

## Introduction:

This is a Hi-fi webstore.

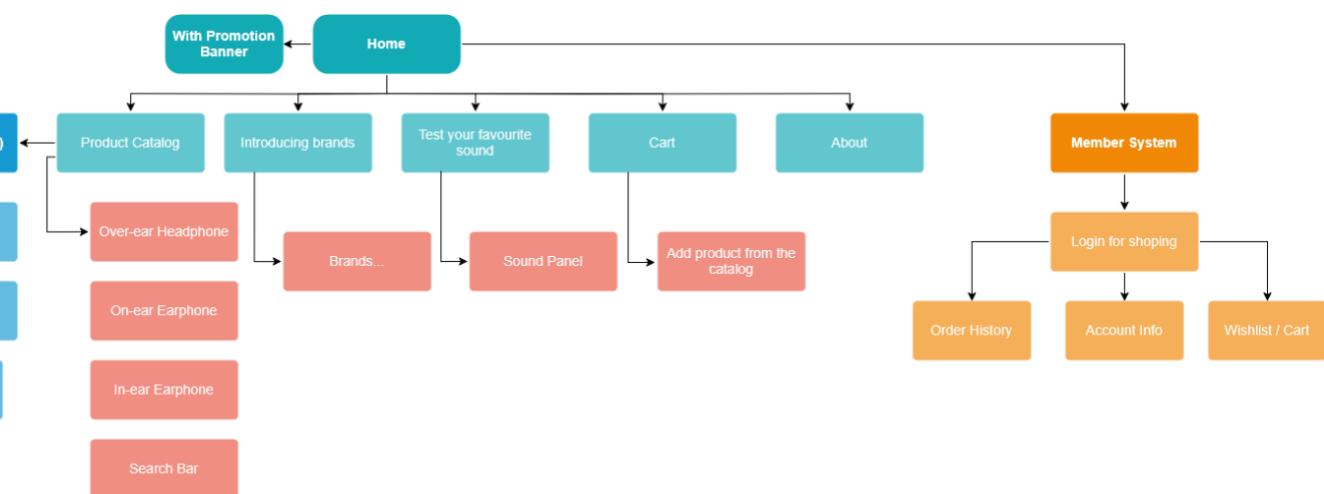
Providing a test soundtrack, which is from online recordings with a human-like headset wearing the headphones from the product catalog, although the actual feeling(feel of listening) will not be the same as the reality, but it is just for a reference.

## The additional function:

There is a page used for choosing different tones, which is an audition room, there is an amplifier setup(decoration) in the middle of the room, which allows users to select different tones to listen to different styles of music they prefer.

Explain: a function to provide different tones for the user to choose, after the selection of those tones, they will be subdivided and ultimately will come out to a few models of headphones for the user to choose from that fits their style. Earphones recommended to the user are according to the three tuning styles of headset e.g. bright, flat, bass, etc.

## Website Structure:



## Website Layout:

# Home



Search Bar



LOGIN/REGISTER



Cart

By Brand

Headphones

Speaker

Amplifier

About

Promotion Banner



## Latest Products



Product 1 Name

\$99 ~~-\$199~~

Add to Cart



Product 2 Name

\$289 ~~-\$10~~

Add to Cart



Product 3 Name

\$169 ~~-\$50~~

Add to Cart

## Sub-menu for headphone

By Brand

Headphones

Speaker

Amplifier

About

In-ear

Half-in ear

Over-ear

## Sub-menu for brands

By Brand

Headphones

Speaker

Amplifier

About

Hot Brands

ALL Brands



# Cart



Search Bar



LOGIN/REGISTER



By Brand

Headphones

Speaker

Amplifier

About

[Continue Shopping](#)

## Cart



Product 1 Name

[Delete](#)

Quantity

\$99 ~~\$199~~

Product 2 Name

[Delete](#)

Quantity

\$29 ~~-\$9~~

Total: HKD\$98

[Checkout](#)

# Register



Search Bar



LOGIN/REGISTER



By Brand

Headphones

Speaker

Amplifier

About

**Register Your Account Here!**

Username:

Email:

Password:

Re-enter your password:

[Create](#)

# LOGIN



Search Bar



LOGIN/REGISTER



By Brand

Headphones

Speaker

Amplifier

About

## Welcome Back!

Username/Email: Password: [Forgot your password?](#)[New user?](#)

## Account Info



Search Bar



LOGIN/REGISTER



By Brand

Headphones

Speaker

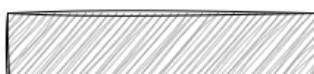
Amplifier

About

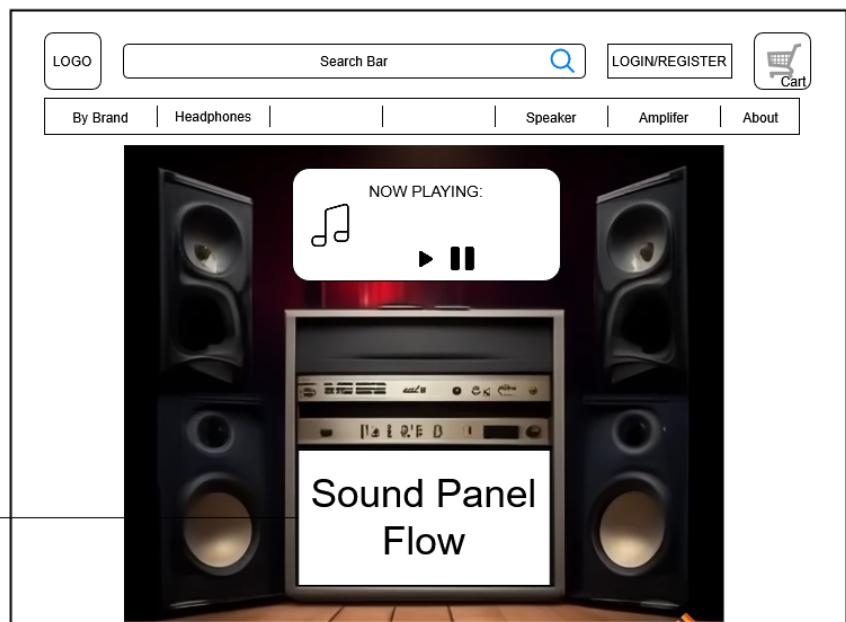
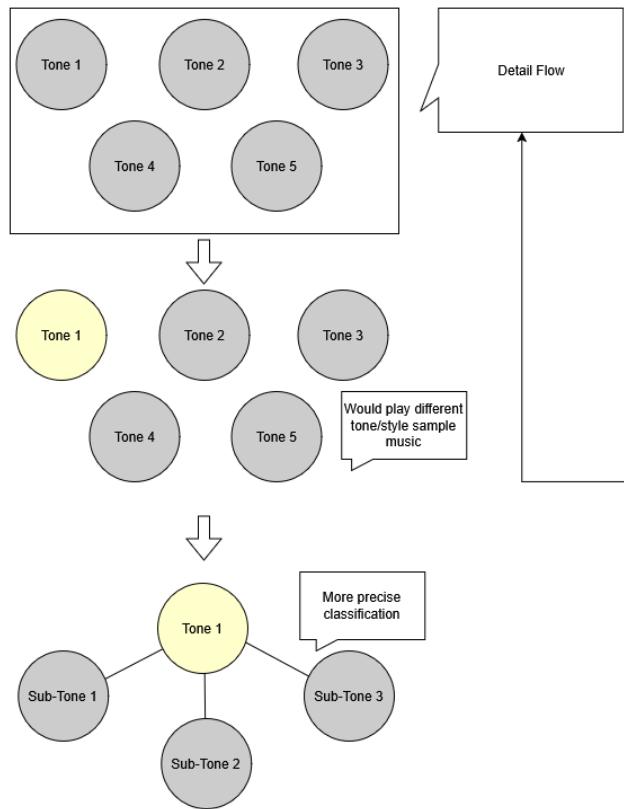
### Account Information

Username: Email: Old Password: New Password: Re-enter your password: 

### Useful functions



# Sound Panel



1) Preparation of multimedia (logo, banner, animation/video)

2) Implementation of Home page and Product page

A. General

The overview of the home page



## Latest Products

**Sennhiser IE800s**  
\$19.99  
What to add  
Add to Cart

**Sony MDR-MV1**  
\$19.99  
What to add  
Add to Cart

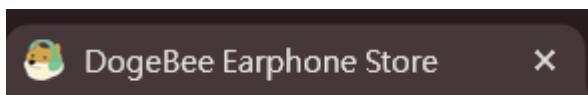
**Philips X3**  
\$19.99  
What to add  
Add to Cart

## HTML (Header)

```
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="./css&js/index_ai.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.7.2/css/all.min.css">
    <link rel="stylesheet" href="./css&js/product.css">
    <link rel="icon" href="photos/doge_logo_v2.png" type="image/png" />
    <title>DogeBee Earphone Store</title>
</head>
```

The following is included in the page inside the `<head>` element.

