



WEB DEVELOPMENT (WDT) GROUP ASSIGNMENT

AAPP009-4-2-WDT

WEB DEVELOPMENT

UCDF2005ICT(SE)

HAND OUT DATE: 20 SEPTEMBER 2021

HAND IN DATE: 14 NOVEMBER 2021

GROUP MEMBERS:

Yip Zi Xian TP059963 (LEADER)

Wong Xie Ling TP060781

Neong Yee Kay TP060276

Table of Contents

1.0	Gantt Chart	3
2.0	Introduction	4
2.1	Objectives and Design Choice	4
3.0	Diagrams and Prototype Designs	7
3.1	Wireframes	7
3.2	Navigational Structure	20
3.3	Flowchart/Activity diagram	21
3.4	Entity Relationship Diagram (ERD)	31
4.0	Implementation	32
5.0	Main section (User Guidance and Sample Screens)	53
5.1	General Functionalities	53
5.2	Admin Functionalities	61
5.3	Customer Functionalities	64
6.0	Conclusion	71
7.0	References	72

1.0 Gantt Chart

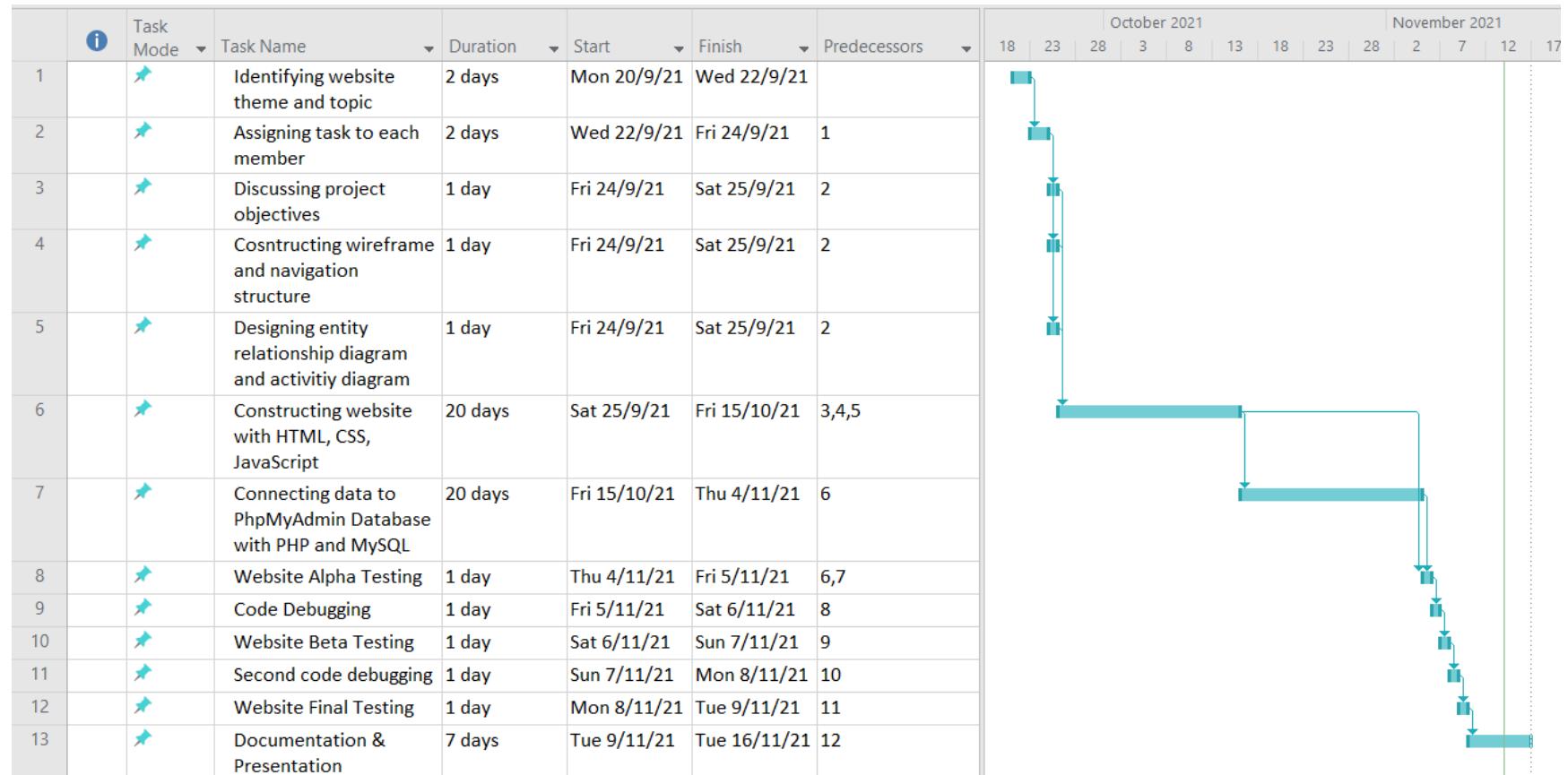


Figure 1: Gantt Chart and Task involved for Les Pet Shop website development

2.0 Introduction

The topic assigned was a web development assignment, in which 3 team members had to design and implement a web application on the topic chosen, which is an Online Pet Store named Les Pet Shop. Requirements for the web application includes the usage of coding languages such as HTML, Cascading Style Sheet (CSS), JavaScript to make it attractive and interactive for features that allow text boxes to ease in or drop-down text blocks to appear when users hover over or click on certain buttons. The web pages are then made dynamic by using PHP to work with database technology. Through this, functions such as customers signing up for an account at Les Pet Shop's website, logging in or admins logging in to access the server database to delete items from the website to update items can be implemented.

2.1 Objectives and Design Choice

A pet shop website where customers can view pets, adopt pets, and purchase pet items such as accessories, pet beds or pet treats is created for Les Pet Shop. Guests visiting the page or customers that are not logged in can only view the website, if they want to make purchases, users will need to login or sign up on the website first.

For a better buying experience, the logged in customers can add or edit their personal information such as their contact information and address. This improves the user experience as users do not have to key in their personal information each time they want to make a purchase because it is already saved and ready to use any time. Users can also view their purchase history by clicking into their user profiles.

Admins have special permissions to make changes to the page, such as adding new products, modifying the products details etc. Each time a user is logged in, a session will start and once the user logs out to end the session, the user will have to log in again. This is a useful security that ensures only authorized users can access the page. This will also make things easier for the users as they do not have to log in anytime they refresh or change pages. Below are the main objectives of our Les Pet Shop website: -

- Design an attractive yet simple interface
- To show all the necessary information such as the product information, customer personal information with a neat and clear view.

- Allow customers to purchase products with the user-friendly add-to-cart and payment features.
- To provide clear instructions and validation when it requires input from users.

The designs for the pages combine color palettes such as elegant yet approachable, minimal yet warm, vibrant, and elegant to create interesting and eye-catching graphics to entice the users to explore the page more.

15 Elegant Yet Approachable



Figure 2: Elegant Yet Approachable palette

19 Minimal Yet Warm



Figure 3: Minimal Yet Warm palette

25 Vibrant and Elegant



Figure 4: Vibrant and Elegant palette

The homepage is more vibrant compared to the other pages, using more graphics and cascading style sheet (CSS) languages to create interactive functions such as hover to view text that is overlayed onto an image, along with a slideshow that changes images when users click the next button. The homepage is also the flashiest so it can leave a good impression on visitors. As for the other pages such as the accessories, pet food and user profile pages, a more minimalistic and aesthetically pleasing design is implemented.

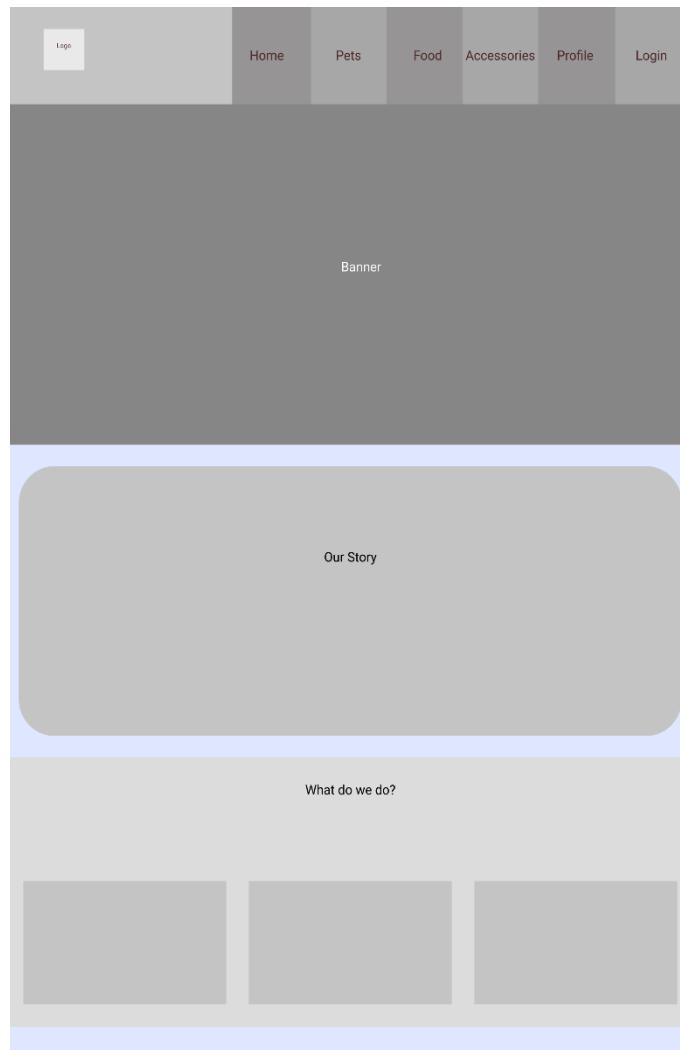
Lesser colors and CSS are used on these pages as they are only meant to display the information in a straightforward fashion, and so users do not get overwhelmed with many functions. Therefore, heavy emphasis was given when presenting the information by performing grouping of information and creating simple, direct yet organized layout. The design for the forms utilized on this website such as the personal information modification form, payment form, product management form and so on are well arranged, providing neat and clear instructions including the input format if necessary to avoid confusion. Placeholders with prompts are also placed within the form input bars to let users know what to input.

The intended audience for the page is meant for everyone, regardless of age, gender, and race, since pets are universally loved by many, hence the mixture of playfully colorful and aesthetically minimalistic designs will surely appeal to both parties. With the easy to understand and direct graphical user interface, it also makes it easier for anyone to you.

3.0 Diagrams and Prototype Designs

3.1 Wireframes

The wireframes below show the basic structure and guideline of the web page that is planned to implement. The wireframe is made to provide a clear view on how the structure of the web page should turn out. It also prevents the people working on the page from straying away from the original idea of how the page should turn out.



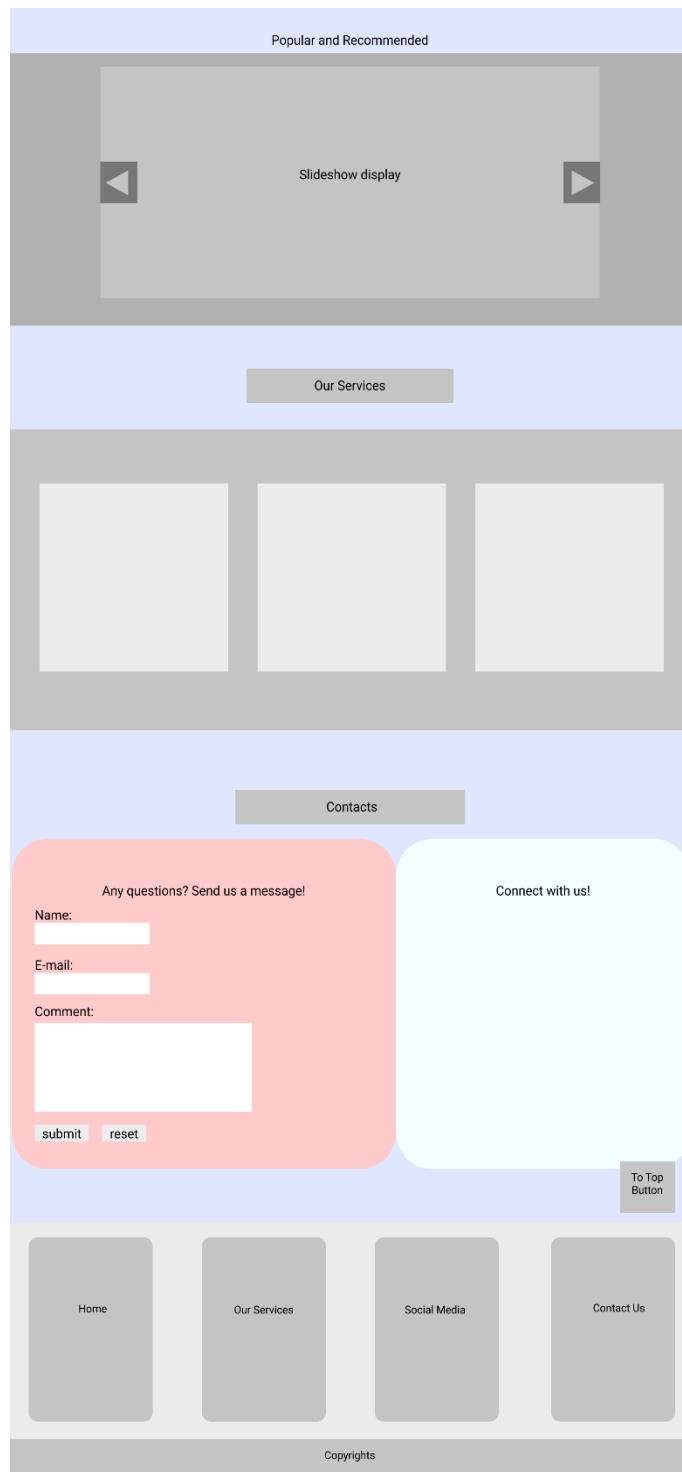


Figure 5: Homepage Wireframe

Wireframe Name	Homepage Wireframe
User(s) with access	Guests, Registered Customers, Admins
Description	<p>This page will be the first thing any page visitor will see upon clicking or typing in the Les Pet Shop link. Just like every page, the homepage contains a navigational bar that allows users to access any user-accessible page on the website just from clicking it. The various navigational buttons include the home button, which redirects back to the homepage, pets, food and accessories button which redirects to the pages where users can add the respective items to cart, profile button where users can click to access their user profile and lastly the login button displays a drop down menu when hovered over where users that are not logged in, users that wished to sign up or admins who wish to access the page can log in. It also has many containers used to display information such as banners with text, slideshows, hover for text overlay images and contact forms. The content mentioned above are used to display useful information. Each page also includes a footer, which contains information such as about us, more from us, social media information and also more contact information.</p>

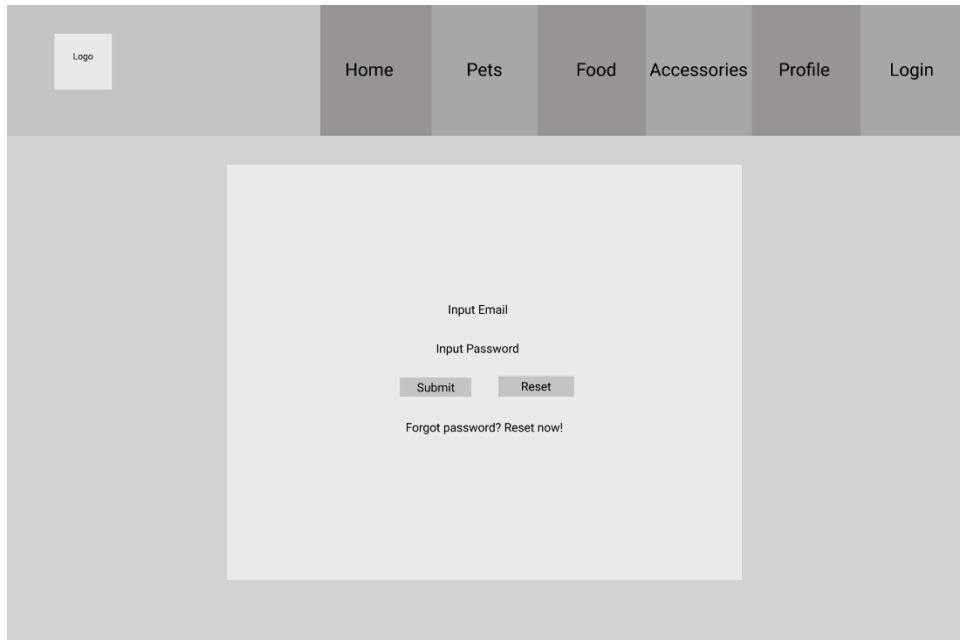


Figure 6: Login page wireframe

Wireframe Name	Login page Wireframe
User(s) with access	Registered customers
Description	The login page can be accessed through the login button from the navigational bar. In this page, users with already registered accounts can use their email and password to log in. If users wish to clear the input in the form, they can click the reset button. Once the information is correct, they can click submit.

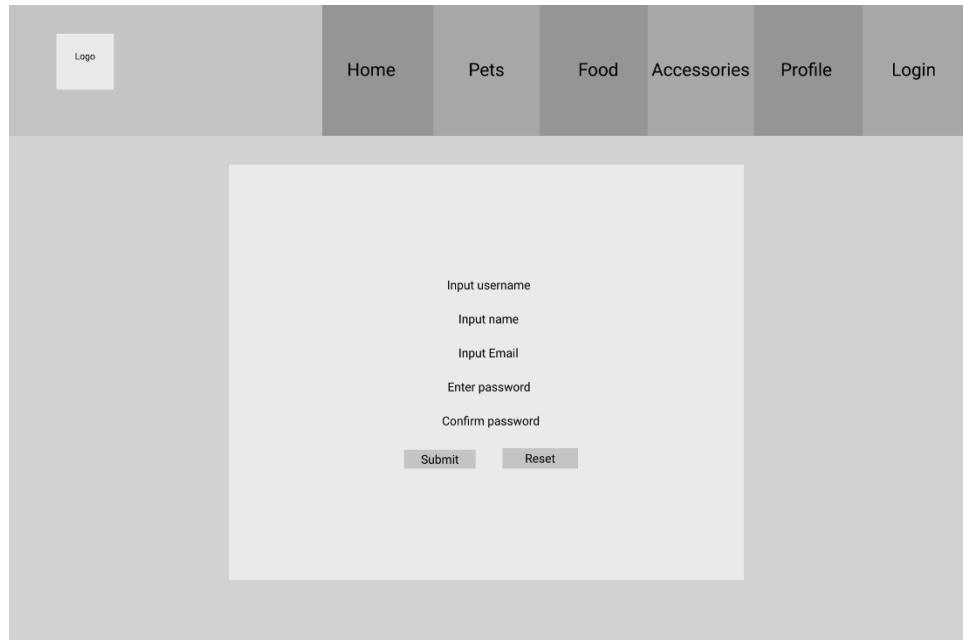


Figure 7: Sign Up page wireframe

Wireframe Name	Sign up page wireframe
User(s) with access	Guests
Description	In this sign-up wireframe, it contains a form that prompts for a username, name, email and password along with a form used to confirm the password entered. Once done, users can submit their information to register an account. A reset button is also added for easy input clearing.

