

```

```
    text-decoration: none;
    cursor: pointer;
}

#cart-button{
    background-color:orange;
}

#cart-button:hover{
    background-color:wheat;
}

#buy-button{
    background-color:#008CBA;
}

#buy-button:hover{
    background-color:#008CBA50;
}

.product-detail-info{
    display: block;
    margin: 100px 80px 100px 80px;
}

.info-content-container{
    display: block;
    margin-bottom: 100px;
}

.detail-info-container h1{
    margin: 0;
    font-weight: 800;
    font-size: 32x;
}

.detail-info-container p{
    margin:0;
    margin-bottom: 80px;
    font-size: 22px;
    font-weight: 300;
}
```

```

td{
  border-bottom: 0.6px solid grey;
  font-size: 20px;
}

.spec-table-key{
  background-color: azure;
  font-weight: 500;
  padding: 20px 40px;
}

.spec-table-value{
  padding: 20px 40px 20px 80px;
}

hr{
  margin-bottom: 30px;
}

input[type="button"]{
  background-color: rgb(165, 174, 184);
  border: none;
  color: white;
  text-decoration: none;
  cursor: pointer;
  font-size: 18px;
  font-weight: 600;
  text-align: center;
}

/Elements/footer.php
<!-- Contact -->
<div id="contact">
  <div id="contact-text">
    <h2>Contact Us</h2>
    <p>
      <!-- (+65) 8888 - 8888 -->
      &#40;&#43;65&#41; 8888 - 8888 <br />
      <!-- support@soundx.com -->
      <a href="mailto:support@soundx.com" style="color:azure;
text-decoration: none;">
        support&#64;soundx.com</a> <br /><br />
      156 Nanyang Cres<br />
      Singapore, 636866
    </p>
  </div>
</div>

```

```

        </p>
    </div>
</div>

/Elements/nav_bar.php
<?php include_once '../db_conn.php';
include_once '../query_utils.php';

$brands = array('Razer', 'Beats', 'Sony', 'SteelSeries', 'Bose');
$types = array('Wired', 'Wireless', 'Gaming', 'Sport');

$brand = select_all_from_table($db, "brand");

?>
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="../css/elements.css" />
</head>

<body>
    <div class="nav-bar-container">
        <!-- Left part -->
        <a class="logo" href="../home/">
            
        </a>
        <div class="brand-container">
            <a class="left-nav brand">Brand</a>
            <!-- TODO: brand dropdown -->
            <div class="brand-dropdown-content">
                <!--Here is something to display => brand-->
                <!-- TODO: style the list -->
                <ul class="nav-dropdown-item">
                    <?php
                    for ($i=0; $i<count($brand); $i++){
                        $brand_row = $brand[$i];
                        $brand_name = $brand_row['brand_name'];
                        $brand_id = $brand_row['brand_id'];
                        echo "<li class='subnav-item'>
                            <div class='item-block'>
                                <a class='item-block-link'

```

```

        href='../product_list/?brand%5B%5D={$brand_id}
'>
        <div class='item-img'><img
src='../img/brand/{$brand_id}.png' alt='SoundX' width='130px'></div>
        <div class='font-item'>{$brand_name}</div>
    </a>
</div>
</li>";
}
?>
</ul>
</div>
</div>
<div class="type-container">
    <a class="left-nav type">Type</a>
    <!-- TODO: type dropdown -->

    <div class="type-dropdown-content">
        <!--Here is something to display => type-->
        <ul class="nav-dropdown-item">
            <li class="subnav-item">
                <div class="item-block">
                    <a class="item-block-link"
                        href=<?php echo
"../product_list/?{$types[0]}=true"; ?>>
                        <div class="item-img"></div>
                        <div class="font-item"><?php echo
$types[0]; ?></div>
                    </a>
                </div>
            </li>
            <li class="subnav-item">
                <div class="item-block">
                    <a class="item-block-link"
                        href=<?php echo
"../product_list/?{$types[1]}=true"; ?>>
                        <div class="item-img"></div>
                        <div class="font-item"><?php echo
$types[1]; ?></div>
                    </a>
                </div>
            </li>

```



```

        <li class="subnav-item">
            <div class="item-block">
                <a class="item-block-link"
                    href=<?php echo
"../product_list/?{$types[2]}=true"; ?>>
                    <div class="item-img"></div>
                    <div class="font-item"><?php echo
$types[2]; ?></div>
                </a>
            </div>
        </li>
        <li class="subnav-item">
            <div class="item-block">
                <a class="item-block-link"
                    href=<?php echo
"../product_list/?{$types[3]}=true"; ?>>
                    <div class="item-img"></div>
                    <div class="font-item"><?php echo
$types[3]; ?></div>
                </a>
            </div>
        </li>
    </ul>
</div>
<div class="product-container">
    <a class="left-nav" href="../product_list/">Product</a>
</div>
<div class="contact-container">
    <a class="left-nav" href="#contact">Contact</a>
</div>
<!-- Right part -->
<div class="right-nav">
    <div class="right-nav-item">
        <a href="../account/">
            </a>
    </div>
    <div class="right-nav-item">
        <a href="../cart/">
            </a>
    </div>
</div>

```

```

        </div>
    </div>

</div>
</body>

</html>

/home/index.php
<?php
require_once '../db_conn.php';
session_start();

if (!isset($_SESSION['user_id'])){
    //If already login, jump to home page directly
    header("Location: http://192.168.56.2/f32ee/EE4717-SoundX/login/");
    exit();
}
?>
<!DOCTYPE html>
<html lang="en">

<head>
    <title>SoundX</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="../css/elements.css" />
    <link rel="stylesheet" href="../css/home-style.css" />
</head>

<body>
    <!-- the navigation bar -->
    <?php include '../Elements/nav_bar.php';?>

    <!-- Big Banner container -->
    <div class="top-banner-container">
        <!-- Full-width images -->
        <div class="banner-page transition">
            <a href="../product_details/?product_id=21">
                
            </a>
            <div class="caption-text">Bose Sport Open Earbuds</div>
        </div>

```

```

    <div class="banner-page transition">
      <a href="../product_details/?product_id=13">
        
      </a>
      <div class="caption-text">IER-H500A h.ear in 2 In-ear
Headphones</div>
    </div>

    <div class="banner-page transition">
      <a href="../product_details/?product_id=3">
        
      </a>
      <div class="caption-text">Razer Hammerhead Duo Console</div>
    </div>

    <!-- Next and previous buttons -->
    <a class="prev" onclick="next(-1)">&#10094;</a>
    <a class="next" onclick="next(1)">&#10095;</a>
  </div>
  <br />

  <!-- Top Sales -->
  <div class="top-sales">
    <div class="title">
      <h1>TOP SALES</h1>
    </div>
    <div class="top-sales-column">
      <div class="product-list">
        <div class="product-list-flex">
          <div class="product-list-item-block">
            <div class="product-list-item-img">
              <a href="../product_details/?product_id=10"
class="item-img-link">
                
              </a>
            </div>
            <div class="product-list-item-info">
              <a class="item-info-link">
                <div class="item-info-body">
                  <p class="item-info-body-title">WF-1000XM4
Wireless Noise Cancelling Headphones</p>

```

```

                <p class="item-info-body-type">Noise
Cancelling</p>
                <p class="item-info-body-
price">$379.00</p>
            </div>
        </a>
    </div>
</div>
<div class="product-list-item-block">
    <div class="product-list-item-img">
        <a href=" ../product_details/?product_id=1"
class="item-img-link">
            
        </a>
    </div>
    <div class="product-list-item-info">
        <a class="item-info-link">
            <div class="item-info-body">
                <p class="item-info-body-title">RAZER
HAMMERHEAD TRUE WIRELESS 2019</p>
                <p class="item-info-body-type">True
Wireless Earbuds</p>
                <p class="item-info-body-
price">$169.90</p>
            </div>
        </a>
    </div>
</div>
<div class="product-list-item-block">
    <div class="product-list-item-img">
        <a href=" ../product_details/?product_id=8"
class="item-img-link">
            
        </a>
    </div>
    <div class="product-list-item-info">
        <a class="item-info-link">
            <div class="item-info-body">
                <p class="item-info-body-title">Beats EP
blue</p>
                <p class="item-info-body-type">Wired
Headphone</p>
            </div>
        </a>
    </div>
</div>

```

```

                <p class="item-info-body-
price">$138.00</p>
            </div>
        </a>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- Full-width banner promotion -->
<div class="topSale-banner-container">
    <div class="topSale-banner-img">
        
    </div>
    <div class="topSale-banner-text-container">
        <div class="topSale-banner-text">
            <div id="topSale-title">
                <p style="font-size:40px; font-weight:700; line-
height:45px; margin:0px 0 10px; color:white">
                    Discover Best Sounds Experience</p>
                <p style="font-size:24px; font-weight:300; margin:10px
0; color:white">Beats</p>
            </div>
            <div>
                <!-- top sales promotion product shop now-->
                <a href="../product_details/?product_id=6"
class="product-button">
                    <span class="product-button-text">
                        Shop Now
                    </span>
                </a>
            </div>
        </div>
    </div>
</div>
</div>
<!-- See More -->
<div class="top-sales">
    <div class="title">
        <h1>SEE MORE</h1>
    </div>
    <div class="top-sales-column">

```

```

<div class="product-list">
  <div class="type-list-flex">
    <div class="type-list-item-block">
      <div id="type-item-img">
        <!--product list pag -->
        <a href="../product_list/?wireless=true"
class="item-img-link">
          
        </a>
      </div>
      <div class="product-list-item-info">
        <a class="item-info-link">
          <div>
            <p class="type-item-title">WIRELESS</p>
          </div>
        </a>
      </div>
    </div>
    <div class="type-list-item-block">
      <div id="type-item-img">
        <!--product list pag -->
        <a href="../product_list/?Gaming=true"
class="item-img-link">
          
        </a>
      </div>
      <div class="product-list-item-info">
        <a class="item-info-link">
          <div>
            <p class="type-item-title">GAMING</p>
          </div>
        </a>
      </div>
    </div>
    <div class="type-list-item-block">
      <div id="type-item-img">
        <!--product list pag -->
        <a href="../product_list/?Wired=true" class="item-
img-link">
          
        </a>

```

```

        </div>
        <div class="product-list-item-info">
            <a class="item-info-link">
                <div>
                    <p class="type-item-title">WIRED</p>
                </div>
            </a>
        </div>
    </div>
    <div class="type-list-item-block">
        <div id="type-item-img">
            <!--product list pag -->
            <a href="../product_list/?Bluetooth=true"
class="item-img-link">
                