- The external stylesheet `index_ai.css`, `all.min.css`, `product.css` is linked in all pages with `<link>` element.
- The title for each page is included using the `<title>` element and the icon for the website is included using the `<link>` element. It will show as below:



Top of the navigation bar:

```
1 <div class="topnav">
2     <nav class="navbar bg-body-tertiary">
3         <div class="container">
4             <a class="navbar-brand" href="#">
5                 
6             </a>
7         </div>
8         <div class="header" style="color:rgb(255, 255, 255);">
9             <h1>DogeBee</h1>
10            <p class="tagline">Welcome to DogeBee Earphone Store!</p>
11        </div>
12    </nav>
13 </div>
```

This allows the logo below to show by using `` pattern.

The `<h1>` and `<p>` are showing the store name and the welcoming sentence.

CSS:

`display: inline-block;` does not add a line-break after the element, so the element can sit next to other elements.

It allows you to set width and height on the element, in the code it is set as `height: 72px;` and `width: auto;`.

```
1  /* ===== Header Styles ===== */
2  #logo {
3      display: inline-block;
4      height: 72px;
5      width: auto;
6  }
7
8  .header {
9      padding: 26px; /* width of logo bar */
10     text-align: left;
11     background: #000;
12 }
13
14 .header h1 {
15     font-size: 30px;
16     color: white;
17 }
18
19 .tagline {
20     margin: 5px 0 0;
21     font-size: 0.9rem;
22     color: #fff;
23 }
```

Effect:



HTML of Navigation Bar:

There are buttons for the Home page, products page, a dropdown list for browsing all the brands, a search bar, cart and user icon.

All the buttons are grouped by a CSS called `search-container`, it plays a part in aligning all the elements in a line.

```
1 <div class="navbar">
2     <a href="#">Home</a>
3     <a href="products.html">Products</a>
4     <div class="dropdown">
5         <button class="dropbtn">Browse by Brand
6             <i class="fa fa-caret-down"></i>
7         </button>
8         <div class="dropdown-content">
9             <a href="#">Sennhiser</a>
10            <a href="#">Audio Technica</a>
11            <a href="#">Shure</a>
12        </div>
13    </div>
14    <a href="products.html">Headphones</a>
15    <a href="about.html">About</a>
16    <div class="nav-utilities">
17        <!-- Search Bar -->
18        <div class="search-container">
19            <input type="text" placeholder="Search..." class="search-input">
20            <button class="search-btn"><i class="fa fa-search"></i></button>
21        </div>
22
23        <!-- Icons -->
24        <div class="nav-icons">
25            <a href="#" class="cart-icon" aria-label="Shopping Cart">
26                <i class="fa-solid fa-cart-shopping"></i>
27                <!-- <span class="cart-count">0</span> -->
28            </a>
29            <a href="#" class="user-icon" aria-label="User Account">
30                <i class="fa-solid fa-user"></i>
31            </a>
32        </div>
33    </div>
34 </div>
```

Effect:



The dropdown menu of `Browse by Brand` button

[Home](#)[Products](#)[Browse by Brand ▾](#)[Headphones](#)[About](#)[Sennhiser](#)[Audio Technica](#)[Shure](#)

1 / 3



CSS:

The buttons in navigation bar will change their text and background colour when the user hovers the cursor shown as above.

The original text and background colour are white and dark grey respectively, the text and background colour changed to black and light pink after hovering.

```
1 .navbar a:hover,
2 .dropdown:hover .dropbtn,
3 .topnav a:hover {
4   background-color: #f9f9f9;
5   color: black;
6 }
7
8 .dropdown-content {
9   display: none;
10  position: absolute;
11  background-color: #f9f9f9;
12  min-width: 170px;
13  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
14  z-index: 2;
15 }
```

Slideshow (by external JavaScript file; .js)

The variable `slideIndex` is to change the index of the slideshow (picture)

The `startSlideShow()` function is for calling the for loop to encounter the operation of slideshow. It changes the slideshow 5 seconds per slide, user can move the slideshow on their own too by the function `plusSlides()`.

HTML:

```

1 <div class="slideshow-container">
2   <div class="mySlides fade">
3     <div class="numbertext">1 / 3</div>
4     
5     <!-- <div class="text">Caption Text</div> -->
6   </div>
7
8   <div class="mySlides fade">
9     <div class="numbertext">2 / 3</div>
10    
11    <div class="text">Promotion</div>
12  </div>
13
14  <div class="mySlides fade">
15    <div class="numbertext">3 / 3</div>
16    
17    <div class="text">Wallpaper</div>
18  </div>
19
20  <a class="prev" onclick="plusSlides(-1)">&#10094;</a>
21  <a class="next" onclick="plusSlides(1)">&#10095;</a>
22 </div>
23 <br>
24
25 <div style="text-align:center">
26   <span class="dot" onclick="currentSlide(1)"></span>
27   <span class="dot" onclick="currentSlide(2)"></span>
28   <span class="dot" onclick="currentSlide(3)"></span>
29 </div>
30
31 <script src=".//css&js/slideshow.js"></script>

```

```

1 let slideIndex = 1;
2 let slideInterval;
3
4 // Initialize the slideshow
5 showSlides(slideIndex);
6 startSlideShow();
7
8 function plusSlides(n) {
9   clearInterval(slideInterval);
10  showSlides(slideIndex += n);
11  startSlideShow();
12 }
13
14 function currentSlide(n) {
15  clearInterval(slideInterval);
16  showSlides(slideIndex = n);
17  startSlideShow();
18 }
19
20 function showSlides(n) {
21  let i;
22  let slides = document.getElementsByClassName("mySlides");
23  let dots = document.getElementsByClassName("dot");
24
25  if (n > slides.length) {slideIndex = 1}
26  if (n < 1) {slideIndex = slides.length}
27
28  for (i = 0; i < slides.length; i++) {
29    slides[i].style.display = "none";
30  }
31
32  for (i = 0; i < dots.length; i++) {
33    dots[i].className = dots[i].className.replace(" active", "");
34  }
35
36  slides[slideIndex-1].style.display = "block";
37  dots[slideIndex-1].className += " active";
38 }
39
40 function startSlideShow() {
41  // Clear any existing interval first
42  clearInterval(slideInterval);
43  // Set new interval (5000ms = 5 seconds)
44  slideInterval = setInterval(function() {
45    plusSlides(1);
46  }, 5000);
47 }

```

`<script> </script>` is to link to the javascript at the right side.

