

ASSIGNMENT 2 FRONT SHEET

Qualification	BTEC Level 5 HND Diploma in Computing		
Unit number and title	10: Website Design & Development		
Submission date	27/08/2022	Date Received 1st submission	
Re-submission Date		Date Received 2nd submission	
Student Name	Bùi Hương Linh	Student ID	GBH200662
Class	GCH1006	Assessor name	Lai Manh Dung
Student declaration <p>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.</p>			
		Student's signature	<i>Linh</i>

Grading grid

P5	P6	P7	M4	M5	D2	D3

⚙ **Summative Feedback:**⚙ **Resubmission Feedback:****Grade:****Assessor Signature:****Date:****Signature & Date:**

Table of Contents

Introduction.....	6
Chapter 1 – Design Plan	6
1. User’s requirements	6
2. Review accessibility standards and guidelines	6
3. Use cases diagram.....	7
4. Wireframes.....	8
4.1. Home page	8
4.2. Register	9
4.3. Log in.....	10
4.4. Product detail	11
4.5. Admin	12
5. Sitemap	13
Chapter 2 – Implementation	14
1. Home page	14
HTML header, logo	15
CSS header, logo.....	15
HTML search.....	15
PHP search.....	16
CSS search	16
HTML login, register.....	16
CSS login, register	17
HTML sidebar.....	18
CSS sidebar.....	18
HTML and PHP ads.....	19
CSS ads	19
PHP and HTML maincontent, all product	20
CSS maincontent and all product.....	20
HTML footer.....	21
CSS footer.....	21
2. Product detail	22

PHP and HTML product detail	22
CSS product detail	23
3. Register	24
HTML and PHP register	24
CSS register	25
4. Login	25
HTML and PHP login	25
CSS Login	26
5. Admin page	27
HTML header, logo, search, logout	27
CSS header, logo, search, logout	28
HTML and PHP maincontent	29
CSS maincontent	30
HTML and PHP add new product, update product	31
CSS add new product and update product	33
HTML and PHP delete product	33
Function session_start	33
My SQL database	34
Chapter 3 – Testing	35
Conclusion	37
Evaluation	37
References	37
Figure 1: Use case diagram	8
Figure 2: Home page web	9
Figure 3: Register	10
Figure 4: Login	11
Figure 5: Product detail	12
Figure 6: Admin	13
Figure 7: Site map	14
Figure 8: Home page	14
Figure 9: HTML header, logo	15
Figure 10: CSS header, logo	15
Figure 11: HTML search	15

Figure 12: PHP search	16
Figure 13: CSS search.....	16
Figure 14: HTML login and register.....	16
Figure 15: CSS login and register.....	17
Figure 16: HTML sidebar	18
Figure 17: CSS sidebar	18
Figure 18: HTML and PHP ads	19
Figure 19: CSS ads	19
Figure 20: PHP and HTML maincontent, all product.....	20
Figure 21: CSS maincontent and all product	20
Figure 22: HTML footer	21
Figure 23: CSS footer	21
Figure 24: Product detail.....	22
Figure 25: PHP and HTML product detail	22
Figure 26: CSS product detail.....	23
Figure 27: HTML and PHP register.....	24
Figure 28: CSS register.....	25
Figure 29: HTML and PHP login	26
Figure 30: CSS Login	27
Figure 31: Admin page	27
Figure 32: HTML header, logo, search, logout	27
Figure 33: CSS header, logo, search, logout.....	28
Figure 34: HTML and PHP maincontent.....	29
Figure 35: CSS maincontent	30
Figure 36: HTML and PHP add new product, update product	32
Figure 37: CSS add new product and update product.....	33
Figure 38: HTML and PHP delete product.....	33
Figure 39: Function session_start	33
Figure 40: My SQL database	34
Table 1: Test case	36

Introduction

Based on what the author has learnt from professors and other sources, the author will discuss website design concepts in her study. The building of an actual website is subsequently entrusted to them by the author. User interface design, website design, and website functionality and performance testing were the author's three divisions of the project. This report will go into great detail on how the author created their website.

Chapter 1 – Design Plan

1. User's requirements

Levents (Including T-shirts, hoodies, jackets, ...)