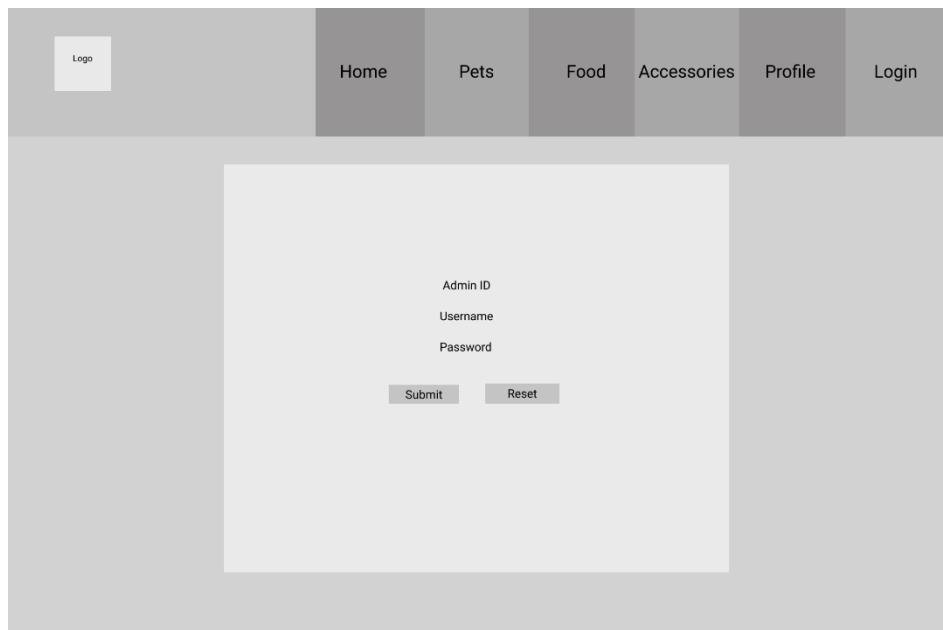


Figure 8: Wireframe for admin access page

Wireframe Name	Admin Access Wireframe
User(s) with access	Admins
Description	This page is used to allow admins to log in. It includes a form that takes input such as admin ID, username, and password. Just like the other two forms, users can submit when done inputting and reset the form as they please.

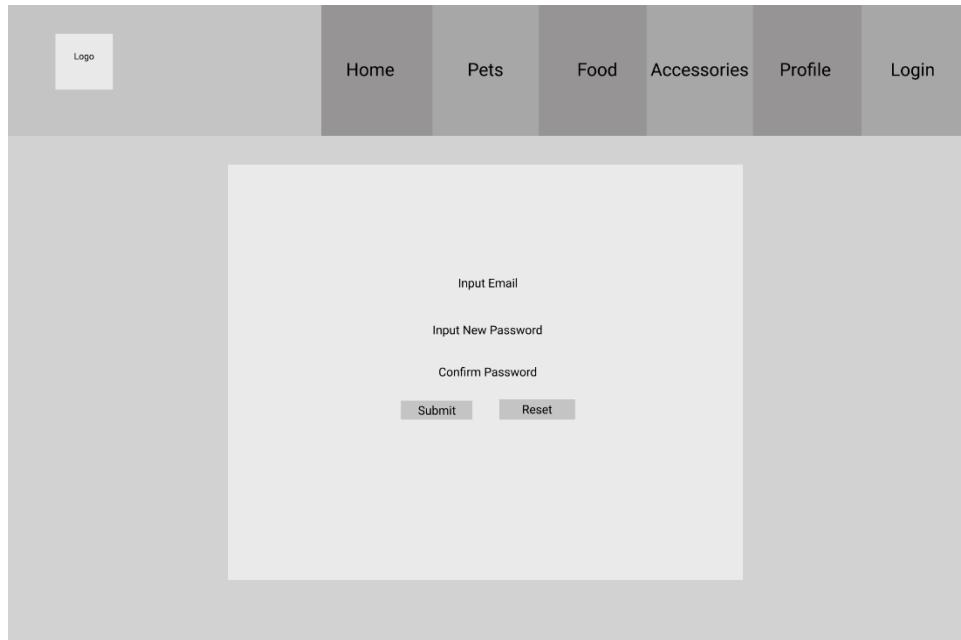


Figure 9: Reset password wireframe

Wireframe Name	Reset Password Wireframe
User(s) with access	Registered Customers
Description	Users can access this page through the login page. If users forget their password, they can click on the hyperlink in the login page to be directed to a page containing a form that allows them to change their password in order to gain access to the logged in user only functionalities again. A navigational bar is also added to all the login/sign up forms so users will not be stuck on the page.

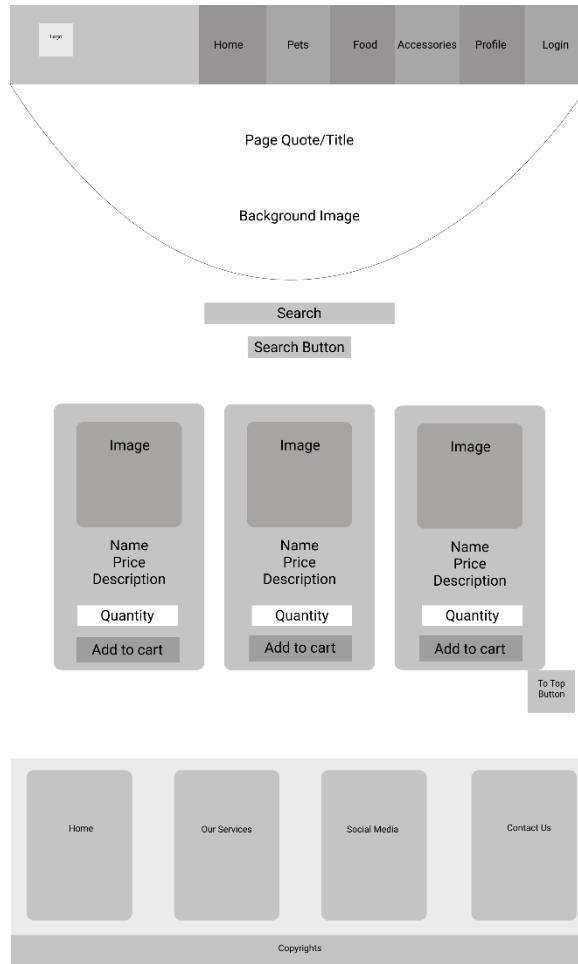


Figure 10: View product page wireframe

Wireframe Name	View products Wireframe
User(s) with access	Guests, Registered Customers, Admins
Description	This product page will show all the information about the products such as price, description, and image to the logged in customers. Each product and its information will be displayed in one box with an input box to receive quantity value and an “add to cart button” at the bottom of the box. There is also a search bar that can assist the users to find the products they want by searching the products’ name.

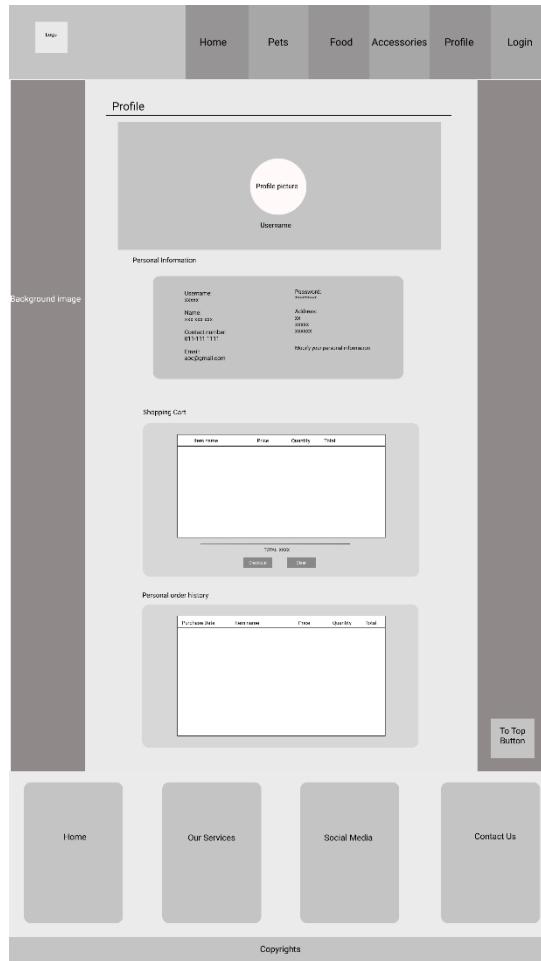


Figure 11: User profile page wireframe

Wireframe Name	User profile page Wireframe
User(s) with access	Registered Customers
Description	<p>The user profile page displays the logged in customers' personal information, shopping cart and purchase history. On the personal information section, a button named "modify personal information" provides an option for the customers to edit their information. The customers can also view the items in their shopping carts as well as the total price. They can clear the cart or checkout to make payment. For the purchase history section, the purchase history information will be displayed in table form.</p>

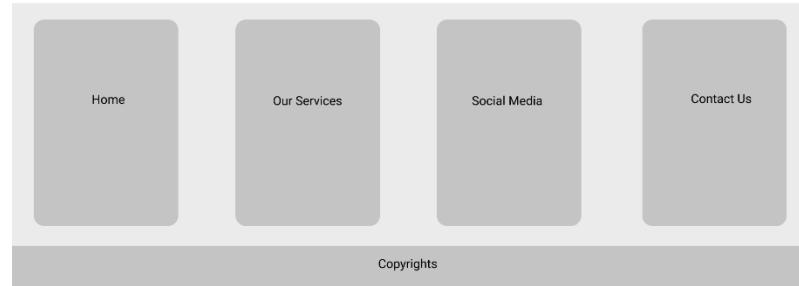
A wireframe for a 'Modify account details' page. On the left is a sidebar titled 'Account Information' containing a placeholder 'profile picture'. The main area is titled 'Modify account details' and contains two columns of form fields. The left column is labeled 'Personal Details' and the right column is labeled 'Delivery Address'. Each column has three rows, each with a label ('Required details:' or '(input field)'), followed by a placeholder '(input field)' enclosed in parentheses. A 'Submit' button is located at the bottom right of the main area.

Figure 12: Manage account page wireframe

Wireframe Name	Manage account page Wireframe
User(s) with access	Registered Customers
Description	This page provides a form for the logged in customers to edit their personal information.

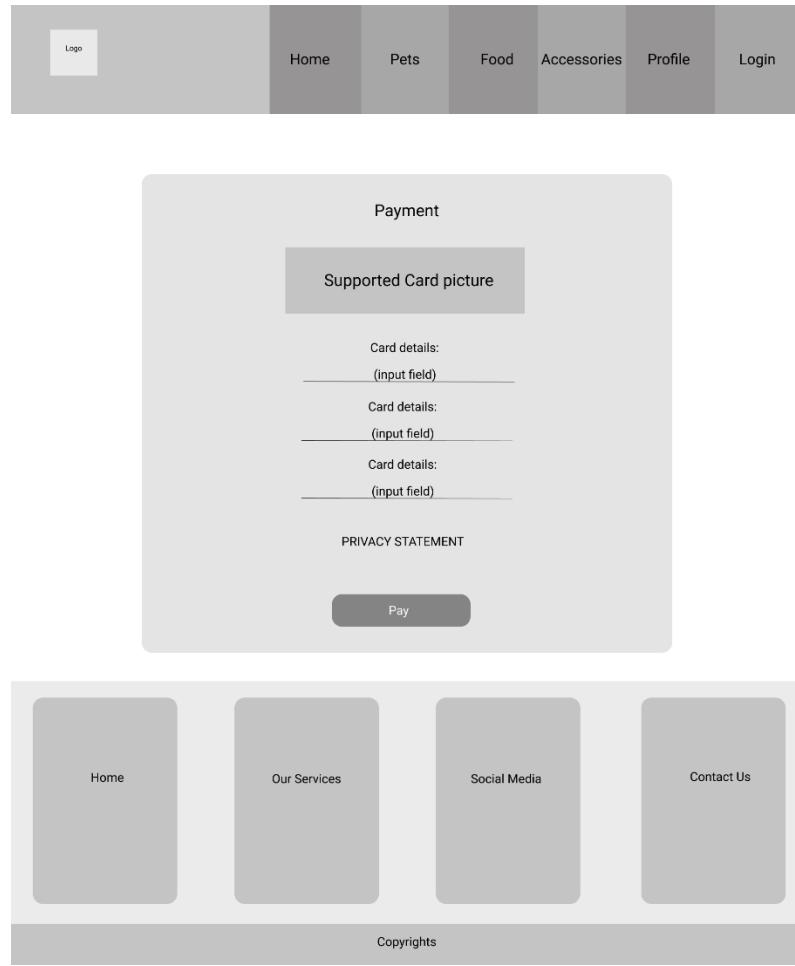


Figure 13: Payment page wireframe

Wireframe Name	Payment page Wireframe
User(s) with access	Registered Customers
Description	The payment page shows a form that receives customers' payment information to process their payment for the shopping cart.

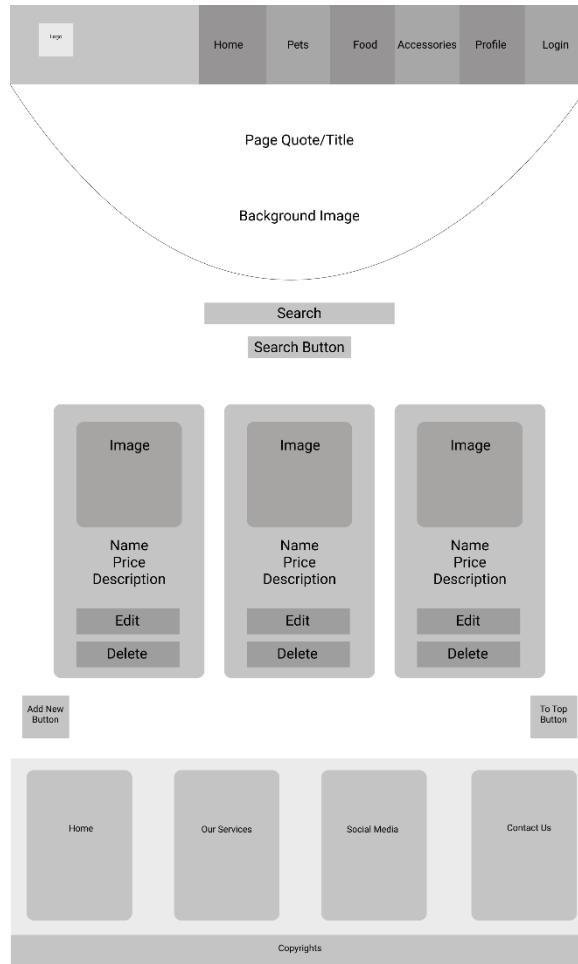


Figure 14: View and edit page for admin

Wireframe Name	View and edit page Wireframe
User(s) with access	Admins
Description	The figure above shows the product page that can be seen and accessed by an admin. This page not only displays the details of the products but also provides edit, delete, and add product information functionalities.

The wireframe illustrates the 'Add new product' page for administrators. At the top, there is a header bar with a 'Logo' icon and six navigation links: 'Home', 'Pets', 'Food', 'Accessories', 'Profile', and 'Login'. Below the header is a large rectangular form area. Inside the form, there are six input fields stacked vertically, each with a placeholder label above it: 'Select Category', 'Select Image', 'Type Name', 'Type Description', 'Type Quantity', and 'Type Stock'. To the right of these input fields, the text 'Background image' is placed. At the bottom of the form are two buttons: 'Submit' and 'Reset'.

Figure 15: Add new product page for admin

Wireframe Name	Add new product Wireframe
User(s) with access	Admins
Description	The add new product page provides a form where admins can enter relevant information such as category, image, product type, description, quantity, and stock for a new product that will be added to Les Pet Shop.