HTML of product cards located at the latest product session:

The css `.row` is to group the product cards below in order to stay in a horizontal alignment, and the `.block` is to set the width of each product card.

```

1 .row {
2   width: 100%;
3   margin: 0 auto;
4   display: flex;
5   justify-content: center; /* for centering 3 blocks in the center */
6   /*justify-content: space-between; /* for space in between */
7 }
8 .block {
9   width: 32%;
10 }

```

```

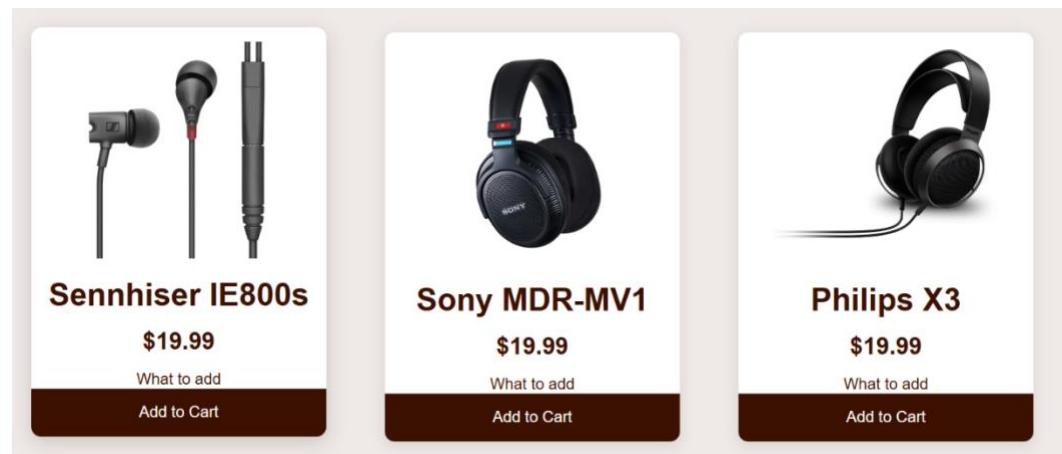
1 <h2>
2   Latest Products
3 </h2>
4 <div class="row">
5   <div class="card block">
6     
7     <h1>Sennhiser IE800s</h1>
8     <p class="price">$19.99</p>
9     <p>What to add</p>
10    <p><button>Add to Cart</button></p>
11  </div>
12  <div class="card block">
13    
14    <h1>Sony MDR-MV1</h1>
15    <p class="price">$19.99</p>
16    <p>What to add</p>
17    <p><button>Add to Cart</button></p>
18  </div>
19  <div class="card block">
20    
21    <h1>Philips X3</h1>
22    <p class="price">$19.99</p>
23    <p>What to add</p>
24    <p><button>Add to Cart</button></p>
25  </div>
26 </div>

```

The product card will shift up a bit and add a shadow effect when user hovers on it.

It is done by the css below.

```
1 .card:hover {  
2     transform: translateY(-5px);  
3     box-shadow: 0 6px 20px rgba(0, 0, 0, 0.15);  
4 }
```



## Products Page

Products filter:

This is the code of the products filtering side bar.

```
1 <div class="container_product">  
2     <!-- Filter Sidebar -->  
3     <div class="filter-sidebar">  
4         <div class="filter-header">  
5             <h2>Filter</h2>  
6             <button class="reset-btn">Reset</button>  
7         </div>  
8         <div class="product-count">3 Products</div>  
9  
10        <div class="filter-group">  
11            <h3>Catagories</h3>  
12            <ul class="filter-options">  
13                <li>  
14                    <input type="checkbox" id="category1" checked>  
15                    <label for="category1">Headphones (3)</label>  
16                </li>  
17                <li>  
18                    <input type="checkbox" id="category2">  
19                    <label for="category2">Earphone (17)</label>  
20                </li>  
21            </ul>  
22        </div>  
23  
24        <div class="filter-group">  
25            <h3>Brand</h3>  
26            <ul class="filter-options">  
27                <li>  
28                    <input type="checkbox" id="brand1" checked>  
29                    <label for="brand1">Sennhiser</label>  
30                </li>  
31                <li>  
32                    <input type="checkbox" id="brand2">  
33                    <label for="brand2">Audio Technica</label>  
34                </li>  
35            </ul>  
36        </div>  
37    </div>
```

Effect:

**Filter**      Reset

3 Products

Catagories

Headphones (3)

Earphone (17)

Brand

Sennhiser

Audio Technica

## Products Sorting:

HTML:

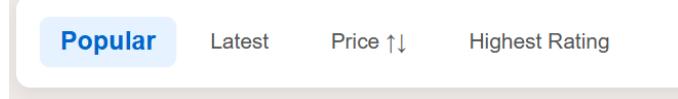
```
1 <div class="products-section">
2   <!-- Sort Options -->
3   <div class="sort-options">
4     <a href="#" class="active">Popular</a>
5     <a href="#">Latest</a>
6     <a href="#">Price ↑↓</a>
7     <a href="#">Highest Rating</a>
8   </div>
```

CSS:

```
1 /* Products Section */
2 .products-section {
3   flex: 1;
4   min-width: 0;
5 }
6
7 .sort-options {
8   background: white;
9   padding: 18px 25px;
10  border-radius: 10px;
11  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.08);
12  margin-bottom: 25px;
13  display: flex;
14  gap: 20px;
15  flex-wrap: wrap;
16 }
17
18 .sort-options a {
19   color: #555;
20   text-decoration: none;
21   font-size: 0.95rem;
22   padding: 6px 12px;
23   border-radius: 6px;
24   transition: all 0.2s ease;
25 }
26
27 .sort-options a:hover {
28   color: #0066cc;
29   background-color: #f0f7ff;
30 }
31
32 .sort-options a.active {
33   color: #0066cc;
34   font-weight: 600;
35   background-color: #e6f2ff;
36 }
```

```
1 /* Filter Sidebar */
2 .filter-sidebar {
3   width: 280px;
4   background: white;
5   padding: 25px;
6   border-radius: 10px;
7   box-shadow: 0 4px 12px rgba(0, 0, 0, 0.08);
8   position: sticky;
9   align-self: flex-start;
10  top: 20px;
11 }
12
13 .filter-header {
14   display: flex;
15   justify-content: space-between;
16   align-items: center;
17   margin-bottom: 18px;
18   padding-bottom: 12px;
19   border-bottom: 1px solid #eee;
20 }
21
22 .filter-header h2 {
23   font-size: 1.4rem;
24   font-weight: 600;
25   color: #222;
26 }
27
28 .filter-group {
29   margin-bottom: 28px;
30 }
31
32 .filter-group h3 {
33   font-size: 1rem;
34   margin-bottom: 15px;
35   color: #444;
36   font-weight: 500;
37   letter-spacing: 0.5px;
38 }
39
40 .filter-options {
41   list-style: none;
42 }
43
44 .filter-options li {
45   margin-bottom: 12px;
46   display: flex;
47   align-items: center;
48   transition: all 0.2s ease;
49 }
50
51 .filter-options li:hover {
52   transform: translateX(3px);
53 }
54
55 .filter-options input[type="checkbox"] {
56   margin-right: 10px;
57   width: 16px;
58   height: 16px;
59   accent-color: #0066cc;
60   cursor: pointer;
61 }
62
63 .filter-options label {
64   cursor: pointer;
65   font-size: 0.95rem;
66   color: #555;
67 }
68
69 .reset-btn {
70   color: #666;
71   background: none;
72   border: none;
73   cursor: pointer;
74   font-size: 0.9rem;
75   transition: color 0.2s;
76   padding: 4px 8px;
77   border-radius: 4px;
78 }
79
80 .reset-btn:hover {
81   color: #0066cc;
82   background-color: #f0f7ff;
83 }
84
85 .product-count {
86   color: #666;
87   font-size: 0.9rem;
88   margin-bottom: 25px;
89   padding-bottom: 15px;
90   border-bottom: 1px solid #eee;
91 }
```

Effect:

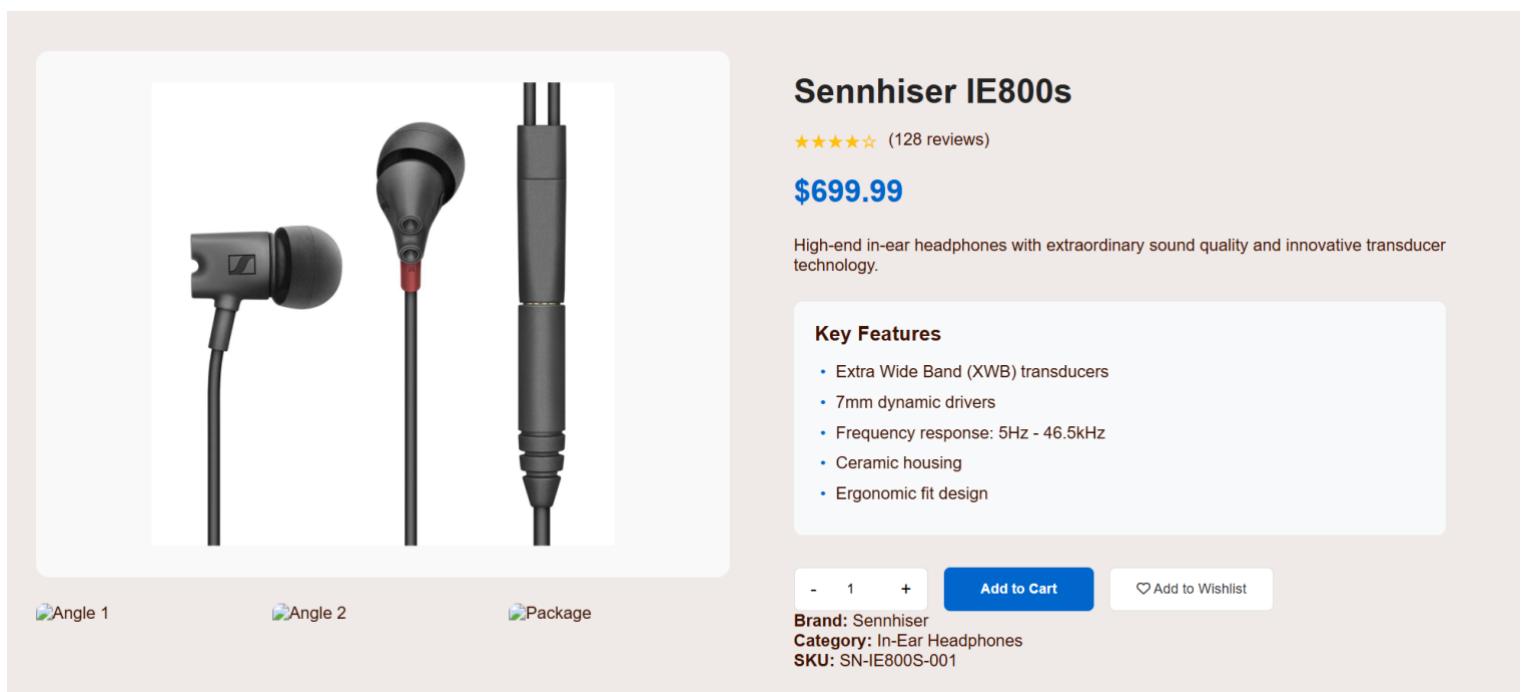


# Individual product page

The html of the product card ; When pressed, it would redirect to the individual product page

```
+1 <a href="./product_ie800s.html">
2   <div class="card block">
3     
4     <h1>Sennhiser IE800s</h1>
5     <p class="price">$19.99</p>
6     <p>What to add</p>
7     <p><button>Add to Cart</button></p>
8   </div>
+9 </a>
```

Overall effect:



This function is not finished:

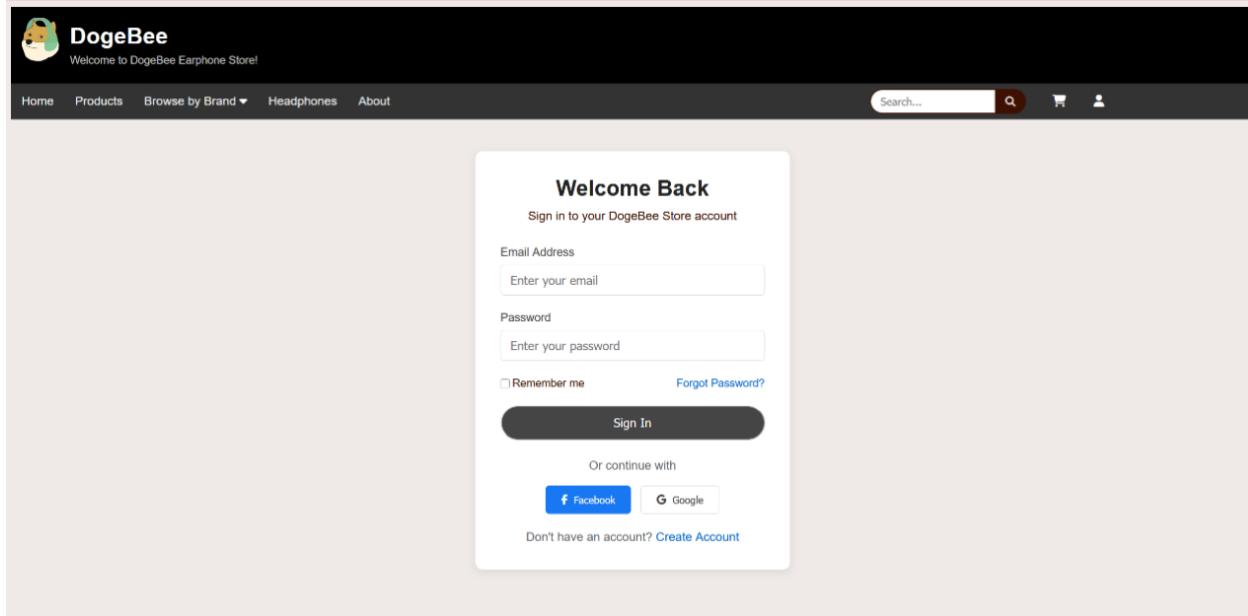
A screenshot of a partially developed product page for the Sony MDR-MV1 headphones. At the top, there are tabs for "Description" (which is underlined in blue), "Specifications", "Reviews (128)", and "Shipping". Below the tabs, a "Detailed Description" section contains the text: "The IE 800 S is the refined version of Sennheiser's legendary IE 800 in-ear headphones...". Underneath this, a "You May Also Like" section is visible, featuring a thumbnail of the Sony MDR-MV1 headphones with the text "Sony MDR-MV1" and the price "\$599.99". An "Add to Cart" button is located at the bottom of this section.



Javascript of quantity button of the product:

```
1  document.querySelectorAll('.quantity-selector').forEach(selector => {
2      const minusBtn = selector.querySelector('.qty-btn:first-child');
3      const plusBtn = selector.querySelector('.qty-btn:last-child');
4      const input = selector.querySelector('input');
5
6      minusBtn.addEventListener('click', () => {
7          const currentValue = parseInt(input.value) || 1;
8          input.value = Math.max(currentValue - 1, parseInt(input.min) || 1);
9          input.dispatchEvent(new Event('change'));
10     });
11
12     plusBtn.addEventListener('click', () => {
13         const currentValue = parseInt(input.value) || 1;
14         input.value = currentValue + 1;
15         input.dispatchEvent(new Event('change'));
16     });
17
18     input.addEventListener('input', () => {
19         const min = parseInt(input.min) || 1;
20         if (parseInt(input.value) < min) {
21             input.value = min;
22         }
23     });
24 });
25 });
```

## Overview of Registration and Login page:



This image compares the login and register pages. On the left, the "LOGIN:" section shows the "Welcome Back" login form. On the right, the "Register:" section shows a similar form titled "Welcome" with fields for "Email Address", "Username", and "Password", and a "Register" button. Both forms include social login options for Facebook and Google, and links for creating an account or logging in.