            </a>
        </div>
        <div class="product-list-item-info">
            <a class="item-info-link">
                <div>
                    <p class="type-item-title">BLUETOOTH</p>
                </div>
            </a>
        </div>
    </div>
</div>
</div>
</div>
<br />
<?php include '../Elements/footer.php'; ?>
</body>
<script src="script.js"></script>

</html>

```

```

/home/script.js

```

```

/*display top banner */
var pageNumber = 1;
display(pageNumber);

// Next/previous controls
function next(n) {

```

```

    display(pageNumber += n);
}

// Thumbnail image controls
function currentSlide(n) {
    display(pageNumber = n);
}

function display(n) {
    var i;
    var slides = document.getElementsByClassName("banner-page");
    if (n > slides.length) {pageNumber = 1;}
    if (n < 1) {pageNumber = slides.length}
    for (i = 0; i < slides.length; i++) {
        slides[i].style.display = "none";
    }
    slides[pageNumber-1].style.display = "block";
    if (pageNumber < slides.length){
        pageNumber++;
    } else {
        pageNumber = 1;
    }
}

/login/index.php

<?php
require_once '../db_conn.php';
session_start();

if (isset($_SESSION['user_id'])){
    //If already login, jump to home page directly
    header("Location: http://192.168.56.2/f32ee/EE4717-SoundX/home/");
    exit();
}

?>
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>SoundX - Login</title>
        <meta charset="utf-8" />
        <link rel="stylesheet" href="../css/login-style.css" />
        <link rel="stylesheet" href="../css/elements.css" />
    </head>

```



```

<body>
  <div class="login-bg">
    <form action="" method="post" class="login-container">
      <div>
        <div class="login-title">
          <h1>Login</h1>
        </div>
        <div class="input-field">

          <label for="Email"><b>Email</b></label>
          <input type="text" placeholder="Email" name="email"
id="Email" required>

          <label for="Password"><b>Password</b></label>
          <input type="password" placeholder="Password"
name="password" id="Password" required>
        </div>
        <div id="sign-up-request">
          <p>New user? <a href="../sign_up">Sign up now!</a></p>
        </div>
        <div>
          <button type="submit" class="login-button">Login</button>
        <?php
          //check whether the user is already in database
          if (isset($_POST['email']) && isset($_POST['password'])) {
            //user has key in email and password, verify in
database
            $query = "SELECT user_id, email, password FROM users
WHERE ";
            $query.= "email='{$_POST['email']}' and
password='{$_POST['password']}'";
            $result = $db->query($query);
            if ($result->num_rows > 0) {
              $row = $result -> fetch_row();
              $_SESSION['user_id'] = $row[0];
              $_SESSION['email'] = $_POST["email"];
              $_SESSION['cart'] = array();
              echo
" <script>alert({$_SESSION['user_id']});</script>";
              header("Location:
http://192.168.56.2/f32ee/EE4717-SoundX/");
            } else {

```

```

        echo "<div class='alert'>&#10069 &nbsp; Input email
or password is not correct.</div>";
    }
    $_POST = array();
}??>
</div>
</div>
</form>
</div>
</body>
</html>

```

/order_details/index.php

```

<?php
    session_start();
    include_once '../db_conn.php';
    include_once '../query_utils.php';
    include_once 'send_email.php';

    if (!isset($_SESSION['user_id'])){
        //If not login, head in to login page
        header("Location: http://192.168.56.2/f32ee/EE4717-SoundX/login/");
        exit();
    }

    $order_id = (int)$_GET['id'];
    $order_details = get_order_details_by_order_id($db, $order_id);
    $order_items = get_order_items_with_product_info_by_order_id($db,
$order_id);

    //order id not exists in database
    if (empty($order_details)){
        header("Location: http://192.168.56.2/f32ee/EE4717-SoundX/orders/");
        exit();
    }

    //order is not created by current user
    if ($order_details['user_id'] != $_SESSION['user_id']){
        header("Location: http://192.168.56.2/f32ee/EE4717-SoundX/orders/");
        exit();
    }
?>

<!DOCTYPE html>

```

```
<html lang="en">

<head>
  <title>Order details</title>
  <meta charset="utf-8" />
  <link rel="stylesheet" href="../css/elements.css" />
  <link rel="stylesheet" href="../css/order-details.css" />
  <style>
    .index-order-detail-container {
      display: block;
      min-width: 1300px;
    }

    .order-item-table {
      display: block;
      margin-bottom: 80px;
    }

    .order-table-main {
      display: block;
      border-collapse: collapse;
      width: 700px;
      margin: auto;
    }

    .item-row-container {
      display: table;
      padding: 10px 20px;
    }

    .item-row-container th {
      height: 60px;
    }

    .item-row {
      align-items: center;
    }

    #img-container {
      width: 50px;
      padding: 10px;
      border-bottom: 1px solid rgb(90, 90, 90);
    }
  </style>
</head>
```

```

#order-img {
    height: 70px;
    width: fit-content;
    overflow: hidden;
}

#product-name-container {
    margin-left: 50px;
    width: 50px;
    height: fit-content;
    text-align: left;
    border-bottom: 1px solid rgb(90, 90, 90);
}

#product-name-container p {
    line-height: 1.5;
    font-family: Gotham, Verdana, sans-serif;
    font-style: normal;
    font-size: 14px;
    font-weight: 800;
}

#qty-container,
#price-container,
#subtotal-container {
    text-align: center;
    font-size: 18px;
    font-weight: 300;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    border-bottom: 1px solid rgb(90, 90, 90);
}

#table-head-item,
#table-head-qty,
#table-head-price,
#table-head-subtotal {
    padding-bottom: 20px;
    padding-top: 20px;
    font-size: 16px;
    font-weight: 600;
    background-color: slategray;
    color: white;
}

```

```

#table-head-item {
    text-align: left;
    padding-left: 30px;
}

#table-head-qty,
#table-head-price,
#table-head-subtotal {
    text-align: center;
}

#order-id-row,
#create-time-row {
    width: 700px;
    font-size: 18px;
}

#total-amount {
    font-size: 20px;
    font-weight: 300;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    text-align: center;
}

#total {
    font-size: 20px;
    font-weight: 700;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    padding-left: 20px;
}

table {
    table-layout: fixed;
    width: 100%;
    border-collapse: collapse;
    border: 1px solid black;
}
</style>

</head>

<body>
    <!-- the navigation bar -->
    <?php include '../Elements/nav_bar.php';?>

```

```

<?php
    if (isset($_GET['from_checkout'])){
        send_email($db, $order_id, $order_items, $order_details);
        echo "<div class='success'>Congrats! Your order has been created
successfully! An email has been sent to your mailbox!</div>";
    }
?>
<div class="content">
    <div id="return_button">
        <a href=" ../orders/index.php">Back</a>
    </div>
    <div class="status-bar-container">
        <?php
$curr_status = (int) $order_details['status'];
$all_status = select_all_from_table($db, 'status');
for ($node = 0; $node < count($all_status); $node++){
    echo "<div class='phase'>";
    if ($node < $curr_status){
        echo "<div class='status-dot done'></div>";
        if ($node != 0){
            echo "<div class='status-line solid'></div>";
        }
    } else {
        echo "<div class='status-dot'></div>";
        echo "<div class='status-line'></div>";
    }
    echo "<div class='status-
text'>{$all_status[$node]['status_name']}</div></div>";
}
?>
</div>
<div class="order-details-container">
    <h1>Order Details</h1>
    <!-- order items table -->
    <?php include 'item_in_order.php';
echo '<div class="index-order-detail-container">';
order_item_table($order_id, $order_details, $order_items);
echo '</div>';
?>
</div>
<div class="delivery-info-container">
    <!-- delivery information row -->
    <h1>Delivery Information</h1>
    <?php

```

```

        echo "<div class='delivery-info'><div id='personal'><p
class='title'>Name</p><p>{$order_details['receiver_name']}</p>";
        echo "<p class='title'>Contact No.
</p><p>{$order_details['receiver_contact']}</p></div>";
        echo "<div id='address'><p class='title'>Address</p>";
        echo "{$order_details['delivery_address_line_1']}<br>";
        if (strlen($order_details['delivery_address_line_2'] > 0)){
            //if delivery address line 2 is set
            echo "{$order_details['delivery_address_line_2']}";
        }
        echo "{$order_details['zip_code']}</div>";
        $track_number = (strlen($order_details['track_number'])>0)?
$order_details['track_number'] : "Not Available Now";
        if ($order_details['track_number'] == NULL){
            echo "<div id='track'><p class='title'>Track
Number</p><p>{$track_number}</p></div></div>";
        }
        ?>
    </table>
</div>
</div>
</body>

</html>

```

/order_details/item_in_order.php

```

<?php
function order_item_table($order_id, $order_details, $order_items){
    $order_id = str_pad($order_id, 8, '0', STR_PAD_LEFT);
    echo "<div class='order-table-main'>
    <div id='order-id-row'>
        <p><strong>Order:</strong> #{$order_id}</p>
    </div>
    <div id='create-time-row'><!-- create time row -->
        <p><strong>Created Time:</strong> {$order_details['create_time']}</p>
    </div>";

    echo "<div class='item-row-container'><!-- items row, contains items table
-->
    <table>
        <tr>
            <td colspan='2' id='table-head-item'>Item</td>
            <td id='table-head-qty'>Quantity</td>

```

```

        <td id='table-head-price'>Price</td>
        <td id='table-head-subtotal'>Subtotal</td>
    </tr>";
$total = 0;
foreach($order_items as $item){
    echo "<tr class='item-row'>";
    echo "<td id='img-container'><a
href='../product_details/?product_id={$item['product_id']}'><img
src='../img/product-snapshot/{$item['product_id']}.png' id='order-
img'></a></td>";//image
    echo "<td id='product-name-
container'><p>{$item['product_name']}</p></td>";//product name
    echo "<td id='qty-container'><p>{$item['qty']}</p></td>";//Qty
    $price = sprintf('%0.2f', $item['price']);
    echo "<td id='price-container'><p>\${$price}</p></td>";//Price
    $subtotal = sprintf('%0.2f', $item['qty'] * $item['price']);
    $total += $item['qty'] * $item['price'];
    echo "<td id='subtotal-
container'><p>\${$subtotal}</p></td>";//Subtotal
    echo "</tr>";
}
echo "<tr class='total-row'><td colspan='4' id='total'><p>Total</p></td>";
echo "<td id='total-amount'><p>\$";
echo sprintf('%0.2f', $total);
echo "</p></td></tr></table></div>";
}
?>

```

/order_details/send_email.php