Users (customers) will be able to:

- View all products of the shop.
- View the product list and the list of utensils for customers.
- View detailed product information.

Users (admins) will be able to:

- Add, delete and update all products.
- Must be logged in with the correct account to use the functions.

2. Review accessibility standards and guidelines

The success of every website depends heavily on its web design. Whether or not it was carefully made by the designer. Usability and utility, not graphic design, decide how well your custom-designed website performs. Because it is the face of your company to the outside world and where the bulk of potential customers will go before visiting your store, your website has been carefully created. The inevitable is the key. Any flaw could harm your company's reputation. It takes a diverse group of people with expertise in several fields to create a decent website. Their combined efforts must be considered when making a key choice. The following list of sound design guidelines will help you make your website more aesthetically pleasing, user-friendly, dependable, and interesting. (He, 2020)

➤ Perceivable:

- ✓ Text alternatives: Make text versions of any content available in any different formats that people may need, such as Braille, large print, voice-over, symbols, or plainer language.
- ✓ Make it simpler for consumers to see and hear content by being distinctive, such as by distinguishing the foreground from the background.
- ✓ Adaptive: Produce content that can be shown in a variety of ways without losing any of its structure or information.
- ✓ Provides time-based media (often video) alternatives, such as subtitles, explained audio, or alternate media.

(He, 2020)

➤ Operable:

- ✓ Make it simpler for users to use input methods other than the keyboard to activate the functionality. Take the minimum size of the touch target as an example.
- ✓ Enough Time: Give users enough time to read and utilize the content. Examples of this include interrupts and timeouts, as well as moving, flashing, scrolling, or automatically updating information.
- ✓ Provide tools that help people navigate their way around, find content, and locate themselves. This area includes crucial user interface choices like information architecture, order of focus, and appropriate headings and labels.
- ✓ Seizures and Physical Reactions: Avoid making materials that could cause seizures or other physical reactions. Typically, this involves limiting flicker and motion animation to reasonable levels.
- ✓ Provides all of the keyboard's features when it is keyboard accessible. This is the ability to use the keyboard interface to switch the focus amongst all page elements. (He, 2020)
- Understandable:
 - ✓ Predictability: Make Web pages look and behave consistently. This includes predictable component interactions and consistent navigation.
 - ✓ Assist users in avoiding and correcting errors through input assistance. This includes detecting input errors and providing appropriate, timely, and relevant instructions or suggestions.
 - ✓ Readability: Make written content understandable and readable. This could be as simple as detecting and explaining idioms or acronyms, or it could be as complex as detecting and explaining idioms or acronyms. (He, 2020)
- Robust:
 - ✓ Compatibility: It is important to maximize compatibility with current and future user agents. This is primarily developer labor to ensure that the product is compatible with assistive technologies. (He, 2020)