LOGIN:

Register:

Code:

The Sign in and register button contain a transaction when hovers on them.

The duration of the transaction is 0.4s while the background colour turns white and the text colour turns black.

```
1 .login-btn {  
2     background-color: #454545;  
3     color: #fff;  
4     width: 100%;  
5     padding: 14px;  
6     border-radius: 50px;  
7     border: solid 10px 15px;  
8     font-size: medium;  
9     transition: 0.4s;  
10 }  
11  
+12 .login-btn:hover {  
13     opacity: 0.9;  
14     background-color: #ffffff;  
15     color: #000000;  
16 }
```

## Overview of Login Page HTML:

```
1  <!-- Login Content -->
2  <div class="login-container">
3      <div class="login-header">
4          <h2>Welcome Back</h2>
5          <p>Sign in to your DogeBee Store account</p>
6      </div>
7
8      <form class="login-form">
9          <div class="form-group">
10             <label>Email Address</label>
11             <input type="email" placeholder="Enter your email" required>
12         </div>
13
14         <div class="form-group">
15             <label>Password</label>
16             <input type="password" placeholder="Enter your password" required>
17         </div>
18
19         <div class="login-options">
20             <div style="display: flex; width: auto;">
21                 <input type="checkbox" style="transform: scale(1); margin-right: 5px;"> <div style="min-width: max-content;">Remember me</div>
22             </div>
23             <a href="#">Forgot Password?</a>
24         </div>
25
26         <button type="submit" class="login-btn" style="width: 100%; padding: 14px">
27             Sign In
28         </button>
29
30         <div class="social-login">
31             <p>Or continue with</p>
32             <div class="social-buttons">
33                 <button type="button" class="social-btn facebook-btn">
34                     <i class="fab fa-facebook-f"></i>
35                     Facebook
36                 </button>
37                 <button type="button" class="social-btn google-btn">
38                     <i class="fab fa-google"></i>
39                     Google
40                 </button>
41             </div>
42         </div>
43
44         <p style="text-align: center; margin-top: 25px; color: #666">
45             Don't have an account? <a href=".register.html" style="color: #0066cc">Create Account</a>
46         </p>
47     </form>
48 </div>
```

## Users Account Database Structure:

The screenshot shows the 'Structure' tab of MySQL Workbench for a table named 'users'. The table has the following structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<b>id</b> 🛡	int(11)			No	None		AUTO_INCREMENT	<span>Change</span> <span>Drop</span> <span>More</span>
2	<b>name</b>	varchar(255)	utf8_general_ci		No	None			<span>Change</span> <span>Drop</span> <span>More</span>
3	<b>email</b>	varchar(255)	utf8_general_ci		No	None			<span>Change</span> <span>Drop</span> <span>More</span>
4	<b>password</b>	varchar(255)	utf8_general_ci		No	None			<span>Change</span> <span>Drop</span> <span>More</span>
5	<b>contact</b>	varchar(255)	utf8_general_ci		No	None			<span>Change</span> <span>Drop</span> <span>More</span>
6	<b>city</b>	varchar(255)	utf8_general_ci		No	None			<span>Change</span> <span>Drop</span> <span>More</span>
7	<b>address</b>	varchar(255)	utf8_general_ci		No	None			<span>Change</span> <span>Drop</span> <span>More</span>

Example of an user data inserted in the database:

phpMyAdmin

Server: 127.0.0.1 » Database: dogebee » Table: users

New

dogebee

New

items

users

information\_schema

mysql

performance\_schema

phpmyadmin

school

shop

test

Browse Structure SQL Search Insert Export Import Privileges Operations

Showing rows 0 - 0 (1 total, Query took 0.0001 seconds.)

SELECT \* FROM `users`

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all Number of rows: 25 Filter rows: Search this table

Extra options

	id	name	email	password	contact	city	address
<input type="checkbox"/>	1	Noah	s201432@ucc.edu.hk	12345678	23436677	Hong Kong	1, Lee Hong Lane

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

Label:  Let every user access this bookmark

Bookmark this SQL query

## User Account Management:

Effect:

### Account Information

View and update your personal details below.

**Email Address**  
eee@c

**Full Name**  
404

**Shipping Address**  
eee

**Contact Number**  
eee

**New Password**  
  
Leave blank if you don't want to change your password.

**Confirm New Password**

**Save Changes**

### Order History

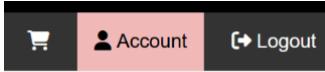
Review your past purchases.

Order ID: #ORD-68b31f2d0a7df Date: August 30, 2025 0

 Bright Sennheiser IE100 Pro  
Quantity: 1 • Price: HK\$780.00 each

Total: HK\$858.00



The interface in  navigation bar is in 'account'.

Code: This php code is to enquire SQL query to the database in order to retrieve the account information.

```

1 // 1. Check if user is logged in.
2 if (!isset($_SESSION["email"])) {
3     header("Location: login.html");
4     exit();
5 }
6
7 // 2. Establish Database Connections
8 $connectdb = mysqli_connect("localhost", "root", "", "dogebee");
9 mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
10 mysqli_set_charset($connectdb, "utf8");
11
12
13 $email = $_SESSION["email"];
14 $update_message = ""; // Variable for feedback messages.
15
16 // 3. Handle Account Update Form Submission (uses 'dogebee' database)
17 if ($_SERVER["REQUEST_METHOD"] == "POST") {
18     // Sanitize and retrieve form data
19     $name = mysqli_real_escape_string($connectdb, $_POST['name']);
20     $address = mysqli_real_escape_string($connectdb, $_POST['address']);
21     $contact = mysqli_real_escape_string($connectdb, $_POST['contact']);
22     $new_password = $_POST['new_password'];
23     $confirm_password = $_POST['confirm_password'];
24
25     $params = [$name, $address, $contact];
26     $types = "sss";
27     $updateQuery = "UPDATE users SET name = ?, address = ?, contact = ? WHERE email = ?";
28
29     // Handle password change
30     if (!empty($new_password)) {
31         if ($new_password === $confirm_password) {
32             $hashed_password = password_hash($new_password, PASSWORD_DEFAULT);
33             $updateQuery = "UPDATE users SET name = ?, address = ?, contact = ?, password = ? WHERE email = ?";
34             $params[] = $hashed_password;
35             $types .= "s";
36         } else {
37             $update_message = "<div class='error-message'>Error: New passwords do not match.</div>";
38         }
39     }
40
41     $params[] = $email;
42     $types .= "s";
43
44     // Proceed with the update if there are no errors
45     if (empty($update_message)) {
46         $stmt = mysqli_prepare($connectdb, $updateQuery);
47         mysqli_stmt_bind_param($stmt, $types, ...$params);
48         if (mysqli_stmt_execute($stmt)) {
49             $update_message = "<div class='success-message'>Account information updated successfully!</div>";
50         } else {
51             $update_message = "<div class='error-message'>An error occurred. Please try again.</div>";
52         }
53         mysqli_stmt_close($stmt);
54     }
55 }

```

## Order history:

Effect:

The screenshot shows a web page titled "Order History" with the subtitle "Review your past purchases.". At the top, it displays "Order ID: #ORD-68b31f2d0a7df" and "Date: August 30, 2025". Below this, there is a product listing for "Bright Sennheiser IE100 Pro" with a quantity of 1 and a price of HK\$780.00 each. The total amount shown is HK\$858.00.

Product	Quantity	Price
Bright Sennheiser IE100 Pro	1	HK\$780.00

Total: HK\$858.00

Code: This is for retrieving orders information in 'orders' table while only selecting the current user's email address

By using JOIN users ON  
orders.email =  
users.email WHERE  
users.email = ?

to join the users table

and connecting the email address from users to orders table.