```

<?php
function send_email($db, $order_id, $order_items, $order_details){
    $to = 'f32ee@localhost';
    $subject = "Your new SoundX order #{$order_id} has been placed!";
    $message = "Order Id : #{$order_id} " . "\r\n";
    $message .= "Order Items: \r\n";
    $total = 0;
    foreach ($order_items as $item){
        $message .= "Product : {$item['product_name']} ";
        $message .= "Qty : {$item['qty']} ";
        $message .= "Price : {$item['price']} \r\n";
        $total += $item['qty'] * $item['price'];
    }
    $message .= "Total : {$total} \r\n\r\n";
}

```



```

$message .= "Delivery Address: \r\n";
$message .= "{$order_details['delivery_address_line_1']} \r\n";
$message .= "{$order_details['delivery_address_line_2']} ";
$message .= "{$order_details['zip_code']} \r\n";
$headers = 'From: f32ee@localhost' . "\r\n" . 'Reply-To: f32ee@localhost' .
"\r\n" . 'X-Mailer: PHP/' . phpversion();

mail($to, $subject, $message, $headers, '-ff32ee@localhost');
}

```

```
?>
```

```
/orders/index.php
```

```

<?php include_once '../db_conn.php';
include_once '../query_utils.php';
include_once '../order_details/item_in_order.php';
session_start();

if (!isset($_SESSION['user_id'])){
    //If not login, head in to login page
    header("Location: http://192.168.56.2/f32ee/EE4717-SoundX/login/");
    exit();
}
?>

```

```

<!DOCTYPE html>
<html lang="en">

```

```

<head>
    <title>My Orders</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="../css/account-order.css" />
    <link rel="stylesheet" href="../css/elements.css" />
    <style>
        .status-bar-container {
            display: block;
            min-width: 1300px;
        }

        .index-order-detail-container {
            display: block;
            min-width: 1300px;
        }
    </style>

```

```

}

.order-item-table {
    display: block;
    margin-bottom: 80px;
}

.order-table-main {
    display: block;
    border-collapse: collapse;
    width: 700px;
    margin: auto;
}

.item-row-container {
    display: table;
    padding: 10px 20px;
}

.item-row-container th {
    height: 60px;
}

.item-row {
    align-items: center;
}

#img-container {
    width: 50px;
    padding: 10px;
    border-bottom: 1px solid rgb(90, 90, 90);
}

#order-img {
    height: 70px;
    width: fit-content;
    overflow: hidden;
}

#product-name-container {
    margin-left: 50px;
    width: 50px;
    height: fit-content;
    text-align: left;
}

```

```

        border-bottom: 1px solid rgb(90, 90, 90);
    }

#product-name-container p {
    line-height: 1.5;
    font-family: Gotham, Verdana, sans-serif;
    font-style: normal;
    font-size: 14px;
    font-weight: 800;
}

#qty-container,
#price-container,
#subtotal-container {
    text-align: center;
    font-size: 18px;
    font-weight: 300;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    border-bottom: 1px solid rgb(90, 90, 90);
}

#table-head-item,
#table-head-qty,
#table-head-price,
#table-head-subtotal {
    padding-bottom: 20px;
    padding-top: 20px;
    font-size: 16px;
    font-weight: 600;
    background-color: slategray;
    color:white;
}

#table-head-item {
    text-align: left;
    padding-left: 30px;
}

#table-head-qty, #table-head-price, #table-head-subtotal {
    text-align: center;
}

#order-id-row, #create-time-row{
    width: 700px;
}

```

```

        font-size: 18px;
    }

    #total-amount{
        font-size:20px;
        font-weight: 300;
        font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
        text-align: center;
    }

    #total{
        font-size:20px;
        font-weight: 700;
        font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
        padding-left: 20px;
    }

    table {
        table-layout: fixed;
        width: 100%;
        border-collapse: collapse;
        border: 1px solid black;
    }
</style>

</head>

<body>
    <?php include '../Elements/nav_bar.php';?>
    <div class="main-content-order">
        <div class="left-nav-column-order">
            <div id="account">
                <a href="../account/">
                    <span>ACCOUNT</span>
                </a>
            </div>
            <div id="order">
                <a href="">
                    <span>MY ORDER</span>
                </a>
            </div>
        </div>
        <div class="right-nav-column-order">
            <div id="order-content">

```

```

        <?php
            $curr_user = $_SESSION['user_id'];
            $query_orders = "SELECT * FROM orders WHERE user_id =
{$curr_user} ORDER BY create_time DESC";
            @$order_result = $db -> query($query_orders);
            echo '<div id="order-item-table">';
            $num_row = $order_result -> num_rows;
            //echo $num_row;
            if($num_row==0){
                echo "<div id='empty_order'><p>You do not have any
order at this time.</p></div>";
            }else{
                while($row = $order_result -> fetch_assoc()){
                    $order_id = $row['order_id'];
                    $order_items =
get_order_items_with_product_info_by_order_id($db, $order_id);
                    order_item_table($order_id, $row, $order_items);
                    echo "<div id='order-view-details'>";
                    echo "<a
href='../order_details/?id={$order_id}'>View Details >></a></div>";
                }
            }
            echo '</div>';
        ?>

    </div>
</div>

```

/product_details/add_to_cart.php

```

<?php include_once '../db_conn.php';
session_start();
$product_id = $_POST["product_id"];
if (!in_array($product_id, $_SESSION["cart"])){
    $_SESSION["cart"][] = $product_id;
    header("Location: http://192.168.56.2/f32ee/EE4717-
SoundX/product_details/index.php?product_id={$product_id}");
} else {
    header("Location: http://192.168.56.2/f32ee/EE4717-SoundX/cart/");
}
exit();
?>

```

```

/product_details/index.php

<?php include_once '../db_conn.php';
include_once '../query_utils.php';
session_start();
$product_id = $_GET["product_id"];
$row = get_details_by_id($db, $product_id);
$added_in_cart = in_array($product_id, $_SESSION['cart']);
?>
<!DOCTYPE html>
<html lang="en">

<head>
    <title>Details</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="../css/elements.css" />
    <link rel="stylesheet" href="../css/product_detail.css">
</head>

<body>
    <!-- navigation bar -->
    <?php include '../Elements/nav_bar.php';?>
    <div class="main_content">
        <div id="return_button" style="margin-left:150px">
            <form>
                <input type="button" value="Back" onclick="history.back()">
            </form>
        </div>
        <div class="product-show">
            <div class="image-and-selector">
                <div class="main-img">
                    <img src=<?php echo "../img/product/{$product_id}-1.jpg";?>
id="product_big_picture" />
                </div>
                <div class="image-selector-small">
                    <?php $num_of_pictures = (int) $row["no_pictures"];
for ($pic = 1; $pic <= $num_of_pictures; $pic++){
    $pic_source = "../img/product/.$product_id."-$pic.".jpg";
    echo '<div class="image-selector-small-item">';
    echo '<img class="pic-thumbnail" src='.$pic_source.'
id="product_pic_'. $pic.'" width="60px" height="60px"
onclick="switchImage(\'product_pic_'. $pic.'\');"/></div>';
    }?>
                </div>
            </div>
        </div>
    </div>

```

```

<div class="text-title-part">
    <h1><?php echo $row["product_name"]; ?></h1>
    <h2><?php echo $row["short_description"]; ?></h2>
    <p class="price"><strong>$<?php echo
$row["price"]; ?></strong></p>
    <p class="stock">Stock Left: <?php echo $row["stock"]; ?></p>
    <div class="button-group">
        <div id="add-to-cart-button">
            <form action="add_to_cart.php" method="post">
                <input type="text" hidden name="product_id"
value="<?php echo $product_id; ?>">
                <button id="cart-button"
onclick="this.form.submit();">
                    <?php
                    if ($added_in_cart){
                        echo "View In Cart";
                    } else {echo "Add to Cart";}
                    ?>
                </button>
            </form>
        </div>
        <div id="buy-now-button">
            <form id="buyNow" action="../checkout/index.php"
method="post">
                <input type="hidden" id="buy-now-select-this"
name="selected[]"
                    value="<?php echo $product_id; ?>">
                <input type="hidden" id="buy-now-qty-one"
name="items[<?php echo $product_id; ?>]"
                    value="1">
                <button id="buy-button"
onclick="buyNow.submit();">Buy Now</button>
            </form>
        </div>
    </div>
</div>
</div>
</div>

<div class="product-detail-info">
    <div class="detail-info-container">
        <h1>Description</h1>
        <hr class="solid">
        <p><?php echo $row["description"]; ?></p>
    </div>
</div>

```

```

    </div>

    <div class="detail-info-container">
        <h1>Specification</h1>
        <hr class="solid">
        <table id="spec-table">
            <?php $json_content = json_decode($row["specification"],
true);
            foreach($json_content as $key => $value){
                echo "<tr class='spec-table'>";
                echo "<td class='spec-table-key'>".$key."</td>";
                echo "<td class='spec-table-value'>".$value."</td>";
                echo "</tr>";
            }
            ?>
        </table>
    </div>
</div>
</div>
<?php include '../Elements/footer.php'; ?>
<script type="text/javascript" src="productdetails.js"></script>
<?php if($row["stock"] == 0){
    echo "<script> noStock(); </script>";
}
?>
</body>

</html>

```

```
/product_details/productdetails.js
```

```
//Product Details Image Selector
```

```
function switchImage(image_id){
    var image = document.getElementById(image_id);
    var source = image.src;
    var big_image = document.getElementById("product_big_picture");
    big_image.src = source;
}

```

```
//Disable add to cart or buy now if no stock
```

```
function noStock(){
    var cartButton = document.getElementById('cart-button');
    var buyNowButton = document.getElementById('buy-button');
    cartButton.disabled = true;
}

```



```

    cartButton.style.backgroundColor = "grey";
    buyNowButton.disabled = true;
    buyNowButton.style.backgroundColor = "grey";
}

/product_list/index.php

<!DOCTYPE html>
<html lang="en">

<head>
    <title>Products</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="../css/product-list-style.css"/>
    <link rel="stylesheet" href="../css/elements.css" />
</head>