3. Use cases diagram

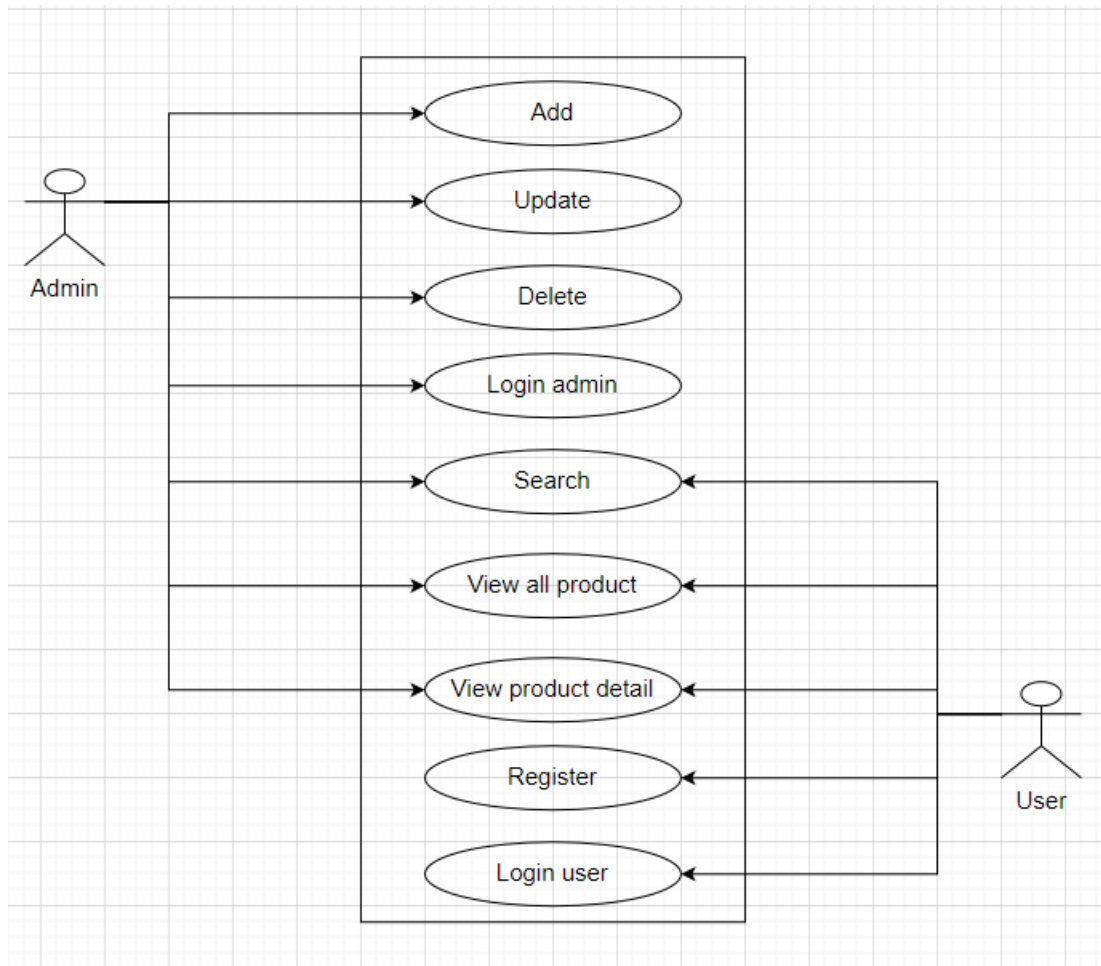


Figure 1: Use case diagram

4. Wireframes

4.1. Home page

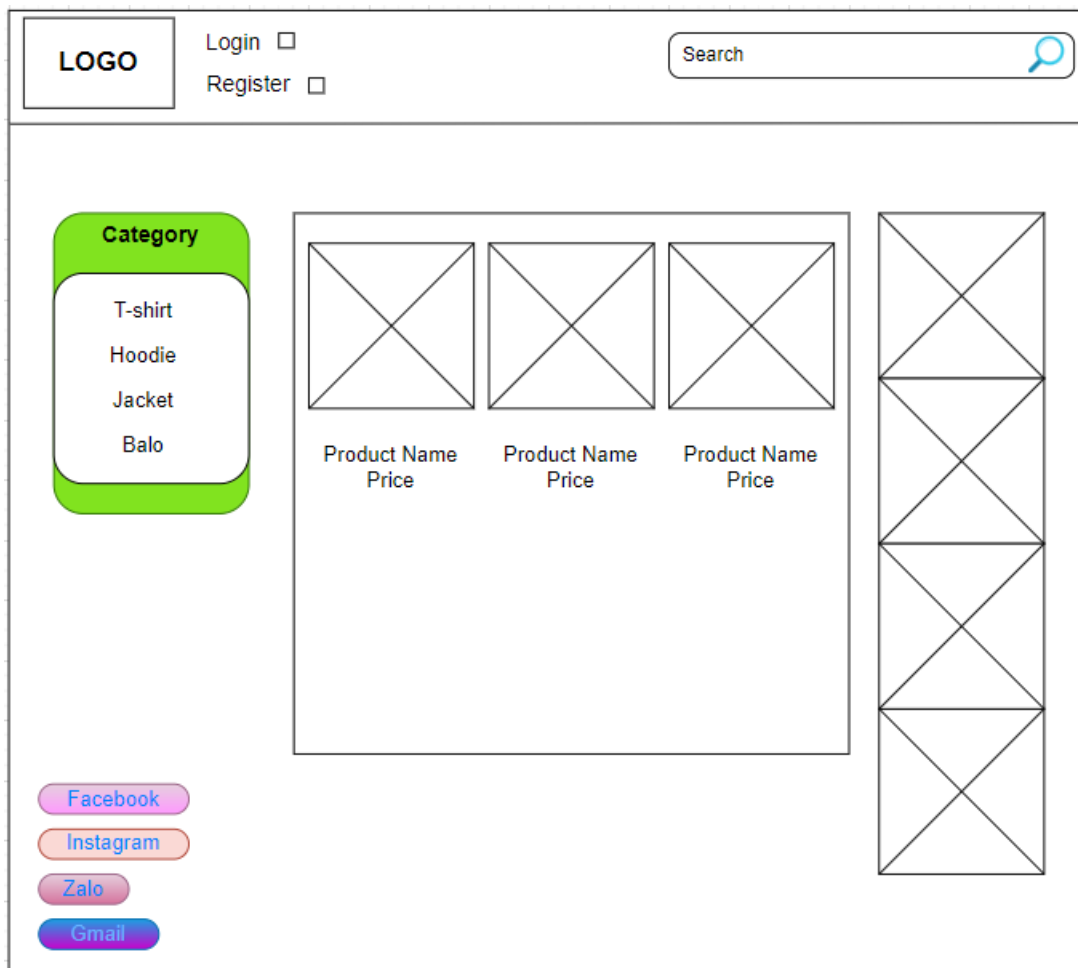


Figure 2: Home page web

4.2. Register

LOGO Login ☐ Register ☒

Search

Category

- T-shirt
- Hoodie
- Jacket
- Balo

Facebook

Instagram

Zalo

Gmail

Register

Username :

Password :

Re-Password :

Register

Product Name Price

Product Name Price

Product Name Price

Figure 3: Register

4.3. Log in

Figure 4: Login

4.4. Product detail

LOGO

Login ☐
Register ☐

Category

T-shirt
Hoodie
Jacket
Balo

Product Name
Price
Colorway
Description

Facebook

Instagram

Zalo

Gmail

Figure 5: Product detail

4.5. Admin

LOGO

Log out

Search

Category

- T-shirt
- Hoodie
- Jacket
- Balo

	Product Name	Price	Colorway	Description	Edit Delete
	Product Name	Price	Colorway	Description	Edit Delete
	Product Name	Price	Colorway	Description	Edit Delete

Add new product

Id:

Product name:

Product image:

Category:

Price:

Colorway:

Description:

Figure 6: Admin

5. Sitemap

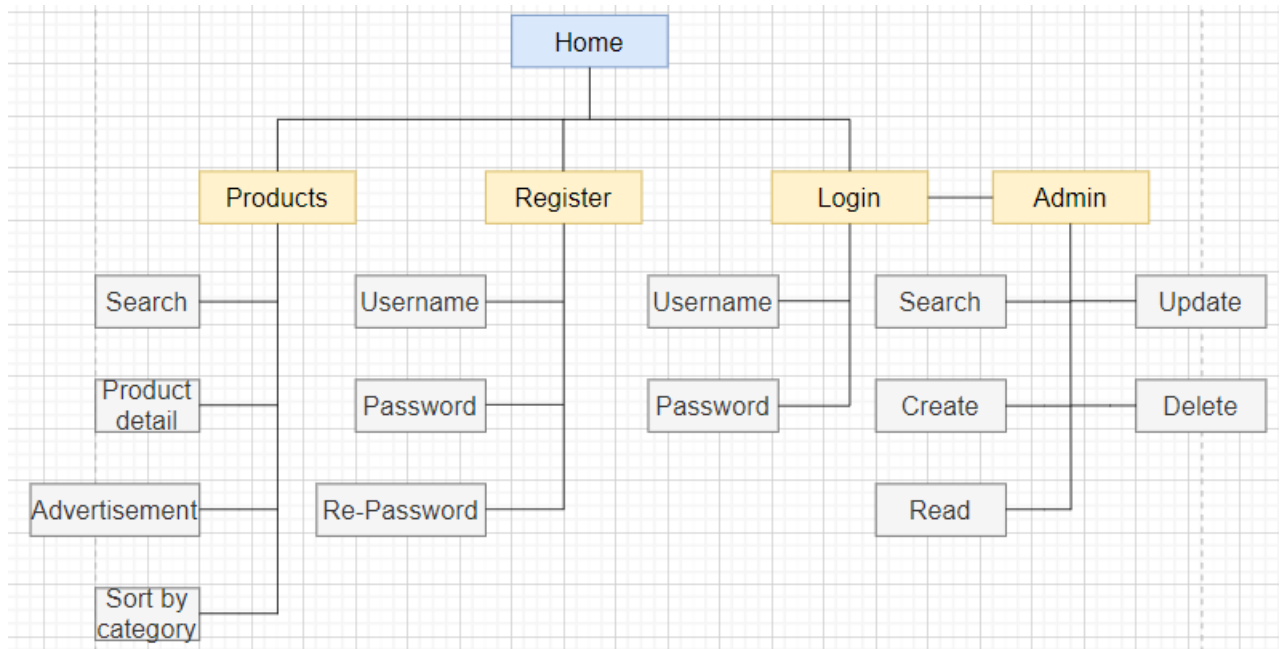


Figure 7: Site map

Chapter 2 – Implementation

1. Home page

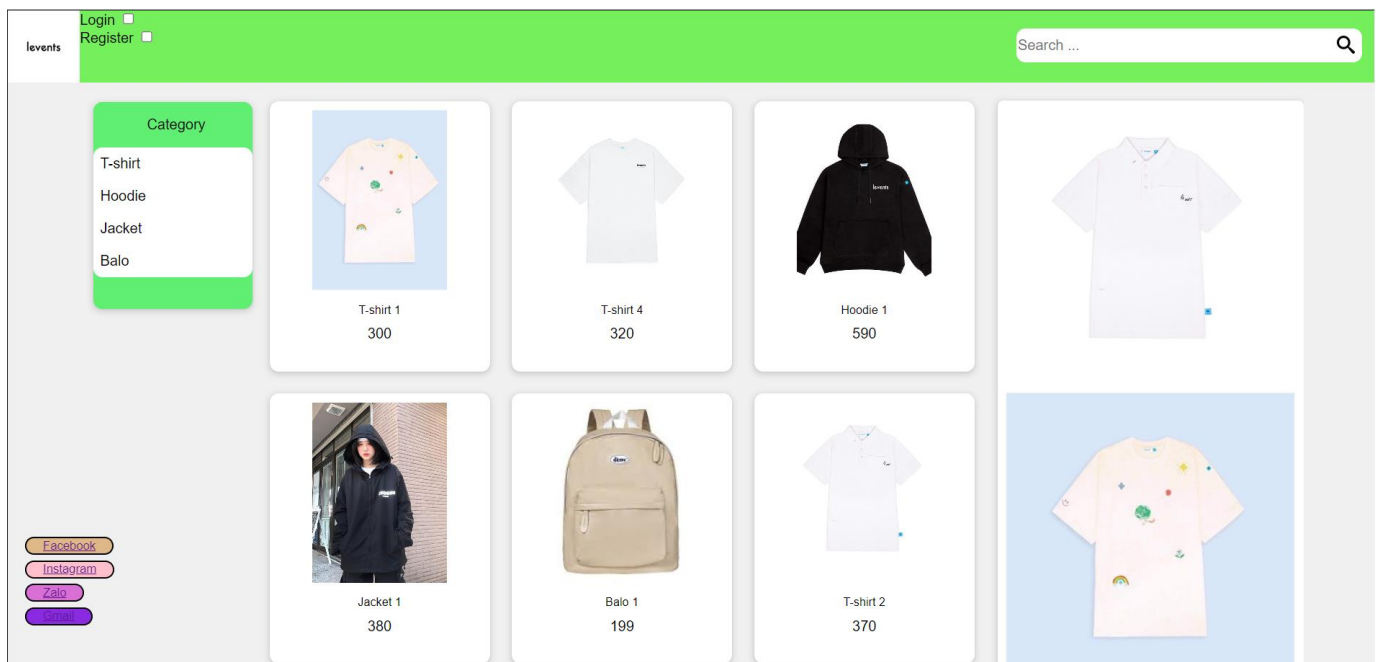


Figure 8: Home page

Users can search for products by searching or selecting the type of product they want on this page, which serves as both the home page and the shopping page.