The while function is to process the results fetched group items by the order\_id.

P.S. The order\_id is generated randomly.

```
1 $sql_fetch_user = "SELECT name, email, address, contact FROM users WHERE email = ?";
2 $stmt_user = mysqli_prepare($connectdb, $sql_fetch_user);
3 mysqli_stmt_bind_param($stmt_user, "s", $email);
4 mysqli_stmt_execute($stmt_user);
5 $result_user = mysqli_stmt_get_result($stmt_user);
6 $user = mysqli_fetch_assoc($result_user);
7 mysqli_stmt_close($stmt_user);
8
9 if (!$user) {
10     session_destroy();
11     header("Location: login.html");
12     exit();
13 }
14
15 // 5. Fetch Order History from the 'orders' database
16 $orders = [];
17 $sql_fetch_orders = "
18     SELECT
19         order_id, order_date, amount, status, quantity, item_name, image_url, price_per_item
20     FROM orders
21     JOIN users
22     ON orders.email = users.email
23     WHERE users.email = ?
24     ORDER BY order_date DESC, order_id DESC";
25
26 $stmt_orders = mysqli_prepare($connectdb, $sql_fetch_orders);
27 mysqli_stmt_bind_param($stmt_orders, "s", $email);
28 mysqli_stmt_execute($stmt_orders);
29 $result_orders = mysqli_stmt_get_result($stmt_orders);
30
31 // Process results and group items by order_id
32 while ($row = mysqli_fetch_assoc($result_orders)) {
33     $order_id = $row['order_id'];
34     if (!isset($orders[$order_id])) {
35         $orders[$order_id] = [
36             'details' => [
37                 'order_date' => $row['order_date'],
38                 'amount' => $row['amount'],
39                 'status' => $row['status']
40             ],
41             'items' => []
42         ];
43     }
44     $orders[$order_id]['items'][] = [
45         'name' => $row['item_name'],
46         'quantity' => $row['quantity'],
47         'price' => $row['price_per_item'],
48         'image_url' => $row['image_url']
49     ];
50 }
```

and

**Admin Functions [admin.php]:**

## Effect (Overview):

**Administration Page**  
Manage your product inventory and settings

40 Total Products		25 Headphones		15 Earphones		
ID	Brand	Name	Price	Category	Image	Actions
1	Sennheiser	IE800S	3,799.0 HKD	Neutral		 
2	Sennheiser	HD 660S	2,490.0 HKD	Neutral		 
3	Sennheiser	HD 490Pro	2,770.0 HKD	Neutral		 
4	Sennheiser	HD 660S2	3,040.0 HKD	Neutral		 
5	Sennheiser	HE-1	570,000.0 HKD	Neutral		 
6	Sennheiser	IE100 Pro	780.0 HKD	Bright		 
7	Sennheiser	IE600	3,580.0 HKD	Bright		 

Editing a product (Green button) [admin\_edit.php]:

**Edit Product**  
Update product information below

<b>Brand</b>	Sennheiser
<b>Product Name</b>	IE800S
<b>Price (HKD)</b>	3799
<b>Category</b>	Earphones (In-ear)
<b>Product Link</b>	./product_ie800s.html
<b>Current Image</b>	
<b>Change Image</b>	<input type="button" value="Browse..."/> No file selected.
<b>Update Product</b> <input type="button" value="Cancel"/>	

Code of fetching information from previous page i.e.

admin.php and database:

The product ID is passed to admin\_edit.php by GET method thereby set as the session product ID

There is a feature that if the product is not found on the database, it redirects back to the admin page.

```

1 // Get product ID safely
2 $product_id = isset($_GET["productID"]) ? (int)$_GET["productID"] : 0;
3 $_SESSION["productID-edit"] = $product_id;
4
5 // Get product data using prepared statement
6 $sql_check = "SELECT * FROM items WHERE id = ?";
7 $stmt = mysqli_prepare($connectdb, $sql_check);
8 mysqli_stmt_bind_param($stmt, "i", $product_id);
9 mysqli_stmt_execute($stmt);
10 $result = mysqli_stmt_get_result($stmt);
11 $row = mysqli_fetch_row($result);
12 mysqli_stmt_close($stmt);
13
14 // If product not found, redirect
+15 if (!$row) {
+16     header("Location: admin.php");
+17     exit();
+18 }
```

Input validation of admin\_edit.php:

This javascript is to prevent misinput and unreasonable inputs being inputted to database.

i.e. The price must be greater than zero.



```
1 <script>
2 // form validation
3 document.querySelector('form').addEventListener('submit', function(e) {
4     const price = document.querySelector('input[name="price"]');
5     if (price.value <= 0) {
6         alert('Price must be greater than 0');
7         e.preventDefault();
8     }
9 });
10 </script>
```

### Admin\_edit\_submit.php:

This is for replacing the image URL of the old one and update it on database, and uploads the new product image to the server.



```
1 // Handle image upload - using column index 6 for image path same as products.php
2 if (isset($_FILES["image"]) && $_FILES["image"]["error"] == UPLOAD_ERR_OK) {
3     $target_dir = "./photos/"; // Match the directory used in products.php
4     $target_file = $target_dir . basename($_FILES["image"]["name"]);
5
6     // Get current image path first
7     $sql_current_image = "SELECT * FROM items WHERE id = ?";
8     $stmt = mysqli_prepare($connectdb, $sql_current_image);
9     mysqli_stmt_bind_param($stmt, "i", $product_id);
10    mysqli_stmt_execute($stmt);
11    $result = mysqli_stmt_get_result($stmt);
12    $row = mysqli_fetch_row($result);
13    mysqli_stmt_close($stmt);
14
15    $current_image = $row[6] ?? '';
16
17    // Delete old image if it exists
18    if (!empty($current_image) && file_exists($current_image)) {
19        unlink($current_image);
20    }
21
22    // Upload new image
23    if (move_uploaded_file($_FILES["image"]["tmp_name"], $target_file)) {
24        $image_path = $target_file;
25    }
26 } else {
27     // Keep existing image if no new one uploaded
28     $sql_current_image = "SELECT * FROM items WHERE id = ?";
29     $stmt = mysqli_prepare($connectdb, $sql_current_image);
30     mysqli_stmt_bind_param($stmt, "i", $product_id);
31     mysqli_stmt_execute($stmt);
32     $result = mysqli_stmt_get_result($stmt);
33     $row = mysqli_fetch_row($result);
34     mysqli_stmt_close($stmt);
35
36     $image_path = $row[6] ?? '';
37 }
38
39 // Update product using prepared statement
40 $sql_edit = "UPDATE items SET
41     brand = ?,
42     products = ?,
43     price = ?,
44     remarks = ?,
45     pictures = ?,
46     link = ?
47     WHERE id = ?";
48
49 $stmt = mysqli_prepare($connectdb, $sql_edit);
50 mysqli_stmt_bind_param($stmt, "ssdsssi",
51     $brand,
52     $name,
53     $price,
54     $remarks,
55     $image_path,
56     $product_link,
57     $product_id);
58
59 mysqli_stmt_execute($stmt);
```

### Admin.php [product deletion]:

```
1 <div class="action-icons">
2   <a href="admin_edit.php?productID=<?= $row[0] ?>" class="edit-btn" title="Edit">
3     <i class="fas fa-pen-to-square"></i>
4   </a>
5   <a href="admin.php?delete=<?= $row[0] ?>" class="delete-btn" title="Delete" onclick="return confirm('Are you sure you want to delete this product?');">
6     <i class="fas fa-trash-can"></i>
7   </a>
8 </div>
```

The GET request is passed to the code below:

```
1 // Handle product deletion
2 if (isset($_GET['delete'])) {
3   $productId = $_GET['delete'];
4   $deleteQuery = "DELETE FROM items WHERE id = ?";
5   $stmt = mysqli_prepare($connectdb, $deleteQuery);
6   mysqli_stmt_bind_param($stmt, "i", $productId);
7   mysqli_stmt_execute($stmt);
8   mysqli_stmt_close($stmt);
9   header("Location: admin.php");
10  exit();
11 }
```

