ASSIGNMENT 2 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	10: Website Design & Development		
Submission date	9/5/2021	**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Phạm Hoàng Long	**Student ID**	200168
Class	GCH0903	**Assessor name**	Long
Student declaration I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.			
		Student's signature	Long

Grading grid

P5	P6	P7	M4	M5	D2	D3

☼ Summative Feedback:	☼ Resubmission Feedback:

Grade:	Assessor Signature:	Date:
Signature & Date:		

Contents

FIGURES

A. Introduction

For the company requirement to create and develop a shopping mall website, the author will make a report to guide users about principles of the website and make presentation for a website implementation. The author also review all appropriate principles, standards and guidelines for website designing and development, evaluate all technical challenges to produce a good design document for the online shopping website with wireframes, functional illustrations and a full set of client and user requirements. To summarise, this report will show users how to effectively create a website for client requirement satisfaction, list the principles of a good website and implement by building a website for example. The website name is “Weapon store fantasy” and use front end HTML, style CSS and PHP for back end.

B. Website design (P5)

I. Client and user requirement analysis

- Customers want to view products list.
- Customers want to view categories list.
- Customers want to filter products based on the category.
- Users can login as admin account if they have.
- Admin can add, view, update and delete products.
- Admin can add, view, update and delete categories.

- Admin can add, view, update and delete users.
- Admin can login and logout their account flexibly.

II. Case diagram

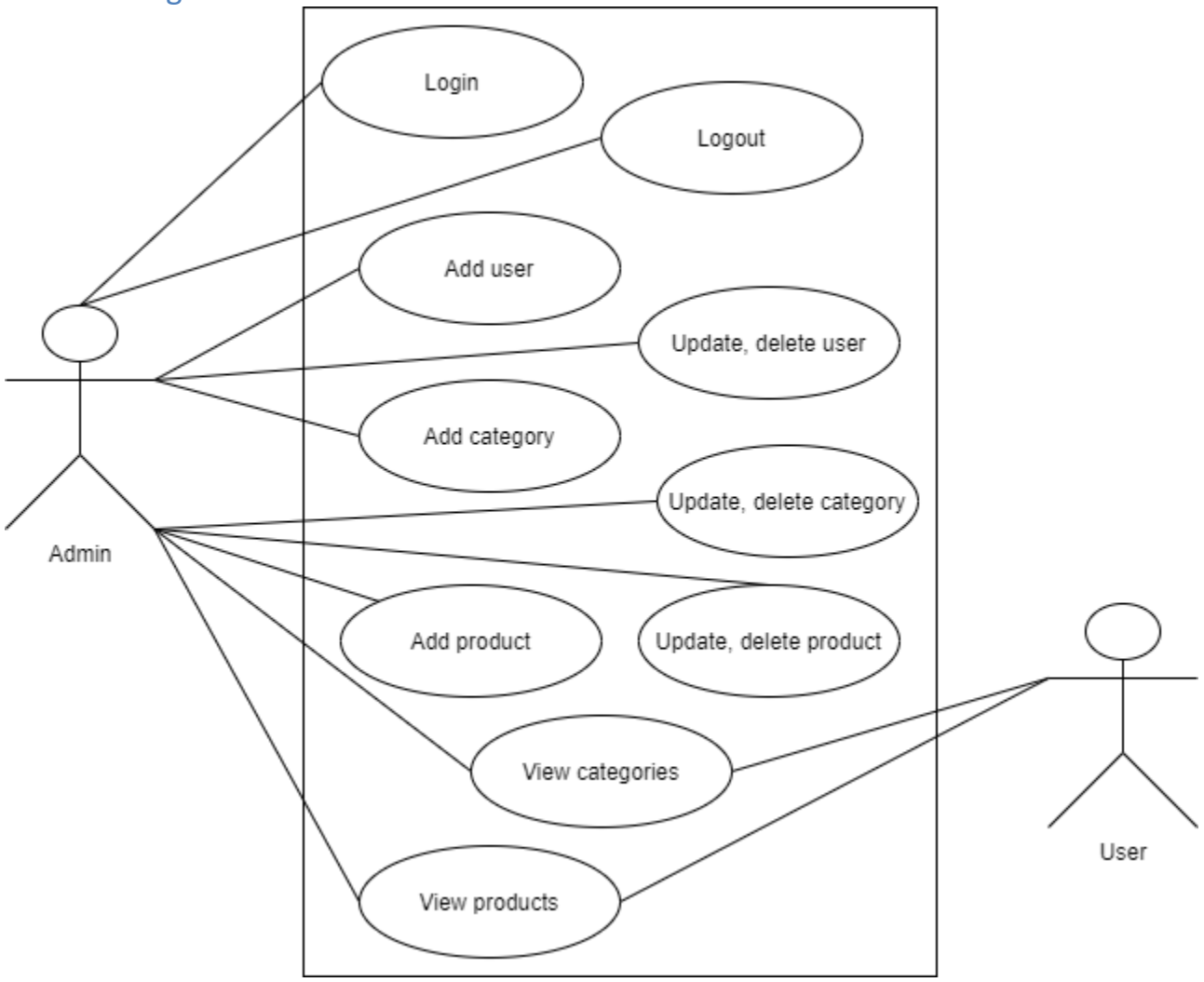


Figure 1. 1. Website case diagram

The case diagram explain the roles of the user and admin when accessing to the website pages. The admin have authorities to login as user with admin role to access to the admin website interfaces and can add new user, product, category and logout. Both admin and user can view the categories and products but the admin can aslo update and delete those current categories and products in their own interface. The admin can update current users and also delete them. User can also view the products that matched with their category by category details.

III. Database diagram

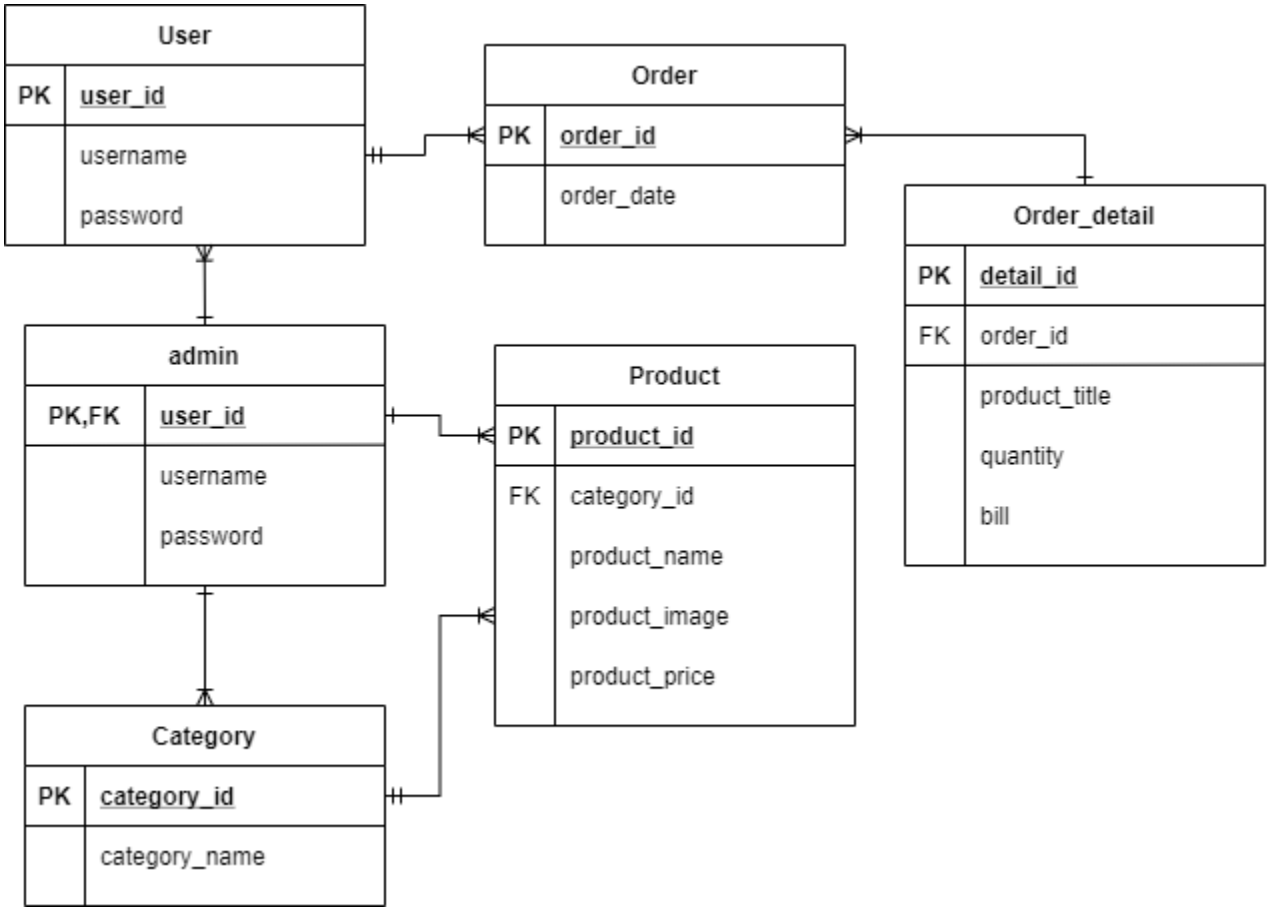


Figure 1. 2. Website database diagram

1. Business rules

- Each user has their unique ID, username and password. User can view categories, products, take orders and view order details.

- Each admin account has their unique ID, name and password. Admin can create user, update and delete accounts, categories, products and view orders, order details.
- Each product has its ID, category ID to classify, name, image and price.
- Each category has its ID and name.
- Each order has its ID and order created date.
- Each order detail has its detail ID, order ID that contained, product title, quantity of products and total bill.

2. Database diagram explanation

- User entity has user_id as primary key because it is the most essensial information to idetify an user.
- Admin entity has user_id as primary and also foreign key from user entity because admin account like user ID with same detail information.
- Category has its own category_id to identify.
- Product has its own product_id to be searched easier. Product entity has its foreign key category_id to classify the products in one category to the others.
- Order has its own ID to identify.
- Order detail has its detail_id to identify. Order details has order_id as foreign key to look for the orders that contained in the detail.
- An admin can create one or many users and one user account can be created by one admin so the relationship of admin-user is M-1.
- An admin can create one or many categories and one category account can be created by one admin so the relationship of admin-category is M-1.

- An admin can create one or many products and one product account can be created by one admin so the relationship of admin-product is M-1.
- An user can take one or many order but an order can only be taken by that user so the relatonship of user-order is M-1.
- A category can contain one or many product inside it but a product is only in one category so the relatonship of category-product is M-1.
- An order detail can have one or many orders and an order can be listed in one order detail. Therefore, the relationship of order-order detail is 1-M.

IV. Site map

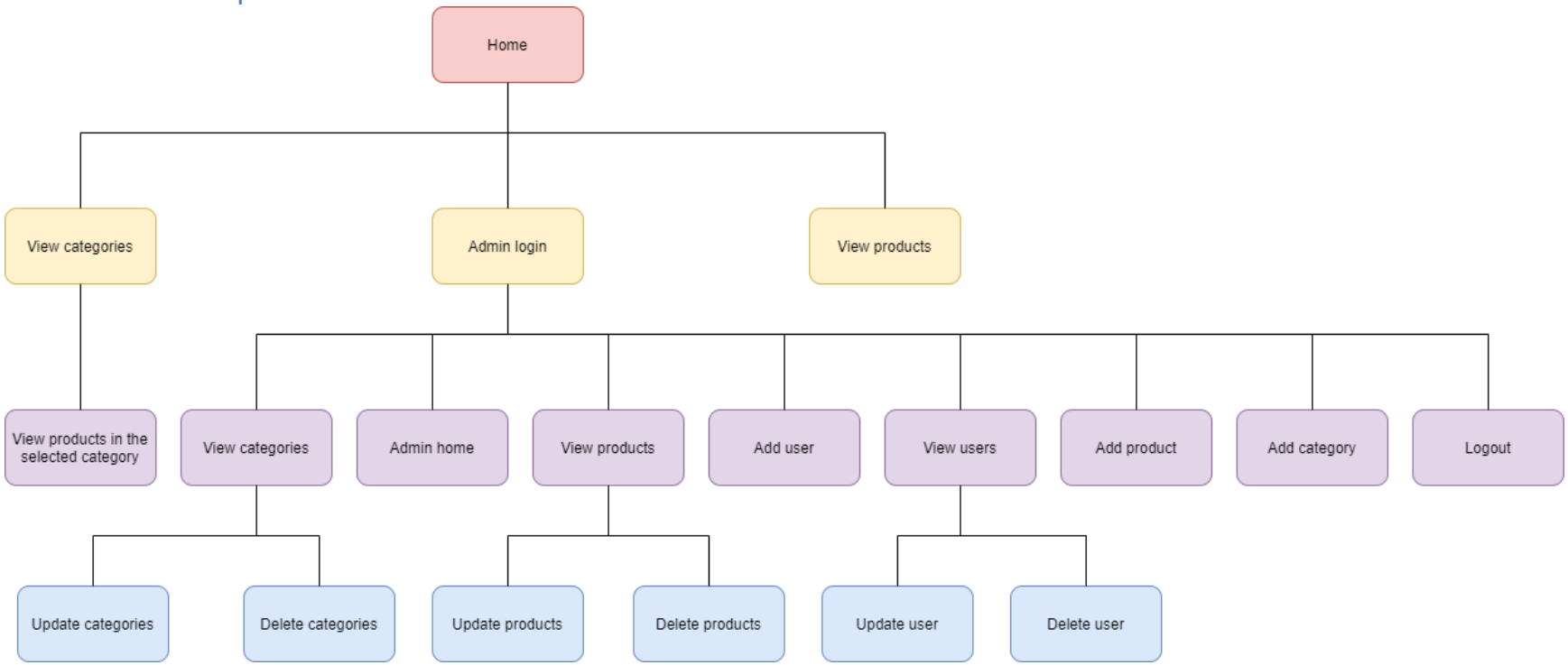


Figure 1. 3. Site map

The site map indicate the website operation of the system interface. When accessing the website, every users will start at the home interface for customers. In the home page, there are three options for user to select: “View categories”, “View products” and “Amin login”. If uer choose to “View categories” option, the website will move forward to the category list for customers that contains added categories. User can select any categories and they will illustrate the products int that category in the page “View products in the category”. Whether user choose “View products” option, they will be sent to the product list page with added products. Else if user choose the last options “Admin login”, they will be sent to the login form that force them to sign in to the “Admin home” page. In the “Admin home”, there are 8 options for users to select as admin authorities: “Admin home”,”Add category”,”View categories”,”Add product”,”View products”,”Add user”,”View users” and “Logout”.

If users choose “Add category” options, they will show the category adding form which contains 2 more options: “Update” and “Delete” that allow them to edit and delete categories . Like “Add category”, “Add product” and “Add user” option will also let user update and delete current products and users. The three options “Add user”,”Add category” and “Add product” allow user to add new user, product and category in their own page. “Logout” option allow user return to the customer “Home” page.

V. Wire frames

1. Customer home wireframe

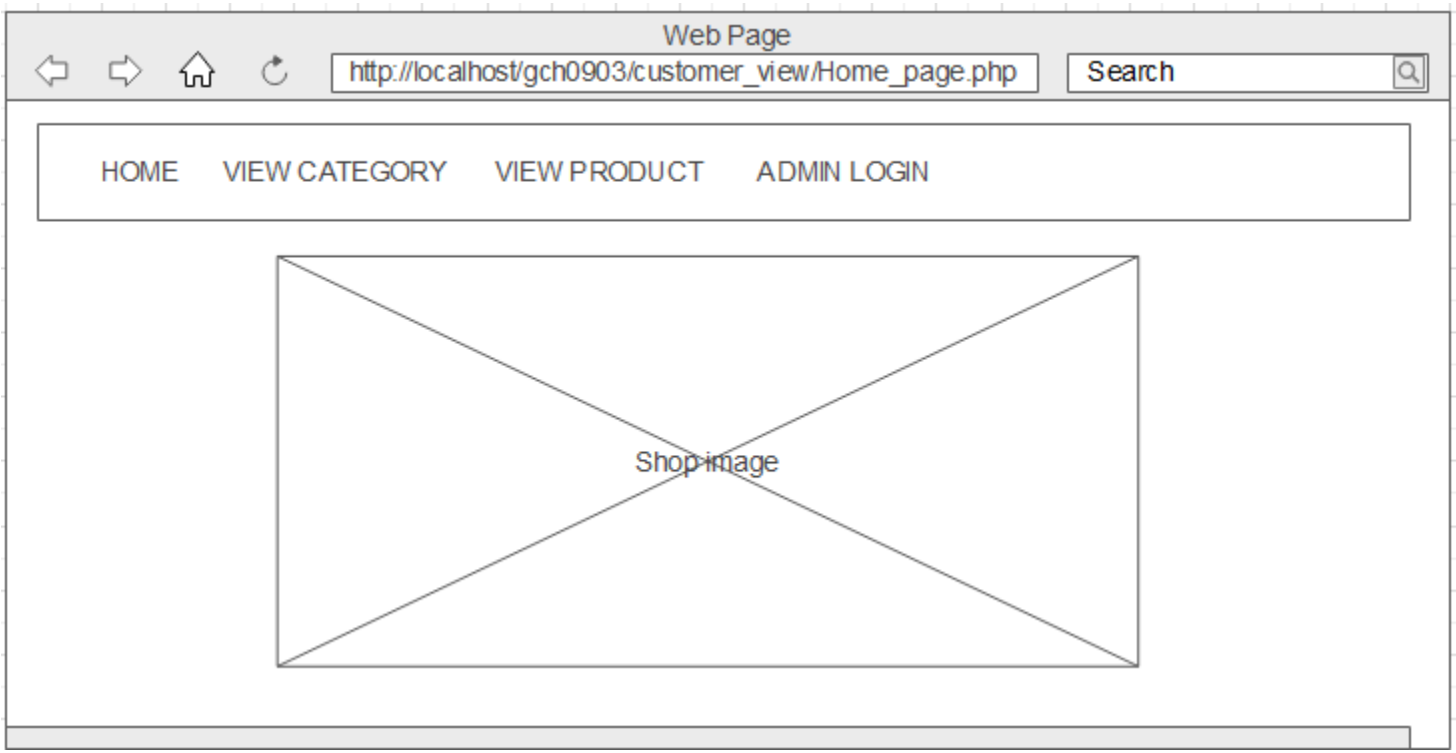


Figure 1. 4. Customer home wireframe

When all users accessing the web, the first page they meet is the home page for customers. It contains four options for customer to choose in the toolbars: HOME, VIEW CATEGORY, VIEW PRODUCT, ADMIN LOGIN.