3.2 Navigational Structure

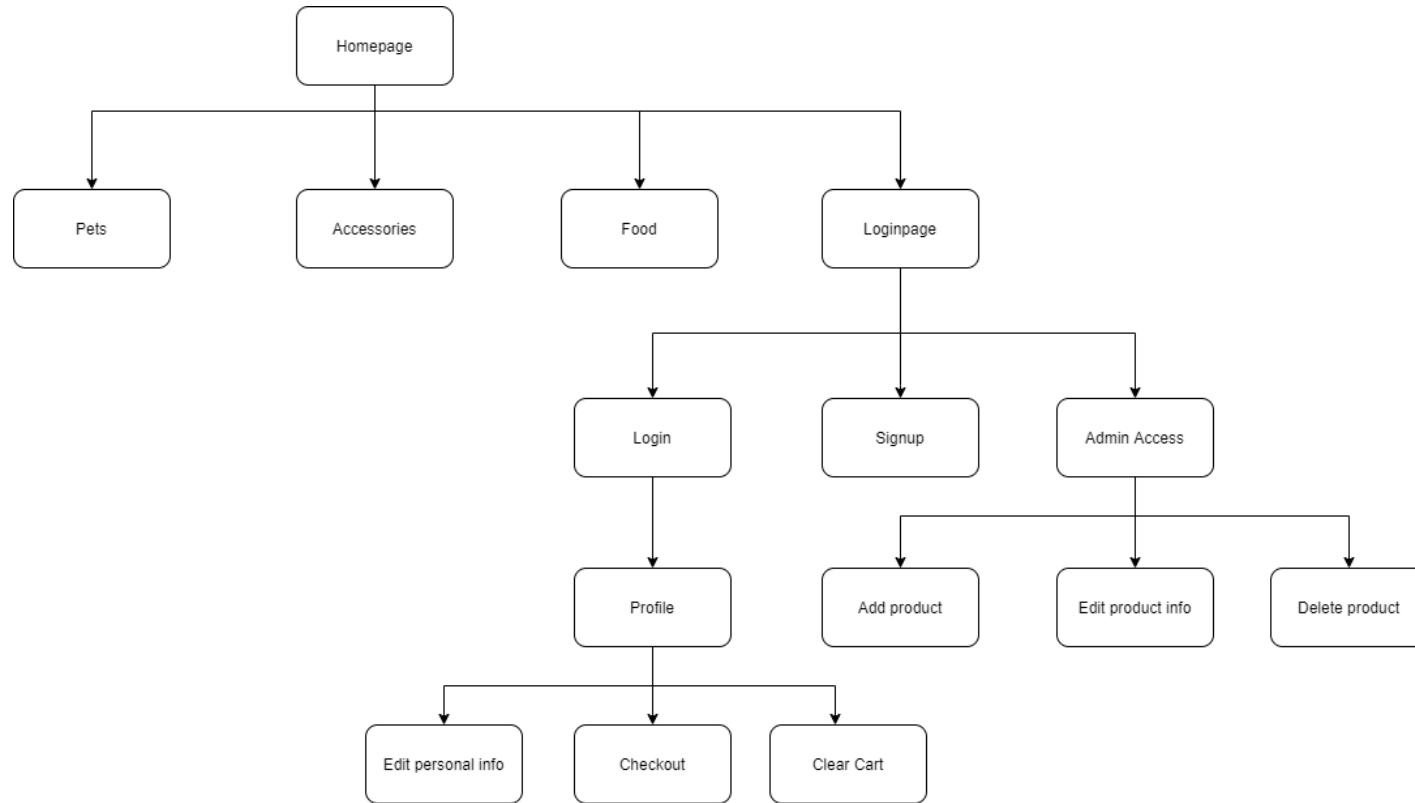


Figure 16: Navigation structure for Les Pet Shop website

3.3 Flowchart/Activity diagram

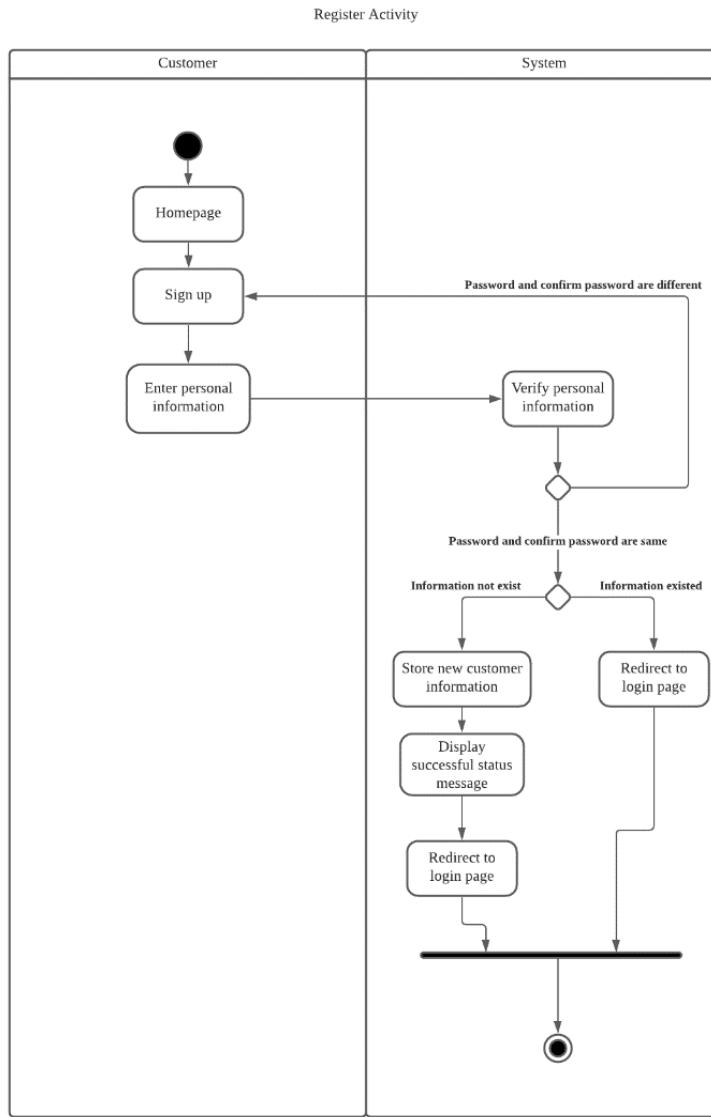


Figure 17: Register activity diagram

Customers can go to the sign-up page through the homepage. Then, they will have to enter their personal information such as email, name, username and password. If the password and the confirm password are the same and the information they entered does not exist in the database, their personal information will be stored into the database as a new customer. A successful status message will be shown, and they will be redirected to the login page to login. However, if the password and confirm password are different, they have to go back to sign up

page again. If the personal information entered already exists, in the database the system will redirect them to the login page.

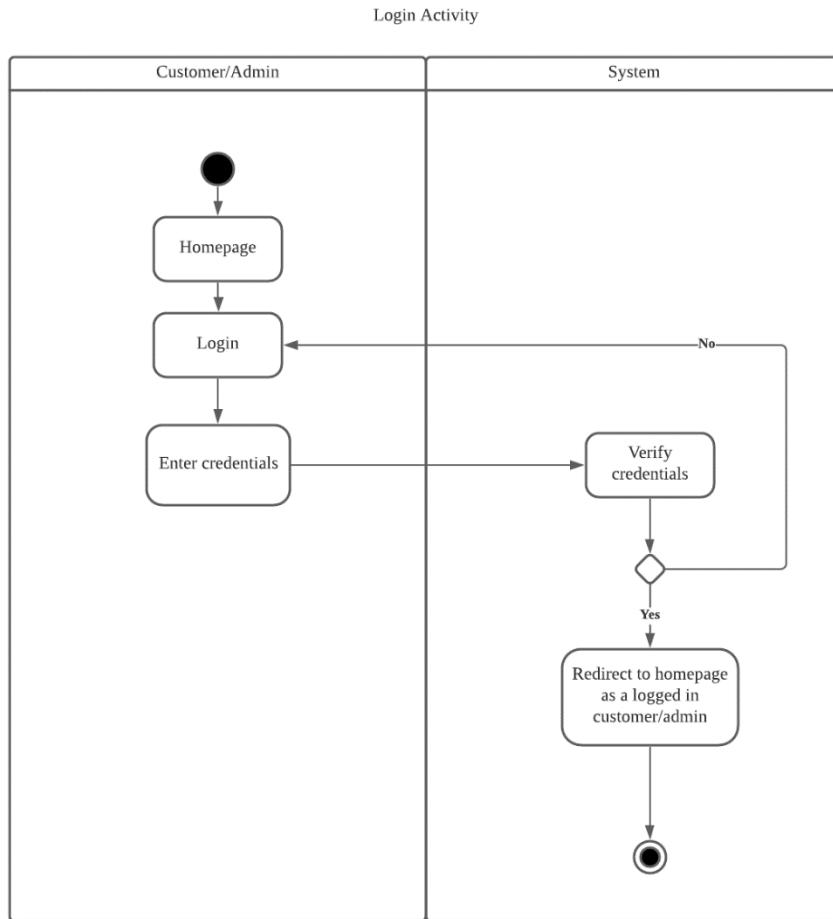


Figure 18: Login activity diagram

The login page can be accessed through the homepage. Admins and registered customers can log into their account by entering their credentials on the login page. Once their credentials are verified by the system, they will be redirected to the homepage as a logged in customer or admin with the given session. If the credentials given are invalid, they will be redirected to the login page.

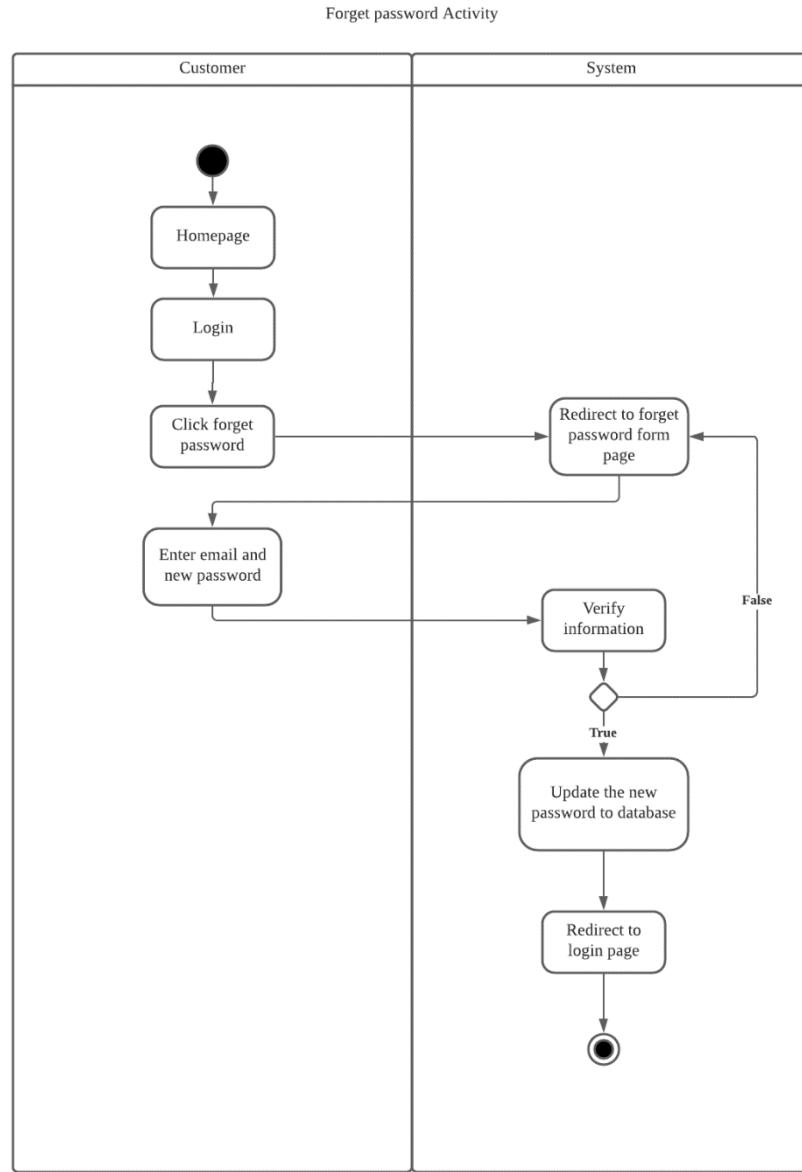


Figure 19: Forget password activity diagram

If customers forget their password, they can go to login page through the homepage and click the forget password hyperlink that will redirect them to the forget password form. After they enter their email and new password, the system will verify the authenticity of the entered information. If it is true, then the new password will be updated to the database, and the customers will be redirected to the login page, else they will be redirected to the forget password form.

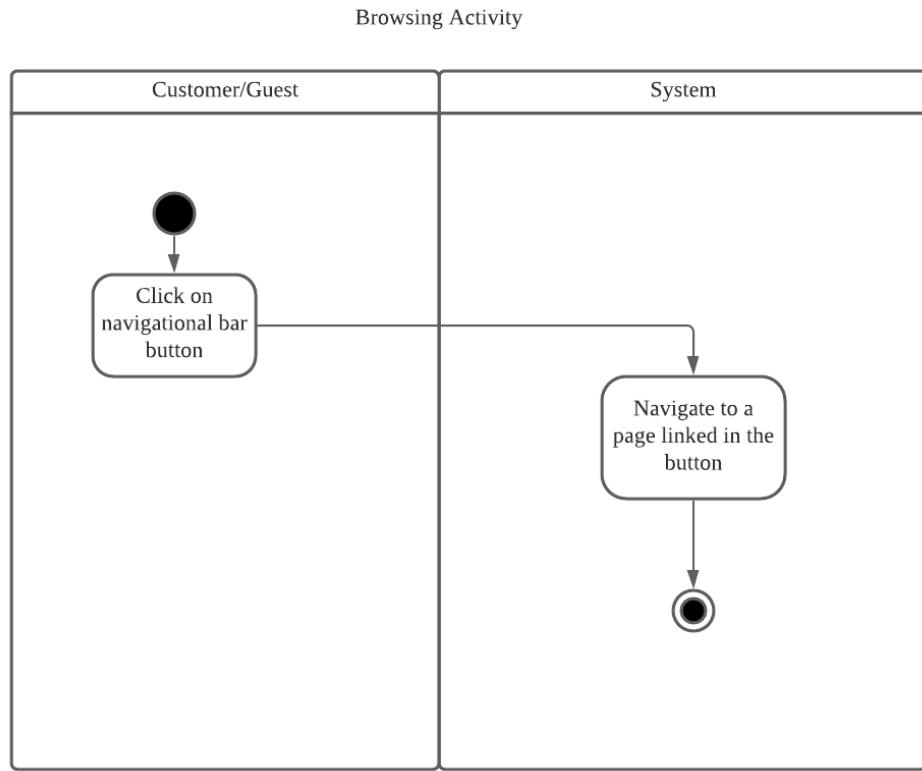


Figure 20: Browsing activity diagram

To browse links on the website, users can do so by clicking various buttons or text that are embedded with links. For example, the navigational bar at the top of each page contains buttons such as home, profile and various purchasable item pages. By clicking on these buttons, users will be redirected to the page.

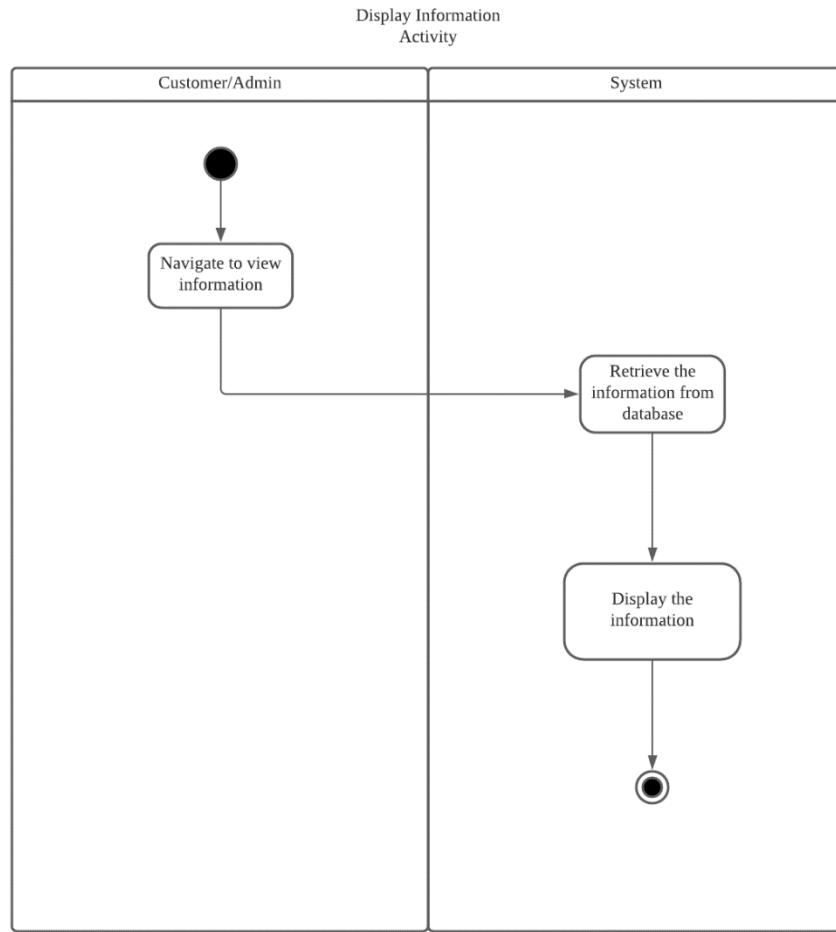


Figure 21: Display information activity diagram

When the admins and customers navigate around to view information such as product information, personal information, shopping cart information or order history, the system will retrieve the relevant information from the database and then display the information.

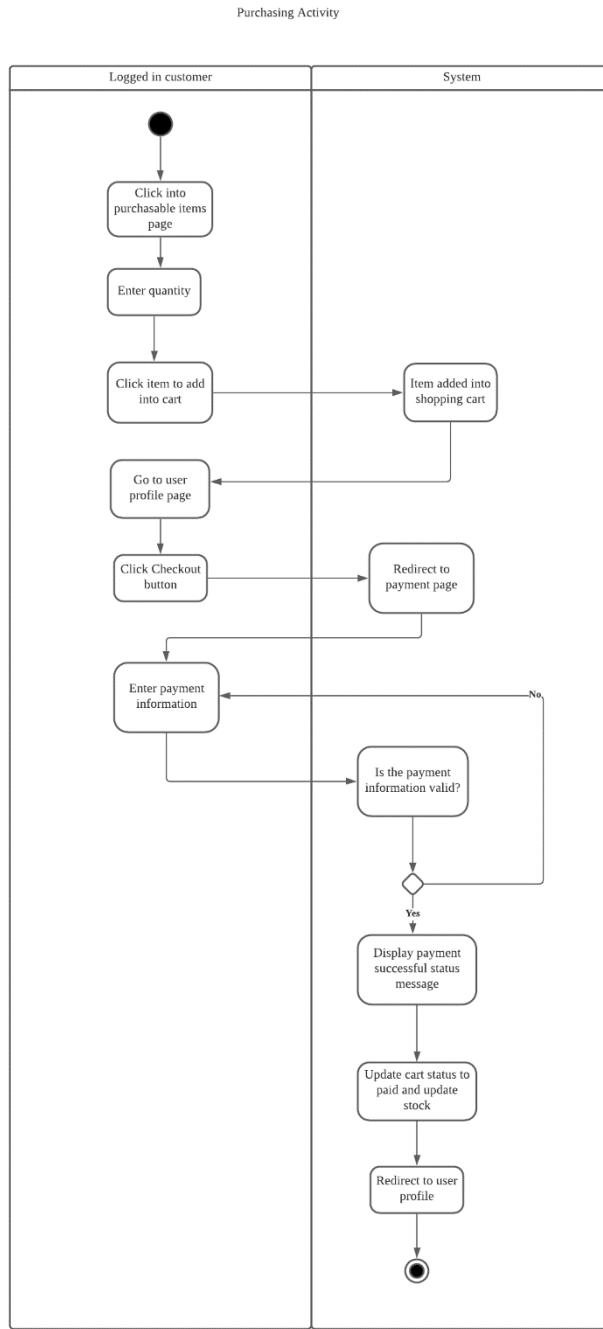


Figure 22: Purchasing activity diagram

The logged in customers can purchase products on Les Pet Shop website on the purchasing items page. They can type the quantity of the products they want to buy and add them into their shopping cart. The customers can click the checkout button on their shopping cart to make payment, then the website will redirect them to the payment page. Once the payment information is entered by the customers, the system will verify the information. If it is valid, a successful status

message will be displayed, and the customer will be redirected to the user profile page. The cart status and product stock will also be updated to the database. Else, it will redirect to the payment page to prompt valid payment information again.