<body>
    <!--navbar-->
    <?php include '../Elements/nav_bar.php';?>
    <div class="filters-container">
        <div class="filters">
            <form id="all_filters" action="" method="get">
                <?php
                if (isset($_GET)){
                    $get_variable = json_encode($_GET);
                    echo "<script>var filter_get = {$get_variable};</script>";
                }?>
                <div class="brand-filters">
                    <div class="filter-title">BRAND</div>
                    <div class="filter-checkbox-list">
                        <div class="checkbox-item">
                            <input type="checkbox" name="brand[]" id="brand1"
value="1">
                                <label for="brand1">Razer</label>
                            </div>
                        <div class="checkbox-item">
                            <input type="checkbox" name="brand[]" id="brand2"
value="2">
                                <label for="brand2">Beats</label>
                            </div>
                        <div class="checkbox-item">
                            <input type="checkbox" name="brand[]" id="brand3"
value="3">
                                <label for="brand3">Sony</label>

```

```

        </div>
        <div class="checkbox-item">
            <input type="checkbox" name="brand[]" id="brand4"
value="4">
            <label for="brand4">SteelSeries</label>
        </div>
        <div class="checkbox-item">
            <input type="checkbox" name="brand[]" id="brand5"
value="5">
            <label for="brand5">Bose</label>
        </div>
    </div>
</div>
<hr id="filter-separation">

<div class="filters">
    <div class="type-filters">
        <div class="filter-title">TYPE</div>
        <div class="filter-checkbox-list">
            <div class="checkbox-item">
                <input type="checkbox" name="Wireless"
id="Wireless" value="true">
                <label for="type1">Wireless</label>
            </div>
            <div class="checkbox-item">
                <input type="checkbox" name="Wired" id="Wired"
value="true">
                <label for="type2">Wired</label>
            </div>
            <div class="checkbox-item">
                <input type="checkbox" name="Earbuds"
id="Earbuds" value="true">
                <label for="type3">Earbuds</label>
            </div>
            <div class="checkbox-item">
                <input type="checkbox" name="Gaming"
id="Gaming" value="true">
                <label for="type4">Gaming</label>
            </div>
            <div class="checkbox-item">
                <input type="checkbox" name="Sport" id="Sport"
value="true">
                <label for="type5">Sport</label>
            </div>
        </div>
    </div>
</div>

```

```

        <div class="checkbox-item">
            <input type="checkbox" name="Bluetooth"
id="Bluetooth" value="true">
            <label for="type6">Bluetooth</label>
        </div>
    </div>
</div>
<div>
<hr id="filter-separation">

<div class="price-filter">
    <div class="filter-title">PRICE</div>
    <div class="filter-price-range">
        <label for="min_price">From</label>
        <input type="text" name="min_price"
onClick="this.select();" id="min_price" size="2" value="0"
            onchange="validateNumberInput(this);">
        <label for="max_price">To</label>
        <input type="text" name="max_price"
onClick="this.select();" id="max_price" size="2" value="1000"
            onchange="validateNumberInput(this);">
    </div>
    <div id="alert-msg" style="margin-right: 400px;"></div>
</div>
    <input id="filter-button" type="submit" value="Apply Filters">
</form>
</div>
</div>
<script src="product_list.js"></script>
<?php
include_once '../db_conn.php';
include_once '../query_utils.php';

$query = "SELECT * FROM products ";
$constrant_set = array();
$filters = $_GET;
if (isset($filters["min_price"])){
    if ($filters["min_price"] != ""){
        $constrant_set[] = " price >= {"$filters["min_price"]} ";
    }
    unset($filters["min_price"]);
}

if (isset($filters["max_price"])){

```

```

    if ($filters["max_price"] != ""){
        $constrant_set[] = " price <= {$filters["max_price"]} ";
    }
    unset($filters["max_price"]);
}

if (isset($filters["brand"])){
    $brands = $filters["brand"];
    $comma_saparate = implode(",", $brands);
    $constrant_set[] = " brand_id IN ({$comma_saparate})";
    unset($filters["brand"]);
}

$type_constraint = array();
foreach ($filters as $key => $value){
    $type_constraint[] = $key."=".$value;
}

if (count($type_constraint) > 0){
    $or_saparate = implode(" or ", $type_constraint);
    $constrant_set[] = "(.$or_saparate.)";
}
if (count($constrant_set) > 0){
    $query .= " WHERE ".implode(" and ", $constrant_set);
}
//var_dump($query);

$result_arr = get_product_list_by_query($db, $query);

echo '<div class="product-result-title">RESULTS</div>';

//if no result is available
if (empty($result_arr)){
    echo '<div id="no_result" style="display: flex; justify-content:
center;"></div>';
}

//echo '<div></div>';
else{
    echo '<div class="filter-result-container">';

    echo '<div class = "product-result">';
    for ($i=0; $i<count($result_arr); $i++){

```

```

        $row = $result_arr[$i];
        $product_id = $row['product_id'];
        $product_name = $row['product_name'];
        $price = $row['price'];

        echo '<a class = "product-card-link"
href=" ../product_details/index.php?product_id='.$product_id.'">';
        echo '<div class = "product-card"
id="product_id_'.$row['product_id'].'">';
        echo '<div>';
        echo '</div>';
        echo '<div class = "product-list-info">';
        echo '<div class="product-list-info-
title"><p>'.$product_name.'</p></div>';
        echo '<div class="product-list-info-
price"><p>$'.$price.'</p></div>';
        //echo '<hr id="filter-separation">';
        echo '</div></div></a>';
    }
    echo '</div></div>';
}
include '../Elements/footer.php';?>
</body>

</html>

```

/product_list/product_list.js

```

function validateNumberInput(new_input){

    if (new_input !== null){
        if (new_input.value == ""){
            new_input.value = (new_input.id == "min_price")? 0 : 1000;
        }
    }
    var min_price_ele = document.getElementById("min_price");
    var max_price_ele = document.getElementById("max_price");
    var pattern = /^[+-]?([0-9]+\.\?[0-9]*|\.[0-9]+)$/;
    if (min_price_ele.value.search(pattern) == 0 &&
max_price_ele.value.search(pattern) == 0){
        clearAlertMessage();
        if (parseFloat(min_price_ele.value)<0){
            min_price_ele.value = 0;

```

```

    }
    if (parseFloat(max_price_ele.value)<0){
        max_price_ele.value = 1000;
    }
} else if (min_price_ele.value.search(pattern) !== 0){
    showAlertMessage();
    min_price_ele.value = 0;
} else if (max_price_ele.value.search(pattern) !== 0){
    showAlertMessage();
    max_price_ele.value = 1000;
}
}
}

validateNumberInput(null);

function showAlertMessage(){
    document.getElementById("alert-msg").innerHTML = "Not a valid number
input, please try again! ";
    document.getElementById("alert-msg").className = "alert";
}

function clearAlertMessage(){
    document.getElementById("alert-msg").innerHTML = "";
    document.getElementById("alert-msg").className = "";
}

if (filter_get !== undefined){

    if (filter_get.brand){
        filter_get.brand.forEach(brand_id => {
            document.getElementById("brand" + brand_id).checked = true;
        });
        delete filter_get['brand'];
    }

    document.getElementById("min_price").value = (filter_get.min_price)?
filter_get.min_price : 0;
    document.getElementById("max_price").value = (filter_get.max_price)?
filter_get.max_price : 1000;
    delete filter_get['min_price'];
    delete filter_get['max_price'];

    for (const [key, value] of Object.entries(filter_get)) {
        document.getElementById(key).checked = true;
    }
}

```

```

    }
}

/sign_up/index.php

<?php
require_once '../db_conn.php';
session_start();

?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>SoundX - Sign Up</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="../css/login-style.css" />
    <link rel="stylesheet" href="../css/elements.css" />
</head>

<body>
    <div class="signup-bg">
        <form action="" method="post" class="signup-container">
            <div class="sign-up-input-block">
                <div class="signup-title">
                    <h1>Member Sign Up</h1>
                </div>
                <div class="input-field">
                    <div id="name-input-row">
                        <div id="name-input">
                            <label for="FirstName"><b>First
Name</b></label>
                            <input type="text" placeholder="First
Name" name="firstName" id="FirstName" required>
                        </div>
                        <div id="name-input">
                            <label for="LastName"><b>Last
Name</b></label>
                            <input type="text" placeholder="Last Name"
name="lastName" id="LastName" required>
                        </div>
                    </div>
                </div>
            </div>
        </form>
    </div>

```

```

        <div id="invalidNameAlert"></div>
        <div>
            <label for="Contact"><b>Contact</b></label>
            <input type="text" placeholder="Your Mobile
No." name="contact" id="Contact" required>

            <div id="invalidContactAlert"></div>

            <label for="Email"><b>Email</b></label>
            <input type="email" placeholder="Your Email
Adress" name="email" id="Email" required>

            <div id="invalidEmailAlert"></div>

            <label for="Password"><b>Password</b></label>
            <input type="password" placeholder="Set
Password." name="password" id="Password" required>

            <label for="ConfirmPassword"><b>Confirm
Password</b></label>
            <input type="password" placeholder="Confirm
Password" name="confirmPassword" id="ConfirmPassword" required>

            <div id="notSamePasswordAlert"></div>
        </div>
    </div>
    <div>
        <button type="submit" id="signup-button"
class="signup-button">Sign Up</button>
<?php
//first check whether the entered email is already in database
    if (isset($_POST['email']) && isset($_POST['password'])) {
        $try_select = "SELECT email FROM users WHERE
email='{$_POST['email']}'";
        $result = $db -> query($try_select);
        if ($result -> num_rows > 0) {
            //the email has already been registereed
            echo '<div class="alert">';
            echo "The email you entered is already registered! ";
            echo '<a href=" ../login">Login here</a>';
            echo '</div>';
        } else {
            $insert_query = "INSERT INTO users (email, password, first_name,
last_name, contact)";

```



```

        $insert_query .= "VALUES({'$_POST['email']}',
'$_POST['password']}', {'$_POST['firstName']}', {'$_POST['lastName']}',
'$_POST['contact']})";
        $insert_res = $db -> query($insert_query);
        echo "<div class='success'>Sign up successfully! Redirecting to
login in 3 seconds...</div>";
        header('Refresh: 3; url=../login/');
    }
}
?>

        </div>
    </form>
</div>
</div>
<script src="signup.js"></script>
</body>
</html>

```

/sign_up/signup.js