2. Customer view products wireframe

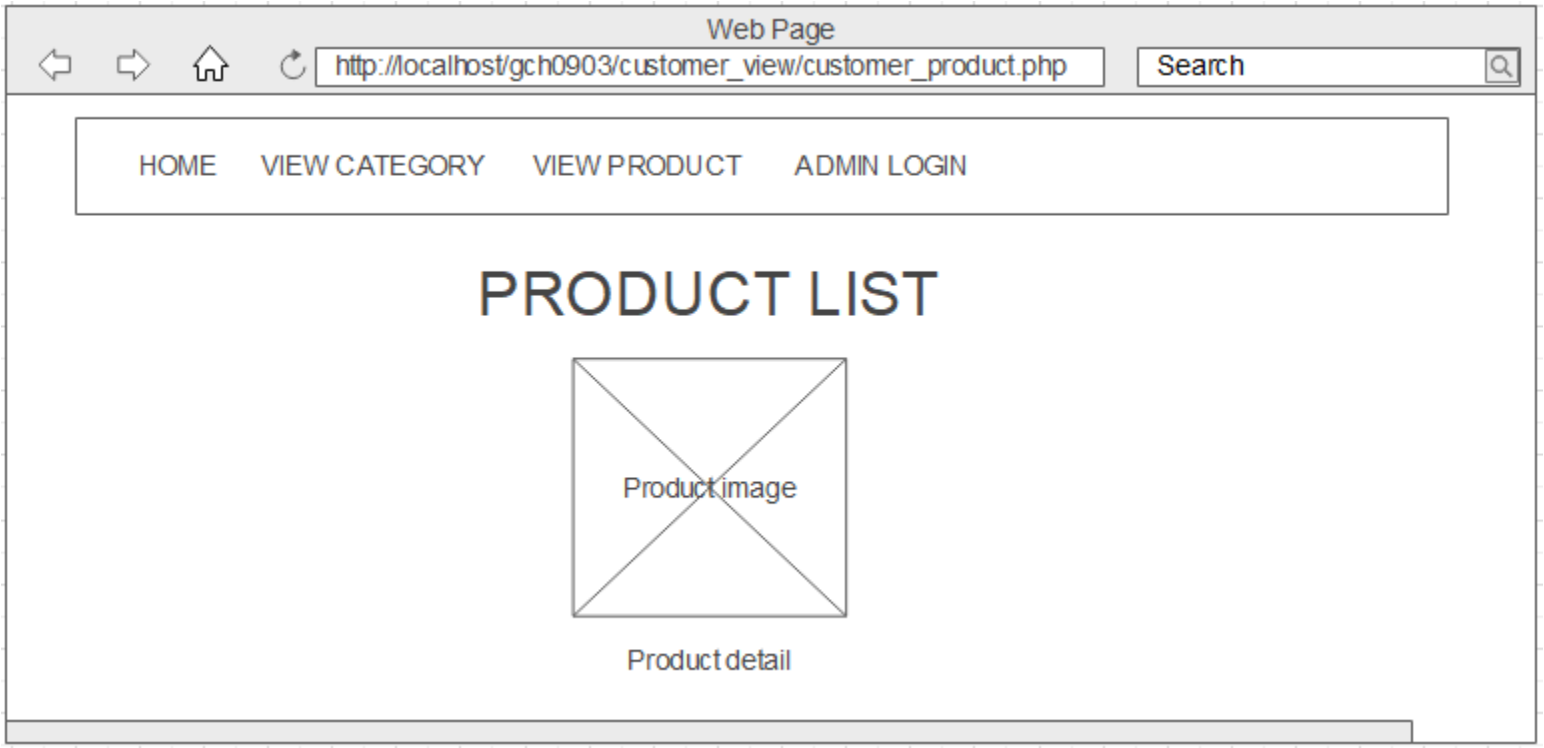


Figure 1. 5. Customer view products wireframe

If users choose VIEW PRODUCTS options, the website will turn into the product list page as above. The products will be shown as a row from up to down by scrolling the mouse or “PgDn” button.

3. Customer view categories wireframe

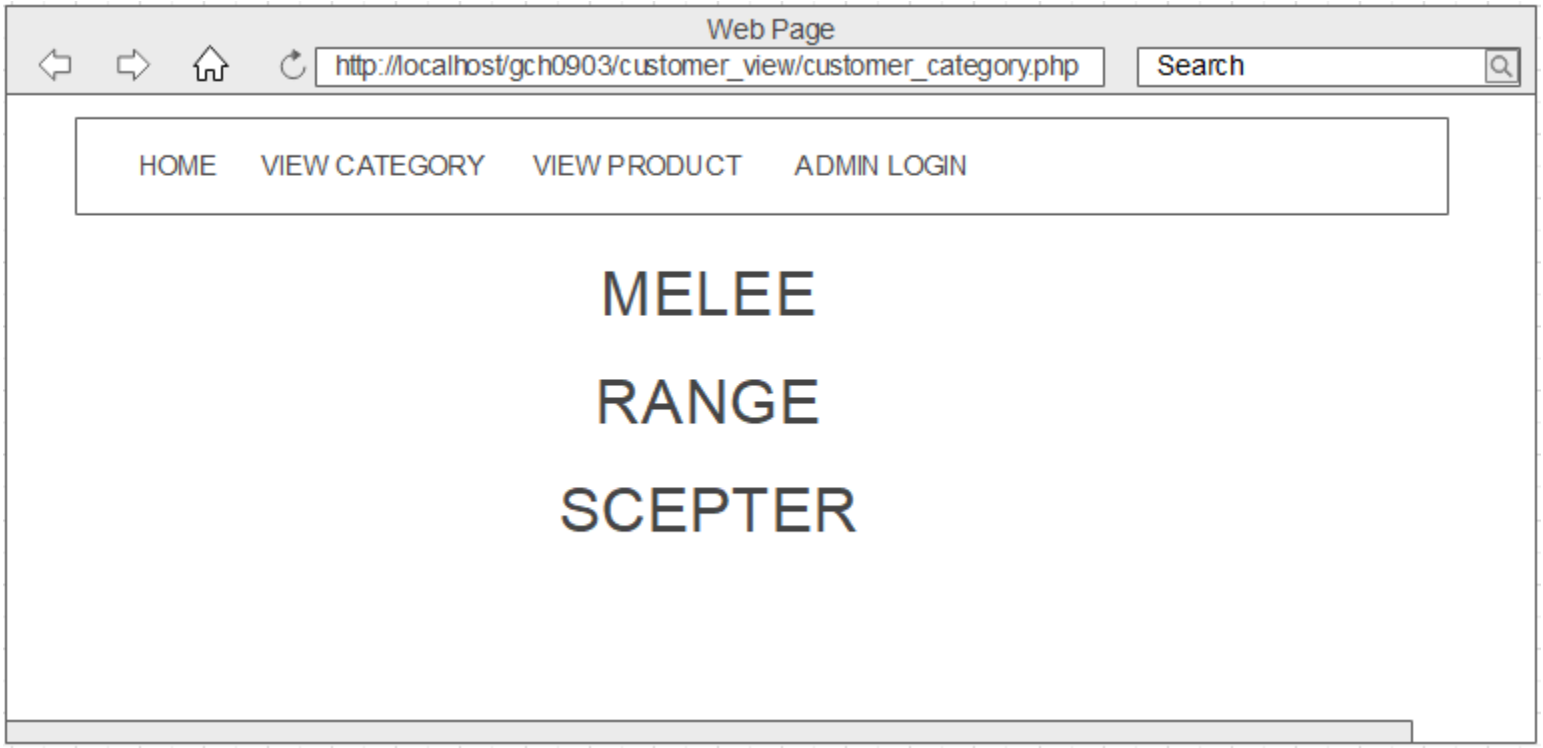


Figure 1. 6. Customer view category wireframe

If user choose to view categories, the website will print out the categories list. All categories are shown as option for customer to view products inside each category. If user click one category to view products, for example, “MELEE” then the page will move to the “MELEE” products page that contains all products of “MELEE” category as below.

4. Customer view products inside one category wireframe

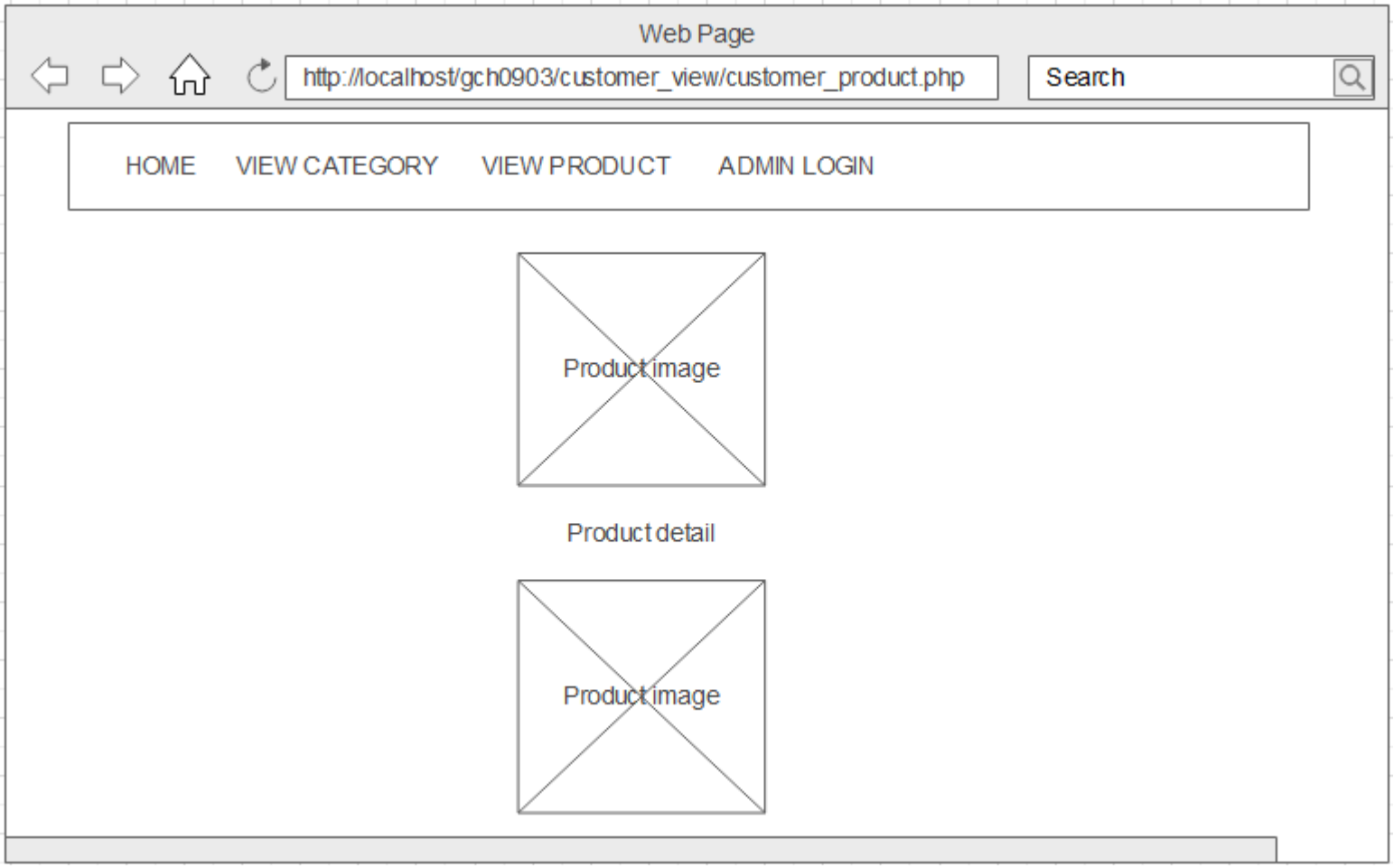


Figure 1. 7. Customer view products inside one category wireframe

The above illustrate the products inside one chosen category by user. The products in that category will be presented from up to down order.

5. User “ADMIN LOGIN” wireframe

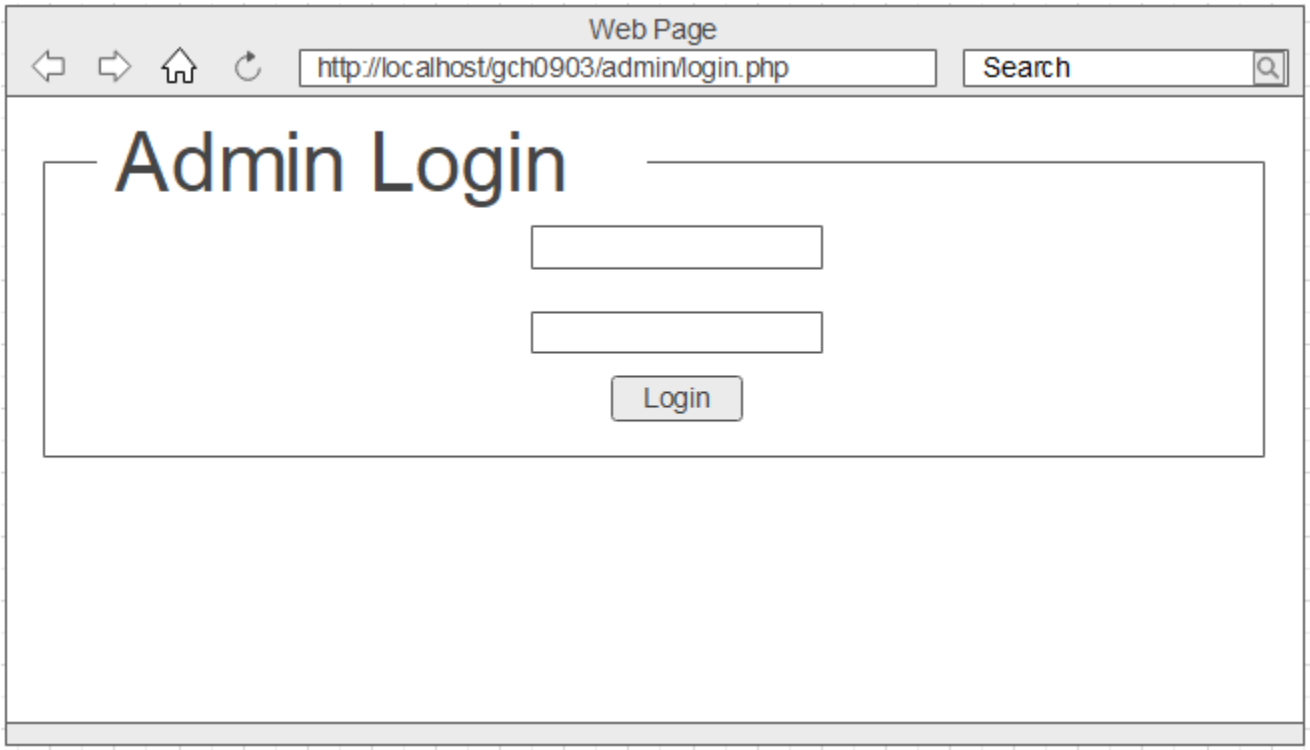


Figure 1. 8. Admin login page wireframe

Back to the Customer home page, if user choose the last option “ADMIN LOGIN”, the web will take user to the “Admin login” page with details. Users have to input information to login as Admin account.

6. Admin homepage wireframe

Figure 1. 9. Admin homepage wireframe

After logging in, user can view the Admin homepage which present and introduce the website. The toolbar show 8 option for admin: HOME as presented above, VIEW CATEGORIES, ADD CATEGORY, VIEW PRODUCTS, ADD PRODUCT, VIEW USER, ADD USER and LOG OUT.