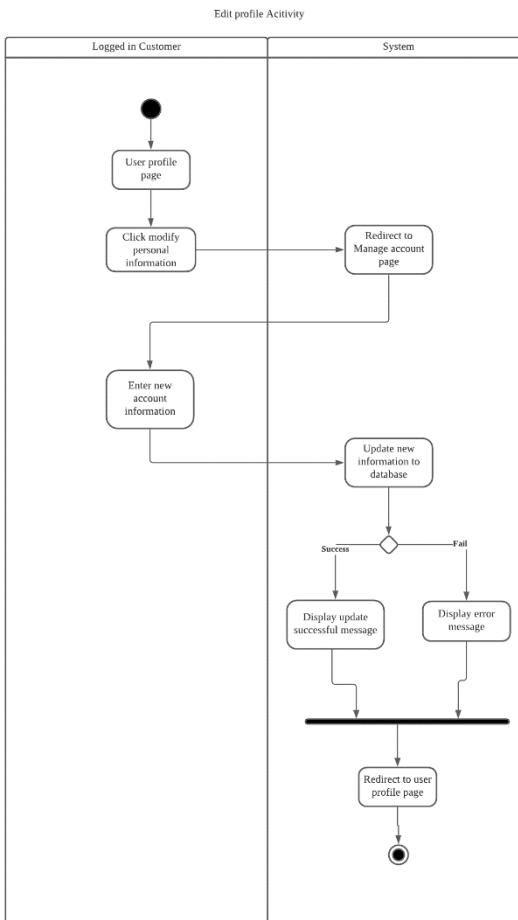


Figure 23: Edit profile activity diagram

Customers that are logged in can edit their profiles through their user profile page. By clicking the button that allows them to modify personal information, they are redirected to the manage account page that allows them to change certain information such as username, address, email etc. After making changes, the system then updates the newly added or changed information into the database. If the update is a success, a status message to inform the users of the update success is shown, whereas an error message will be shown to the users if the information has failed to update. After changes are made, users are redirected to the profile page.

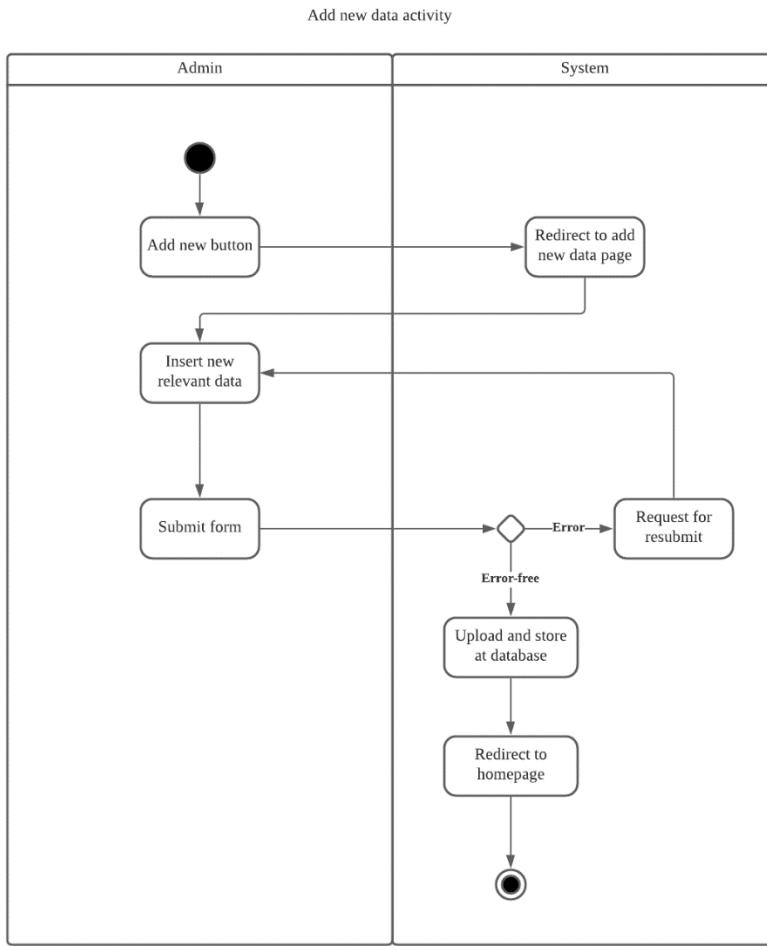


Figure 24: Add new data activity diagram

Admins can add new product data on Les Pet Shop by clicking “add new” button. This button will redirect the admins to add new data page where they can inset new relevant data for the product in the form such as product name, description, price and so on. After that they can submit the form to be uploaded and stored at database, then the admins will be redirected to the homepage. If an error occurs during the submission, the system will request for resubmit and redirect them to insert form page.

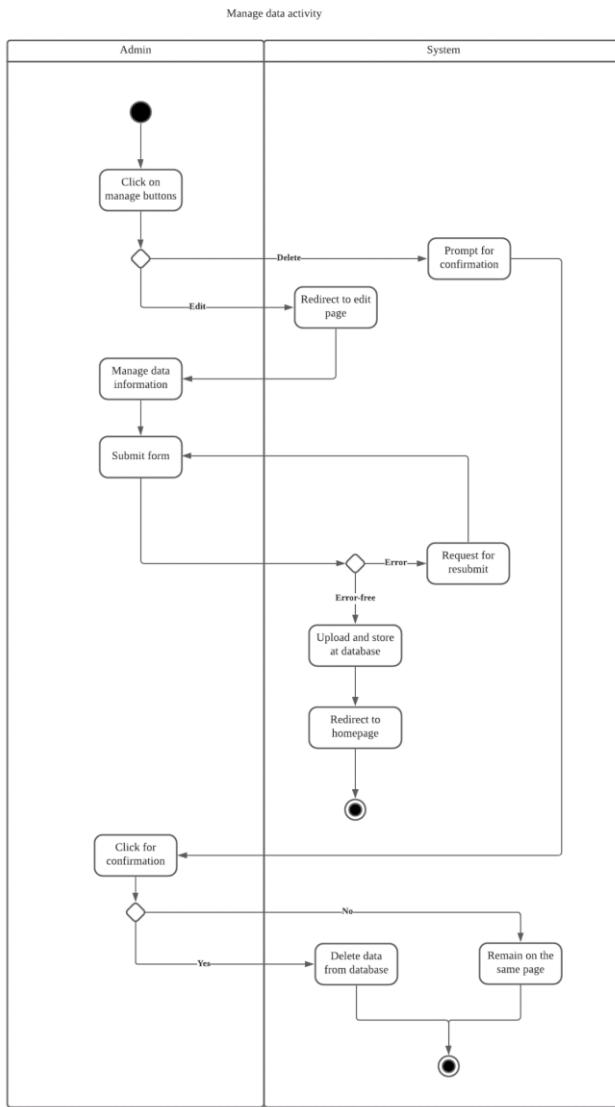


Figure 25: Manage data activity diagram

To manage data in the database, admins need to first click on the manage button, where they can choose to either edit or delete data. By clicking the delete button, the admins are prompted for confirmation, to prevent from accidentally deleting data. If admins confirm to delete data, the data is deleted from the database, but if the admin does not confirm to delete, they will remain on the same page. When the admins choose to edit data instead, they will be redirected to an edit page. Through the edit page, they can change the data such as item description. After editing, admins need to submit form. If an error arises, they will be prompted to request for resubmit, but if it is error free, the newly updated data will be uploaded and stored in the database, and once completed, admins will be redirected to the homepage.

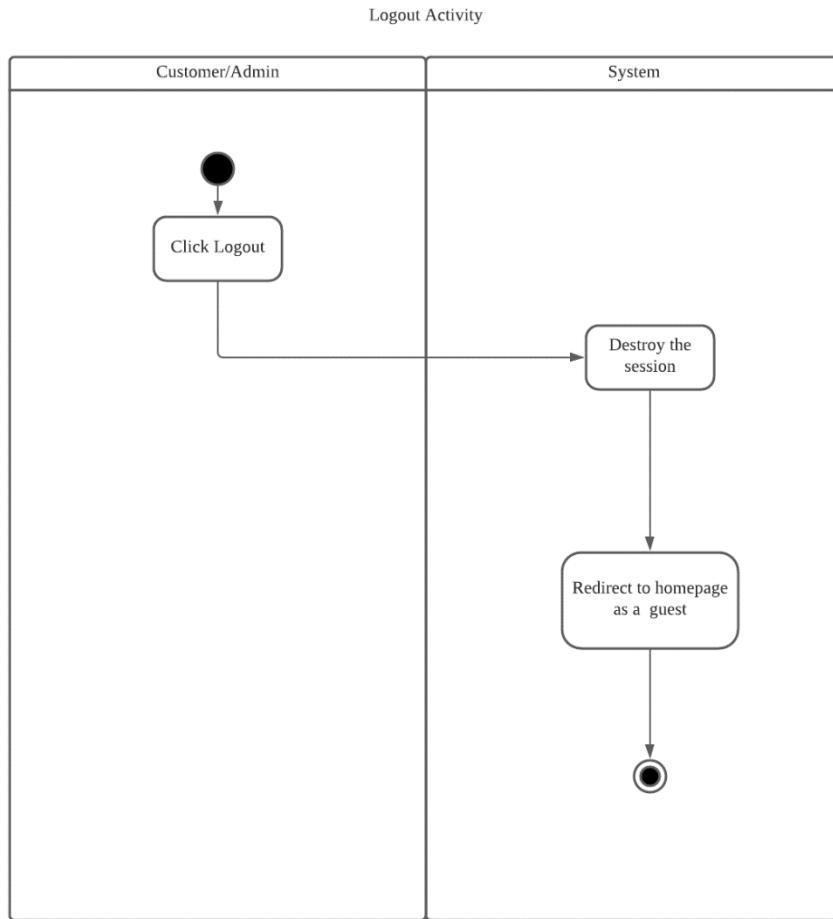


Figure 26: Logout activity diagram

Both customers and admins can just log out by clicking the logout button on the navigational bar. The system will destroy the session created for them and then redirect them back to the homepage as a guest.

3.4 Entity Relationship Diagram (ERD)

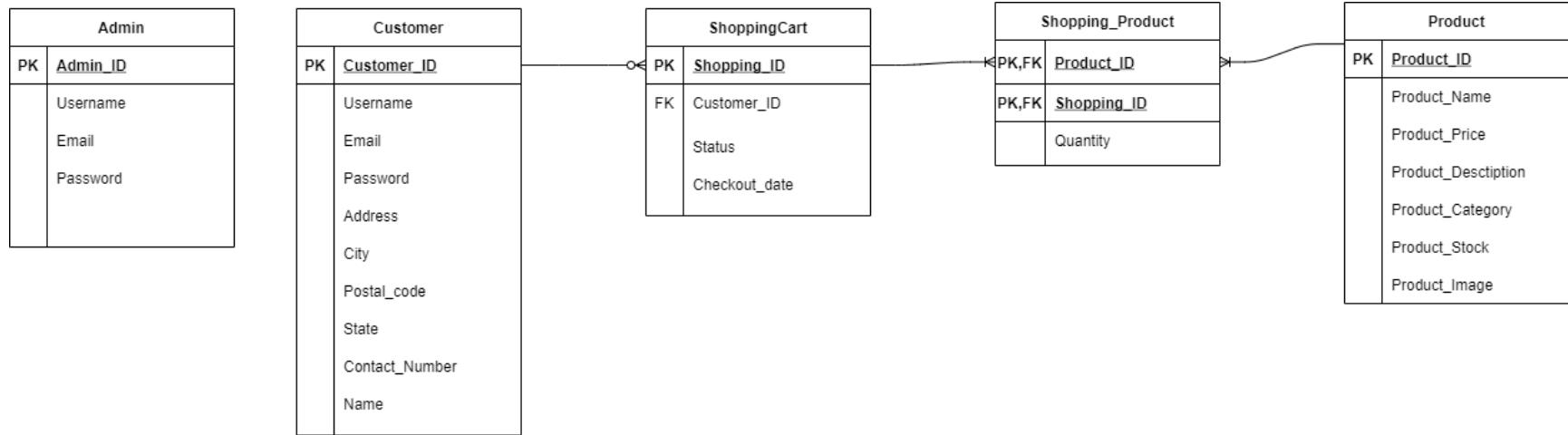


Figure 27: Entity Relationship Diagram of Les Pet Shop

4.0 Implementation

1) Create

```
<!--Sign Up Form-->
<div class="container">
  <div class="selectcont">
    <div class="formtext">
      <table>
        <h2>Wish to create an account?<br>Sign Up here!</h2>
        <form method="post" action="php/signup.php">
          <tr>
            <td>Username :</td>
            <td><input type="text" name="username" required placeholder="Choose a username"></td>
          </tr>
          <tr>
            <td>Name :</td>
            <td><input type="text" name="name" required placeholder="Enter your name"></td>
          </tr>
          <tr>
            <td>Email :</td>
            <td><input type="email" name="email" required placeholder="Enter your email"></td>
          </tr>
          <tr>
            <td>Password :</td>
            <td><input type="password" name="password" required placeholder="Enter a password"></td>
          </tr>
          <tr>
            <td>Confirm Password :</td>
            <td><input type="password" name="checkpassword" required placeholder="Re-enter your password"></td>
          </tr>
        <tr>
          <td></td>
          <td><input type="submit" value="Submit" name="submitinput">
              <input type="reset" value="Reset" name="resetinput">
            </td>
        </tr>
      </table>
    </div>
  </div>
</div>
```

Figure 28: HTML Form for signing up as a new customer

The HTML form for signing up as a new customer requires input such as username, name, email, password and confirm password. The form is set to required, so users will need to fill it in completely to successfully log in. Placeholders are also put to prompt users for the right data. Submit and reset buttons are added to allow users to submit their information as well as reset the whole form.

```

?php
include('conn.php');

$email = mysqli_real_escape_string($con, $_POST['email']);
$password = mysqli_real_escape_string($con, $_POST['password']);
$checkpassword = mysqli_real_escape_string($con, $_POST['checkpassword']);
$checkemail = "SELECT * FROM customer WHERE email = '$email'";
$validate = mysqli_query($con, $checkemail);
$insert="INSERT INTO customer (username, name, email, password) VALUES ('$_POST[username]', '$_POST[name]', '$_POST[email]', '$_POST[password]');";

```

Figure 29: PHP code to create new row for new customer

The php code starts by including the connection to the database. After that, variables are set to get the users input that are submitted through the form. This is done using the \$_POST super global variable. After that, validation to check whether the password is the same as the one entered as well as if the email is already existing or not is added. This can be seen in the \$checkpassword and \$checkemail variables. The \$checkpassword variable retrieves the user's form input while the \$checkemail variable will retrieve data from the database. The sql query to add the data for the new customer calls to insert the user input data into the respective colums in the customer table.

2) Read

```

<!--Content Flexboxes-->


Figure 30: HTML for displaying products available



In this HTML form, various division classes to align the containers and content along with text formatting are done to display products available for the food, pet and accessories page.


```

```
<!--Search activation-->
<?php
    include("conn.php");

    $searchText = "";

    if(isset($_POST["search"])) {
        $searchText = $_POST['searchText'];
    }

    $searchResult = mysqli_query($con, "SELECT * FROM product WHERE Product_Category='accessory' AND Product_Name LIKE '%$searchText%' ORDER BY Product_Name");
?>
```

Figure 31: Search bar to display wanted products and all products by default

The \$searchText variable will receive the searchText form input from the search bar in the accessory page. The default value for this variable is “” so that every product will be displayed when the users haven’t performed the search function. For the \$searchResult variable, it is a MySQL query to select everything from the product table where the product category is under the accessory page and the product name has the same pattern as the search text input in the search bar. It is then ordered by product name.

```

<!--Content Flexboxes-->
<div class='content-flexbox-container'>

<?php
    // Get and display data
    while ($row = mysqli_fetch_array($searchResult)){
        $display = '

        <div class="content-card">

            <div class="content-imagebox" >

            </div>

            <div class="content-descriptionbox">

                <h1>' . $row['Product_ID'] . '</h1>

                <h2>' . $row['Product_Name'] . '</h2>

                <h6>' . $row['Product_Description'] . '</h6>

                <p>RM ' . $row['Product_Price'] . '</p>

                <form method="post" action="accessoriesToCart.php">

                    <input type="hidden" name="Product_ID" value=' . $row['Product_ID'] . ' >

                    <input type="number" name="Quantity" value=1>

                    <input type="submit" name="submit" value="Take it home">

                </form>

            </div>

        </div>

    ';

        echo $display;
    }

```

Figure 32 – 33: PHP code to display all products from database

In this code, `mysqli_fetch_array()` is used to fetch rows of data from the `$searchResult` variable as an array. A while loop is used to loop through blocks of codes when there is still result that can be fetched. Inside the blocks of codes, the corresponding product image, product id, product name, product description and product price from the specified rows will be displayed.

CSS styling with div classes is used along with added images. The form method post is used to submit the data and it is sent to accessoriesToCart.php to be processed. Lastly, the variable display is echoed to display the products from the database.