HTML header, logo

```
<div class="header">
  <a class="logo" href="index.php"></a>

  <div class="search">
    <form action="" method="get" id="ex_fsearch">
      <input type="text" name="keyword" placeholder="Search ..." />
      <button type="submit" name="btn" onclick="window.location.href = 'index.php'" id="ex_sbm">
        <path d="M15.504 13.6161-3.79-3.223c-0.392-0.353-0.811-0.514-1.149-0.499 0.895-1.048 1
        </svg> </button>
    </form>
  </div>
</div>
```

Figure 9: HTML header, logo

CSS header, logo

```
.header{
  border: 0 solid #000;
  width: 90%;
  height: 50px;
  background-color: #7f7f7f;
}
```

```
.logo img {
  width: auto;
  height: 80px;
  float: left;
}
```

Figure 10: CSS header, logo

When the user hovers over the logo icon, the browser takes them back to the home page.

HTML search

```
<div class="search">
  <form action="" method="get" id="ex_fsearch">
    <input type="text" name="keyword" placeholder="Search ..." />
    <button type="submit" name="btn" onclick="window.location.href = 'index.php'" id="ex_sbm">
      <path d="M15.504 13.6161-3.79-3.223c-0.392-0.353-0.811-0.514-1.149-0.499 0.895-1.048 1
      </svg> </button>
    </form>
  </div>
```

Figure 11: HTML search

The browser will display the corresponding products when the user clicks on the search box for the product they want to find.

PHP search

```
<?php
if (isset($_GET['keyword']) && !empty($_GET['keyword']))
    $sql = "select * from product where productname like '%" . $_GET['keyword'] . "%'";
else
    $sql = "select * from Product";
$rows = query($sql);
?>
```

Figure 12: PHP search

CSS search

```
.search{
    float: right;
    width: 380px;
    padding: 0 15px 0 10px;
}
.search input[type=text]{
    padding: 5px -1px 5px 12px;
    width: 100%;
    border: 0px;
    font-size: 16px;
    height: 35px;
    border-radius: 10px;
    float: right;
}
```

Figure 13: CSS search

HTML login, register

```
<div class="loginregister">
    <div class="login" method="POST">
        <label for="dangnhap">Login</label>
        <input type="checkbox" id="dangnhap" >

        <div class="formlogin" id="id01">
            <?php
                require("admin/login.php");
            ?>
        </div>
    </div>
    <div class="register" method="POST">
        <label for="dangki">Register</label>
        <input type="checkbox" id="dangki" >

        <div class="formregister" id="id02">
            <?php
                require("admin/register.php");
            ?>
        </div>
    </div>
</div>
```

Figure 14: HTML login and register

The browser will display the corresponding window when the user clicks on the register or login button.

CSS login, register

```
.formlogin {
  display: none;
  position: absolute;
  background-color: #afcaef;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgb(0 0 0 / 20%);
  padding: 12px 16px;
  z-index: 1;
  color: white;
  top: 70px;
  right: 1137px;
  border-radius: 4px;
}

.Login input[type=checkbox]:checked ~ .formlogin{
display: block;
}

.formregister {
  display: none;
  position: absolute;
  background-color: #88eb64;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgb(0 0 0 / 20%);
  padding: 12px 16px;
  z-index: 1;
  color: white;
  top: 81px;
  left: 1026px;
  border-radius: 4px;
}

.register input[type=checkbox]:checked ~ .formregister{
display: block;
}
```

Figure 15: CSS login and register

HTML sidebar

```
<div class="main">
  <div class="sidebar">
    <p class="sidebar-sanpham"> Category</p>
    <ul class="list-sidebar">
      <?php
        $sql = "Select * From Category";
        $rows = query($sql);
        for($i=0; $i<count($rows); $i++)
        {
          ?>
          <li><a href="index.php?catid=<?=$rows[$i][0]?>"><?=$rows[$i][1]?></a></li>
          <?php
            }
          ?>
        </ul>
      </div>
    </div>
```

Figure 16: HTML sidebar

Displays a list of products from which users can select.