7. Admin view category option

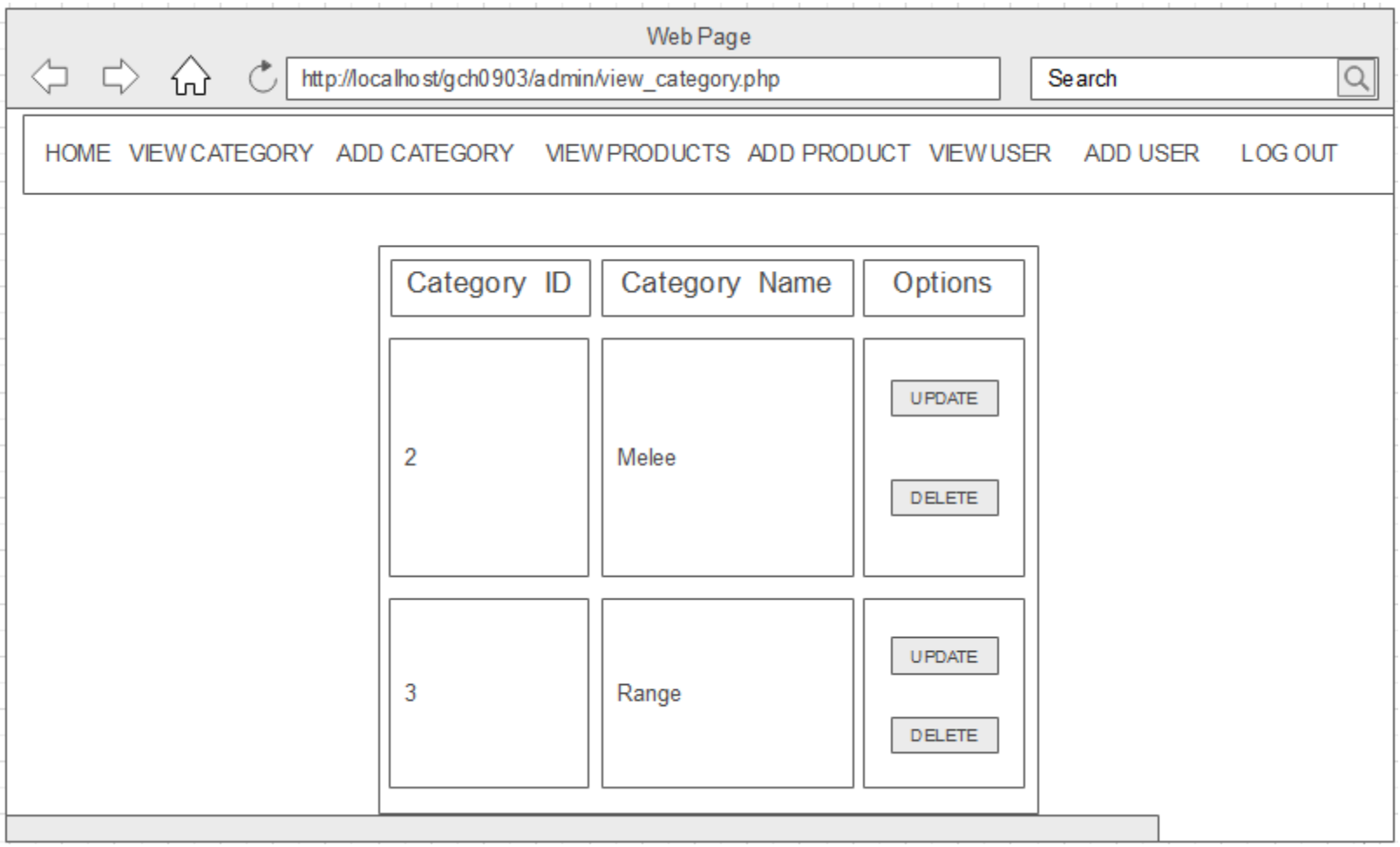


Figure 1. 10. Admin View category option

When selecting VIEW CATEGORY from Admin home toolbar, the website will present the category list for admin as table above. The admin also can update and delete category by selecting “UPDATE” or “DELETE” option.

8. Admin update category wireframe

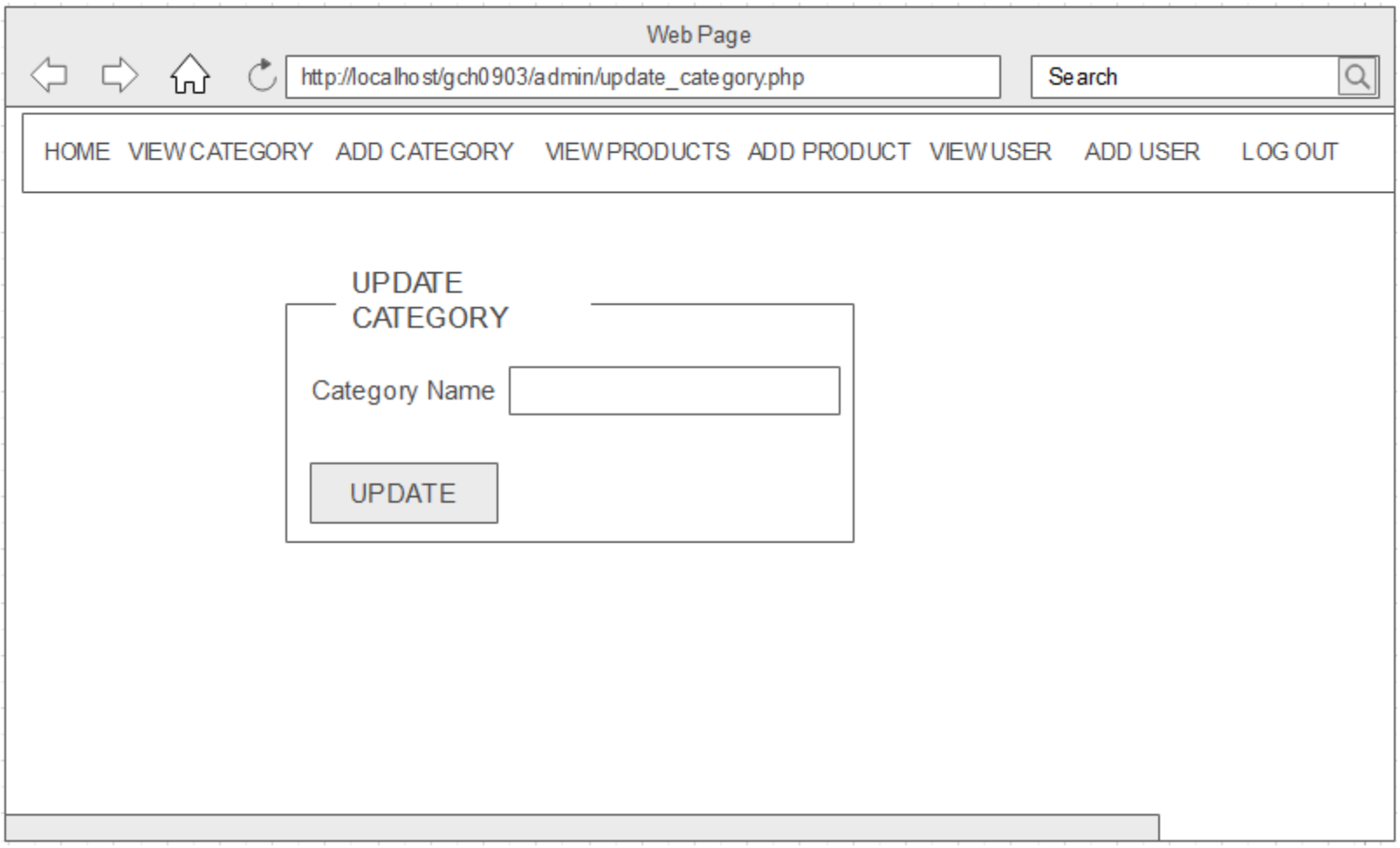


Figure 1. 11. Admin update category wireframe

If admin choose one category to update, the page will move to the Update category detail page for admin to update that category. With DELETE button, when user click on DELETE, the localhost bar will present options for user to confirm “DELETE” or not.

9. Admin add category wireframe

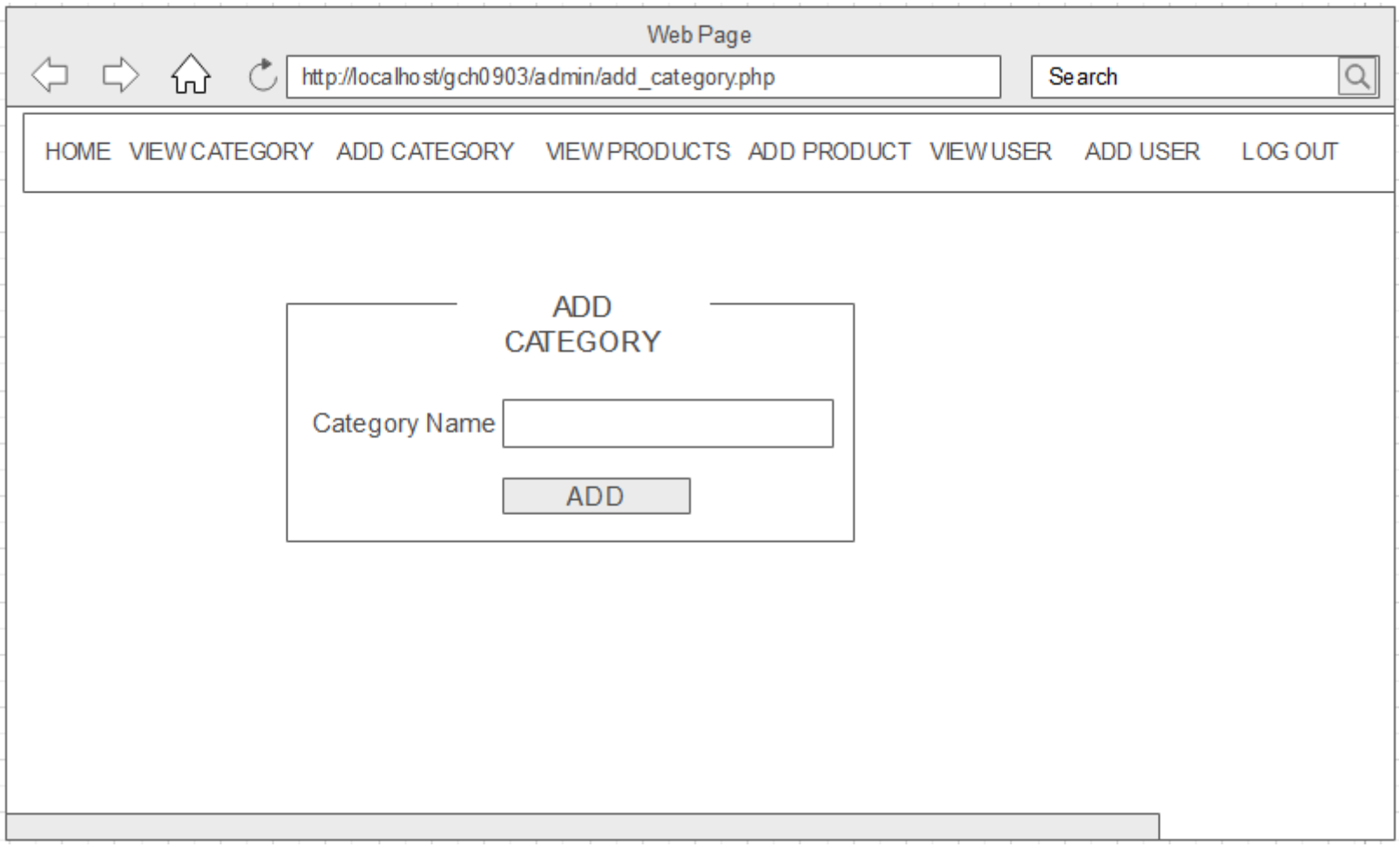


Figure 1. 12. Admin add category wireframe

When user choose ADD CATEGORY option, the website will move to the Add category webpage. The page contains detail for category name that user want to add by selecting “ADD” option.

10. Admin VIEW PRODUCT wireframe

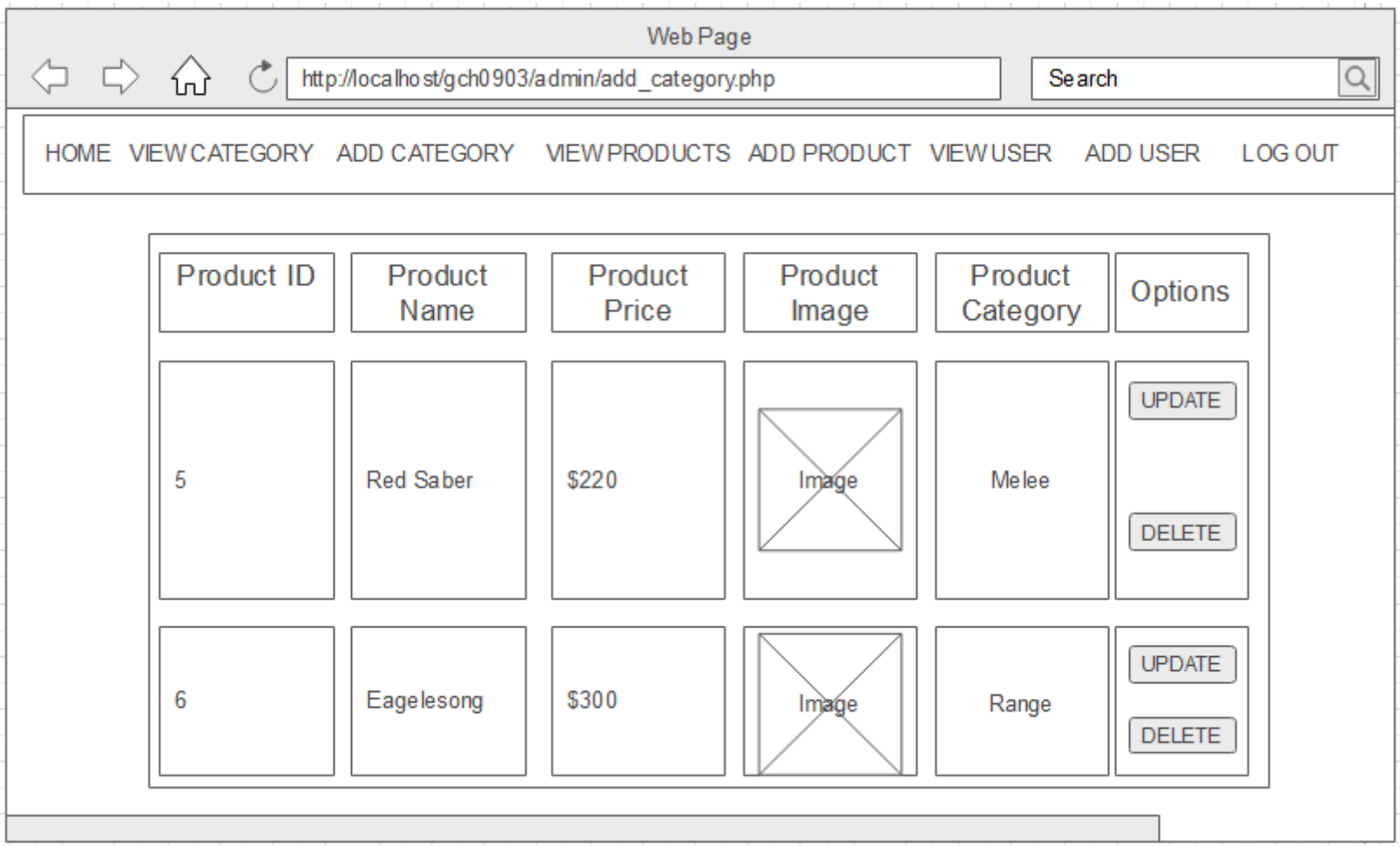


Figure 1. 13. Admin View product wireframe

If user choose to VIEW PRODUCT option, it will print out the the products list with its information. Admin can also have authorities to update and delete products by selecting UPDATE and DELETE button. If users choose DELETE button the website will ask them to confirm DELETE or not.

11. Admin Update product wireframe

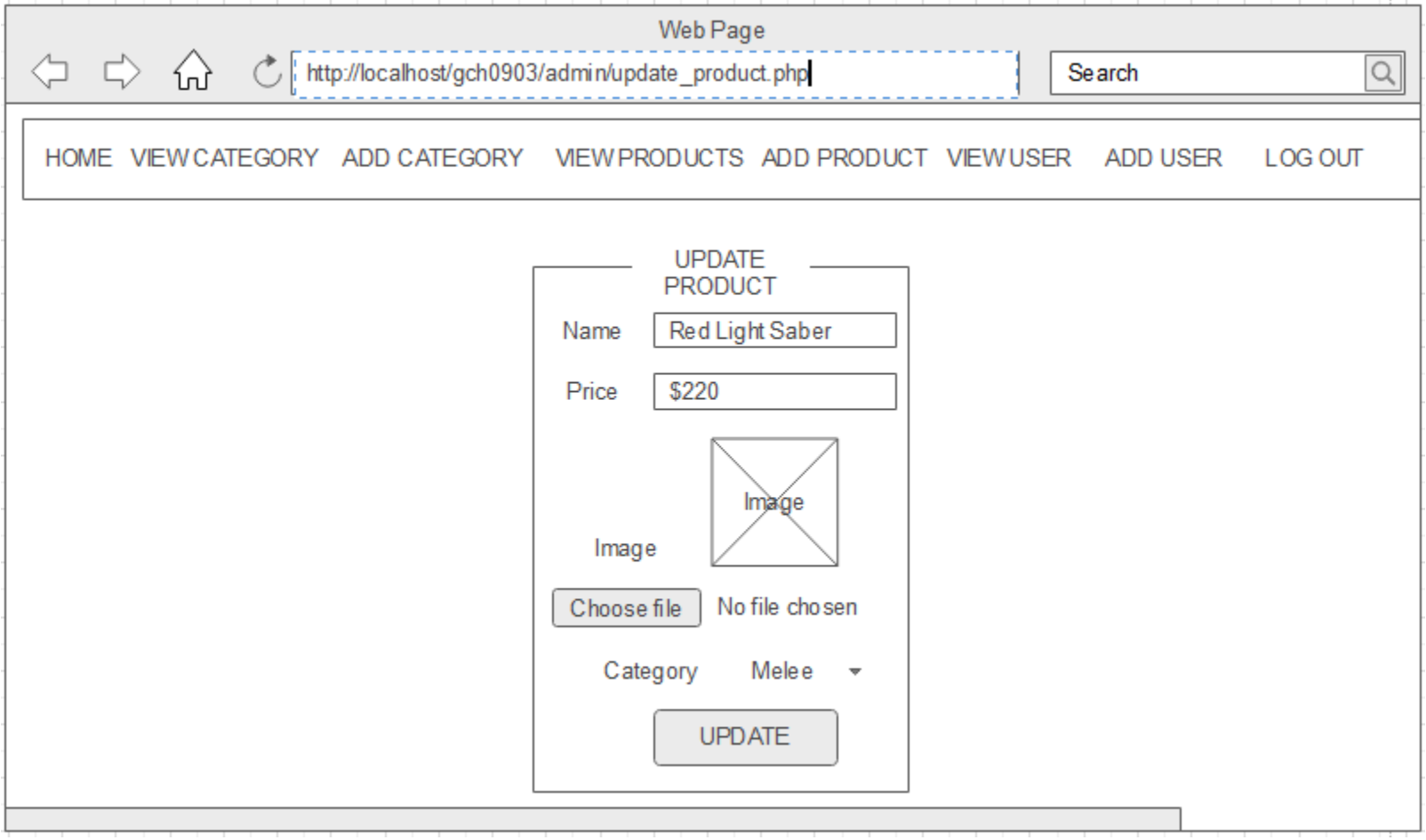


Figure 1. 14. Admin update product wireframe

From product list, if user want to update product they click the UPDATE button from the page above then the details will be print out for them to update the selected product.