3) Update

```
<?php
include("conn.php");
$customer_ID=$_SESSION['Customer_ID'];
$result = mysqli_query($con,"SELECT * FROM customer WHERE Customer_ID=$customer_ID");
$row = mysqli_fetch_array($result);
>


<div class="tab-container">
<div class="pfp">
|   
</div>
<div class="tab" id="tab1">Account Information</div>
</div>

<form class="content" style="background-image: url('../art/white_texture.png')" id="personal-info-form" action="update_personal_info.php">
<div class="form-left">
<h2>Modify Account Information</h2>
<h3>Personal details</h3>
<div>Username:</div>
<input type="text" name="user_username" required="required" value="<?php echo $row['Username']?>">
<div>Name:</div>
<input type="text" name="user_name" required="required" value="<?php echo $row['Name']?>">
<div>Contact number:</div>
<input type="text" name="user_contact_number" required="required" pattern="[0][1][0-9]-[0-9]{4}[0-9]{4}|[0][1][0-9]-[0-9]{3}[0-9]{3}">
<div>E-mail:</div>
<input type="email" name="user_email" required="required" value="<?php echo $row['Email']?>">
</div>
<div class="form-right">


```

```
<div class="form-right">
    <h2>&nbsp;</h2>
    <h3>Delivery Address</h3>
    <div>Address:</div>
    <input type="text" name="user_address" required="required" value=<?php echo $row['Address']?>>
    <div>City:</div>
    <input type="text" name="user_city" required="required" value=<?php echo $row['City']?>>
    <div>Postal code:</div>
    <input type="text" name="user_postcode" required="required" value=<?php echo $row['Postal_code']?>>
    <div>State:</div>
    <Select name="user_state">
        <option value="Johor" <?php if ($row['State'] == "Johor") { ?>
            selected="selected"
            <?php } ?>>Johor</option>
        <option value="Kedah" <?php if ($row['State'] == "Kedah") { ?>
            selected="selected"
            <?php } ?>>Kedah</option>
        <option value="Kelantan" <?php if ($row['State'] == "Kelantan") { ?>
            selected="selected"
            <?php } ?>>Kelantan</option>
        <option value="Melacca" <?php if ($row['State'] == "Melacca") { ?>
            selected="selected"
            <?php } ?>>Melacca</option>
        <option value="Negeri Sembilan" <?php if ($row['State'] == "Negeri Sembilan") { ?>
            selected="selected"
            <?php } ?>>Negeri Sembilan</option>
        <option value="Pahang" <?php if ($row['State'] == "Pahang") { ?>
            selected="selected"
            <?php } ?>>Pahang</option>
        <option value="Penang" <?php if ($row['State'] == "Penang") { ?>
            selected="selected"
            <?php } ?>>Penang</option>
        <option value="Perak" <?php if ($row['State'] == "Perak") { ?>
            selected="selected"
            <?php } ?>>Perak</option>
        <option value="Perlis" <?php if ($row['State'] == "Perlis") { ?>
            selected="selected"
            <?php } ?>>Perlis</option>
        <option value="Sabah" <?php if ($row['State'] == "Sabah") { ?>
            selected="selected"
            <?php } ?>>Sabah</option>
        <option value="Sarawak" <?php if ($row['State'] == "Sarawak") { ?>
            selected="selected"
            <?php } ?>>Sarawak</option>
        <option value="Selangor" <?php if ($row['State'] == "Selangor") { ?>
            selected="selected"
            <?php } ?>>Selangor</option>
        <option value="Terengganu" <?php if ($row['State'] == "Terengganu") { ?>
            selected="selected"
            <?php } ?>>Terengganu</option>
        <option value="Kuala Lumpur" <?php if ($row['State'] == "Kuala Lumpur") { ?>
            selected="selected"
            <?php } ?>>Kuala Lumpur</option>
        <option value="Labuan" <?php if ($row['State'] == "Labuan") { ?>
            selected="selected"
            <?php } ?>>Labuan</option>
        <option value="Putrajaya" <?php if ($row['State'] == "Putrajaya") { ?>
```

```
        <option value="Putrajaya" <?php if ($row['State'] == "Putrajaya") { ?>
                selected="selected"
            <?php } ?>>Putrajaya</option>
    </Select>
    <br><br>
    <div><input type="submit" value="submit" name="submit"></div>
</div>
</form>
<?php
    mysqli_close($con);
?>
/div>
```

Figure 34 - 38: HTML form used to edit account information

In the HTML form, it requests input such as username, name, contact number and e-mail etc. that are mandatory for users to fill in. Validations are included in these input fields by specifying the types and patterns of the required information to eliminate human errors. If there is existing information, it will be displayed in the input field using the php code for the customers to edit.

```

<?php
session_start();
$customer_id = $_SESSION['Customer_ID'];

include("conn.php");
$sql = "UPDATE customer SET Username='".$POST['user_username']."' , Name='".$POST['user_name']."' ,
Contact_No='".$POST['user_contact_number']."' ,
Email='".$POST['user_email']."' ,
Address='".$POST['user_address']."' ,
City='".$POST['user_city']."' ,
Postal_code='".$POST['user_postcode']."' ,
State='".$POST['user_state']."' 
WHERE Customer_ID=$customer_id";

if (mysqli_query($con, $sql)) {
    echo'<script>
    alert("Changes are made successfully.");
    window.location.href="userprofile.php";
    </script>';
}
else{
    echo'<script>
    alert("Failed to update your information.");
    window.location.href="userprofile.php";
    </script>';
}
mysqli_close($con);
?>

```

Figure 39: PHP codes used to edit account information

To edit the account information, a user needs to be logged in, which will cause a session to start. Once the session is started, the customer id from the session is derived. The connection to the database is also included, and can be used later by typing \$con, which is the variable for the connection set in the conn.php file.

After that, the structured query language, SQL query to update the existing information is set into a variable, as shown in the image provided above. The \$sql variable set will update the customer table with input from customers such as username, name, contact number, email, address, city, postal code, and city. A where clause is added to ensure that the only information of that particular customer will be changed. This is done by ensuring that the customer id from the session is the same as the customer id in the database.

After that, an if else statement is used, where if the query is successfully executed, an alert message will pop up to inform the users of the success in editing the information. If the query is

not executed, a different alert message telling the users that the information failed to update will be shown. Both “if else” statements will redirect the users back to the userprofile.php page.

4) Delete

```
</table>
<div class="cart-bottom">
    <hr>
    <p>Total: RM <?php echo $checkout_price ?></p>
    <button onclick="document.Location = '../payment.html';" class="cart-button">Checkout</button>
    <button onclick="document.Location = 'resetcart.php';" class="cart-button">Clear</button>
</div>
</div>
```

Figure 40: HTML button to delete all unpaid products in user profile

The button to delete all unpaid products in the shopping cart is the “Clear” button. When clicked, it will be redirected to the resetcart.php where it will process the data and in return, clear the shopping cart.

```
<?php
session_start();
include("conn.php");

$customer_id = $_SESSION['Customer_ID'];

$delete1 = mysqli_query($con, "DELETE FROM shopping_product WHERE
Shopping_ID=(SELECT Shopping_ID FROM shoppingcart WHERE Customer_ID = $customer_id AND Status ='unpaid')");

if ($delete1) {
    $delete2 = mysqli_query($con, "DELETE FROM shoppingcart WHERE Customer_ID = $customer_id AND Status = 'unpaid'");
    echo "<script>alert('All items in your cart have been cleared.')";
    window.location.href ='userprofile.php'</script>";
    mysqli_close($con);
}
else {
    echo"<script>alert('Failed to clear the cart.')";
    window.location.href ='userprofile.php'</script>";
    mysqli_close($con);
}?

```

Figure 41: PHP used to delete all items in the shopping cart

To delete all items from the shopping cart, a session must be started, followed by including the conn.php file and also getting the customer Id from the session. Moving on, a sql query \$delete1 is defined and executed. In the query, records from the **shopping product** table will be deleted

where conditions such as the unpaid status as well as whether the customer id of the session user is the same as the record in database are met.

If \$delete1 is executed, it will move on to execute \$delete2, where a query to delete records from the **shopping cart table** where the table's customer id matches the session user's customer id as well as if the status matches unpaid. After the \$delete2 is executed, an alert message is displayed, showing that all the items are successfully cleared. Both the if and else statements have the same redirect outcome, where users will be redirected to the user profile php page and the opened SQL database will be closed with mysqli_close().

5) Login

```
<!--Login Form-->
<div class="container">
  <div class="selectcont">
    <div class="formtext">
      <table>
        <tr>
          <td>Have an existing account?<br><a href="login.php">Login here!</a></td>
        </tr>
        <tr>
          <td>Email</td>
          <td>:</td>
          <td><input type="email" name="email" required="required" placeholder="Enter your email"></td>
        </tr>
        <tr>
          <td>Password</td>
          <td>:</td>
          <td><input type="password" name="password" required="required" placeholder="Enter your password"></td>
        </tr>
        <tr>
          <td>&ampnbsp</td>
          <td>&ampnbsp</td>
          <td>
            <input type="submit" value="Login">
            <input type="reset" value="Reset">
          </td>
        </tr>
      </table>
      <a href="changepassword.html">Forgot your password? Click here!</a>
    </div>
  </div>
</div>
```

Figure 42: HTML and form for login

The HTML form for login's action is in the login.php file and uses the post method. The login form requests for email and password. All inputs are required, and users cannot proceed without giving those 2 inputs. Placeholders such as “Enter your email” and “Enter your password” are included so users that might be confused will know what to type.

```

<?php
include ('conn.php');
session_start();
if($_SERVER["REQUEST_METHOD"] == "POST")
{
    $pemail = $_POST['email'];
    $ppassword = $_POST['password'];

    $email=mysqli_real_escape_string($con,$pemail);
    $password=mysqli_real_escape_string($con,$ppassword);

    $login ="SELECT * FROM customer WHERE email='".$email' and
password='".$password'";

    $retrieve = "SELECT Customer_ID FROM customer WHERE email = '$pemail'";
    $sql = mysqli_query($con, $retrieve);
    if($sql){
        $fetch = mysqli_fetch_assoc($sql);
    }

    if ($result=mysqli_query($con,$login)) {
        $rowcount=mysqli_num_rows($result);
    }
    if($rowcount==1) {
        $_SESSION['Customer_ID'] = $fetch['Customer_ID'];
        header("location: homepage.php");
        echo '<script>
            alert ("Login Successful. Welcome back!")
            </script>';
    }
    else {
        echo
        '<script>
            alert ("Invalid credentials, please try again.")
            window.location.href: ../loginform.html";
            </script>';
    }
}
mysqli_close($con);
?

```

Figure 43: PHP codes used for login

In the php codes for login, like always it starts by including the database connection, and for the session to start. Moving on, if the request method made by the server is post, the codes will execute. The variables \$pemail, \$ppassword, \$email and \$password to get the password and email input as well as escape special characters are set.

After that, a sql query is set where all the data from the table customer is selected, where the email and password from the database matches with the input given by the user. Another sql query to retrieve records by selecting customer id from the customer table where the email in the database matches the email input by the user. The retrieve sql code is put into the variable \$sql as a parameter for mysqli_query to retrieve data from the database. The \$sql is then put into a \$fetch variable so it can fetch resulting rows as associative arrays.

Moving on, if the login query is run against the database, the number of rows for the \$login variable will be counted. Using a “==” comparison operator, both sides are compared with each other to make sure that the number of rows counted is 1. If it is true, the customer id is the same as the customer id fetched from the database. The users will then be redirected to the homepage.php page.

They will not be redirected to a html page because they are considered logged in now, meaning sessions will be used to provide different features functionalities. One example of that is the login button in navigation bar for logged in customers will no longer have a dropdown and will change to log out. If the login query is not executed and it passed from the if to else statement, an alert message that states “Invalid credentials, please try again” will be echoed. The users will then be redirected to the loginform.html to try again. Lastly, the connection to the database will be closed.

6) Signup

```
<!--Sign Up Form-->
<div class="container">
  <div class="selectcont">
    <div class="formtext">
      <table>
        <tr>
          <td>Wish to create an account?  
Sign Up here!</td>
        <tr>
          <td>Username :</td>
          <td><input type="text" name="username" required placeholder="Choose a username"></td>
        <tr>
          <td>Name :</td>
          <td><input type="text" name="name" required placeholder="Enter your name"></td>
        <tr>
          <td>Email :</td>
          <td><input type="email" name="email" required placeholder="Enter your email"></td>
        <tr>
          <td>Password :</td>
          <td><input type="password" name="password" required placeholder="Enter a password"></td>
        <tr>
          <td>Confirm Password :</td>
          <td><input type="password" name="checkpassword" required placeholder="Re-enter your password"></td>
        <tr>
          <td>
            <input type="submit" value="Submit" name="submitinput">
            <input type="reset" value="Reset" name="resetinput">
          </td>
        </tr>
      </table>
    </div>
  </div>
</div>
```

Figure 44: HTML and form for Signup

To signup, users will have to fill in a form with their username, name, email, password and confirm their password by typing it in again. Every input is required and is accompanied with a placeholder to show what needs to be used as input.

```

?php
include('conn.php');

$email = mysqli_real_escape_string($con, $_POST['email']);
$password = mysqli_real_escape_string($con, $_POST['password']);
$checkpassword = mysqli_real_escape_string($con, $_POST['checkpassword']);
$checkemail = "SELECT * FROM customer WHERE email = '$email'";
$validate = mysqli_query($con, $checkemail);
$insert="INSERT INTO customer (username, name, email, password) VALUES ('$_POST[username]','$_POST[name]', '$_POST[email]', '$_POST[password]')";

if($password == $checkpassword){
    if(mysqli_num_rows($validate) == 0){
        if (mysqli_query($con, $insert)) {
            echo
            '<script>
            alert("Thank you for signing up! Please login to access the other features!");
            window.location.href = "../loginform.html";
            </script>';
        }
        else {
            echo
            '<script>
            alert("Sign up failed. Please try again.");
            window.location.href = "../signupform.html";
            </script>';
        }
    }
    else {
        echo
        '<script>
        alert ("An account under this email already exists. Please log in instead.")
        window.location.href = "../loginform.html";
        </script>';
    }
}
else{
    echo
    '<script>
    alert ("Please enter the same password!")
    window.location.href = "../signupform.html";
    </script>';
};

mysqli_close($con);
??

```

Figure 45: PHP used for signing up

Variables \$email, \$password and \$checkpassword are used to receive the input from the submitted form. Other than that, the \$checkemail variable selects all from the customer table where the email in the database is equals to the email input submitted by the user. The \$checkemail is then put into a mysqli_query function under the variable \$validate to perform the query on the database. The other variable is to insert the data into the database. Using the INSERT INTO statement, the values, which are input from the forms submitted by the users are inserted into the customer table.

Validation is also added. This includes the checking of passwords to make sure the initial password and the second password input is the same. If the initial password is not the same with the second password, the users will be informed that the sign up failed and they should try again before redirecting them back to the signupform.html page. If this validation passes, the next if

statement will execute, where the email input by the users will be cross checked with the email in the database. This is to prevent multiple different users from signing up with the same email address. Should the data not pass this validation, an alert message that says “An account under this email already exists. Please log in instead.” will be alerted to the users, then they will be redirected to the loginform.html page to log in instead. If the data also passes through this validation, it will then execute the query to insert the data into the database. After the data is inserted into the database, an alert message will be shown, stating that the sign up was successful and to ask the users to log in to access other features. The user will then be redirected to the loginform.html to log in. After running through the if else statements, the connection to the database will close.

7) Self-created CSS and JavaScript

```
.content-flexbox-container {  
    margin: 50px;  
    height: auto;  
    width: auto;  
    display: flex;  
    flex-wrap: wrap;  
    align-content: center;  
    justify-content: center;  
    align-items: center;  
}  
  
.content-card {  
    margin: 25px;  
    padding: 25px;  
    height: auto;  
    width: 350px;  
    background-color: #EEE2DC;  
    display: flex;  
    align-content: center;  
    align-items: center;  
    flex-direction: column;  
    border-radius: 25px;  
}  
  
.content-imagebox {  
    height: auto;  
    width: 350px;  
    display: flex;  
    justify-content: center;  
}  
  
.content-imagebox img {  
    max-width: 100%;  
    border-radius: 10px;  
}  
  
.content-descriptionbox {  
    height: auto;  
    width: 350px;  
    display: none;  
    align-content: center;  
    align-items: center;  
    flex-direction: column;  
    text-align: center;  
}  
  
.content-descriptionbox h1 {  
    display: none;  
}
```

Figure 46: CSS codes used to display content

```
.content-descriptionbox h2 {  
    font-size: 20px;  
    font-weight: bolder;  
    font-family: 'Montserrat', sans-serif;  
    letter-spacing: 1px;  
}  
  
.content-descriptionbox h6 {  
    margin: 0px;  
    font-size: 14px;  
    font-weight: bold;  
    font-family: 'Rubik', sans-serif;  
    letter-spacing: 0.5px;  
}  
  
.content-descriptionbox p {  
    font-size: 18px;  
    font-weight: bold;  
    font-family: 'Rubik', sans-serif;  
    letter-spacing: 1px;  
}  
  
.content-descriptionbox form input[type=submit] {  
    margin-top: 15px;  
    padding: 10px;  
    font-size: 20px;  
    font-weight: bolder;  
    font-family: 'Montserrat', sans-serif;  
    letter-spacing: 0.5px;  
    color: #EAЕ7DC;  
    background-color: #24305E;  
    border: none;  
    cursor: pointer;  
    animation-name: color-change-pddb;  
    animation-duration: 2.5s;  
    animation-fill-mode: forwards;  
    animation-iteration-count: infinite;  
    animation-direction: alternate-reverse;  
}  
  
@keyframes color-change-pddb {  
    from {  
        box-shadow: transparent;  
    }  
    to {  
        box-shadow: 5px 5px #374785, -5px -5px #A8D0E6;  
    }  
}  
  
.content-card:hover .content-descriptionbox {  
    display: flex;  
}
```

Figure 47: Continuation of CSS codes used to display content

```
.content-card:hover .content-descriptionbox {
    display: flex;
}

.content-descriptionbox form input[type=number] {
    padding: 10px;
    font-size: 20px;
}

.content-descriptionbox form input[type=number]:focus {
    animation-name: color-fill;
    animation-duration: 1s;
    animation-fill-mode: forwards;
    background-color: transparent;
}

@keyframes color-fill {
    from {
        background-color: #ffffff;
    }
    to {
        background-color: #A8D0E6;
    }
}
```