**Adding new product [admin\_add.php & admin\_add\_submit.php]:**

Button from admin.php

42	er	er	20.0 HKD	Neutral			
----	----	----	----------	---------	---	---	---

+ Add New Product

Effect:

## + Add New Product

Enter product information below

**Brand**  
Enter Brand

**Product Name**  
Enter Product Name

**Price (HKD)**  
Enter Price

**Style**  
neutral

**Category**  
Headphones

**Product Link**  
Enter Product URL

**Product Image**  
Browse... No file selected.

Add Product    Cancel

Code of admin\_add\_submit.php:

```

1 // Handle form submission
2 if ($_SERVER["REQUEST_METHOD"] == "POST") {
3     // Get form data
4     $brand = mysqli_real_escape_string($connectdb, $_POST["brand"]);
5     $name = mysqli_real_escape_string($connectdb, $_POST["name"]);
6     $price = (float)$_POST["price"];
7     $style = mysqli_real_escape_string($connectdb, $_POST["style"]);
8     $remarks = mysqli_real_escape_string($connectdb, $_POST["remarks"]);
9     $product_link = mysqli_real_escape_string($connectdb, $_POST["product_link"] ?? '');
10
11    // Handle file upload
12    $image_path = '';
13    if (is_uploaded_file($_FILES["image"]) && $_FILES["image"]["error"] == UPLOAD_ERR_OK) {
14        $target_dir = "photos/";
15        $target_file = $target_dir . basename($_FILES["image"]["name"]);
16        $imageFileType = strtolower(pathinfo($target_file, PATHINFO_EXTENSION));
17
18        // Check if image file is a actual image
19        $check = getimagesize($_FILES["image"]["tmp_name"]);
20        if ($check !== false) {
21            // Generate unique filename
22            $new_filename = uniqid() . '.' . $imageFileType;
23            $target_file = $target_dir . $new_filename;
24
25            if (move_uploaded_file($_FILES["image"]["tmp_name"], $target_file)) {
26                $image_path = $target_file;
27            }
28        }
29    }
30
31    // Insert into database - corrected to match the form fields and database columns
32    $insertQuery = "INSERT INTO items (brand, products, price, style, remarks, pictures, link)
33                  VALUES (?, ?, ?, ?, ?, ?, ?)";
34    $stmt = mysqli_prepare($connectdb, $insertQuery);
35    mysqli_stmt_bind_param($stmt, "ssssss", $brand, $name, $price, $style, $remarks, $image_path, $product_link);
36    mysqli_stmt_execute($stmt);
37    mysqli_stmt_close($stmt);
38
39    header("Location: admin.php");
40    exit();

```

## Admin Orders Management [admin\_orders.php]:

Effect:

The screenshot shows the 'Order Management' section of the admin dashboard. At the top, there's a search bar labeled 'Search Orders' and a dropdown menu 'Filter by Status' set to 'All Statuses'. Below this, a table lists three orders:

Order ID	Customer	Date	Items	Amount	Status	Actions
#ORD-68b31f2d0a7df	404 eee@c	Aug 30, 2025	1 item(s) Bright Sennheiser IE100 Pro (Qty: 1)...	HK\$858.00	0	<a href="#">View</a> <a href="#">Edit</a>
#ORD-68b31ef83156a	Noah s201432@ucc.edu.hk	Aug 30, 2025	2 item(s) Neutral Sennheiser HD 490Pro (Qty: 1), Neutral S...	HK\$5,786.00	0	<a href="#">View</a> <a href="#">Edit</a>
#ORD-68b31ee241458	Noah s201432@ucc.edu.hk	Aug 30, 2025	1 item(s) Neutral Sennheiser IE800S (Qty: 1)...	HK\$4,179.00	0	<a href="#">View</a> <a href="#">Edit</a>

Code of fetching orders from the database:

There is a search for orders function which is done by the highlighted code below:

```
1 // Handle search and filter
2 $search_term = "";
3 $status_filter = "";
4 $where_conditions = [];
5
6 if (isset($_GET['search']) && !empty($_GET['search'])) {
7     $search_term = mysqli_real_escape_string($connectdb, $_GET['search']);
8     $where_conditions[] = "(orders.order_id LIKE '%$search_term%' OR orders.email LIKE '%$search_term%' OR orders.item_name LIKE '%$search_term%')");
9 }
10
11 if (isset($_GET['status']) && !empty($_GET['status'])) {
12     $status_filter = mysqli_real_escape_string($connectdb, $_GET['status']);
13     $where_conditions[] = "orders.status = '$status_filter'";
14 }
15
16 // Build the WHERE clause
17 $where_clause = "";
18 if (!empty($where_conditions)) {
19     $where_clause = "WHERE ". implode(" AND ", $where_conditions);
20 }
21
22 // Fetch all orders
23 $sql_orders = "
24     SELECT
25         orders.order_id,
26         orders.email,
27         users.name as customer_name,
28         orders.order_date,
29         orders.amount,
30         orders.status,
31         GROUP_CONCAT(CONCAT(orders.item_name, ' (Qty: ', orders.quantity, ')') SEPARATOR ', ') as items,
32         COUNT(*) as item_count
33     FROM orders
34     JOIN users ON orders.email = users.email
35     $where_clause
36     GROUP BY orders.order_id
37     ORDER BY orders.order_date DESC, orders.order_id DESC
38 ";
39
40 $result_orders = mysqli_query($connectdb, $sql_orders);
41 $orders = mysqli_fetch_all($result_orders, MYSQLI_ASSOC);
42
43 // Get unique status values for filter dropdown
44 $status_result = mysqli_query($connectdb, "SELECT DISTINCT status FROM orders ORDER BY status");
45 $status_options = [];
46 while ($row = mysqli_fetch_assoc($status_result)) {
47     $status_options[] = $row['status'];
48 }
```

## Checkout function in cart.php:

Effect; the second picture is the effect after pressing “Proceed to checkout” button:

Noted that the `order_id` is generated randomly.

Your Shopping Cart

Product	Total
Sennheiser HD 490Pro Neutral 2,770.0 HKD	2,770.0 HKD

- 1 + Update

**Order Summary**

Subtotal	2,770.0 HKD
Shipping	Free
Tax	277.0 HKD
<b>Total</b>	<b>3,047.0 HKD</b>

**Proceed to Checkout**

Continue Shopping



## Thank You For Your Order!

Your order has been placed successfully. A confirmation has been sent to your email.

Your Order ID is:

**ORD-68b812c502009**

[Continue Shopping](#)

[View Order History](#)

Pressing the “View Order History” button can be redirected to account.php:

The effect is shown below.

## ⌚ Order History

Review your past purchases.

Order ID: #**ORD-68b812c502009** Date: **September 3, 2025** 0

**Neutral Sennheiser HD 490Pro**  
Quantity: 1 • Price: HK\$2,770.00 each

**Total: HK\$3,047.00**

Code:

The order ID is generated randomly by the uniqid('ORD-') function in line 16.

The order date is stored as the format of [Year-Month-Day Hour:Minute:Second] by using the date function in php in line 17.

The insert query to the orders database table of the order is located in line 21 to 23.