12. Admin Add product wireframe

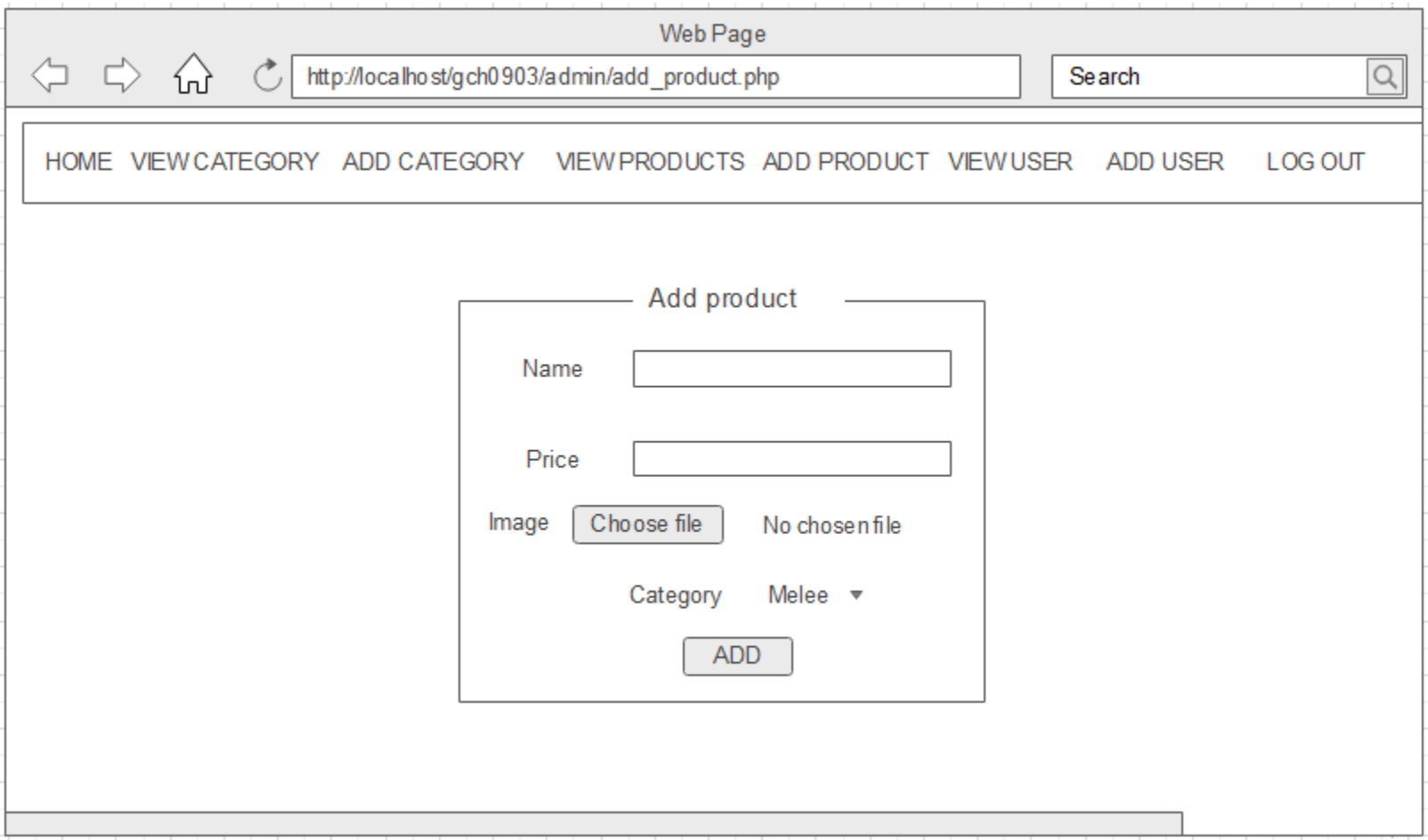


Figure 1. 15. Admin add product wireframe

Whether user choose to add product, the website will present the Add product detail for user to add information of a new product. After the new one is added, it will move to the product list page.

13. Admin view user wireframe

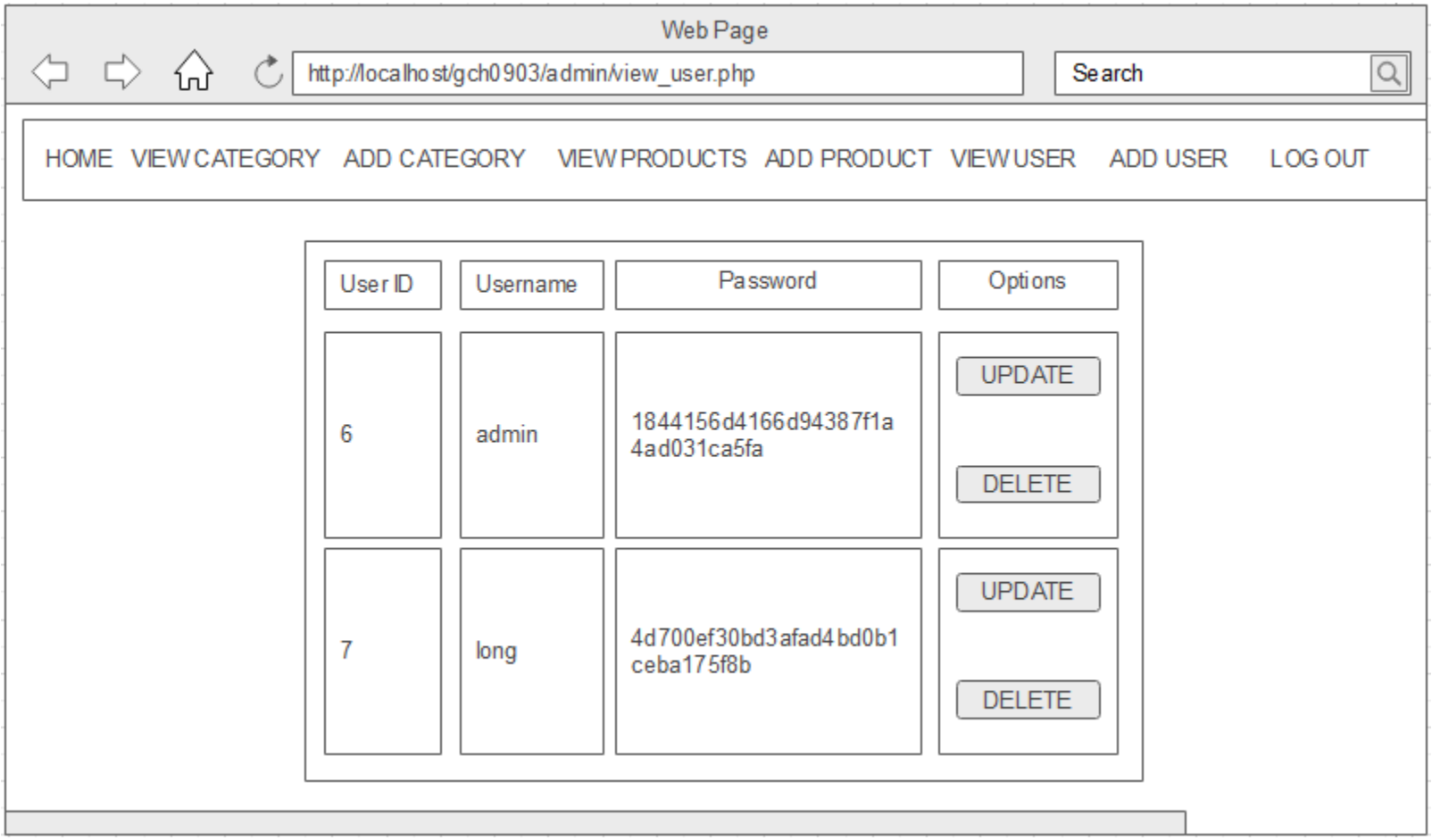


Figure 1. 16. Admin view user wireframe

If user choose to view user by clicking VIEW USERS option, the website will move to the user list page for admin to view. Admin can also update and delete current user by choosing UPDATE or DELETE button of each user in the table list.

14. Admin add user wireframe

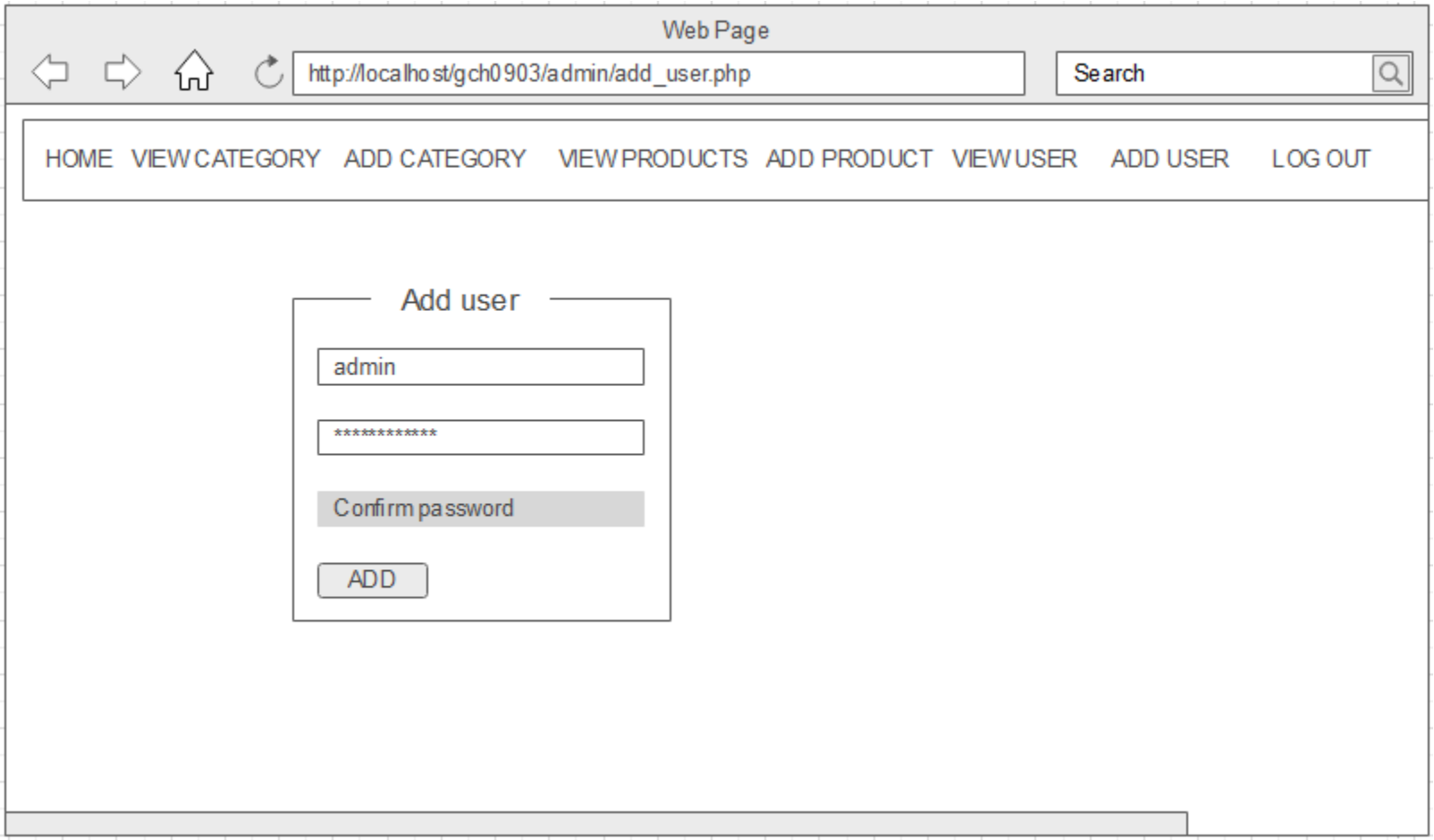


Figure 1. 17. Admin add user wireframe

If user choose to add user, the website will move to the Add user page for admin to input detail. After selecting ADD button, the localhost will show message that user is input or not. If input succeed, the website will move to the user list page.

Finally, if user selecting LOGOUT option, the website will move out to the customer home page.

C. Website implementation (P6)

VI. Web principles, standards and guidelines

1. Consistency principle

The consistency of a website's design is extremely important. Pay close attention to how the design features on - page align. It's obvious that the fonts, sizes, headings, subheadings, and button types must be consistent across the board. All should be planned ahead of time. Finish the fonts and colors for your texts, buttons, and other elements, and stick to them in the creation process. CSS (Cascading Style Sheets) will be useful for storing all of the details about design elements and styles (Pawar, 2018).

The webiste use the consistency principle that fonts, sizes, headings, sub-headings, and button styles are the same throughout the website and planned ahead of time. The home page, view products and categories use same heading and styles and sizes.