Figure 48: Continuation of CSS codes used to display content

As shown in the multiple figures above, those are the CSS codes that are used to style the page used to display content. Selectors along with the declaration of properly and values are used. These CSS codes were used to align, change color of both text and background, position content,

choose how content will be displayed and even the height and width are specified. Many different division classes are set so it can be used easily in HTML or PHP.

```
<!--Content Flexboxes-->
<div class='content-flexbox-container'>
    <div class='content-card'>
        <div class='content-imagebox'>
            <img src='art/catharness.jpg'>
        </div>
        <div class='content-descriptionbox'>
            <h2>Cat Harness</h2>
            <h6>"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.
                Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
                Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."</h6>
            <p></p>
            <button>TAKE IT HOME!</button>
        </div>
    </div>
    <div class='content-card'>
        <div class='content-imagebox'>
            <img src='art/dogharness.jpg'>
        </div>
        <div class='content-descriptionbox'>
            <h2>Dog Harness</h2>
            <h6>"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.
                Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
                Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."</h6>
            <p></p>
            <button>TAKE IT HOME!</button>
        </div>
    </div>
    <div class='content-card'>
        <div class='content-imagebox'>
            <img src='art/catsweater.jpg'>
        </div>
        <div class='content-descriptionbox'>
            <h2>Cotton Sweater for Cat</h2>
            <h6>"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.
                Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
                Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."</h6>
            <p></p>
            <button>TAKE IT HOME!</button>
        </div>
    </div>
</div>
```

Figure 49: HTML, using CSS to display content

Placing the content within the division class, any text inside the division class will follow the styling, alignment, colors, etc set. If there are any inline styles, the inline styles will take priority over the division classes referenced from the CSS files.

```
/* Back to Top Button */
#back2top-btn {
    padding: 10px;
    display: none;
    position: fixed;
    z-index: 100;
    bottom: 20px;
    right: 20px;
    border: none;
    border-top-left-radius: 25px;
    border-bottom-right-radius: 25px;
    color: #E98074;
    background-color: #24305E;
    cursor: pointer;
}

#back2top-btn:hover {
    color: #AC3B61;
}
```

Figure 50: Back to top button

The figure above shows the CSS codes used to style the back to top button. CSS styling such as position fixed ensures that the button follows the user wherever it scrolls, the z-index indicates the button's priority when being displayed, a higher number means it will be displayed on top of more things. When users hover over the button, their cursor will change to a pointer.

```
// Back to Top Button
window.onscroll = function() {back2topFunction()};

function back2topFunction() {
    var back2top = document.getElementById('back2top-btn');
    if (document.documentElement.scrollTop > 500) {
        back2top.style.display = 'block';
    } else {
        back2top.style.display = 'none';
    }
}
```

Figure 51: JavaScript for back to top button

Right when users scroll, a function will execute. The function is the back2topfunction, where if users scroll to a certain extent, the button will appear at the bottom right side of the screen and if clicked, will bring the users back to the top of the page. However, if the users are right at the top, the back to top button will not appear.

5.0 Main section (User Guidance and Sample Screens)

5.1 General Functionalities

The screenshot shows the homepage of the Les Pet Shop website. At the top, there is a navigation bar with links for HOME, PETS, FOOD, ACCESSORIES, PROFILE, and LOGIN. A small logo featuring a puppy and the text 'Les Pet Shop' is located in the top left corner. The main banner has a dark blue background with white stars and a central illustration of various pets including a cat, a dog, a bird, and a lizard. The text 'Les Pet Shop' is written in a large, stylized font, followed by 'WHERE YOU'LL FIND YOUR NEW BEST FRIEND!' Below the banner, there is a section titled 'Our Story' containing text about the founders' mission to source pets from ethical breeders. Another section titled 'Why Us?' provides information about the ethical sourcing of animals. The bottom of the page features a decorative footer with illustrations of various pet faces.

Les Pet Shop

WHERE YOU'LL FIND YOUR NEW BEST FRIEND!

Our Story

The founders of Les Pet Shop wanted to provide the most decadent treats, luxurious accessories and most comfortable beds so owners everywhere can spoil their beloved companions. On top of that, we here at Les Pets Shop sources our wide selection of animals, that ranges from cats, dogs to sugar gliders and iguanas from ethical breeders, so soon-to-be pet owners can rest easy knowing their pets did not come from breeding mills.

Why Us?

As mentioned earlier, we source our pets from ethical breeders and not breeding mills. This means that the code of ethics were closely followed when handling these animals. Supply sourcing also comes from high quality production companies that tailor to your pet's best interest.



Figure 54 - 56: Les Pet Shop Homepage Design

Home Page

The core of our Les Pet Shop Website is the homepage. The homepage is where we let customers learn more about us, what we do, and who we are by displaying them in containers and aligning them. Recommendation advertisements for new customers or existing customers about our pet food and accessories are done and some are static, some are included as slideshows for our website design layout.

In the popular and recommended section's slides banners that advertise Les Pet Shop's items are displayed. By clicking on the arrows located at the end of each banner image, users can navigate to and from banner slides. The container below it is the "our services" container. Here, there are three boxes displaying more advertisements and when hovered over, a text overlay will ease in. Users can also click on the text in the text overlay to be redirected to the corresponding page. At the end of the page before the footer are contact forms. Here, users can send the staff at Les Pet Shop a message if they have any inquiries. Once sent, they a status message showing the status of the submit message will be shown. Besides, it includes the phone number and various links that users can click on to view Les Pet Shop's social media.

As an online pet shop, we also let customers browse our pet available in our physical pet shop as we can deliver to them whenever they purchased online during this pandemic. We also have some rare and exotic animals which we will need to the customer to apply for a one-year license from the Department of Wildlife and National Parks (PERHILITAN) and personally come to our shop with the permit as prove.

Login/Signup

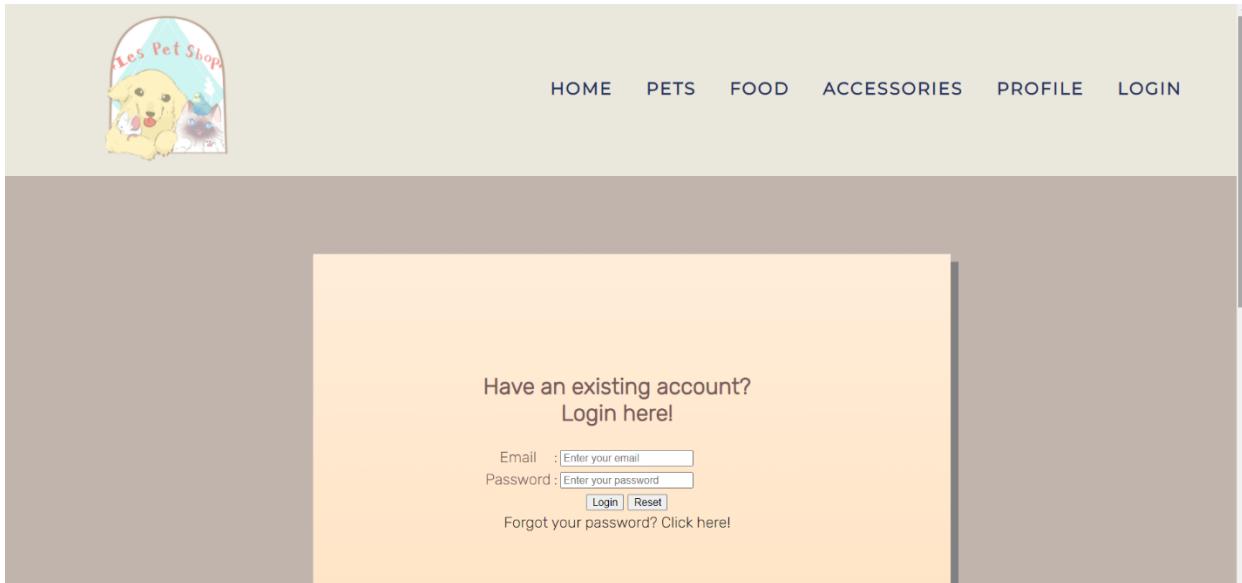


Figure 57: Les Pet Shop login page with ‘forgot password’ function

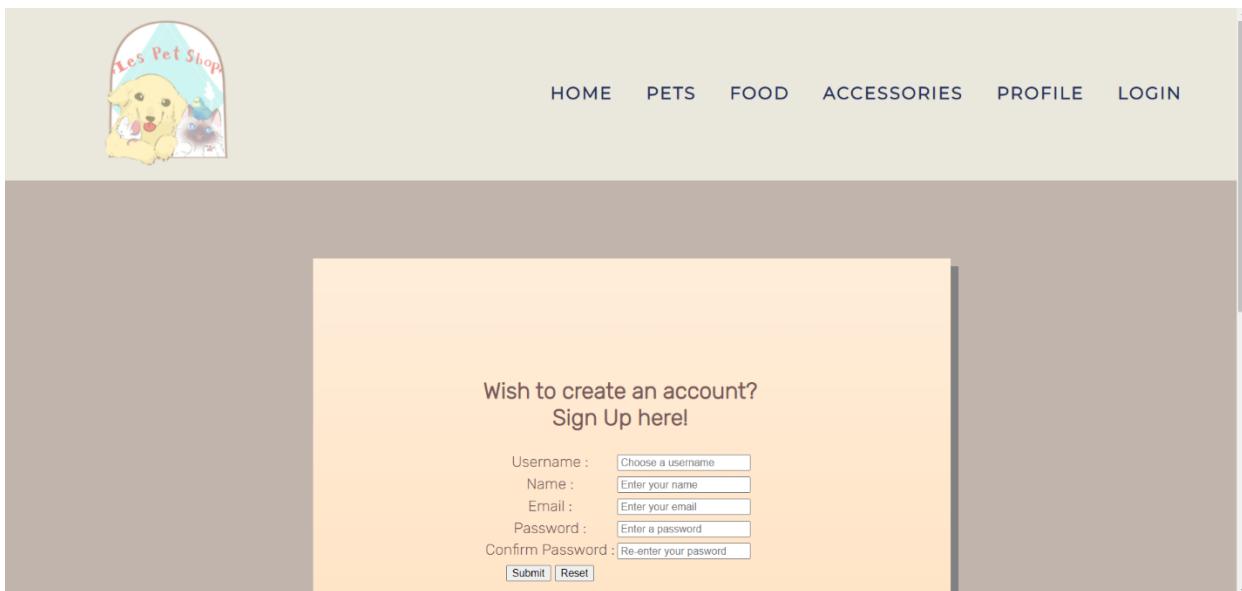


Figure 58: Les Pet Shop signup page

Before new or existing customer and even admin of Les Pet Shop will need to login to our website in order to have other registered customer functions like browsing, and purchasing for customer, adding, and editing products for admin. If customers who are new to our website, they will need to provide their username, name, email address, and their intended password. Users who forgot their password will also be able to change it by clicking on the “Forgot your password? Click here!” button.

Exclusively for admin, only one admin can access the website in admin mode. The Admin ID is 1, username is petadmin, password is lespetshop. For future, we might consider recruiting more trustworthy and friendly admin to access and help out in our Les Pet Shop website.

After logging in, the login button in the navigational bar for users will change to logout, whereas for admins, the profile button will disappear as admins do not have a profile and the login button will change to a logout button as well.

View product

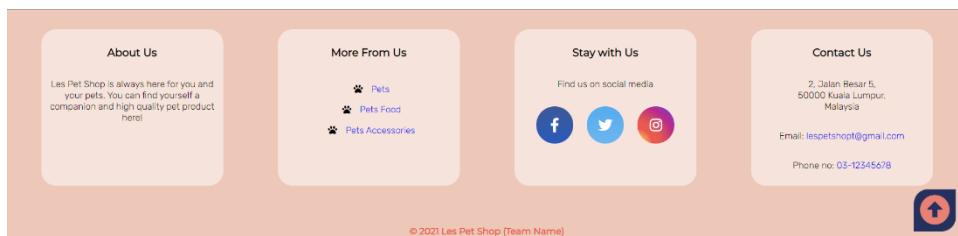
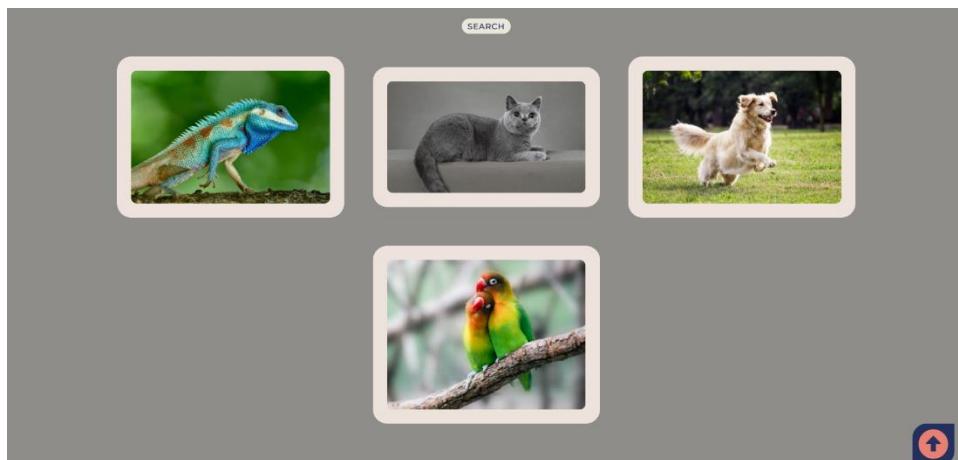
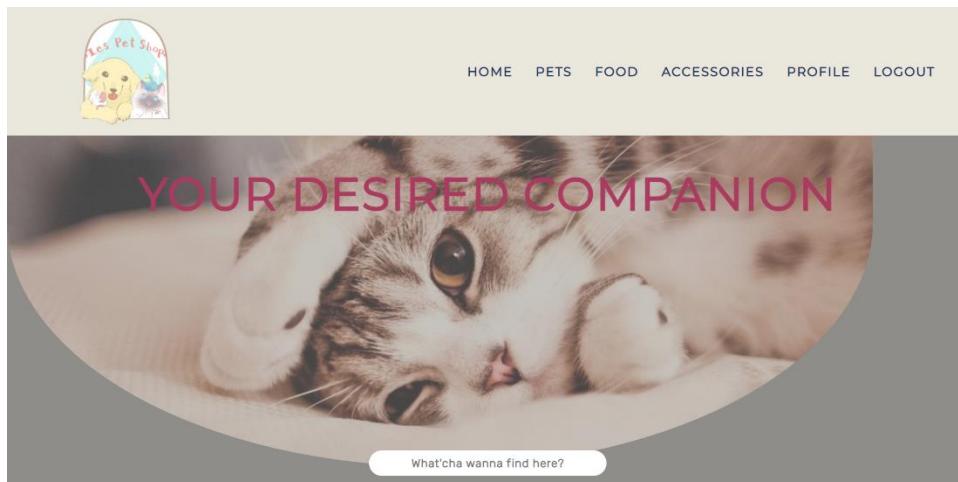


Figure 60: View pets page

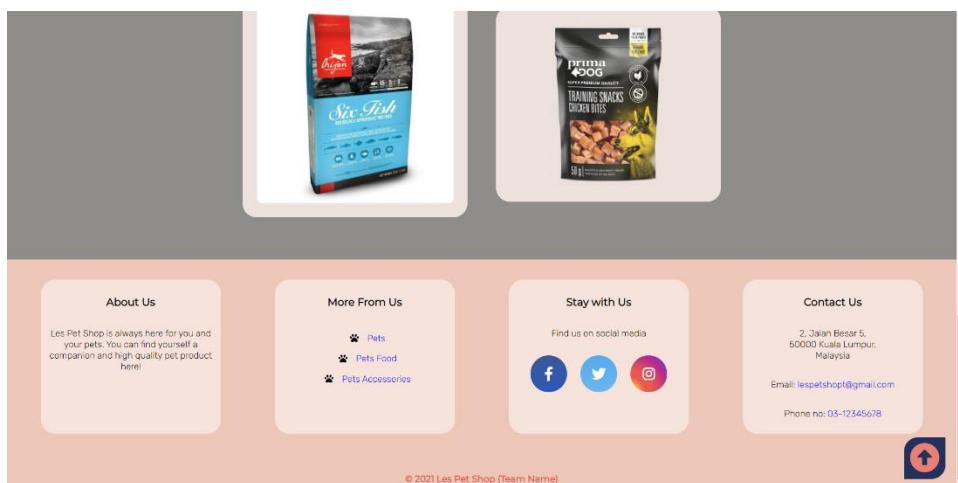
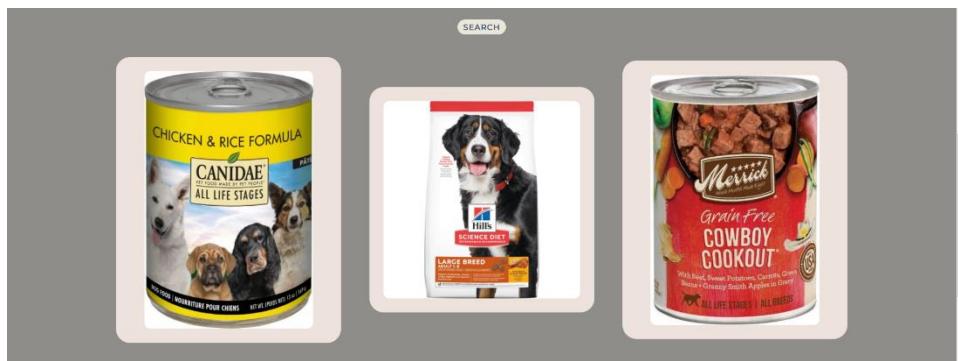
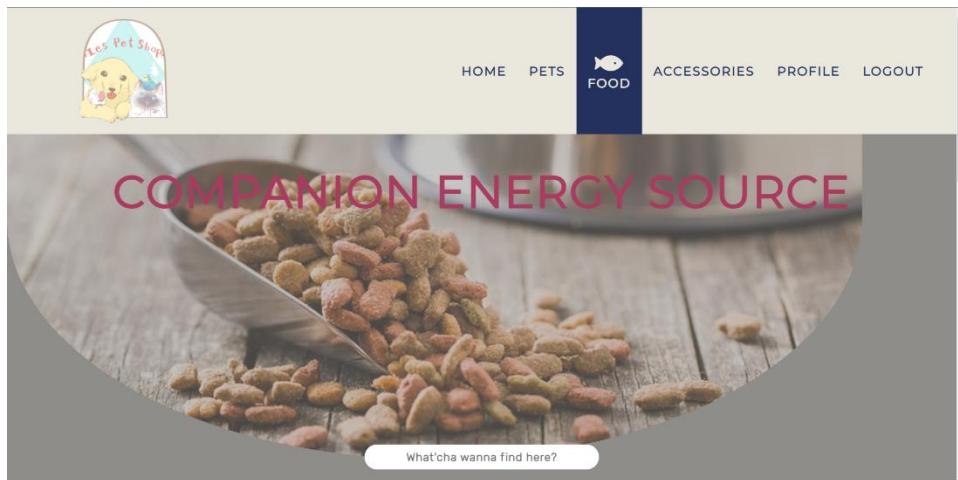


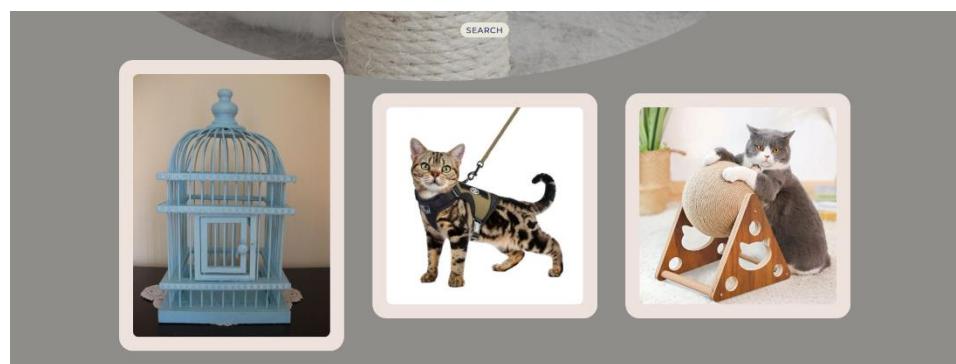
Figure 61: View food page

Les Pet Shop

HOME PETS FOOD ACCESSORIES PROFILE LOGOUT

LUXURIOUS PET LIFE

What'cha wanna find here?