```

var signUpButton = document.getElementById("signup-button");
var passwordField = document.getElementById('Password');
var confirmPasswordField = document.getElementById('ConfirmPassword');
var contactField = document.getElementById("Contact");
var emailField = document.getElementById("Email");
var firstNameField = document.getElementById("FirstName");
var lastNameField = document.getElementById("LastName");

firstNameField.addEventListener("change", validateName);
lastNameField.addEventListener("change", validateName);
passwordField.addEventListener("change", confirmPassword);
confirmPasswordField.addEventListener("change", confirmPassword);
contactField.addEventListener("change", validateContactNumber);
emailField.addEventListener("change", validateEmail);

function validateName(e){
    let name = e.currentTarget.value;
    let alert = document.getElementById("invalidNameAlert");
    if (name.search(/^[a-zA-Z\s-]+$/) !== 0){
        alert.className = "alert";
        alert.innerHTML = "Name should only contain letters, space or -";
        signUpButton.disabled = true;
    }
}

```

```

    } else {
        alert.className = "";
        alert.innerHTML = "";
        signUpButton.disabled = false;
    }
}

function validateContactNumber(){
    let contact = contactField.value;
    let alert = document.getElementById("invalidContactAlert");
    if (contact.search(/^\d{8}$/) !== 0){
        alert.className = "alert";
        alert.innerHTML = "A valid phone number contains 8 digits.";
        signUpButton.disabled = true;
    } else {
        alert.className = "";
        alert.innerHTML = "";
        signUpButton.disabled = false;
    }
}

function validateEmail(){
    let email = emailField.value;
    console.log(email);
    let alert = document.getElementById("invalidEmailAlert");
    if (/^[^\w\.-]+\@([\w]+\.)\{1,3\}[\w]+\$/ .test(email)){
        alert.innerHTML = "";
        alert.className = "";
        signUpButton.disabled = false;
    } else {
        alert.className = "alert";
        alert.innerHTML = "Please enter a valid email.";
        signUpButton.disabled = true;
    }
}

function confirmPassword(){
    var password = document.getElementById('Password').value;
    var confirmedPassword = document.getElementById('ConfirmPassword').value;
    let alert = document.getElementById("notSamePasswordAlert");
    if (password.length > 0 && confirmedPassword.length > 0){
        if (confirmedPassword !== password){
            alert.className = "alert";

```

```

        alert.innerHTML = "Password input is not matched. Please try
again.";
        signUpButton.disabled = true;
    } else {
        alert.className = "";
        alert.innerHTML = "";
        signUpButton.disabled = false;
    }
}
}
}

```

```
/sql/tables_import.sql
```

```
-- Table Name: brand.sql
```

```

DROP TABLE IF EXISTS brand;
CREATE TABLE IF NOT EXISTS brand (
    brand_id int(11) NOT NULL PRIMARY KEY,
    brand_name varchar(20) NOT NULL
);

```

```

INSERT INTO brand (brand_id, brand_name) VALUES
(1, 'Razer'),
(2, 'Beats'),
(3, 'Sony'),
(4, 'SteelSeries'),
(5, 'Bose');

```

```
-- Table Name: orders.sql
```

```

DROP TABLE IF EXISTS orders;
CREATE TABLE IF NOT EXISTS orders(
    order_id                INT NOT NULL PRIMARY KEY
,create_time                TIMESTAMP default CURRENT_TIMESTAMP
,order_id                  INT NOT NULL
,status                     INT
,delivery_address_line_1  VARCHAR(100)
,delivery_address_line_2  VARCHAR(100)
,zip_code                  VARCHAR(10)
,receiver_name             VARCHAR(50)
,receiver_contact          VARCHAR(10)
,payment_method            VARCHAR(10)
,track_number              VARCHAR(10)
);

```

```

INSERT INTO orders (order_id, create_time, user_id, status,
delivery_address_line_1, delivery_address_line_2, zip_code, receiver_name,
receiver_contact, payment_method) VALUES
(24053638, '2021-10-01 15:49:06', 1, 1, '156 Nanyang Cres', '', '636866',
'Haoran Wei', '88888888', 'visa');

```

```
-- Table Name: order_items.sql
```

```
DROP TABLE IF EXISTS order_items;
```

```

CREATE TABLE IF NOT EXISTS order_items(
  order_id INT NOT NULL
  ,product_id INT NOT NULL
  ,qty INT NOT NULL
);

```

```
INSERT INTO order_items (order_id, product_id, qty) VALUES (24053638, 2, 1);
```

```
-- Table Name: products.sql
```

```
DROP TABLE IF EXISTS products;
```

```

CREATE TABLE IF NOT EXISTS products(
  product_id INTEGER NOT NULL PRIMARY KEY
  ,product_name VARCHAR(60) NOT NULL
  ,short_description VARCHAR(70) NOT NULL
  ,no_pictures INTEGER NOT NULL
  ,brand_id INTEGER NOT NULL
  ,price NUMERIC(6,2) NOT NULL
  ,wireless BOOLEAN NOT NULL
  ,wired BOOLEAN NOT NULL
  ,earbuds BOOLEAN NOT NULL
  ,gaming BOOLEAN NOT NULL
  ,sport BOOLEAN NOT NULL
  ,bluetooth BOOLEAN NOT NULL
  ,stock INTEGER NOT NULL
  ,description VARCHAR(1000) NOT NULL
  ,specification VARCHAR(2000) NOT NULL
);

```

```
INSERT INTO
```

```

products(product_id,product_name,short_description,no_pictures,brand_id,price,
wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (1,'RAZER HAMMERHEAD TRUE WIRELESS 2019','True Wireless
Earbuds',5,1,169.9,1,0,1,0,0,1,10,'Low latency connection, Water-resistant
IPX4 design, Custom-tuned 13mm drivers','{"HEADPHONE FREQUENCY RESPONSE": "20
Hz - 20 KHz", "HEADPHONE IMPEDANCE": "32 Ohm (1 kHz)", "HEADPHONE
SENSITIVITY": "91 &#177; 3 dB (1 kHz)", "HEADPHONE INPUT POWER": "8 mW",
"HEADPHONE DRIVERS": "13 mm", "HEADPHONE CONNECTOR": "Bluetooth", "CABLE

```

```
LENGTH": "None", "WEIGHT": "45 g", "MICROPHONE PICK UP PATTERN":  
"Omnidirectional", "MICROPHONE SIGNAL-TO-NOISE RATIO": "55 dB", "MICROPHONE  
SENSITIVITY (@1KHZ)": "42 &#177; 3 dB (1 kHz)", "COMPATIBILITY": "Devices with  
Bluetooth connections, Smartphone app available for Android 8.0 Oreo and iOS  
11 (or higher)"}');
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,  
wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)  
VALUES (2,'Razer Kraken Kitty - Black','Razer Kitty Ear USB Headset with  
Chroma',2,1,239.9,0,1,0,1,0,0,20,'Kitty Ears and Earcups Powered by Razer  
Chroma , Stream Reactive Lighting ,Cosplay Mode',{'FREQUENCY RESPONSE': "20  
Hz - 20 kHz", "IMPEDANCE": "32 Ohm (1 kHz)", "SENSITIVITY": "109 dB (1 kHz)",  
"DRIVER SIZE - DIAMETERS (MM)": "50 mm", "DRIVER TYPE": "Custom Tuned",  
"EARCUPS": "Cooling Gel-Infused Cushions", "INNER EARCUP DIAMETER": "Major  
Diameter 65 mm, Minor Diameter 44 mm", "EARPADS MATERIAL": "Heat-Transfer  
Fabric / Leatherette / Memory Foam", "NOISE CANCELLING": "Active noise-  
canceling Microphone", "CONNECTION TYPE": "USB Digital", "CABLE LENGTH": "4.27  
ft. / 1.3 m", "WEIGHT": "0.90 lbs / 408 g", "MICROPHONE STYLE": "Retractable  
Unidirectional Microphone", "PICK-UP PATTERN": "Unidirectional ECM boom",  
"MICROPHONE FREQUENCY RESPONSE": "100 Hz - 10 kHz", "MICROPHONE SENSITIVITY  
(@1KHZ)": "-45 &#177; 3 dB", "VIRTUAL SURROUND ENCODING": "THX Spatial",  
"VOLUME CONTROL": "Yes (VOL + and VOL buttons)", "OTHER CONTROLS": "On  
Microphone Mute Function: Click-to-mute", "On-earcup: THX Spatial on/off  
toggle": "BATTERY LIFE", "None": "LIGHTING", "Stream reactive kitty ears":  
"Underglow earcups", "COMPATIBILITY": "Devices with USB Connector"}');
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,  
wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)  
VALUES (3,'Razer Hammerhead Duo Console - Green','Dual Driver In-  
Ears',5,1,79.9,0,1,0,0,0,0,22,'Dual Driver Technology, Three Silicone Tip  
Sizes, Aluminum Frame & Braided Cables',{'HEADPHONE FREQUENCY RESPONSE': "20  
Hz-20 KHz", "HEADPHONE IMPEDANCE": "32 Ohm &#177; 15%", "HEADPHONE  
SENSITIVITY": "112 &#177; 3 dB (Max SPL)", "HEADPHONE INPUT POWER": "10 mW/20  
mW", "HEADPHONE DRIVERS": "Dynamic + Balanced Armature", "HEADPHONE  
CONNECTOR": "3.5 mm angled jack", "CABLE LENGTH": "1.2 m", "WEIGHT": "17 g",  
"MICROPHONE FREQUENCY RESPONSE": "100 Hz -10 kHz", "MICROPHONE PICK UP  
PATTERN": "Omni-directional", "MICROPHONE SIGNAL-TO-NOISE RATIO": " 58 dB",  
"MICROPHONE SENSITIVITY (@1KHZ)": "-40 &#177; 3 dB", "COMPATIBILITY": "Devices  
with 3.5 mm audio jack, Devices with 3.5 mm audio + microphone combined  
jack"}');
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,  
wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)  
VALUES (4,'Razer Kaira Pro for Xbox - White','Wireless Headset for Xbox Series
```