Figure 1. 18. Customer home webpage

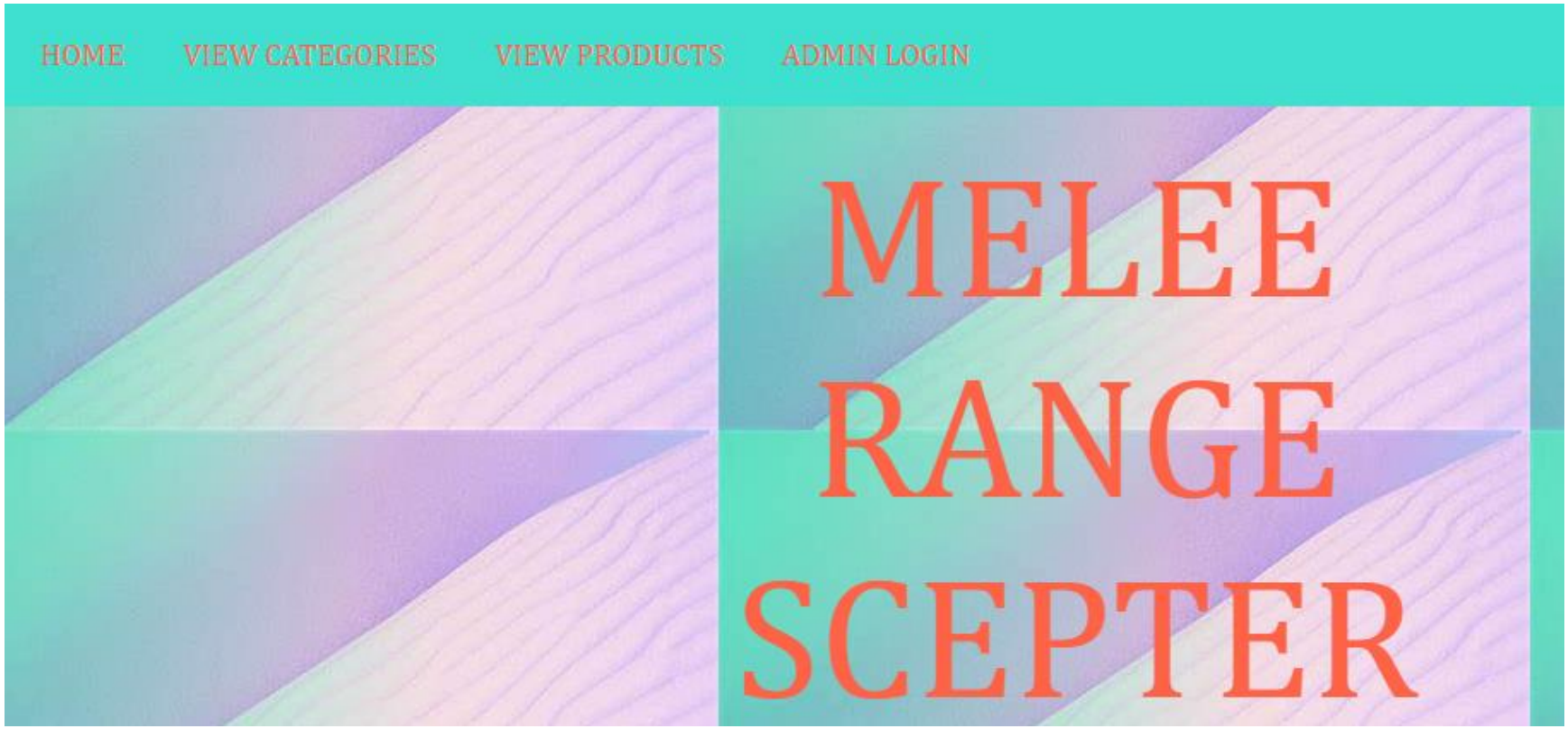


Figure 1. 19. Customer view category webpage

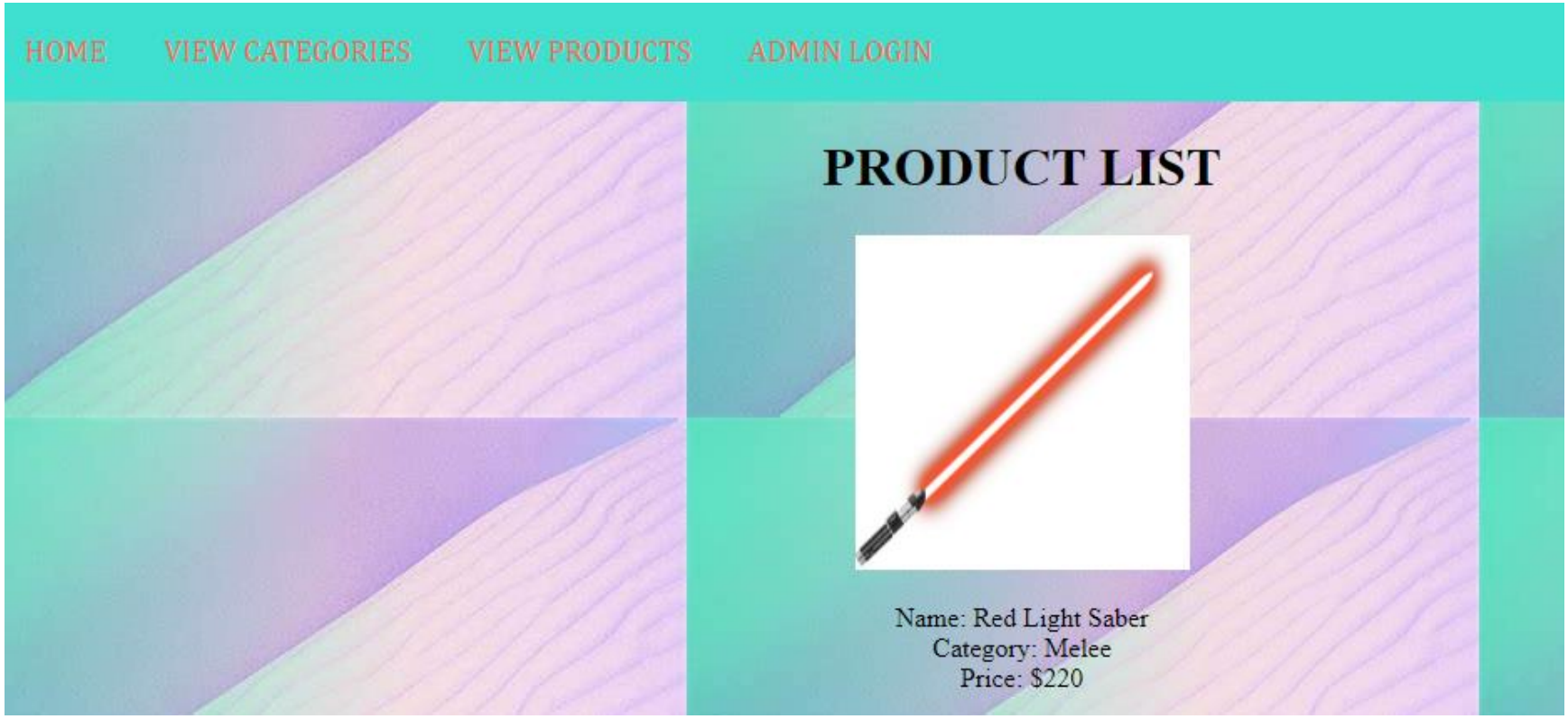


Figure 1. 20. Customer view products from selected category webpage

2. Easy navigation

Visitors spend more time on websites with simple navigation, according to research. Consider building a logical page hierarchy, using bread scrums, and designing clickable buttons for efficient navigation. The "three-click-rule" should be followed so that tourists can get the details they need in three clicks (Kawar, 2018).

The website use simple navigation that all in one toolbar and appear in every page for user and admin to search and choose easily.

Figure 1. 21. Customer toolbar

Figure 1. 22. Admin toolbar

3. Visual hierachy

The arrangement of elements in visual hierarchy is based on their significance. Scale, color, imagery, contrast, typography, whitespace, texture, and design are all used to achieve this. Establishing a focal point, which tells tourists where the most relevant information is, is one of the most important features of visual hierarchy (Mariane, 2021).

The website use visual hierachy by creating options for user to understand easily and take them to the page they need. For example, if they want to view products, the is a VIEW PRODUCTS in the toolbar and using style to make it clear to see as the focal point.

4. Content

Great design and content go hand in hand in a successful web design. Great content can attract and influence tourists, turning them into customers, by using persuasive words.

5. F-Shaped pattern reading

The F-based pattern is the most popular way for website visitors to search text. According to eye tracking research, the majority of what people see is on the top and left sides of the screen. In the West, our normal reading style is formed like an F. (left to right and top to bottom). A well-designed website would follow the normal pattern of a reader scanning the page (Marianne, 2021).

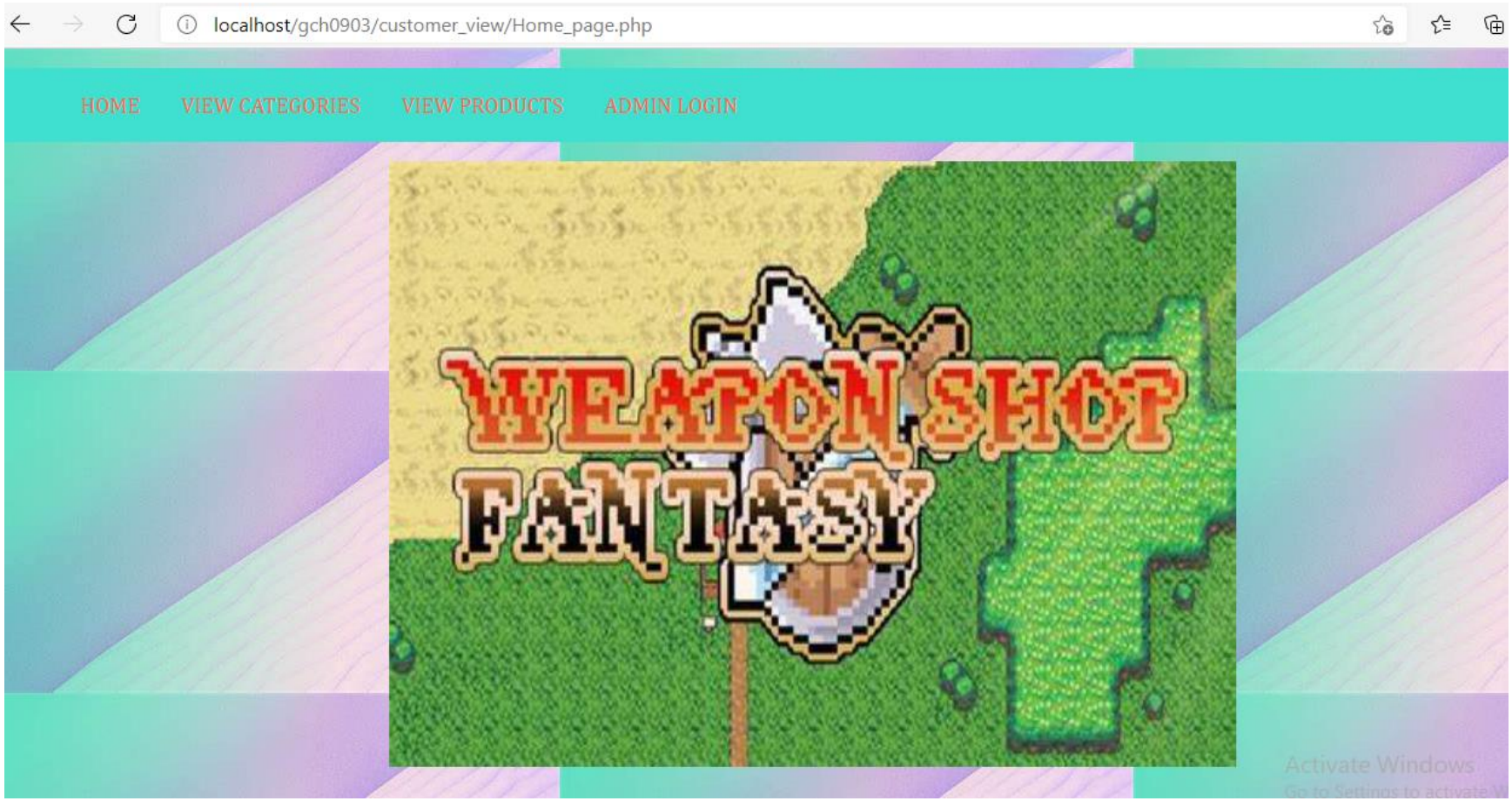
6. Grid based layout

Grids aid in the structure of your design and the organization of your content. The grid aids in the alignment of items on the website and helps to keep it tidy. The grid-based architecture organizes content into a neat, rigid grid structure with columns and sections that line up, feel balanced, and enforce order, resulting in a visually appealing website.

7. Load time

Visitors will leave if they have to wait for a website to load. Nearly half of web visitors expect a site to load in two seconds or less, and they can abandon a site that takes longer than three seconds to load. Increasing the size of your images will make your site load faster (Marianna, 2021).

The website use this principle by using a large-size image in homepage to make site load faster.



8. Mobile compatibility

With the increasing use of smartphones, laptops, and phablets, web design must be adaptable to different screen sizes. If your website design does not accommodate all screen sizes, you risk losing the fight to your rivals. There are many web design studios or

support points where you can get your desktop design converted into a responsive and adaptive design for all screen sizes (Kawar, 2018).

9. Color palette and imagery

A good color combination attracts users, while a bad color combination will confuse them. This necessitates selecting a color palette for your website that can create a pleasant environment and thereby make a positive impression on visitors. To give your website design a balanced look, choose a complementary color palette to enhance users' experience. Remember to use white spaces to avoid visual clutter and mess on your website. Often, stay away from using too many shades. For an attractive and simple style, 3 or 4 tones for the entire website are sufficient. The same can be said for photographs that do not use a lot of colorful pictures (Kawar, 2018).

The website use sufficient and simple colors to make it clear to see and contains only one picure for the home page.

10. Communication

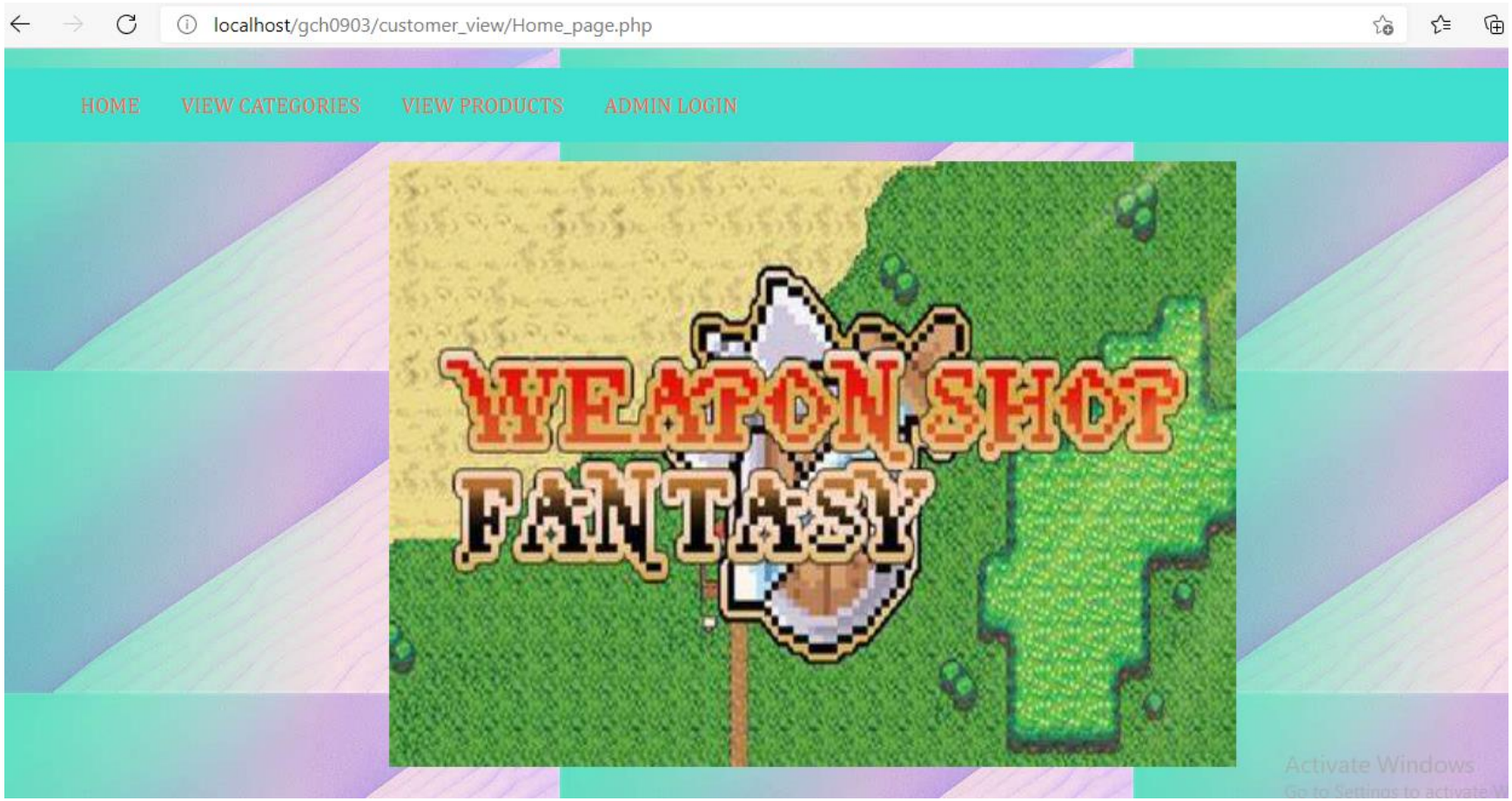
The primary goal of visitors is to obtain information, and if your website is able to effectively interact with them, they will most likely spend more time on it. Organizing information by making good use of headlines and sub-headlines, cutting the waffle, and using bullet points are all tricks that can work to create easy contact with visitors.

Figure 1. 23. Admin homepage

VII. Functional screen shot of website

1. Customer's site

a. Customer homepage



To create customer homepage, it requires function of Home page and also Index customer of the website.

- Customer index

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Web administration</title>
    <link rel="stylesheet" href="../css/style_admin.css">
    <style>
    body {
        background-image: url("https://th.bing.com/th/id/OIP.BnCXNWKbnhY2_A_bBMMjBgHaEK?pid=ImgD
    }
</style>
</head>
<body>
    <nav>
        <ul>
            <li><a href="../customer_view/Home_page.php">Home</a></li>
            <li><a href="../customer_view/customer_category.php">View categories</a></li>
            <li><a href="../customer_view/customer_product.php">View products</a></li>
            <li><a href="../admin/login.php">Admin login</a></li>
        </ul>
    </nav>
</body>
</html>
```

The index for customer contains all front-end functions of the customer webpage with all options in the home toolbars and the transport web for each of them when selected.

The <head> content of the header show the title of the website, inherit the style from the css webpage and insert background image for the homepage website. The navigation toolbar <nav> indicate the options shown in the homepage toolbar with the transport link of each webpage for each option. For example, pressing the Admin login options and the function of the website will move to the created transport website by card <a href="../admin /login.php">

- Home