About Us
Les Pet Shop is always here for you and your pets. You can find yourself a companion and high quality pet product here!

More From Us
Pets
Pets Food
Pets Accessories

Stay with Us
Find us on social media
[f](#) [t](#) [i](#)

Contact Us
2, Jalan Besar 5,
50000 Kuala Lumpur,
Malaysia
Email: lespetshop@gmail.com
Phone no: 03-12345678

© 2021 Les Pet Shop (Team Name)

Figure 62: View accessories page

Admin and customers both can view our Les Pet Shop available products, which initially will display the product image.

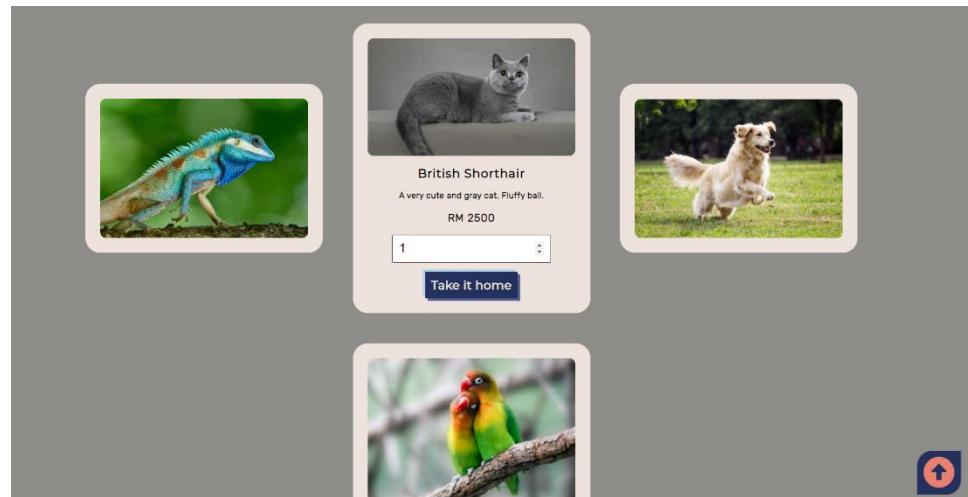


Figure 63: Product details shown when hovered

When users hover over it, it will display the product's name, description, price tag in Malaysian Ringgit, and the exclusive button for customer and admin.

Our Les Pet Shop product is categorized into three sections, Pets, Food, and Accessories. In Pets Section, users can view all the available pets for sale. In Food Section, users can see all the pet food which includes dry and wet food or even snack bites for their pets. In Accessories Section, they will be surprised with a variety of fashionable pet clothes, kennels, cages, toys for pets and much more.

To cope with our variety of products, we have implemented a Search bar for customer to browse for the product that wanted or seeking for the product they hope to find in our Les Pet Shop website. Each search bar will only retrieve the product of their category respectively.

[**Logout**](#)



Figure 64: Logout button

If a user wants to logout from our website, they can just simply click on the “LOGOUT” button. Clicking on this button will destroy the session the users are in. After that, they will be redirected back to the homepage where the login button in the navigation bar will appear again for future login.

5.2 Admin Functionalities

Add new product

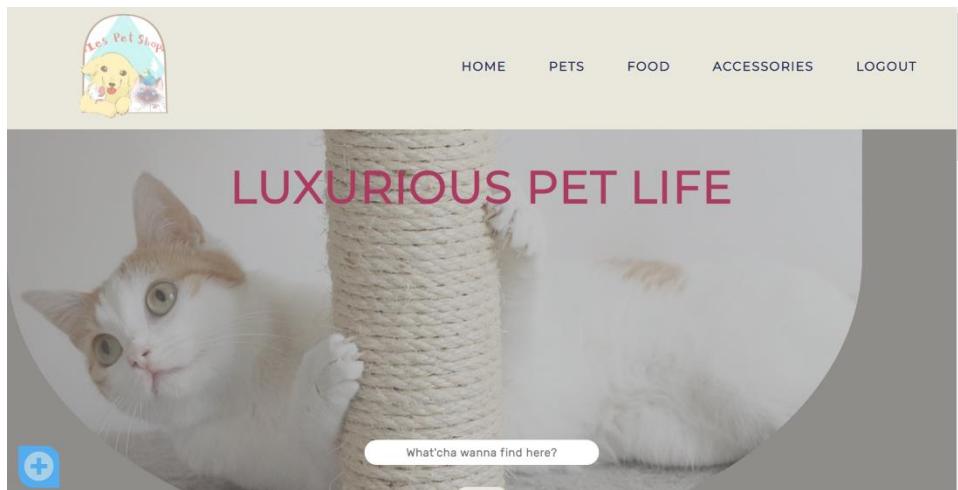


Figure 65: Add button at left bottom corner

A screenshot of a "Add new product" form page. The top navigation bar includes links for HOME, PETS, FOOD, ACCESSORIES, PROFILE, and LOGOUT. The main content area has a light beige background. At the top of the content area, the text "WHAT DO YOU WANT TO ADD?" is centered. Below it is a row of three radio buttons labeled "NEW PETS", "NEW FOOD", and "NEW ITEMS/FASHION". A "Display Image:" label is followed by a file input field containing the placeholder "Choose File | No file chosen". There are two text input fields: one for "Name:" and one for "Description:", both with their respective placeholder text. The overall layout is clean and organized.

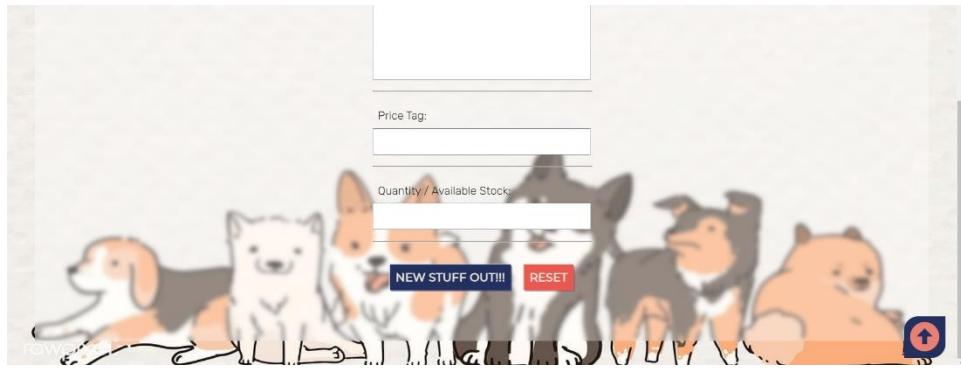


Figure 66: Add product form

When there is new stock or products to be launched or to be sold in our physical shop, we will also update it into our website database, so that our faraway customers are able to browse our newest product without the trouble of visiting our physical shop.

As in admin form or login as admin, when browsing or checking in either pets, food, or accessories Section, there will be blue sticky button on the bottom left corner. That is where admin get directed to the page which they can add a new product.

In the adding new product page, admin will be able to choose the category they want, the product name, description, price tag, and stock available. After including the new product details, which is a must or admin cannot submit the new product to insert into database, admin can click on the “NEW STUFF OUT!” button to display it to our customer.

Edit product details

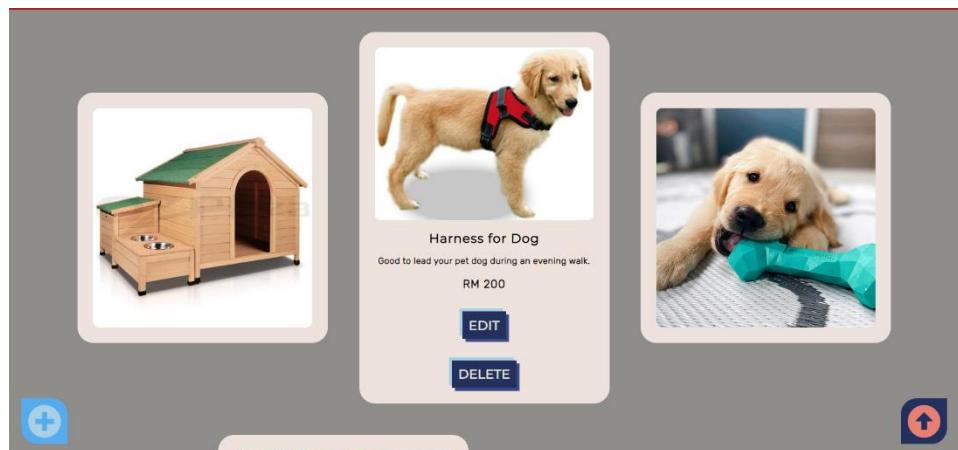


Figure 67: Edit and Delete buttons

Aside from adding new products, admin is able to edit or delete the existing products in order to keep in line with the society trend. In admin mode, when hover over the product image, the button available for admin will be “EDIT” and “DELETE”.

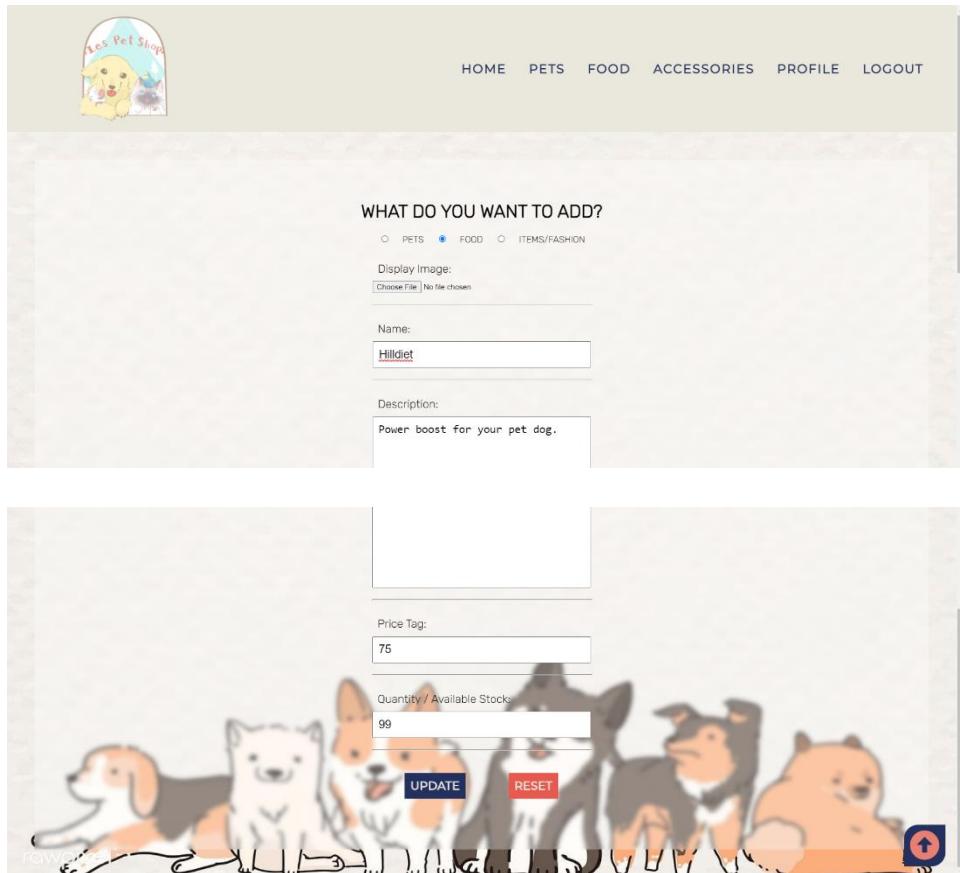


Figure 67: Edit product page

When they click on “EDIT”, they will get prompted to edit the product before being redirected to the same page as adding new product, but the existing information will be included for editing except for the description as admin might need to change it. After editing, admin will need to click on “UPDATE” to update the data existed in the database.

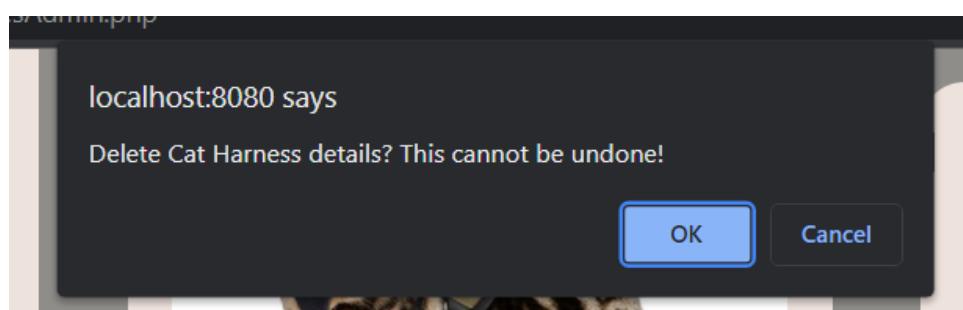


Figure 68: Delete confirmation

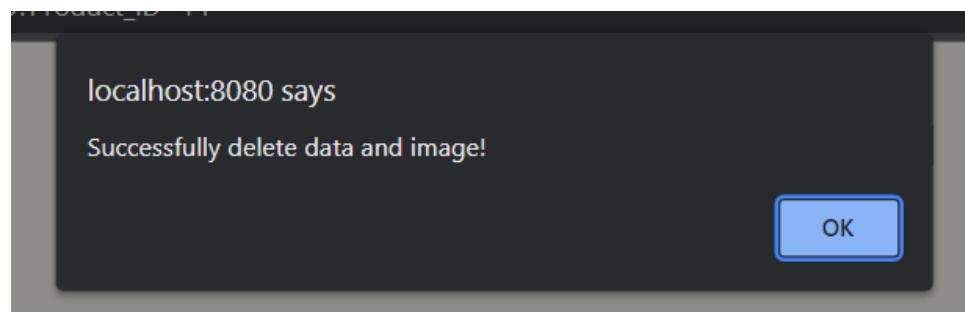


Figure 69: Delete successful alert

When they click on “DELETE”, they will get a “Confirm delete” message for safety and accident purposes. If they confirm, then the page will refresh, and the product will be gone. If they cancel, they will just remain on the same page.

5.3 Customer Functionalities

User profile

A screenshot of a user profile page for a pet store. At the top, there is a navigation bar with links for HOME, PETS, FOOD, ACCESSORIES, PROFILE, and LOGOUT. On the left side, there is a sidebar with a profile picture placeholder labeled "Profile" and "Personal Information". The main content area has a decorative background of paw prints and a cat icon. It displays personal information fields: Username (Ekay), Name (chan), Contact number, E-mail (hello@hotmail.com), Password (*****), and Address. A link to "Modify your personal information" is also present. Below this, there is a "Shopping Cart" section with a table showing items: Item Name (Happy), Price (Pet), Quantity (Shopping), and Total (0). The total value is "Total: RM 0".