```
X and mobile Xbox gaming',4,1,229.9,1,0,0,1,0,1,9,'Razer TriForce Titanium
50mm Drivers, Raze HyperClear Supercardioid Mic, Dedicated mobile
mic','{"FREQUENCY RESPONSE": "20 Hz - 20 kHz", "IMPEDANCE": "32 at 1 kHz",
"SENSITIVITY": "108 dB", "DRIVER SIZE - DIAMETERS (MM)": "50 mm", "DRIVER
TYPE": "Razer TriForce Titanium", "EARCUPS": "Oval ear cushions", "INNER
EARCUP DIAMETER": "2.2 x 2.64, 56 mm x 67 mm", "EARPADS MATERIAL": "FlowKnit
memory foam ear cushions", "NOISE CANCELLING": "None", "CONNECTION TYPE":
"Direct to Xbox / via Xbox Wireless Adapter for Windows 10, Wireless range: 10
m / 30 ft, Wireless frequency: 2.4 GHz / 5 GHz", "CABLE LENGTH": "None",
"WEIGHT": "0.80 lbs / 365 g", "MICROPHONE STYLE": "None", "PICK-UP PATTERN":
"Unidirectional", "VIRTUAL SURROUND ENCODING": "None", "MICROPHONE FREQUENCY
RESPONSE": "100 Hz - 10 kHz", "VOLUME CONTROL": "Yes (VOL + and VOL Wheel)",
"OTHER CONTROLS": "On Earcup", "BATTERY LIFE": "Up to 15 hours (with Chroma
Lighting), 20 hours (without Chroma Lighting)", "LIGHTING": "Razer Chroma RGB
underglow lighting", "COMPATIBILITY": "Xbox One / Xbox One S / Xbox One X/
Xbox Series X|S, PC (Windows 10 or higher) **May require Xbox Wireless Adapter
for Windows 10 (not included/ Sold Separately)", "MICROPHONE SENSITIVITY
(@1KHZ)": "54 &#177; 3 dB"}');
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,
wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
```

```
VALUES (5,'Beats Flex - Black','BEATS FLEX: FLEX ALL
```

```
DAY',5,2,99.99,0,1,0,0,1,1,26,'In your ears or around your neck, Beats Flex
are as versatile as the life you lead. Whether you are listening to music,
taking calls, or scrolling your social feed, you will always be ready for what
s next. Magnetic earbuds make listening that much easier by automatically
playing music when they are in your ears and pausing when they are attached
around your neck. The Flex-Form cable provides all-day comfort with durable
Nitinol construction while four eartip options offer a personalized fit. And
when you are not wearing them, the magnetic earbuds keep Beats Flex tangle-
free as they easily coil up into your pocket or purse.','{"Other Features":
"Bluetooth, With Mic, Inline Volume Control, Wireless", "Form Factor": "In
Ear", "Connections": "Bluetooth, Wireless", "Power Source": "Battery",
"Batteries": "Rechargeable lithium-ion", "Height": "0.6 in / 1.6 cm",
"Length": "34 in / 86.4 cm", "Width": "4.2 in / 10.6 cm", "Weight": "18.6 g /
0.66 oz"}');
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,
wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
```

```
VALUES (6,'PowerBeats Pro - Red','POWERBEATS PRO: POWER.
```

```
PLAY.',4,2,328,1,0,1,0,1,1,36,'Powerbeats Pro, powered by the Apple H1
headphone chip, will revolutionise the way you work out. Built for elite
athletes, these totally wireless earphones have no wires to hold you back. The
adjustable, secure-fit earhooks are customisable with multiple ear-tip options
```

for extended comfort and are made to stay in place, no matter how hard you go. This lightweight earphone is built for performance with a reinforced design for IPX4-rated sweat and water resistance, so you can take your workouts to the next level. With up to 9 hours of listening time in each earbud and powerful, balanced sound, you will always have your music to motivate you.', '{"Other Features": "Bluetooth, Wireless", "Form Factor": "Ear hook, In Ear", "Connections": "Bluetooth, Wireless", "Batteries": "Rechargeable lithium-ion", "Height": "2.3 cm / 0.9 in. (bud), 4.3 cm / 1.7 in. (case)", "Length": "5.9 cm / 2.3 in. (bud), 7.7 cm / 3 in. (case)", "Width": "3.8 cm / 1.5 in. (bud), 7.7 cm / 3 in. (case)", "Weight": "11g (bud), 80g (case)"}');
INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (7,'Beats Solo Pro - Ivory','NOISE CANCELLING WIRELESS HEADPHONES TO STAY INSPIRED',6,2,429,1,0,0,0,0,1,8,'Get inspired with Solo Pro wireless headphones. To deliver sound how you want it, Solo Pro features two listening modes: Active Noise Cancelling (ANC) and Transparency mode. Beats Pure ANC gives you the space to create with fully immersive sound, while Transparency mode helps you stay aware of your surroundings. Every detail of Solo Pro has been carefully considered, right down to the intuitive way the headphones turn on and off via folding. The ergonomic design delivers exceptional comfort for extended wear and sleek style. And with up to 22 hours of battery life, you can keep the music going no matter where your day takes you.', '{"Form Factor": "On Ear", "Connections": "Bluetooth, Wireless", "Power Source": "Battery", "Batteries": "Rechargeable lithium-ion", "Height": "17.9 cm/7.05 in.", "Weight": "267 g/9.42 oz."}');
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (8,'Beats EP - blue','WIRED HEADPHONES BUILT FOR LIFE',5,2,138,0,1,0,1,0,0,25,'Beats EP On-Ear Headphones deliver masterfully tuned sound. The battery-free design offers unlimited playback and their sleek, durable frame is reinforced with lightweight stainless steel. Beats EP is an ideal introduction to Beats for any music lover seeking a dynamic listening experience.', '{"Form Factor": "On Ear", "Cable length": "133.9 cm / 52.7 in.", "Height": "6.6 cm / 2.6 in.", "Length": "17.8 cm / 7 in.", "Width": "15.4 cm / 6 in.", "Weight": "200g"}');
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (9,'PowerBeats - White','LIGHTWEIGHT WIRELESS EARPHONES FOR ALL?DAY ACTIVITY',4,2,199,0,1,0,0,1,1,26,'Powered by the Apple H1 headphone chip, high-performance wireless Powerbeats earphones are built to keep you moving. The adjustable, secure-fit earhooks maximise comfort and stability, while a
```

reinforced design for IPX4-rated sweat and water resistance⁵ lets you take it to the next level. And with up to 15 hours of listening time,¹ you can stay charged through multiple workouts and fuel your training with powerful, balanced sound.', '{"Other Features": "Bluetooth, Wireless, Noise Isolation", "Form Factor": "Ear hook, In Ear", "Connection": "Bluetooth, Wireless", "Power Source": "Battery", "Batteries": "Rechargeable lithium-ion", "Height": "2.4 cm / 0.9 in.", "Length": "5.9 cm / 2.3 in.", "Width": "3.8 cm / 1.5 in.", "Weight": "27g"}');

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,
wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (10,'WF-1000XM4 Wireless Noise Cancelling Headphones','Your sound.
Nothing else.',5,3,379,1,0,1,0,1,1,24,'An exceptional listening experience
that/ s tailored just for you. The WF-1000XM4 truly wireless headphones take
industry-leading noise cancelling1 and audio quality to the next level. Made
to fit every ear, they offer a personalised experience that adjusts to every
situation.', '{"WEIGHT": "Approx. 41 g", "HEADPHONE TYPE": "Truly Wireless",
"DRIVER UNIT": "6 mm", "MAGNET": "High power neodymium magnets", "FREQUENCY
RESPONSE(BLUETOOTH COMMUNICATION)": "20Hz - 20,000Hz(44.1kHz sampling) / 20Hz
- 40,000Hz(LDAC 96kHz sampling, 990kbps)", "WATERPROOF": "Yes(IPX4)", "DSEE
EXTREME": "Yes", "BATTERY CHARGE TIME": "Approx. 1.5 hrs", "BATTERY
LIFE(CONTINUOUS MUSIC PLAYBACK TIME)": "Max. 8 hrs(NC On) / Max. 12 hrs(NC
Off)", "BATTERY LIFE(CONTINUOUS COMMUNICATION TIME)": "Max. 5.5 hrs(NC On) /
Max. 6.0 hrs(NC Off)", "BLUETOOTH VERSION": "BLUETOOTH Specification
Version5.2", "PROFILE": "A2DP, AVRCP, HFP, HSP", "SUPPORTED AUDIO FORMAT(S)":
"SBC, AAC, LDAC", "SUPPORTED CONTENT PROTECTION": "SCMS-T", "AMBIENT SOUND
MODE": "Yes", "QUICK ATTENTION": "Yes"}');
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,
wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (11,'WF-SP900 Sports Wireless Headphones','Music to take you
further',3,3,399,1,0,1,0,1,1,38,'On the track, in the pool or under the ocean,
the WF-SP900 headphones are all you need to take you further than ever before.
With a truly wireless design and 4GB of on-board storage so you can listen
anywhere.', '{"WEIGHT": "Approx. 7.3g x 2 (not including the arc supporters)",
"HEADPHONE TYPE": "Closed Balanced Armature", "DRIVER UNIT": "Balanced
Armature", "FREQUENCY RESPONSE(BLUETOOTH COMMUNICATION)": "20Hz - 20,000Hz
(44.1kHz Sampling)", "VOLUME CONTROL": "Yes (Tap sensor)", "WATERPROOF": "IPX5
/ IPX8 (Headset only)2", "NFC": "Yes (Charging Case)", "MEMORY SIZE": "4GB3",
"MUSIC PLAY MODE": "Normal / Repeat /Shuffle / Shuffle all / Repeat 1 Song /
Folder / Playlists / Album", "AUDIO PLAYBACK": "MP3 ( .mp3) WMA ( .wma) FLAC
( .flac) WAV ( .wav) AAC ( .mp4, .m4a, .3gp)4", "AMBIENT SOUND MODE": "Yes",
"BATTERY CHARGE TIME": "Approx. 2.5 Hours (Full charge )", "BATTERY CHARGE
METHOD": "USB charging (with Charging Case)", "BATTERY LIFE(CONTINUOUS MUSIC
```