```php
<?php
require_once "index.php";
?>
<center>
<img src="https://th.bing.com/th/id/OIP.im2S1kqoYP62m1ZLNI3C7AHaEP?pid=ImgDet&rs=1"
alt="" width="700" height="500">
</center>
```

This is the function of the home page body. The order "require_once" make the homepage use to check that function are created or not and also use the function of the customer index. In the body of customer homepage website, an image is inserted in the center of the page with edit width and height using front-end function.

b. Customer view products

From the homepage, if user choose option VIEW PRODUCTS, it will move to the view product page for customer.

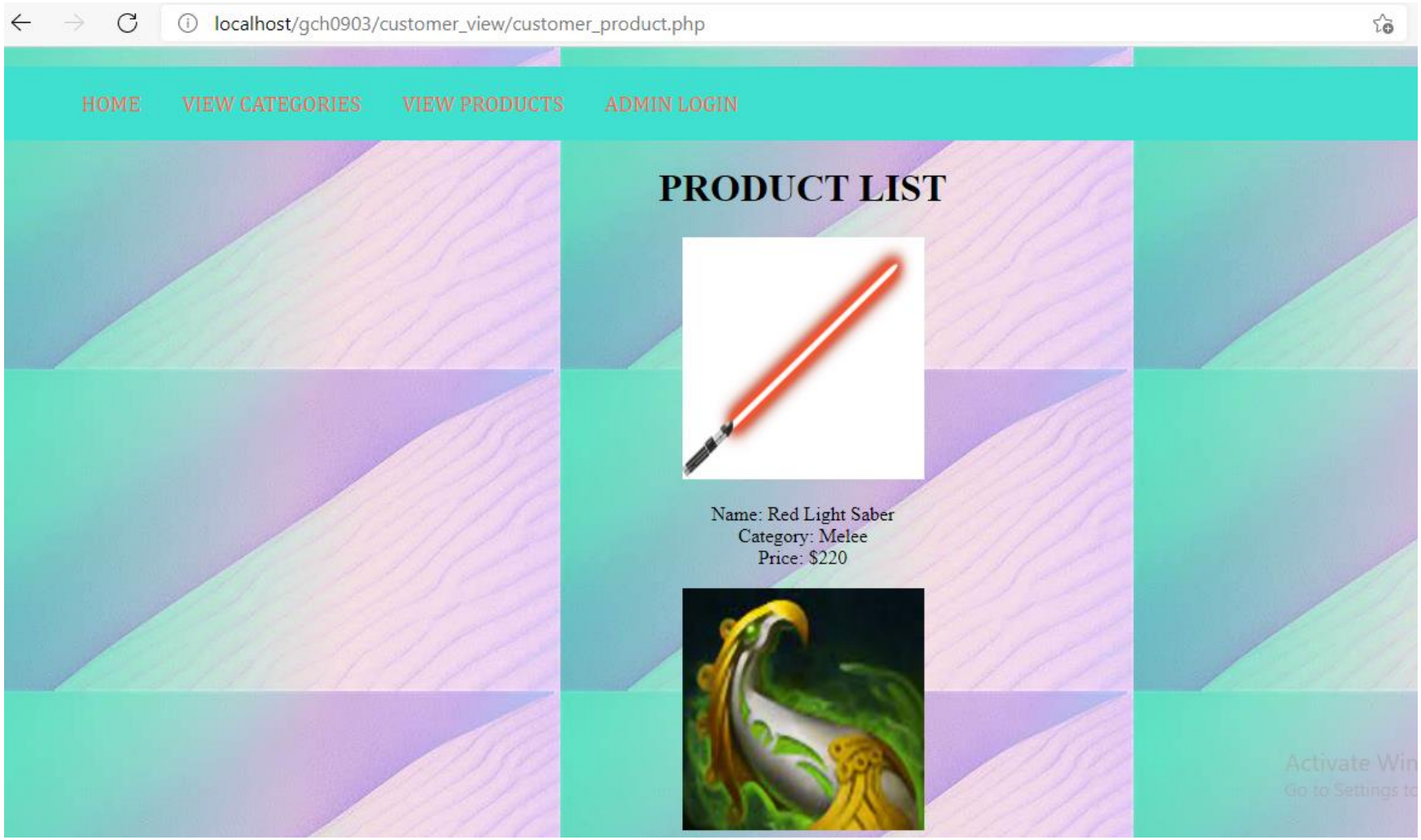


Figure 1. 24. Customer view product page

The view products page was created by its function.

- Product index for customer

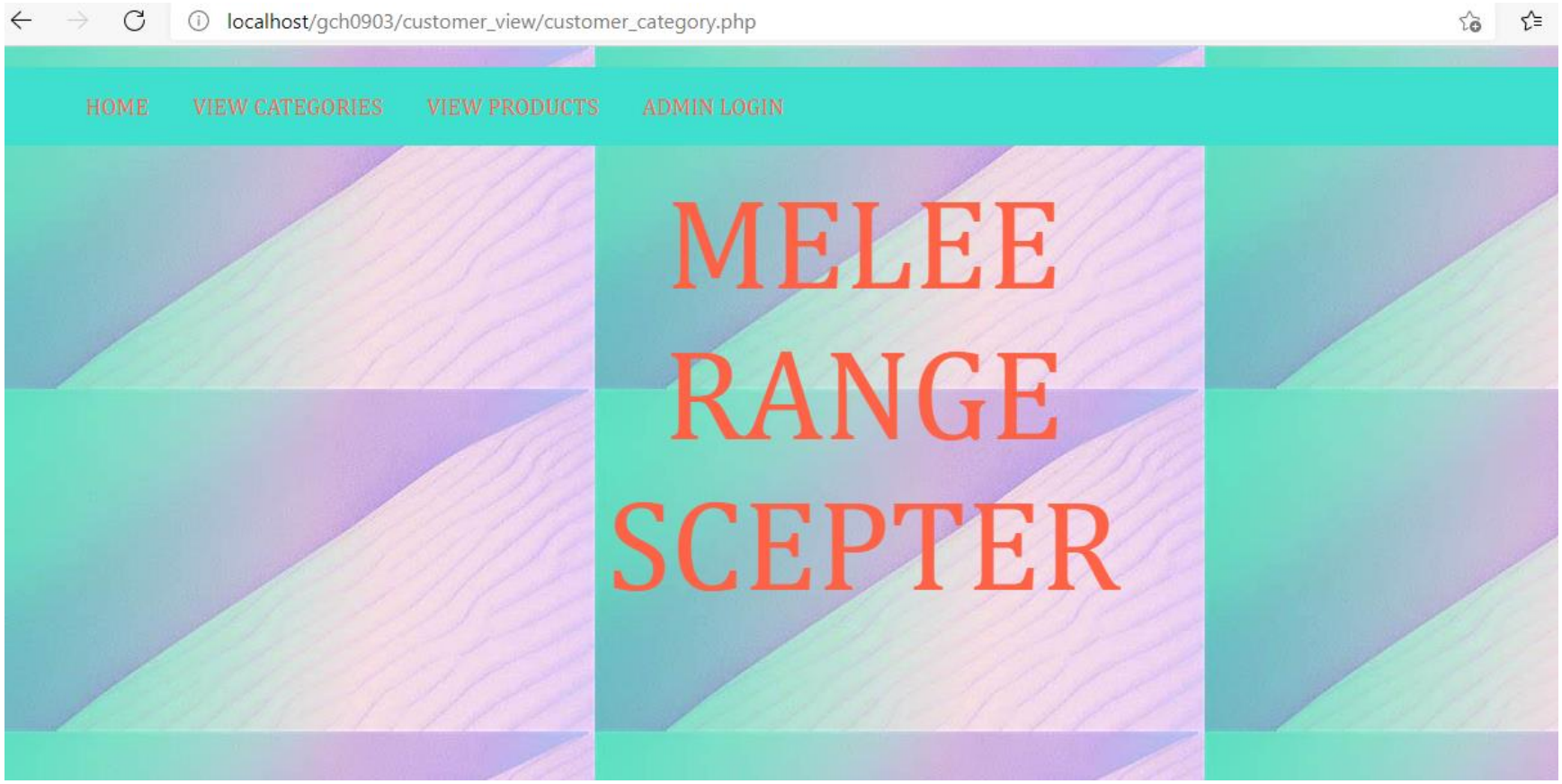
```php
<?php
require_once "index.php";
?>
<center>
        <tr>
            <th>Product List</th>
        </tr>
        <?php
        $sql = "SELECT * FROM  product";
        $run = query($sql);
        while ($product = mysqli_fetch_array($run)) {
            ?>
        <tr>
        <td>
                <a href="product_detail.php?ProductID=<?= $product[0] ?>">
                <img src="images\<?= $product[3] ?>" width="150" height="150">
                </a>
            </td>
            <td><?= $product[1] ?> </td>
            <td><?= $product[2] ?> </td>
            <td> <?= $cls[0] ?> </td>
        </tr>
        <?php
        } ?>
</center>
```

The product index inherit the toolbars from the customer home by function “require_once ‘index.php,” as the backend. The whole products are present in the center body with title “Product List” before starting query. In the backend query, to print out all the current products the website will select all inserted products from the product database then run and start the while loop. In the while loop, $product present for all information of the product entity then connect with the front end structures. The function < a href= “product_detail.php?ProductID=<?=$product[0]?>”> select data from the product database of product_detail file that set them before (product_detail get product information database as below).

```php
<?php
require_once "functions.php";
$id = $_GET['ProductID'];
$sql = "SELECT * FROM product WHERE product_id = '$id'";
$run = query($sql);
$std = mysqli_fetch_array($run);
?>
```

Next, the function <img src="images\<?= $product[3] ?>" width="150" height="150"> select images from product image entity and edit then side. The order product[1], product[2] present the order of the product information from database.

c. Customer view categories



When user choose to VIEW CATEGORIES option from toolbar, the categories list will be shown as also an options for user to view the products inside one category and others. Therefore, if user click one category, for example, “MELEE” category then the website will show the products in category “MELEE”.

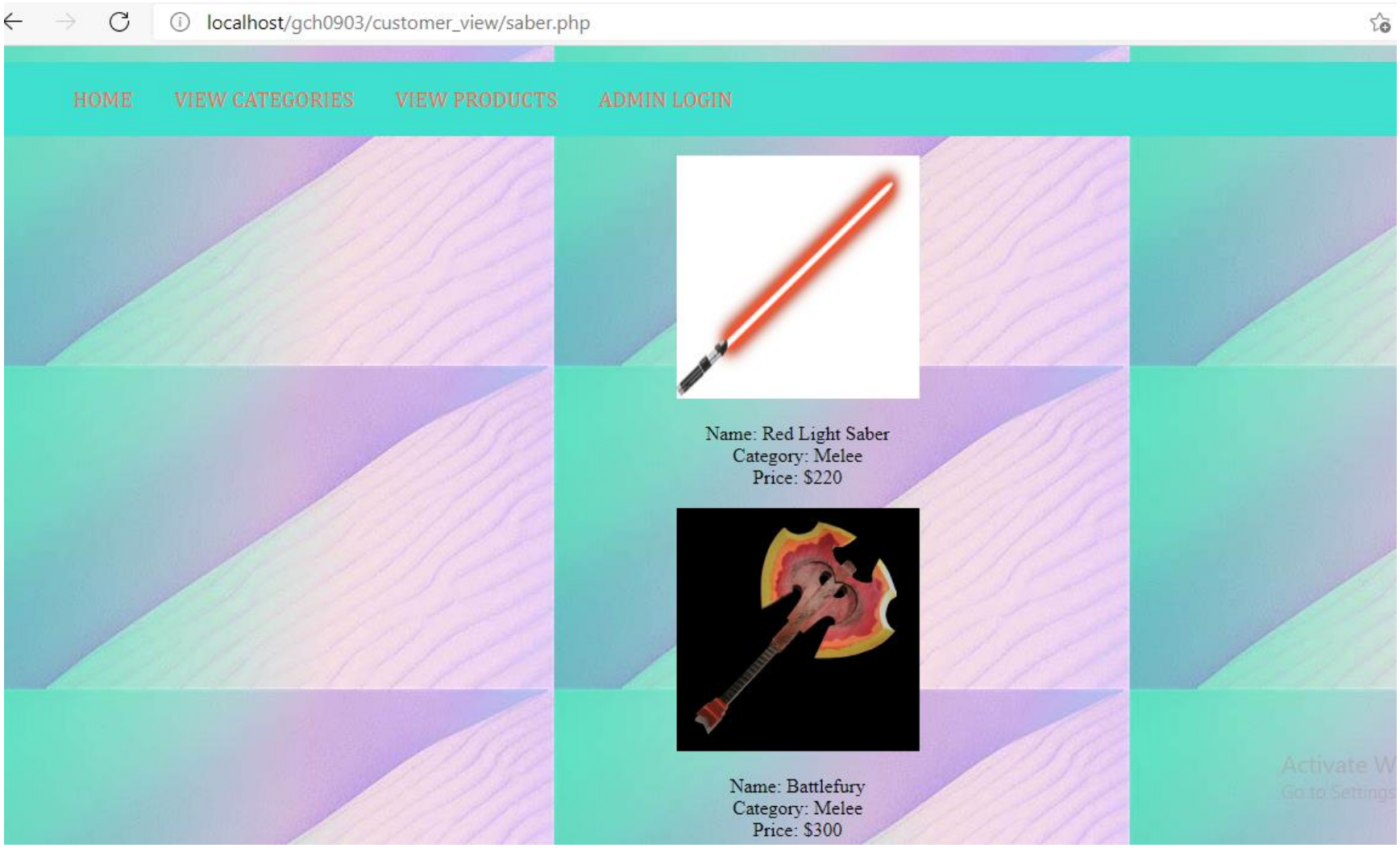


Figure 1. 25. Customer view category in "MELEE" category webpage

For their function code, starting with create category list the website will select the category name from the category entity then run the query as the same as products query using while loop to print the inserted category.

- Customer view category function

```php
<?php
require_once "index.php";

$sql = "SELECT category_name FROM category";
$run = query($sql);
?>
<center>
<table border="1">
    <?php
    while ($category = mysqli_fetch_array($run)) {
        ?>
    <tr>
        <td><a href="../customer_view/saber.php"><?= $category[0] ?></td>
    </tr>
    <?php
    }
    ?>
</table>
</center>
```

Then to print out the products from that category, website need one query to print them out in the saber_php function code that the category option bring them to.

- Customer view products in one category function:

```php
<?php
require_once "index.php";
?>
<center>
        <?php
        $sql = "SELECT * FROM  product";
        $run = query($sql);
        while ($product = mysqli_fetch_array($run)) {
            ?>
        <tr>
        </td>
            <?php
                $sql1 = "SELECT category_name FROM category WHERE category_id = '$product[4]'";
                $run1 = query($sql1);
                $cls = mysqli_fetch_array($run1);
                ?>
            <td> <?= $cls[0] ?> </td>
        <td>
                <a href="product_detail.php?ProductID=<?= $product[0] ?>">
                <img src="images\<?= $product[3] ?>" width="150" height="150">
                </a>
            </td>
```

```
            <td><?= $product[1] ?> </td>
            <td><?= $product[2] ?> </td>
            <td> <?= $cls[0] ?> </td>
        </tr>
        <?php
        } ?>
</center>
```

The web function need a query to print the products to the category list then it start to refine them to selected category that contains. Therefore, it refine products through the product_category entity (foreign key reference from category entity) then run the second query to print out all the products in that category and its detail. After being refined, the products and their details in the selected category will be print out.

d. Admin login

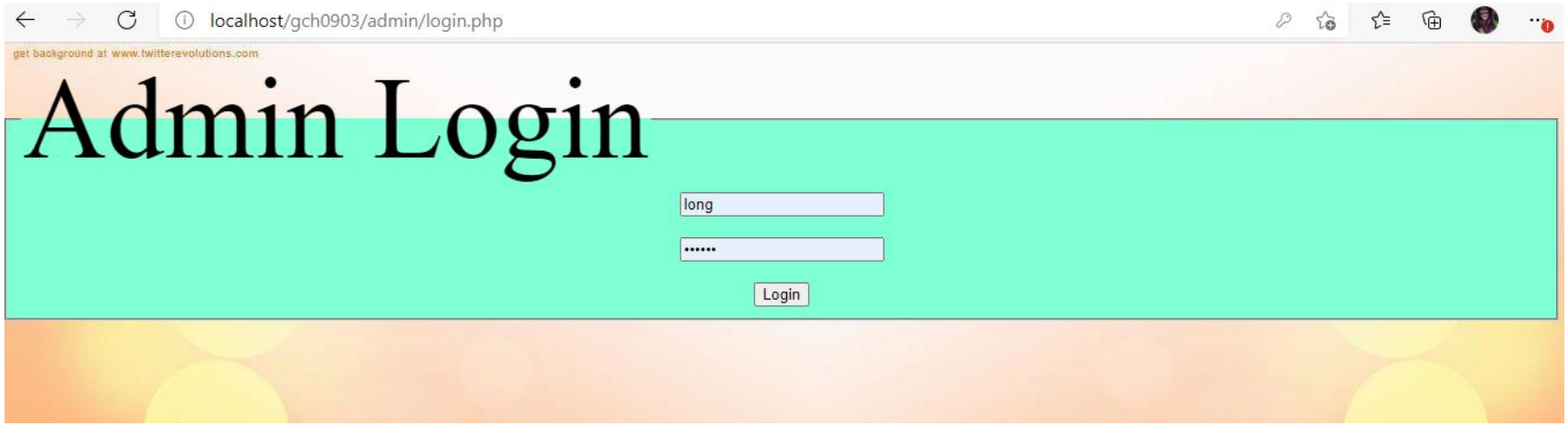


Figure 1. 26. Admin login webpage

When user choose AMIN LOGIN option in the customer home toolbars, it will move directly to the admin login interface. The interface has its own front-end and back-end function.

- Login function code

```php
<?php
session_start();
require_once "db_connect.php";

if (isset($_POST['login'])) {

$username = $_POST['username'];
$password = $_POST['password'];
$pass = md5($password);

$sql = "SELECT * FROM user WHERE user_name = '$username' AND password = '$pass'";
$run = $connection->query($sql);
$check = mysqli_fetch_array($run);
if (is_array($check)) {
    $_SESSION['username'] = $username;
    $_SESSION['password'] = $pass;
    ?>
    <script>
        alert("Login succeed !");
        window.location.href = "home.php";
    </script>
```

```
<?php } else { ?>
    <script>
        alert("Login failed !");
        window.location.href = "";
    </script>
<?php }
}
else {

?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <link rel="stylesheet" href="style_admin.css">
    <style>
    body {
        background-image: url("https://th.bing.com/th/id/Rd0ea4edd1209491510bf5233012e220f?rik=3
    }
</style>
```

```
</head>
<body>
<center>
    <form action="" method="post">
        <fieldset style="background-color:aquamarine">
            <legend style="font-size: 100px;">Admin Login</legend>
            <input type="text" name="username" placeholder="Enter username here"
            id="" required>
            <br> <br>
            <input type="password" name="password" id=""
            placeholder="Enter password here" required>
            <br> <br>
            <input type="submit" value="Login" name="login">
        </fieldset>
    </form>
</center>
</body>
</html>

<?php
}
?>
```

First of all, the function run query session_start () to run the webpage. To do so it need function from the db_connect file which contains the web localhost accessment to connect to the website network.

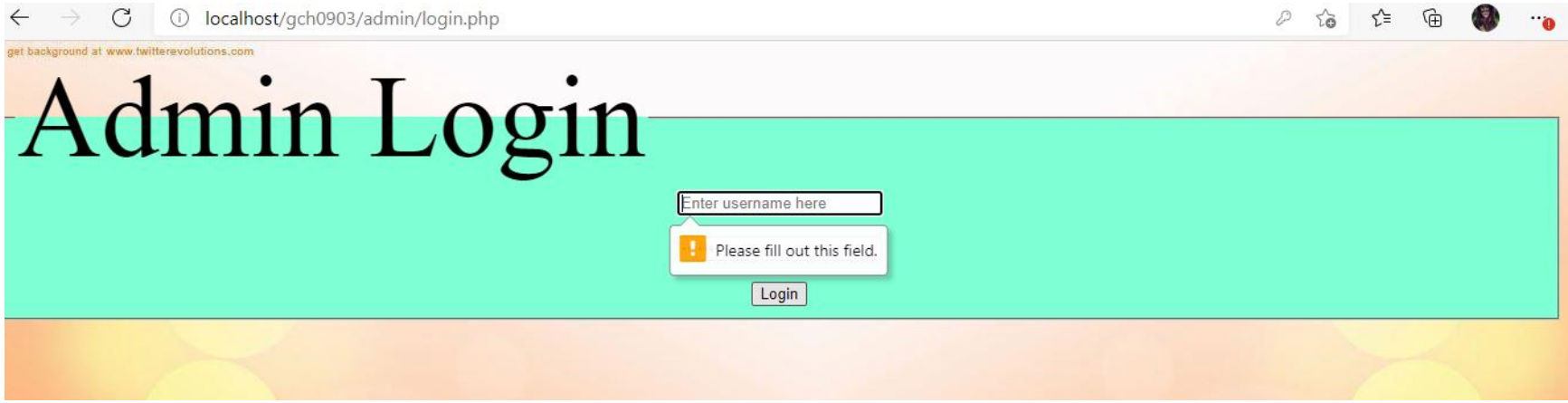
- DB connect function code

```php
<?php

$host_name = "localhost";
$sql_username = "root";
$sql_password = "root";
$db_name = "gch0903";
$db_port = "3306";