Eventually, we have to clear the shopping cart after checking out successfully so in line 53, we are using the unset function in the session to clear the cart history.

```

1 // Store cart contents and total before clearing the session cart
2 $processed_cart = $_SESSION['cart'];
3 $subtotal = 0;
4 foreach ($processed_cart as $item) {
5     $subtotal += $item['price'] * $item['quantity'];
6 }
7 $total_amount = $subtotal * 1.1; // Assuming 10% tax as in cart.php
8
9 // Database Connection
10 $connectdb = mysqli_connect("localhost", "root", "", "dogebee");
11 mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
12 mysqli_set_charset($connectdb, "utf8");
13
14 // Prepare data for insertion
15 $email = $_SESSION["email"];
16 $order_id = uniqid('ORD-'); // Generate a unique order ID, e.g., "ORD-631a0b3de8e7b"
17 $order_date = date('Y-m-d H:i:s'); // Current timestamp
18 $status = 'Processing';
19
20 // Insert each item into the 'orders' table
21 $insertQuery = "INSERT INTO orders
22     (order_id, email, order_date, amount, status, quantity, item_name, image_url, price_per_item)
23     VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
24
25 $stmt = mysqli_prepare($connectdb, $insertQuery);
26
27 foreach ($processed_cart as $item) {
28     $itemName = $item['brand'] . ' ' . $item['name'];
29     $itemPrice = $item['price'];
30     $itemQuantity = $item['quantity'];
31     $itemImage = $item['image'];
32
33     mysqli_stmt_bind_param(
34         $stmt,
35         "ssssdisss",
36         $order_id,
37         $email,
38         $order_date,
39         $total_amount,
40         $status,
41         $itemQuantity,
42         $itemName,
43         $itemImage,
44         $itemPrice
45     );
46     mysqli_stmt_execute($stmt);
47 }
48
49 mysqli_stmt_close($stmt);
50 mysqli_close($connectdb);
51
52 // Clear the shopping cart after successful checkout
53 unset($_SESSION['cart']);

```

## Navigation Bar Product Search function [search.php]:

Search Bar Effect:

Code:

After pressing the search button, the searching parameter is passed to seach.php in a format of search.php?q={search parameter} received as the \$search\_query variable. Therefore seach the product to see if it exists in the database.

```
1 require_once 'config.php'; // Database connection
2
3 // Get search query from URL parameter
4 $search_query = isset($_GET['q']) ? trim($_GET['q']) : '';
5
6 // Initialize variables
7 $products = [];
8 $message = '';
9
10 // If there's a search query, search the database
11 if (!empty($search_query)) {
12     try {
13         // Prepare SQL query to search products
14         $sql = "SELECT * FROM items
15             WHERE products LIKE :query
16             OR brand LIKE :query
17             OR style LIKE :query
18             ORDER BY products";
19
20         $stmt = $pdo->prepare($sql);
21         $stmt->execute([':query' => "%$search_query%"]);
22         $products = $stmt->fetchAll(PDO::FETCH_ASSOC);
23
24         if (empty($products)) {
25             $message = "No products found for '$search_query'";
26         }
27     } catch (PDOException $e) {
28         $message = "Error searching products: " . $e->getMessage();
29     }
30 } else {
31     $message = "Please enter a search term";
32 }
```

## Special Features:

It is a function for finding the most suitable earphone for the users. Users have to answer few questions to find the most suitable earphone.

Effect:

Step 1

### Find Your Perfect Earphones

Answer a few questions to discover earphones that match your preferences.

Let's Get Started

Next →

Step 2

### Feel the sound difference, choose the most suitable sound.

Please choose your favourite sound according to what you have listened.



Clear and Bright

Good Treble



Warm and Thick

Good in Middle and Bass



Neutral

Presenting the true sound



Bass

Great Dynamics



Spacious

Spacial Audio

← Back

Next →

Step 3

## Which sound signature do you prefer?

Strong bass

Balanced sound

Bright treble

V-shaped  
(emphasized bass  
and treble)

← Back

Next →

Step 4

## What's your preferred earphone type?

In-ear

Earbuds

Over-ear

True wireless

← Back

Next →

Step 5

## What's your budget range?

Under \$500

\$500 - \$1500

\$1500 - \$3000

\$3000+

← Back

Next →

## Step 6

### Which feature is most important to you?

Noise cancellation

Battery life

Comfort

Sound quality

← Back

See Results →

## Step 7

### Your Perfect Earphones!

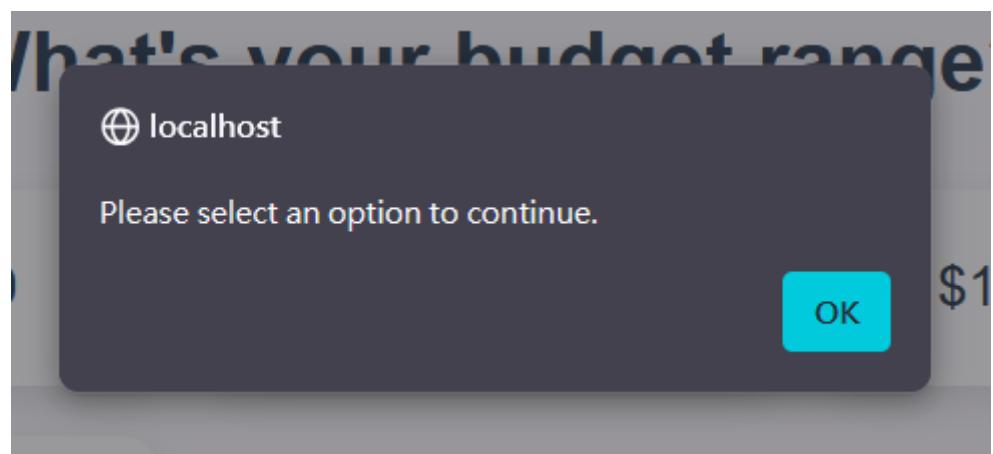
Based on your preferences, we recommend the following:

Audio-Technica ATH-M50x

These professional monitor headphones provide exceptional clarity with deep, accurate bass response. They're perfect for critical listening and studio work.

Start Over C

If there are no options chosen, the server will alert that after pressing the next button.



## Code:

```
1 $(document).ready(function() {
2     const entryElements = $('.entry');
3     const nextBtn = $('.next_btn');
4     const backBtn = $('.back_btn');
5     const restartBtn = $('#restart');
6     const progressBar = $('.progress-bar');
7     let entryIndex = 0;
8     const userSelections = {};
9
10    function updateUI() {
11        entryElements.hide();
12        $(entryElements[entryIndex]).show();
13
14        // Progress Bar
15        const progress = (entryIndex / (entryElements.length - 2)) * 100;
16        progressBar.css('width', progress + '%');
17
18        // Button Visibility
19        backBtn.toggle(entryIndex > 0 && entryIndex < entryElements.length - 1);
20        nextBtn.toggle(entryIndex < entryElements.length - 1);
21        restartBtn.toggle(entryIndex === entryElements.length - 1);
22
23        // Update Next Button Text
24        if (entryIndex === entryElements.length - 2) {
25            nextBtn.html('See Results <i class="fas fa-arrow-right"></i>');
26        } else {
27            nextBtn.html('Next <i class="fas fa-arrow-right"></i>');
28        }
29    }
30
31    $('.option-card').click(function() {
32        $(this).closest('.options-grid').find('.option-card').removeClass('selected');
33        $(this).addClass('selected');
34
35        const question = $(this).closest('.entry').find('h1').text();
36        userSelections[question] = $(this).data('value');
37    });
38
39    nextBtn.click(function() {
40        const currentEntry = $(entryElements[entryIndex]);
41        if (currentEntry.find('.option-card').length > 0 && currentEntry.find('.option-card.selected').length === 0) {
42            alert('Please select an option to continue.');
43            return;
44        }
45        if (entryIndex < entryElements.length - 1) {
46            entryIndex++;
47            updateUI();
48        }
49    });
50
51    backBtn.click(function() {
52        if (entryIndex > 0) {
53            entryIndex--;
54            updateUI();
55        }
56    });
57
58    restartBtn.click(function() {
59        entryIndex = 0;
60        $('.option-card').removeClass('selected');
61        for (let key in userSelections) {
62            delete userSelections[key];
63        }
64        updateUI();
65    });
66
67    // Initial UI setup
68    updateUI();
69});
```