```
PLAYBACK TIME)": "Max. 3 Hours (Ambient sound mode OFF ) *via Bluetooth
playback5678", "BATTERY LIFE(CONTINUOUS MUSIC PLAYBACK TIME WITH INTERNAL
MEMORY)": "Max. 6 Hours (Ambient sound mode OFF) *internal memory
playback5678", "BATTERY LIFE(CONTINUOUS COMMUNICATION TIME)": "Max. 2.5
Hours", "BLUETOOTH VERSION": "Bluetooth Specification Version 4.0", "EFFECTIVE
RANGE": "Line of sight approx.30ft (10m)", "FREQUENCY RANGE": "2.4GHz band
(2.4000GHz-2.4835GHz)", "PROFILE": "A2DP (Advanced Audio Distribution
Profile) ,AVRCP (Audio Video Remote Control Profile) ,HFP (Hands-free
Profile) ,HSP (Headset Profile)9", "SUPPORTED AUDIO FORMAT(S)": "SBC,
AAC1011", "SUPPORTED CONTENT PROTECTION": "SCMS-T"}');
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,
wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (12,'WI-1000XM2 Wireless Noise Cancelling In-ear Headphones','Only
music. Nothing else.','7,3,449,0,1,0,0,0,1,7','Discover a world of
uncompromising silence where you can enjoy music without noise or distraction
in all-day comfort. Industry-leading noise cancellation1, superior sound
quality and the comfort of an angled earphone design take listening to a whole
new level.','{"WEIGHT": "Approx. 58 g (main unit approx. 44 g)", "HEADPHONE
TYPE": "Closed,Hybrid", "DRIVER UNIT": "Hybrid", "MAGNET": "Neodymium",
"IMPEDANCE (OHM)": "50 ohm(1kHz)(when connecting via the headphone cable with
the unit turned on), 17 ohm(1kHz)(when connecting via the headphone cable with
the unit turned off)", "FREQUENCY RESPONSE": "3 Hz-40,000 Hz", "FREQUENCY
RESPONSE (ACTIVE OPERATION)": "20Hz - 20,000Hz (44.1kHz Sampling) / 20Hz -
40,000Hz (LDAC 96kHz Sampling, 990kbps)", "FREQUENCY RESPONSE(BLUETOOTH
COMMUNICATION)": "20Hz - 20,000Hz (44.1kHz Sampling) / 20Hz - 40,000Hz (LDAC
96kHz Sampling, 990kbps)", "SENSITIVITIES (DB/MW)": "99 dB/mW (when connecting
via the headphone cable with the unit turned on), 94 dB/mW (when connecting
via the headphone cable with the unit turned off)", "VOLUME CONTROL": "Yes",
"CORD TYPE": "Single-sided (detachable)", "CORD LENGTH": "approx. 1m", "PLUG":
"Gold-plated L-shaped Stereo Mini", "INPUT(S)": "Stereo Mini", "WEARING
STYLE": "Neckband", "NFC": "Yes", "DSEE HX": "Yes", "PASSIVE OPERATION":
"Yes", "BATTERY CHARGE TIME": "Approx. 3.5 hours, This unit can be used for 80
minutes after 10 minutes charging.", "BATTERY CHARGE METHOD": "USB", "BATTERY
LIFE(CONTINUOUS MUSIC PLAYBACK TIME)": "Max. 10 hours (NC ON), Max. 12 hours
(Ambient Sound Mode), Max. 15 hours (NC OFF)", "BATTERY LIFE(CONTINUOUS
COMMUNICATION TIME)": "Max. 9 hours (NC ON), Max. 10 hours (Ambient Sound
Mode), Max. 13 hours (NC OFF)", "BATTERY LIFE(WAITING TIME)": "Max. 11 hours
(NC ON), Max. 13 hours (Ambient Sound Mode), Max. 100 hours (NC OFF)",
"BLUETOOTH VERSION": "Bluetooth Specification Version5.0", "EFFECTIVE RANGE":
"Line of sight approx.30ft (10m)", "FREQUENCY RANGE": "2.4GHz band (2.4000GHz-
2.4835GHz)", "PROFILE": "A2DP, AVRCP, HFP, HSP", "SUPPORTED AUDIO FORMAT(S)":
"SBC, AAC, LDAC", "SUPPORTED CONTENT PROTECTION": "SCMS-T", "NOISE CANCELING
ON/OFF SWITCH": "Yes", "AMBIENT SOUND MODE": "Yes"}');
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,
wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (13,'IER-H500A h.ear in 2 In-ear Headphones','Compact. Colourful.
Crystal-clear audio.',3,3,129,0,1,0,0,0,0,5,'Experience the superior detail of
High-Resolution Audio with an in-ear design that comes in a range of colours
to suit your style.', '{"WEIGHT": "Approx. 5g (0.2 oz) without cable",
"HEADPHONE TYPE": "Closed, dynamic", "DRIVER UNIT": "9mm, dome type (CCAW
Voice Coil)", "MAGNET": "Neodym", "IMPEDANCE (OHM)": "16ohmsat 1kHz",
"DIAPHRAGM": "PET", "FREQUENCY RESPONSE": "5Hz-40,000Hz", "SENSITIVITIES
(DB/MW)": "103dB / mW", "CORD TYPE": "Y-type", "CORD LENGTH": "Approx. 1.2m
(47 1 / 4 in) silver-coated OFC Litz wire", "PLUG": "L-shaped gold-plated 4-
pole mini plug"}');
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,
wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (14,'WH-1000XM4 Wireless Noise Cancelling Headphones','Silent
White',6,3,499,1,0,0,0,0,1,38,'Designed in a new Silent White colour to
reflect the concept of silence and serenity of the noise cancelling
headphones.', '{"HEADPHONE TYPE": "Closed, dynamic", "DRIVER UNIT": "40mm, dome
type (CCAW Voice coil)", "MAGNET": "Neodymium", "DIAPHRAGM": "Aluminum coated
LCP", "FREQUENCY RESPONSE": "4Hz-40,000Hz", "FREQUENCY RESPONSE (ACTIVE
OPERATION)": "4Hz-40,000Hz", "FREQUENCY RESPONSE(BLUETOOTH COMMUNICATION)":
"20Hz - 20,000Hz (44.1kHz Sampling) / 20Hz - 40,000Hz (LDAC 96kHz Sampling,
990kbps)", "SENSITIVITIES (DB/MW)": "105dB / mW (1kHz) (when connecting via
the headphone cable with the unit turned on) , 101dB / mW (1kHz) (when
connecting via the headphone cable with the unit turned off)", "CORD TYPE":
"Single-sided (detachable)", "CORD LENGTH": "Headphone cable (approx. 1.2m,
OFC strands, gold-plated stereo mini plug)", "PLUG": "Gold-plated L-shaped
stereo mini plug", "WEARING STYLE": "Circumaural", "NFC": "Yes", "DSEE
EXTREME": "Yes", "PASSIVE OPERATION": "Yes"}');
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,
wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (15,'MDR-1AM2 Headphones','Take music
anywhere',5,3,429,0,1,0,0,0,0,17,'The precision sound and full-spectrum
clarity of the MDR-1AM2 headphones captures the hearts of music lovers
everywhere. And with a lightweight design and soft cushioned earpads, you can
listen for hours in total comfort.', '{"WEIGHT": "187g", "HEADPHONE TYPE":
"Closed Dynamic", "DRIVER UNIT": "40mm (Dome Type)", "MAGNET": "Neodymium",
"IMPEDANCE (OHM)": "16 Ohm", "DIAPHRAGM": "Aluminium-Coated LCP", "FREQUENCY
RESPONSE": "3-10000Hz2", "SENSITIVITIES (DB/MW)": "98dB / mW", "CORD TYPE":
"Detachable / Silver coated OFC", "CORD LENGTH": "1.2m", "PLUG": "Gold-plated
L-shaped Stereo Mini", "WEARING STYLE": "Circumaural"}');
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,
wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (16,'ARCTIS PRO WIRELESS - black','High fidelity audio comes to gaming
for the first time',4,4,369.99,1,0,0,1,0,1,11,'Dual Wireless: 2.4G lossless +
Bluetooth, Industry-leading hi-res capable speaker drivers, Lightweight
aluminum alloy and steel construction, Swappable dual-battery
system',{'Neodymium Drivers': "40mm", "Headphone Frequency Response": "10 -
40,000 Hz", "Headphone Sensitivity": "102 dB SPL", "Headphone Impedance": "32
Ohm", "Headphone Total Harmonic Distortion": "< 1%", "Microphone Type":
"Retractable Boom", "Microphone Polar Pattern": "Bidirectional Noise-
Canceling", "Microphone Frequency Response": "100 - 10,000 Hz", "Microphone
Sensitivity": "-38 dBV/Pa", "Microphone Impedance": "2200 Ohm", "Microphone
Noise Cancellation": "Yes", "Wireless 2.4G range": "40 ft, 12m", "Battery
Life": "20 hours (10 per battery)", "Bluetooth Version": "4.1", "Bluetooth
Profiles": "A2DP, HFP, HSP"}');
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,
wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (17,'TUSQ','In-ear mobile gaming headset',4,4,39.9,1,0,0,1,0,0,9,'Tusq
gaming earbuds are designed with clear audio, upgraded mic quality, and a
comfortable adjustable design to bring the all perks of a gaming headset to
mobile earphones.',{'Headphone Style': "In-ear (canalphone)", "Headphone
Frequency Response": "20 - 20000Hz", "Headphone Sensitivity": "102 dB SPL @ 1
kHz, 1 mW", "Headphone Total Harmonic Distortion": "<1%", "Microphone
Frequency Response": "100 - 10,000 Hz", "Microphone Pattern":
"Omnidirectional", "Microphone Sensitivity": "-44 dBV/Pa", "Microphone
Location": "Detachable", "Connector Type": "Single 3.5mm, 4-pole plug", "Cable
Length": "1.20m / 4ft.", "Cable Material": "Rubber"}');
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,
wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (18,'ARCTIS PRO - black','Peerless High Resolution PC Gaming
Headset',4,4,199.99,1,0,0,1,0,0,11,'Industry-leading hi-res capable speaker
drivers, Lightweight aluminum alloy and steel construction, DTS Headphone:X
v2.0 surround sound, ClearCast, the best mic in gaming',{'Neodymium Drivers':
"40mm", "Headphone Frequency Response": "10 - 40,000 Hz", "Headphone
Sensitivity": "102 dB SPL", "Headphone Impedance": "32 Ohm", "Headphone Total
Harmonic Distortion": "< 1%", "Microphone Type": "Retractable Boom",
"Microphone Polar Pattern": "Bidirectional Noise-Canceling", "Microphone
Frequency Response": "100 - 10,000 Hz", "Microphone Sensitivity": "-38
dBV/Pa", "Microphone Impedance": "2200 Ohm", "Microphone Noise Cancellation":
"Yes", "Version Support": "3.12.0 +", "Platform Support": "Windows 7+, Mac OS
X 10.9+ (DTS Headphone:X available on Windows 7+ only)"}');
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,
wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (19,'ARCTIS 5','Design Exclusively for PC
Gaming',5,4,99.99,0,1,0,1,0,1,11,'Specifically designed for the PC Gamer,
Arctis 5 combines independent game and chat control, cutting-edge DTS surround
sound, and dual zone RGB illumination to create the perfect audio solution for
your battlestation.', '{"Neodymium Drivers": "40mm", "Headphone Frequency
Response": "20 - 2000Hz", "Headphone Sensitivity": "98db", "Headphone
Impedance": "320hm", "Headphone Total Harmonic Distortion": "< 3%", "Headphone
Volume Control": "On Ear Cup", "Microphone Frequency Response": "100Hz -
10000Hz", "Microphone Pattern": "Bidirectional", "Microphone Sensitivity": "-
48db", "Microphone Impedance": "2200 Ohm", "Microphone Noise Cancellation":
"Yes", "Microphone Location": "Retractable", "Microphone Mute Toggle": "On Ear
Cup", "Connector Type": "USB or 4-Pole 3.5mm via included adapter", "Cable
Length": "30m / 100ft", "Cable Material": "Rubber", "Adapter": "Single 3.5mm,
4-Pole Plug", "Share Jack": "Yes", "Detachable Cable": "Yes"}');
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,
wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (20,'Bose Frames Soprano-black','Nothing in or on your
ears',5,5,224,0,1,0,0,0,1,31,'The original Bose Frames debuted in 2019 and
quickly became one of our most popular products. Now the sequel is here.
Thoughtfully refined and strikingly elegant, Bose Frames Soprano flaunt
polarized lenses and premium craftsmanship, while exclusive Bose Open Ear
Audio technology produces sound you'd never expect from
sunglasses. It's a jaw-dropping experience that leaves you free to
engage with the world around you, all while discreetly listening to music. As
Esquire noted in choosing Bose Frames for Best Smart Accessory in their 2020
Gadget Guide, "These things are about the coolest wearable tech out
there."', '{"Frames Soprano": "55 mm x 17 mm x 136 mm(Lens width x Distance
between lenses x Temple length)", "Weight": "1.76 oz", "Carrying case
dimensions": "6.85 in L x 2.48 in W x 2.28 in", "Frames": "TR-90 nylon with
IPX2 water resistant rating", "Lenses": "Shatter- and scratch-resistant
premium plastic", "Carrying case": "Protective plastic with soft fabric
lining", "Battery life": "Up to 5.5 hours", "Battery charging time": "Up to 1
hour", "Battery charge method": "Custom 4-pin charging cable", "Bluetooth
range": "30 ft (9 m)", "Bluetooth version": "5.1", "Input": "Pogo pin charging
cable"}');
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,
wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (21,'Bose Sport Open Earbuds','HEAR THE MUSIC.AND THE
MOMENT.',6,5,199,0,1,1,0,0,1,14,'Bose Sport Open Earbuds are a truly different
```

kind of sport headphone. That s because they are the first of their kind that don't go in your ears. Instead, they combine exclusive Bose OpenAudio technology with a unique open-ear design to produce high-quality sound without covering your ears. The city. The surf. Your running buddy. Hear it all and your music at the same time with these entirely new, entirely open sport earbuds.'