$connection = new mysqli($host_name,$sql_username,$sql_password,$db_name,$db_port);

?>
```

After the connection is activated, it starts the conditional statement if to set the user detail to input $username, $password (password is also set md5 mode with new card $pass). Then it will run the query to select the information from user database to print them out before put in the check statement.

The check statement if(is_array($check)) will make sure that the input information match the ones in inserted database or not. If they match, the function code print out the notification that “Login succeed” using <script> type and so does if “Login failed” status in the conditional check statement. Else if “Login succeed” they will move to the “home.php” page – page for admin. The next front end function present the interface of Admin login details with inserted image.

The function code also check for empty input in the blank detail, if user do not type in then the website will print out notification for user to fill in.



The center body present the field for detail and using style from “style_admin.css” file. The <form> and <fieldset> create the content and interface of the detail field using css style. In the field, three <input> functions present the form for user to type information and submit.

2. Admin’s site

a. Admin home page

The admin home webpage was created using to php function files: header and home for admin.

- Header code

```php
<?php
require_once "functions.php";

?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Web administration</title>
    <link rel="stylesheet" href="../css/style_admin.css">
    <style>
    body {
        background-image: url("https://th.bing.com/th/id/Rd0ea4edd1209491510bf5233012e220f?rik=3
    }
</style>
</head>
```

```
<body>
    <nav>
        <ul>
            <li><a href="../admin/home.php">Home</a></li>
            <li><a href="../admin/view_category.php">View categories</a></li>
            <li><a href="../admin/add_category.php">Add category</a></li>
            <li><a href="../admin/view_product.php">View products</a></li>
            <li><a href="../admin/add_product.php">Add product</a></li>
            <li><a href="../admin/view_user.php">View user</a></li>
            <li><a href="../admin/add_user.php">Add user</a></li>
            <li><a href="../customer_view/Home_page.php">Logout</a></li>
        </ul>
    </nav>
</body>
</html>
```

To set connection, the header need query from the funtion file using require_once”functions.php”.

- Functions code

```php
<?php
require_once "db_connect.php";
require_once "forbidden.php";

function query ($sql) {
    global $connection;
    return $connection->query($sql);
}


function encrypt ($pass) {
    return md5($pass);
}
```

The function also require connection from db_connect to run the query to set connection to global before running. The second function is used to set the password as md5 form for the admin login password.

The rest of the header webpage is the front end using “style_admin.css” style with all options printed in the toolbars and set background image. The toolbar using navigation front end type to intput the option with link php file for the transport page of each option.

- Home code

```php
<?php
require_once "header.php";
session_start();
if (isset($_SESSION['username']) && isset($_SESSION['password'])) { ?>
    <center>
        <a style="font-size:xx-large">Weapon Store</a>
        <br><br>
        <h1>This web allow people to buy weapons and use them for good purpose!</h1>
        <h2>
        </h2>
        <img src="https://th.bing.com/th/id/OIP.im2S1kqoYP62m1ZLNI3C7AHaEP?pid=ImgDet&rs=1"
        alt="" width="700" height="500">
    </center>
<?php } else { ?>
     <script>
         alert("You are not allowed to access this page. Please login first !");
         window.location.href = "login.php";
     </script>
<?php } ?>
```

The home page require header toolbars and options to start query. Because it is a forbidden webpage to prevent intrusions so it will force user to login first before accessing to this page, for it belongs to admin authorities. The front end show the content of the homepage with introduced words and inserted image writed at the center of the page. The <script> type present notification for users who want to access this page that they have to login before accessing then move them to the admin login webpage.

b. View categories (Admin)

Figure 1. 27. View categories (admin) webpage

The View categories page for admin will be shown after selecting “VIEW CATEGORIES” option from toolbars. This webpage was created by the view_category function file.

```php
<?php
require_once "header.php";
$sql = "SELECT * FROM category";
$run = query($sql);
?>
<center>
<table border="1">
    <tr>
        <th>Category ID</th>
        <th>Category Name</th>
        <th>Options</th>
    </tr>
    <?php
    while ($category = mysqli_fetch_array($run)) {
        ?>
    <tr>
        <td><?= $category[0] ?></td>
        <td><?= $category[1] ?></td>
        <td class="options">
            <form class="inline" action="update_category.php" method="post">
                <input type="hidden" name="id" value="<?= $category[0] ?>">
                <input type="submit" value="UPDATE">
            </form>
```

```
            <form class="inline" action="delete_category.php" method="post"
            onsubmit="return confirm_delete()">
                <input type="hidden" name="id" value="<?= $category[0] ?>">
                <input type="submit" value="DELETE">
            </form>
        </td>
    </tr>
    <?php
    }
    ?>
</table>
</center>
<script>
    function confirm_delete() {
        var del = confirm("Do you want to delete this category ?");
        if (del) {
            return true;
        } else {
            return false;
        }
    }
</script>
```

Firstly, the function using require_once “header.php” to confirm that user selected VIEW CATEGORIES option from header toolbar. Then, it starts a query function to select inserted category information from database category. In the center, the website will illustrate category ID, category name and options with matched content in the next query. The while loop start to search all the category detail

from the database then print it to the created blank that used <td> column function. For Options, the website use form class " " action " " method " " function to make the button for Update and Delete. Each of these options will have the executing file to move in like "update_category.php" and "delete_category.php". The <script> type show notification for confirmation that user really want to delete category or not.

c. Admin update category

When user choose update a category from category list, it will move to their own execution file from the UPDATE button shown above. The update webpage has its own function code.

```php
<?php
require_once "header.php";

$id = $_POST['id'];
if (isset($_POST['update'])) {
    $name = $_POST['name'];
    $sql = "UPDATE category SET category_name = '$name' WHERE class_id ='$id'";
    $run = query($sql);
    echo $run;
    if ($run) { ?>
        <script>
            alert ("Update category succeed !");
            window.location.href = "view_category.php";
        </script>
    <?php } else { ?>
        <script>
            alert ("Update category failed !");
            window.location.href = "";
        </script>
```

```php
<?php } } else {
    $sql1 = "SELECT * FROM category WHERE category_id = '$id'";
    $run1 = query($sql1);
    $class = mysqli_fetch_array($run1);
    ?>
<form action="" method="post">
    <fieldset>
        <legend>UPDATE CATEGORY</legend>
        Category Name: <input type="text" name="name" id=""
        minlength="3" maxlength="10" value="<?= $class[1] ?>" required>
        <br><br>
        <input type="hidden" name="id" value="<?= $id ?>">
        <input type="submit" value="UPDATE" name="update">
    </fieldset>
</form>
<?php } ?>
```

The update funtion also require header function of toolbars and confirmation for updating. Continuously, it set $id to mark the function information to update the running the query to select the category entity to update. The conditional statement was made to check for update succeed or not using <script> notification else it will return the first information of that category if user do not want to update anymore. If update succeed it will move to the customer category list web page using window.location.href = “view_category.php” function. Else then the website will run second query to allow user to input the category name and update button function to submit.

d. Admin add category

Figure 1. 28. Admin add category webpage

The admin add category use the function from add_category php file.

```php
<?php
require_once "header.php";

$id = $_POST['id'];
if (isset($_POST['add'])) {
    $name = $_POST['name'];
    $sql = "INSERT INTO category (category_name) VALUES ('$name')";
    $run = query($sql);
    if ($run) { ?>
        <script>
            alert ("Add category succeed !");
            window.location.href = "view_category.php";
        </script>
    <?php } else { ?>
        <script>
            alert ("Add category failed !");
            window.location.href = "";
        </script>
<?php } }else { ?>
<center>
```

```
    <form action="" method="post">
        <fieldset>
            <legend>ADD CATEGORY</legend>
            Category Name: <input type="text" name="name" id=""
            minlength="3" maxlength="10" required>
            <br><br>
            <input type="submit" value="ADD" name="add">
        </fieldset>
    </form>
</center>
<?php } ?>
```

The webpage inherit toolbar from header web function before setting query for selecting category information from the databse in the conditional statement. The statement also shown notification that adding new category successful or not using <script> type. Then it will present the form in else statement for user to input category detail then submit by the coded ADD button function.

e. View products (Admin)

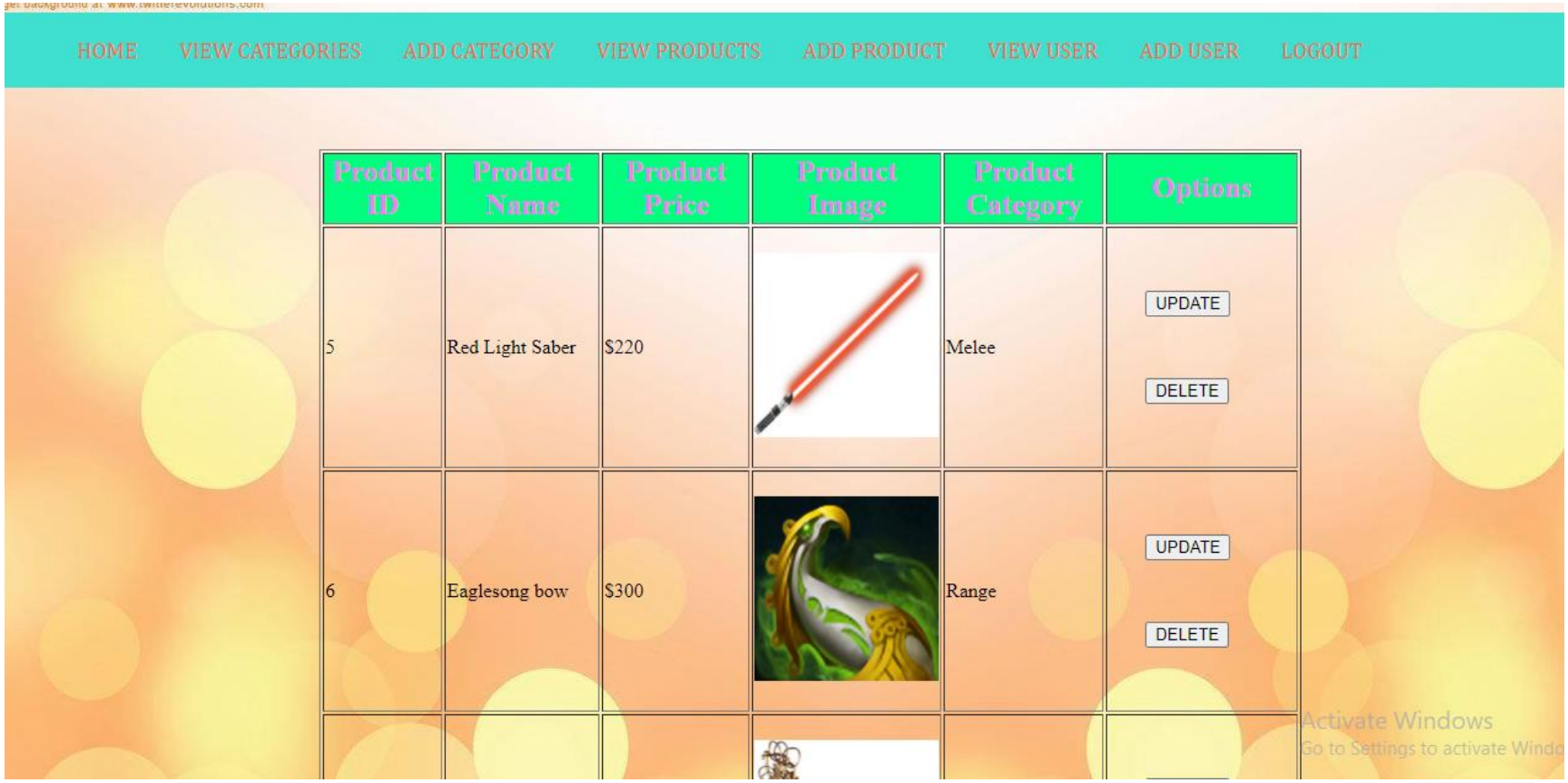


Figure 1. 29. Admin view products webpage

The products list webpage has its function coded in the view_product.php file.

```php
<?php
require_once "header.php";
?>
<center>
    <table border="1">
        <tr>
            <th>Product ID</th>
            <th>Product Name</th>
            <th>Product Price</th>
            <th>Product Image</th>
            <th>Product Category</th>
            <th>Options</th>
        </tr>
        <?php
        $sql = "SELECT * FROM  product";
        $run = query($sql);
        while ($product = mysqli_fetch_array($run)) {
            ?>
        <tr>
            <td><?= $product[0] ?> </td>
            <td><?= $product[1] ?> </td>
            <td><?= $product[2] ?> </td>
            <td>
                <a href="product_detail.php?ProductID=<?= $product[0] ?>">
                <img src="images\<?= $product[3] ?>" width="150" height="150">
                </a>
```

```
            </td>
            <?php
                $sql1 = "SELECT category_name FROM category WHERE
                category_id = '$product[4]'";
                $run1 = query($sql1);
                $cls = mysqli_fetch_array($run1);
                ?>
            <td> <?= $cls[0] ?> </td>
            <td class="options">
                <form action="update_product.php" method="post">
                    <input type="hidden" name="id" value="<?= $product[0] ?>">
                    <input type="submit" value="UPDATE">
                </form>
                <form action="delete_product.php" method="post"
                onsubmit="return confirm_delete()">
                    <input type="hidden" name="id" value="<?= $product[0] ?>">
                    <input type="submit" value="DELETE">
                </form>
            </td>
        </tr>
        <?php
        } ?>
    </table>
```

```
</center>
<script>
    function confirm_delete() {
        var del = confirm ("Do you want to delete this product ?");
        if (del) {
            return true;
        } else {
            return false;
        }
    }
</script>
```

The view products page require toolbars and confirmation from header page by “require_once ‘header.php’;” function. In the center of the page, it will print out all the titles of the product entity in one column. To make the product information match with the titles, the function use query to select products information from database and print out all of them in ascending order depend on the product ID in auto_increment mode. Therefore, in the while loop each product detail has their order sequence and product image will be selected from their matched ID. Because category_id is the foreign key reference from category entity, the code will run another query to select all category ID from the database and print them out with contained products. The next two function print out the option “UPDATE” and “DELETE” with the selected product and move them to the execution file. The <script> notification is used for check for user confirmation to delete product or not.

f. Admin update product

Figure 1. 30. Admin update product webpage

Update product webpage has its function coded in update_product.php file.

```php
<?php
require_once "header.php";

$id = $_POST['id'];
if ($_POST['change']) {
    $name = $_POST['name'];
    $price = $_POST['price'];
    $image = "";
    $current_image = $_POST['current_image'];
    $category = $_POST['category'];


    if (isset($_FILES['image']) && $_FILES['image']['size'] != 0) {

        $temp_name = $_FILES['image']['tmp_name'];

        $img_name = $_FILES['image']['name'];

        $parts = explode(".", $img_name);

        $extension = end($parts);

        $random = rand(1, 100);
        $image = $name . "_" . $class . "_" . $random . "." . $extension;

        $path = "images/" . $image;
```

```
        move_uploaded_file($temp_name, $path);
        $sql1 = "UPDATE product SET product_name = '$name',
                 product_price = '$price', product_image = '$image',
                  product_category = '$category'
                 WHERE product_id = '$id'";
    } else {
        $sql1 = "UPDATE product SET product_name = '$name',
                 product_price = '$price', product_image = '$current_image',
                  product_category = '$category'
                 WHERE product_id = '$id'";
    }
    $run1 = query($sql1);
    if ($run1) { ?>
        <script>
            alert ("Update product succeed !");
            window.location.href = "view_product.php";
        </script>
<?php } else { ?>
        <script>
            alert ("Update product failed !");
            window.location.href = "";
        </script>
<?php } } else {
        $sql2 = "SELECT * FROM product WHERE product_id = '$id'";
        $run2 = query($sql2);
```

```
        $std = mysqli_fetch_array($run2);
    ?>
<center>
    <form action="" method="post" enctype="multipart/form-data">
        <fieldset>
            <legend>Update product</legend>
            Name: <input type="text" name="name" required minlength="10"
            maxlength="30" value="<?= $std[1] ?>">
            <br><br>
            Price: <input type="price" name="price" value="<?= $std[3] ?>">
            <br><br>
            Image:
            <img src="images\Weapon<?= $std[5] ?>" width="100" height="100"> <br>
            <input type="file" name="image" accept="image/*">
            <br><br>
            Category:
            <select name="category">
                <?php
                $sql = "SELECT * FROM category";
                $run = query($sql);
                while ($cls = mysqli_fetch_array($run)) {

                    if ($cls['category_id'] == $std['product_category']) { ?>
                        <option selected value="<?= $cls['category_id']?>">
                            <?= $cls['category_name'] ?>
```