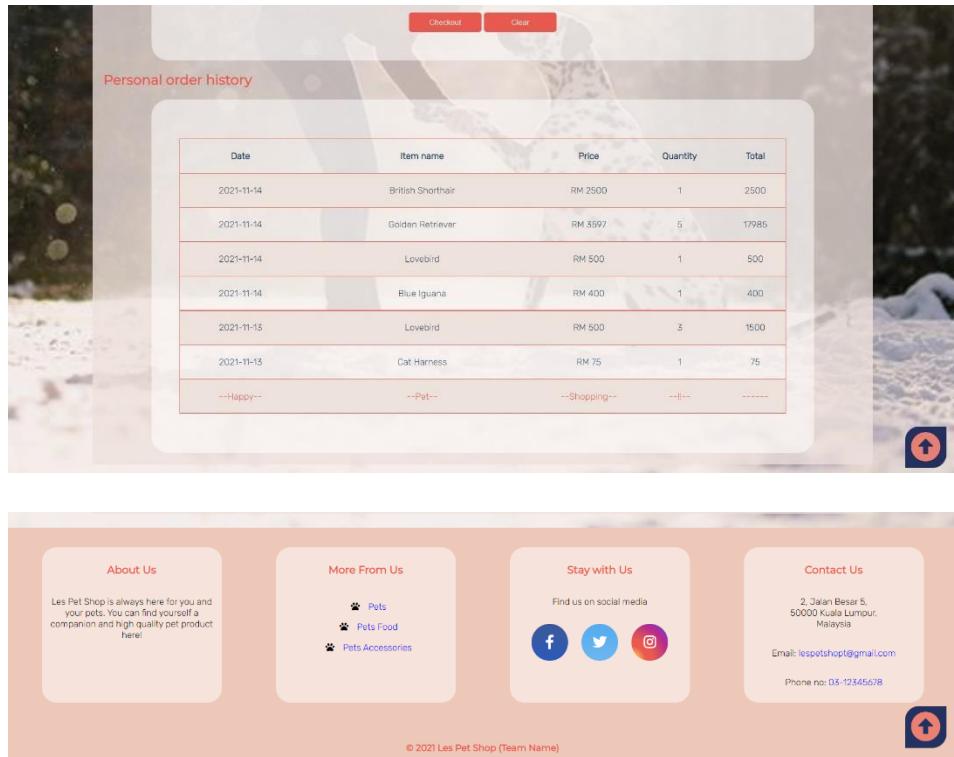


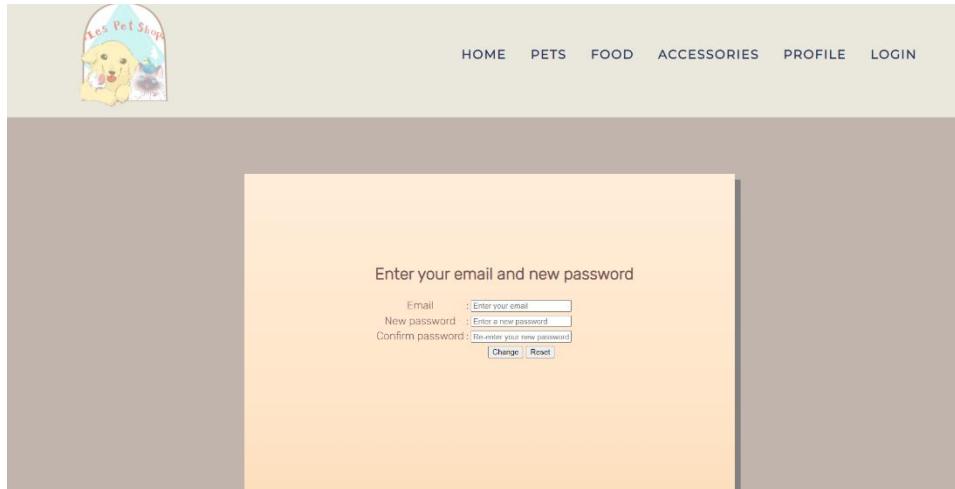
Figure 70: User profile page

Before registering, anytime users click on the profile page, they will be redirected to the login page. After registering, users can log in with their email and password to access other pages, after successfully accessing the user profile page, users can see a profile page containing information such as personal information, shopping cart and personal order history.

At the top of the page, a cat profile picture and a banner are displayed. Below it is a container for personal information such as username, name, contact number, e-mail, password, and address. The username, name, e-mail, and password automatically update itself after users register. Other information such as contact number and address can be added by clicking on the modify your personal information button. Aside from adding the contact number and address, users can also change their other personal information through that button.

Upon scrolling further down, students can see their shopping cart displayed in table form. The columns of the table include item name, price, quantity and total. Below the shopping cart, users can see their own personal order history. Any transactions made will be displayed here. The columns for this table are date, item name, price, quantity and total.

Forgot password



The screenshot shows a web page for 'Pet Pet Shop'. At the top, there's a logo of a puppy and a menu bar with links for HOME, PETS, FOOD, ACCESSORIES, PROFILE, and LOGIN. The main content area has a light orange background and contains the following text and fields:

Enter your email and new password

Email:

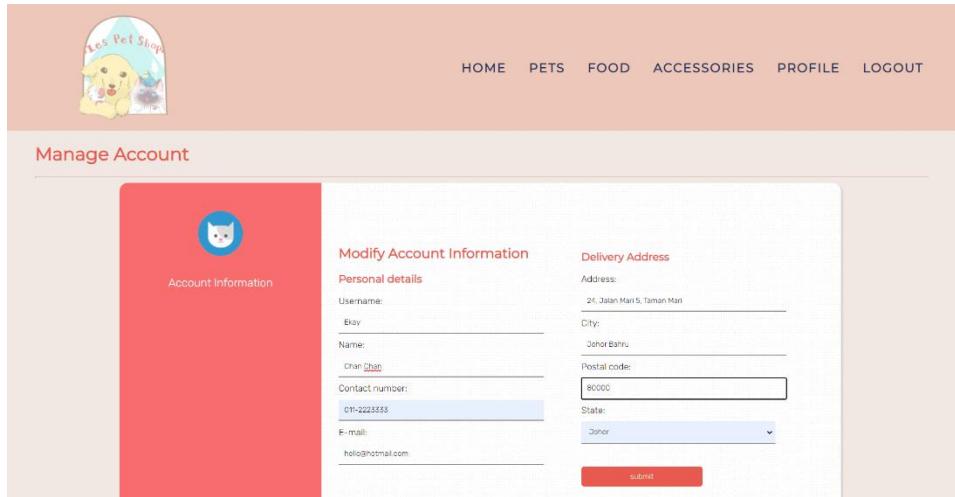
New password:

Confirm password:

Figure 71: Forgot password form

Of course, we do let our customers reset their password if they forgot their initial password. They can just click on “Forgot your password” at the login page, then they will be redirected to the forgot password page where they need to include their email address and their new password.

Modify personal details



The screenshot shows a 'Manage Account' page for 'Pet Pet Shop'. The top navigation bar includes a logo, a menu bar with links for HOME, PETS, FOOD, ACCESSORIES, PROFILE, and LOGOUT, and a red header bar with the text 'Manage Account'. The main content area has a red sidebar on the left labeled 'Account Information' with a blue cat icon. The main form area contains two sections: 'Modify Account Information' and 'Delivery Address'.

Modify Account Information

Personal details

Username:

Name:

Contact number:

E-mail:

Delivery Address

Address:

City:

Postal code:

State:

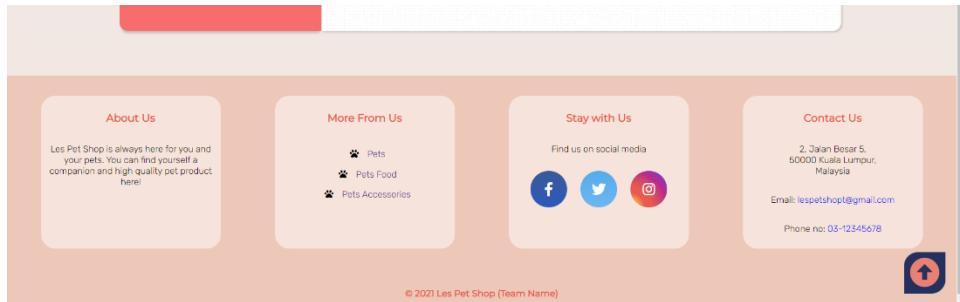


Figure 72: Account edit page

To modify the account details, users need to click on the modify personal information button. Upon clicking it, users will be redirected to a page to manage their account. Here, they will be able to see a form to modify their personal details and delivery address. They can change their username, name and e-mail address but they cannot change their password through this page.

Users can add on their contact number, address, city, postal code and select the state they live in. Once submitted, the changes can be seen when users get redirected back to the user profile page as the information will be displayed in the personal information container.

Shopping cart and checkout

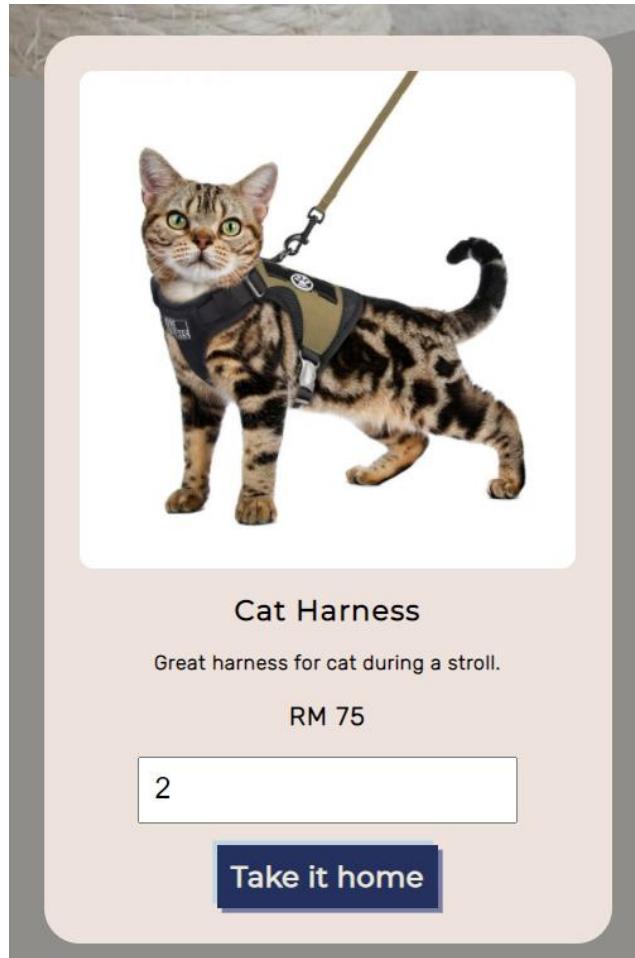


Figure 73: Set quantity and add item to cart

Shopping Cart			
Item Name	Price	Quantity	Total
British Shorthair	RM 2500	1	RM 2500
Cat Harness	RM 75	2	RM 150
Hilldiet	RM 130	1	RM 130
--Happy--	--Pet--	--Shopping--	-- --

Total: RM 2780

[Checkout](#) [Clear](#)

Figure 74: Shopping cart

While shopping, users can add items with the respective quantity to their shopping cart. Once users add anything from any one of the pets, food and accessories page, the item's information will be added into the shopping cart table. The total cost for all the items will be displayed at the bottom of the page. If the quantity of the items increases or decreases, the total at the bottom will increase and decrease accordingly.

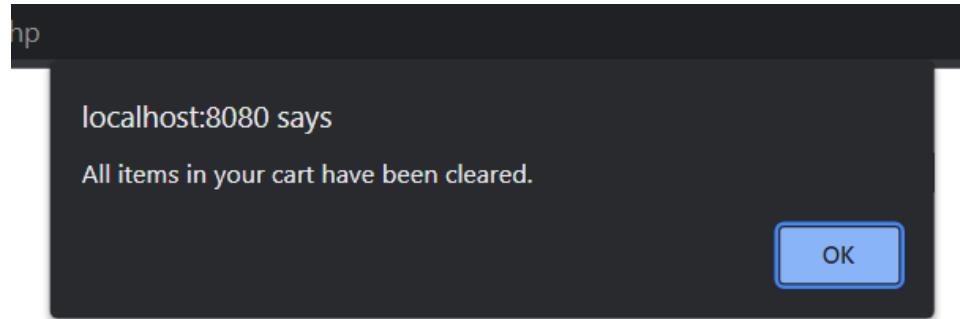


Figure 75: Clear Cart Alert

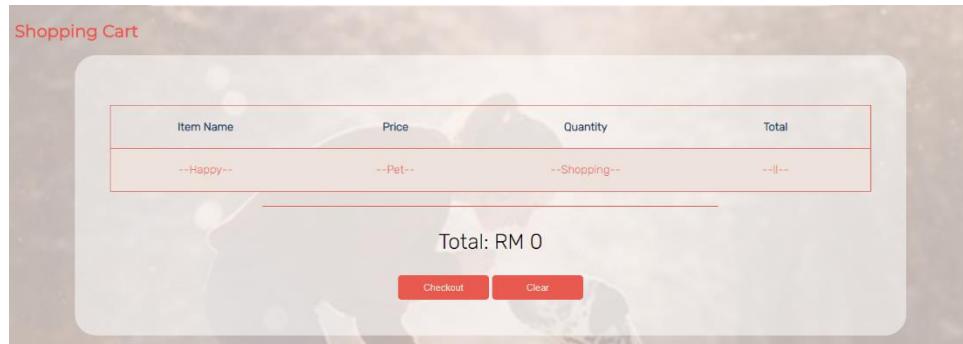


Figure 76: Shopping cart after cleared

Users can clear all the information in their cart by clicking the clear button. They can also checkout the items by clicking on the checkout button. When users click on the checkout button, they will be redirected to the payment page.

Payment

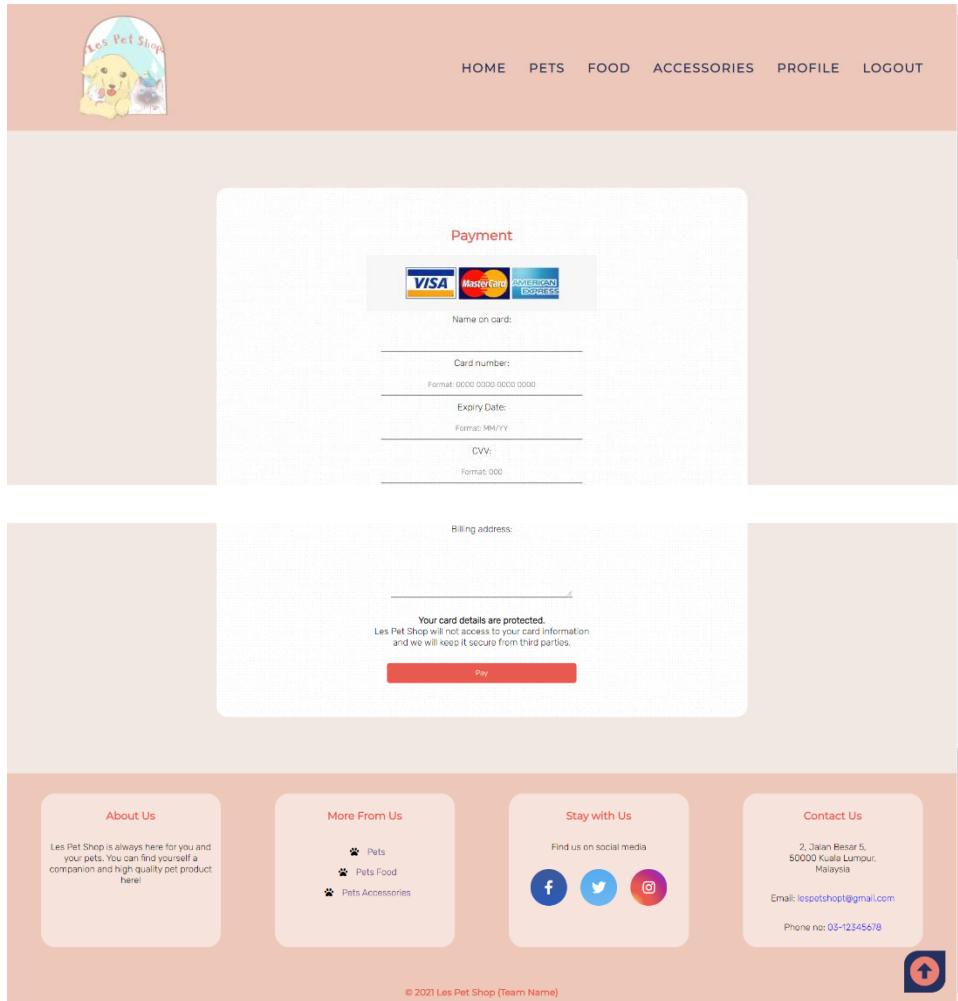


Figure 77: Payment page

After being redirected from clicking the checkout button, users can see a form where they will have to fill in their payment details. Payment information such as name on card, card number, expiry date, CVV and billing address is needed to complete the transaction.

The types of cards accepted by Les Pets Shop, which is the visa card, master card and American Express card is shown at the top. A message is displayed above the pay button, stating that the card details are confidential and will be kept secure from third parties. Placeholders to prompt users for the right input are also placed. Validation to check the input is also added. For example, users can only enter a certain range of dates for expiry date. Any input other than the one specified will not be processed, and users are made to retype the input again.

6.0 Conclusion

Limitations

The assignment given was difficult as we had to use various programming languages that we were unfamiliar with all at once. Some functions and features had to be discarded because there were errors with it and due to time constraints and lack of solutions to the specific problems, we were unable to resolve. This led to us deleting the functions completely. Discarded functions that were almost included were a profile picture selection feature, where users can select one out of two preset profile pictures. Other than that, a slide out menu that appears upon clicking for the login, sign up and admin access forms were also discarded due to alignment issues.

Assumptions

Besides our limitations of our Les Pet Shop website, we also assumed a few things.

- i. We assume that our website is completely secured and well-protected from malicious sources and hackers. This is because our website verification is rather simple and easy to get access.
- ii. We assume that our shop supplies will get restock every week or two, so that customers will not face the issue of out-of-stock or ran out-of-stock due to slow payment transaction.
- iii. We assume that the payment transaction is able to run and processed by bank as this is only a university assignment and there is no need for an actual verification of credit/debit card.

7.0 Appendix

7.1 References

Refsnes Data. (n.d.). *Online web tutorials*. <https://www.w3schools.com/>

Achour, M., Betz, F., Dovgal, A., Lopes, N., Magnusson, H., Richter, G., et al. (2021, October 29). *PHP manual*. <https://www.php.net/manual/en/index.php>

TutorialsPoint. (n.d.). *PHP – function references*.

https://www.tutorialspoint.com/php/php_function_reference.htm

Google Fonts. (n.d.). *Montserrat*. <https://fonts.google.com/specimen/Montserrat>

Google Fonts. (n.d.). *Rubik*. <https://fonts.google.com/specimen/Rubik?query=rubik>

Fonticons, Inc. (n.d.). *Font awesome*. <https://fontawesome.com/v5.15/icons?d=gallery&p=2>

Chibana, N. (2016, September 4). *50 gorgeous color schemes from award-winning websites*.

Visme. <https://visme.co/blog/website-color-schemes/>

7.2 Workload Matrix

Name	Workload
Yip Zi Xian	Documentation, Content page for customer and admin, Admin editing and adding product page
Neong Yee Kay	Documentation, User profile page (Checkout, Clear), Payment page, Edit account page
Wong Xie Ling	Documentation, Home page, Login page, sign up, Session, Admin access, Forgot password