```
{
  "Earbuds dimension": "2.2 inch H x 1.9 inch W x 0.75 inch D (0.49 oz each earbud)",
  "Carrying case dimension": "1.6 inch H x 4.4 inch W x 2.8 inch D (2.4 oz)",
  "Charging base dimension": "1.2 inch H x 3.8 inch W (1.7 oz)",
  "Earbuds material": "PC-ABS plastic composite",
  "Carry case material": "Plastic exterior, fabric inside, rubberized polyurethane",
  "Charging base material": "Plastic, rubber footing, metal charging pins",
  "Battery life": "Up to 8 hours",
  "Battery charging time": "2 hours",
  "Quick 30-minute charge": "Up to 3 hours",
  "Battery charge method": "Connect charging base to USB-A port on computer or standard wall charger (e.g. phone charger)",
  "Bluetooth range": "Up to 30 ft (9 m)",
  "Bluetooth version": "5.1 (A2DP, HFP, AVRCP, BLE)",
  "Codec": "SBC and AAC"}
};
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (22,'Bose Noise Cancelling Headphones 700 - black','PREMIUM QUIET. BOLD SOUND.',6,5,419,0,1,0,0,0,1,20,'World-class adjustable noise cancellation with situational awareness for when you want to let the world in. High-fidelity audio with adjustable EQ so you can tune music to your liking. Unrivalled voice pickup for the clearest calls. And protein leather cushions for all-day comfort. It s everything you demand from wireless Bluetooth headphones amplified. Bose Noise Cancelling Headphones 700, our most advanced headphones ever. And now they are bundled with the Headphones 700 Charging Case, making them an even smarter and more convenient choice.',{"Battery life": "Up to 20 hours", "Charging case battery life": "Up to 40 hours", "Battery charging time": "Up to 2.5 hours", "Case charging time": "Up to 3 hours", "Quick 15-minute charge": "For up to 3.5 hours", "Battery charge method": "USB-C", "Bluetooth range": "Up to 33 ft (10 m)", "Bluetooth version": "5.0 (including all headphone profiles)", "Codec": "Codec SBC and AAC", "Microphone": "8 total microphones, 6 microphones for Active Noise Cancelling (ANC), 4 microphones for Voice Pickup (2 are shared with ANC)"});
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (23,'Bose QuietComfort Earbuds','RULE THE QUIET',6,5,279,0,1,1,0,0,1,39,'Better sound begins with better silence. That s why we engineered QuietComfort Earbuds with the world s most effective noise cancelling and high-fidelity audio, plus StayHear Max tips for extra comfort. Because when you eliminate distractions, your music takes centerstage, and so does your passion. Roller skating, street art, woodworking and every other
```

thing that makes you, you. It is an experience you won't find in any other wireless earbud. And now for a limited time, QuietComfort Earbuds are available in two new fashion-forward colors, Stone Blue or Sandstone, exclusive styles only from Bose. Just another way you will stand out from the crowd.'

```
{
  "Earbuds": "Plastic/gold plating/polymer coating",
  "Eartips": "Silicone",
  "Case": "Hard plastic",
  "Battery life": "Up to 6 hours",
  "Earbud battery charge time": "2 hours",
  "Charging case battery charge time": "3 hours",
  "Quick-charge time": "15 minutes for 2 hours",
  "Battery charge method": "USB-C or Qi certified wireless charging pad",
  "Battery type (earbud)": "Lithium-ion (metal-enclosed coin cell)",
  "Battery type (case)": "Lithium-ion (pouch cell)",
  "Battery feature": "Automatic on/off feature to preserve battery life",
  "Bluetooth range": "Up to 30 ft (9.144 m)",
  "Bluetooth version": "5.1",
  "Codec": "SBC and AAC",
  "Microphone": "4 total microphones"}
};
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (24,'SoundSport wireless headphones','Engineered to push you forward, not hold you back',3,5,129,0,1,1,0,1,1,18,'Exercise is a demanding activity. And you demand wireless earbuds that are up to the challenge. SoundSport wireless headphones keep you moving with powerful audio and earbuds that stay secure and comfortable.',{"Headphones": "1.2 inch H x 1 inch W x 1.2 inch D (0.8 oz)", "Battery": "Rechargeable lithium-ion battery", "Charging time": "2 hours", "Battery life": "6 hours per full charge"});
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (25,'Bose QuietComfort 35 II Gaming Headset?','2-in-1 gaming and lifestyle headset with acclaimed lifelike sound',6,5,329,1,0,0,1,0,0,7,'Introducing the Bose QuietComfort 35 II Gaming Headset, engineered to play everything. With our award-winning noise cancelling technology and a detachable gaming module, this is the first 2-in-1 gaming and lifestyle headset from Bose. Designed for hours of competitive gaming and music listening, with a classic look and comfortable fit to wear at home, work, and everywhere in-between.',{"Battery life": "Up to 20 hours wireless, up to 40 hours gaming/wired", "Battery charging time": "2.25 hours", "Quick 15-minute charge": "Up to 2.5 hours wireless, up to 5 hours gaming/wired", "Battery charge method": "Micro USB B", "Microphone": "7 total microphones (1 removable in boom), 4 microphones for Active Noise Cancelling (ANC), 2 microphones for voice pickup, 1 microphone for gaming in the boom mic", "Bluetooth range": "Up to 33 ft (10 m)", "Bluetooth version": "4.2 (A2DP, HFP, AVRCP, BLE)", "Codec": "SBC and AAC"});
```

INSERT INTO

```
products(product_id,product_name,short_description,no_pictures,brand_id,price,
```

```
wireless,wired,earbuds,gaming,sport,bluetooth,stock,description,specification)
VALUES (26,'SoundComm B40 Headset','A professional communication
headset',6,5,849,1,0,0,1,0,0,15,'Bose Sport Open Earbuds are a truly different
kind of sport headphone. That s because they are the first of their kind that
do not go in your ears. Instead, they combine exclusive Bose OpenAudio
technology with a unique open-ear design to produce high-quality sound without
covering your ears. The city. The surf. Your running buddy. Hear it all and
your music at the same time with these entirely new, entirely open sport
earbuds.','{"Microphone type": "150-ohm Dynamic Noise Cancelling", "Headset
variants": "4-pin female XLR connector, 5-pin male XLR connector", "SoundComm
B40 Headset": "8 inch H x 6.5 inch W x 3.5 inch D", "On-head weight (not
including cable) Single earcup": "10 oz", "On-head weight (not including
cable) Dual earcup": "12.7 oz", "Cable length - Headset to control module":
"4.8 ft", "Cable length - Control module to XLR connector": "10 inch"}');
```

```
-- Table Name: status.sql
```

```
DROP TABLE IF EXISTS status;
```

```
CREATE TABLE IF NOT EXISTS status (
  status_id int(1) NOT NULL PRIMARY KEY,
  status_name varchar(20) NOT NULL
);
```

```
INSERT INTO status (status_id, status_name) VALUES
(1, 'paid'),
(2, 'confirmed'),
(3, 'shipped'),
(4, 'received');
```

```
-- Table Name: users.sql
```

```
DROP TABLE IF EXISTS users;
```

```
CREATE TABLE IF NOT EXISTS users(
  user_id INT NOT NULL PRIMARY KEY AUTO_INCREMENT
  ,email VARCHAR(50) NOT NULL
  ,password VARCHAR(50) NOT NULL
  ,first_name VARCHAR(50) NOT NULL
  ,last_name VARCHAR(50) NOT NULL
  ,contact VARCHAR(10) NOT NULL
);
```

```
INSERT INTO users (user_id, email, password, first_name, last_name, contact)
VALUES (1, 'haoranwei@test.com', 'haoranwei@test.com', 'haoran', 'wei',
'88888888');
```

```
INSERT INTO users (user_id, email, password, first_name, last_name, contact)
VALUES (2, 'sining040@gmail.com', 'sining040@gmail.com', 'sining', 'lin',
'88888888');
```