```
                        </option>
                        ?>

                    <?php
                     } else { ?>
                        <option value="<?= $cls['category_id']?>">
                            <?= $cls['category_name'] ?>
                        </option>
                <?php }
                }?>
            </select>
            <br><br>
            <input type="hidden" name="id" value="<?= $id ?>">
            <input type="hidden" name="current_image" value="<?= $std[5] ?>">
            <input type="submit" value="UPDATE" name="change">
        </fieldset>
    </form>
</center>
<?php } ?>
```

Liked other webpage, Add product page also require header toolbars and confirmation that user selected ADD PRODUCT option. Then it will set all the product entities as variables like $id, $name, etc and $id required to start the statement to execute the query. The image variable are set separately from other entities to code the image upload and execution and check that user select image file that have more than 0 byte by conditional statement “if (isset($_FILES['image']) && $_FILES['image']['size'] != 0)”. In the if statement, first it will declare variable to save image in a temporary link by function “$temp_name = $_FILES['image']['tmp_name']”.

Then it will declare declare variable used to save image name : “$img_name = $_FILES['image']['name'];”. The next step is to detach image name based on the dots using function by “ $parts = explode(".", $img_name); “ function and get the image file extension by “$extension = end($parts);” function. Then it will create new address for image using “$random = rand(1, 100); $image = $name . "_" . $class . "_" . $random . "." . $extension; $path = "images/" . $image; “functions then move image files from temporary link to the created address by function “move_uploaded_file($temp_name, $path)”.

After setting for images, it will run query for setting new image and other product detail from the new ID, else it will show products detail but with current image because user do not update image but select current image. The check statement is created to see that new product is added successfully or not, if success it will move forward to the view product page by function “window.location.href = ‘view_product.php’”, else it will return to the current add product webpage. Else it will run query for selecting inserted products and its information from the database to print out to the body.

The center body create form and fieldset for details for products to update: Name, Price,form edited Image, Category name that selected based on the category ID of that product. For category name, it will run a query to select category name from category database based on the ID to run in the while loop with if - else conditional statement inside to select category. The UPDATE button is code for user to submit the updated product detail for checking before updating to the product list.

g. Admin add product

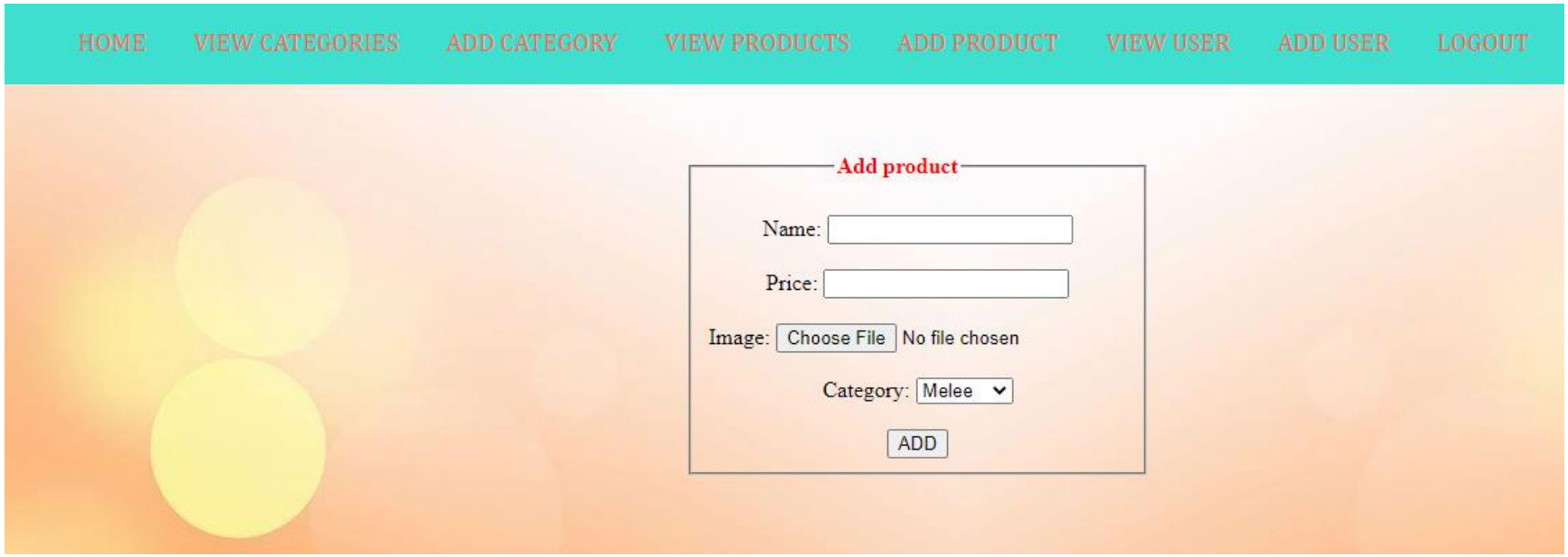


Figure 1. 31. Admin add product webpage

The admin add product webpage has its function coded in add_product.php file.

```php
<?php
require_once "header.php";
if ($_POST['add']) {
    $name = $_POST['name'];
    $price = $_POST['price'];
    $image = "";
    $category = $_POST['category'];

    if (isset($_FILES['image']) && $_FILES['image']['size'] != 0) {

        $temp_name = $_FILES['image']['tmp_name'];

        $img_name = $_FILES['image']['name'];

        $parts = explode(".",$img_name);

        $extension = end($parts);

        $random = rand(1,100);
        $image = $name . "_" . $class . "_" . $random . "." . $extension;

        $path = "images/" . $image;

        move_uploaded_file($temp_name,$path);
```

```
    }
    $sql1 = "INSERT INTO product (product_name, product_price,
                                  product_image, product_category) VALUES ('$name','$price',
                                   '$image','$category')";
    $run1 = query($sql1);
    if ($run1) { ?>
        <script>
            alert ("Insert new product succeed !");
            window.location.href = "view_product.php";
        </script>
<?php } else { ?>
        <script>
            alert ("Insert new product failed !");
            window.location.href = "";
        </script>
<?php } } else { ?>
<center>
    <form action="" method="post" enctype="multipart/form-data">
        <fieldset>
            <legend>Add product</legend>
            Name: <input type="text" name="name"
            required minlength="10" maxlength="30"><br><br>
            Price: <input type="price" name="price"> <br><br>
            Image: <input type="file" name="image" accept="image/*" required> <br><br>
```

```
            Category:
            <select name="category">
                <?php
                $sql = "SELECT * FROM category";
                $run = query($sql);
                while ($cls = mysqli_fetch_array($run)) { ?>

                    <option value="<?= $cls['category_id']?>">
                        <?= $cls['category_name'] ?>
                    </option>
                <?php } ?>
            </select>
            <br><br>
            <input type="submit" value="ADD" name="add">
        </fieldset>
    </form>
</center>
<?php } ?>
```

Like update products function, add product webpage also need header toolbars, confirmation and set variables for each entity of product database. The image style and function are set as the same as the update product web page. Then it will run query to select product entities from database to add. The <script> style in the conditional check statement show notification that new product is inserted successfully or not. In the center, the front end code print out the product detail titles and the back end run query to match the product information with each detail title by selecting category ID and product detail from their database. The ADD button is code to make sumission for adding new product.

h. View user (Admin)

Figure 1. 32. Admin view user webpage

The web page used function of view_user.php file.

```php
<?php
require_once "header.php";
?>
<center>
<table border="1">
  <tr>
    <th>User ID</th>
    <th>Username</th>
    <th>Password</th>
    <th>Options</th>
    </tr>
    <?php
    $sql = "SELECT * FROM user";
    $execute = query($sql);
    while ($user = mysqli_fetch_array($execute)) {
    ?>
    <tr>
    <td><?= $user[0] ?></td>
    <td><?= $user[1] ?></td>
    <td><?= $user[2] ?></td>
    <td class="options">
      <form class="inline" action="../admin/update_pass.php" method="post">
        <input type="hidden" name="id" value="<?= $user[0] ?>">
        <input type="submit" value="UPDATE" name="update">
      </form>
```

```
        <form class="inline" action="delete_user.php" method="post"
        onsubmit="return confirm_delete()">
            <input type="hidden" name="id" value="<?= $user[0] ?>">
            <input type="submit" value="DELETE" name="delete">
        </form>
    </td>
    </tr>
    <?php
    }
    ?>
</table>
</center>
<script>

  function confirm_delete() {
    var del = confirm("Are you sure to delete this user ?");
    if (del) {
      return true;
    } else {
      return false;
    }
  }
</script>
```

The web user header toolbar and confirmation for VIEW USER button of admin then it runs the center contains titles of user: User ID, Username, Password and Options. To match the information of user database with these titles, the function run query that select

current user in database their start the while loop to print them out. For button, they use form style to create options UPDATE and DELETE. For DELETE option, function use <script> style to make confirmation that admin want to delete selected user or not.

i. Admin add user.

Figure 1. 33. Admin add user webpage

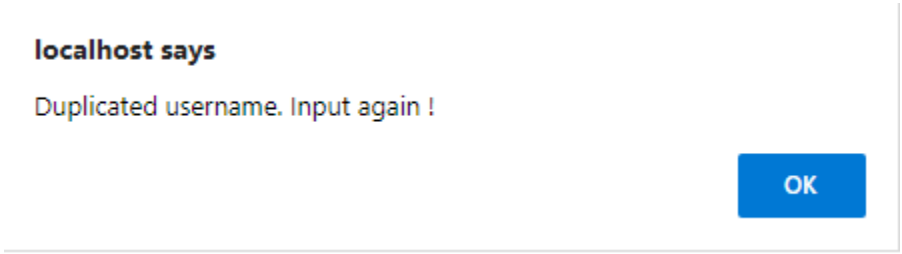
Admin add user webpage has its functions based on add_user.php file.

```php
<?php
require_once "header.php";

if (isset($_POST['add'])) {

    $username = $_POST['username'];
    $password = $_POST['password'];
    $confirm = $_POST['confirm'];

    $sql = "SELECT user_name FROM user WHERE user_name = '$username'";
    $run = query($sql);
    $check = mysqli_fetch_array($run);
    if (is_array($check)) { ?>
        <script>
            alert("Duplicated username. Input again !");
            window.location.href = "";
        </script>
    <?php } else {
      if ($password != $confirm) { ?>
        <script>
            alert("Passwords are not similar. Input again !");
            window.location.href = "";
        </script>
```

```
    <?php } else {

        $pass = encrypt($password);

        $sql1 = "INSERT INTO user (user_name, password) VALUES ('$username','$pass')";
        $run1 = $connection->query($sql1);
        if ($run1) { ?>
          <script>
              alert("Add new user succeed !");
              window.location.href = "view_user.php";
          </script>
        <?php } else { ?>
          <script>
              alert("Add new user failed !");
              window.location.href = "";
          </script>
    <?php }
    }
} else {
?>
```

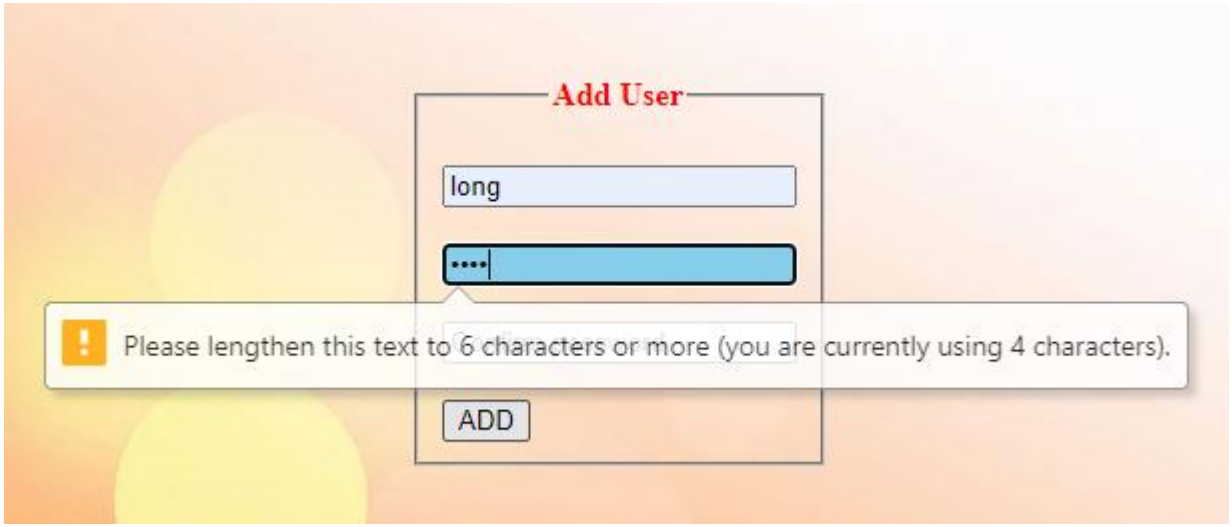
```
<form action="" method="post">
    <fieldset>
        <legend>Add User</legend>
        <input type="text" name="username" placeholder="Username" id="" required>
        <br><br>
        <input type="password" name="password" placeholder="Password"
        minlength="6" maxlength="20" required>
        <br><br>
        <input type="password" name="confirm"
        placeholder="Confirm password" minlength="6" maxlength="20" required>
        <br><br>
        <input type="submit" value="ADD" name="add">
    </fieldset>
</form>
<?php } ?>
```

After creating function of inheriting toolbar and confirmation from header, it start to use conditional statement to set query and variable for usename, password and confirm password. Then it run the query to select username from variable $username to check that it is duplicated the already names in the database or not. If it dubplicated, the conditional statement will print out notification using <script> style the the name is duplicated and user have to input again.



Else in the conditional statement, funtion also check that the confirmed password match with the input password or not. If it does not match, <script> type show notification make user to input again.

The function also check for word limits in the input field that name or password is to short or contain special type. For example, if password is too short, they will use the required function password to make user lengthen them.



Then it will run one more query to check that new user is input successfully or not. If it succeed, it will move new user to the user list webpage by function “window.location.href = "view_user.php";”, else notificate that “Add new user failed” and return to the add user webpage.

In the field, functions make details for fieldset that let user input user information and create submit button “ADD”.

Finally, if admin to LOGOUT button it will move to the customer view webpage by function “
Logout” that already show in the header.

D.Website tesing (P7)

No	Test case	Input data	Expected output	Actual output	Evaluation

1	Admin add products	New product name, product price, image, select category for new product	Localhost say: "Add new product suceed" And the new product is added to the view product list page	Localhost say: "Add new product suceed" And the new product is added to the view product list page	The website run as expected so it meet the requirement (PASS)
2	Test link image	Choose image in the product file	The image is printed out in the product detail	The image is printed out in the product detail	The website run as expected so it meet the requirement (PASS)
3	Add category	New category name	Localhost say: Add new category suceed" and the new category is added to the view category page	Localhost say: Add new category suceed" and the new category is added to the view category page	The website run as expected so it meet the requirement (PASS)
4	Add user	User name, password and confirmation password	Localhost say: "Add new user suceed" and the new user is added to the view user page	Localhost say: "Add new user suceed" and the new user is added to the view user page	The website run as expected so it meet the requirement (PASS)
5	Update category	New category name	Localhost say: "Update cateory succeed" and the new	Localhost say: "Update cateory succeed" and	The website run as expected so it meet the

			category is added to the view category page that replace the old one	the new category is added to the view category page that replace the old one	requirement (PASS)
6	Update product	New product name or product price or image or select category for updated product	Localhost say: "Update product suceed" and the new product is added to the view product page and replace the old one	Localhost say: "Update product suceed" and the new product is added to the view product page and replace the old one	The website run as expected so it meet the requirement (PASS)
7	Update user	New user name, password and confirmation password	Localhost say: Update user succeed" and the new user is added to the view user page and replace the old one	When clickling UPDATE button it only print notification the "Update product succeed" but no details are printe out for user to input new user information	Some the function did not work as expected and the queries for update user was not correct so it does not run as well (FAILED)

8	Delete user	Press DELETE button for the selected user	Localhost say: "Are you sure to delete this user" and print two options YES, NO. If choosing YES then localhost say: "Delete user succeed" and that user will be disappeared from the user list	Localhost say: "Are you sure to delete this user" and print two options YES, NO. If choosing YES then localhost say: "Delete user succeed" and that user will be disappeared from the user list	The website run as expected so it meet the requirement (PASS)
9	Delete category	Press DELETE button for the selected category	Localhost say: "Are you sure to delete this category" and print two options YES, NO. If choosing YES then localhost say: "Delete category succeed" and that catgory will be disappeared from the category list	Localhost say: "Are you sure to delete this category" and print two options YES, NO. If choosing YES then localhost say: "Delete category succeed" and that catgory will be disappeared from the category list	The website run as expected so it meet the requirement (PASS)

10	Delete product	Press DELETE button for the selected product	Localhost say: "Are you sure to delete this product" and print two options YES, NO. If choosing YES then localhost say: "Delete product succeed" and that product will be disappeared from the product list	Localhost say: "Are you sure to delete this product" and print two options YES, NO. If choosing YES then localhost say: "Delete product succeed" and that product will be disappeared from the product list	The website run as expected so it meet the requirement (PASS)
11	Test link home	Press HOME option in the toolbar	The home page is shown with its details	The home page is shown with its details	The website run as expected so it meet the requirement (PASS)
12	Logout	Press LOGOUT in the toolbar of admin interface	The web will move to the home page for customers	The web will move to the home page for customers	The website run as expected so it meet the requirement (PASS)
13	Customer view products	Press VIEW PRODUCTS from the toolbar of customer interface	The web will move to the view products list of customer that show the list of products	The web will move to the view products list of customer that show the list of products	The website run as expected so it meet the requirement (PASS)

			and their details	and their details	
14	Customer view categories	Press VIEW CATEGORIES from the toolbar of customer interface	The web will move to the view categories list of customer that show the list of categories	The web will move to the view categories list of customer that show the list of categories	The website run as expected so it meet the requirement (PASS)
15	Customer view products in filtered category	In the categories list for customer, select one category by clicking it	The web will move to the product list of selected category	The web will move to the product list of selected category	The website run as expected so it meet the requirement (PASS)
16	Admin view user	Press VIEW USERS from the toolbar of admin interface	The web will move to the view users list of admin that show the list of users	The web will move to the view users list of admin that show the list of users	The website run as expected so it meet the requirement (PASS)
17	Admin login	Input admin name and passoword to details of ADMIN LOGIN	Localhost say :" Login succeed" for the correct details then move user to the admin webpace, else localhost say: "Login failed" for the incorrect detail then return to	Localhost say :" Login succeed" for the correct details then move user to the admin webpace, else localhost say: "Login failed" for the incorrect detail then return to	The website run as expected so it meet the requirement (PASS)

			the ADMIN LOGIN detail page to make user input again	the ADMIN LOGIN detail page to make user input again	

E. Reference

Marianne, 2021. [online] Available at: <https://www.feelingpeaky.com/9-principles-of-good-web-design/#:~:text=PRINCIPLES%20OF%20GOOD%20WEBSITE%20DESIGN%20An%20effective%20website,and%20functionality%20all%20contribute%20to%20good%20website%20design.> [Accessed 9 May 2021].

Pawar, S., 2018. *8 Principles of Good Website Design*. [online] Astra. Available at: <https://wpastra.com/good-website-design/> [Accessed 9 May 2021].