CSS sidebar

```
.main {
  width: 1360px;
  height: auto;
  margin-left: auto;
  margin-right: auto;
  overflow: hidden;
  position: relative;
  background-position: top center;
  background-repeat: no-repeat;
  background-size: 100% auto;
}

.sidebar {
  width: 13%;
  height: 230px;
  float: left;
  text-align: center;
  box-shadow: 0px 0px 0px 0px, 0px 0px 0px 0px;
  border-radius: 10px;
  margin-left: 15px;
  margin-top: 21px;
  background-color: #60ef73;
}

p.sidebar-sanpham {
  margin-left: 8px;
}
```

```
.list-sidebar {
  padding: 0;
  margin: 0;
  list-style: none;
  line-height: 20px;
  background-color: white;
  border-radius: 10px;
}

.list-sidebar li {
  margin: 0;
  padding: 0;
  display: block;
  border: none;
}

.list-sidebar li a {
  text-align: left;
  text-decoration: none;
  display: block;
  color: rgb(0, 0, 0);
  padding: 8px;
}

.list-sidebar li a:hover {
  background: #eaeaea;
}
```

Figure 17: CSS sidebar

HTML and PHP ads

```
<div class="ads">
<?php
$sql = "select * from product";
$rows = query($sql);
$section = array_rand($rows);
for($i=0;$i<5;$i++){
    ?>
    ">
<?php
}
?>
</div>
</div>
```

Figure 18: HTML and PHP ads

Randomly display images products.

CSS ads

```
.ads{
    margin-top: 20px;
    width:25%;
    height: auto;
    float: right;
    text-align: center;
    border-radius: 6px;
    box-shadow: 0px 1px 2px 0px, 0px 1px 2px 0px, 0px 1px 2px 0px, 0px 1px 2px 0px, 0px 1px 2px 0px;
    background-color: #ffffff;
}

img.advertisement{
    width: 320px;
    height: auto;
}
```

Figure 19: CSS ads

PHP and HTML maincontent, all product

```
<?php
if(isset($_GET['productid']))
    require_once('pages/product-detail.php');
else
    require_once('pages/main-product.php');
?>
```

```
<div class="maincontent">
<?php
if (isset($_GET['catid']))
    $sql = "select * from Product where CatID ='" . $_GET['catid'] . "'";
elseif (isset($_GET['productid']))
    $sql = "select * from Product where Productid ='" . $_GET['productid'] . "'";
elseif (isset($_GET['keyword']))
    $sql = "select * from product where productname like '%" . $_GET['keyword'] . "%' or productid like '%" . $_GET['keyword'] . "%' ";
else
    $sql = "select * from Product";
$rows = query($sql);

for($i=0; $i<count($rows); $i++)
{
    ?>

    <ul class="product">
    <li><a href="index.php?productid=<?=$rows[$i][0]>?>"> ?>"></a></li>
    <li class="nameproduct"><?=$rows[$i][1]>?></li>
    <li><?=$rows[$i][3]>?></li>
    </ul>
    <?php
}
?>
</div>
```

Figure 20: PHP and HTML maincontent, all product

Show all product and product detail.

CSS maincontent and all product

```
.maincontent {
width: 60%;
height: auto;
float: left;
margin-left: 7px;
margin-top: 8px;
}
```

```
.product li {
text-align: center;
list-style: none;
margin-top: 10px;
display: block;
}
li.nameproduct {
width: 100%;
font-size: 13px;
}
.product {
background-color: white;
height: 300px;
width: 30%;
float: left;
margin: 1.5%;
box-shadow: rgb(60 64 67 / 10%) 0px 1px 2px 0px, rgb(60 64 67 / 15%) 0px 2px 6px 2px;
border-radius: 10px;
padding: 0px;
}
.product li img{
height: 200px;
width: 150px;
object-fit: cover;
}
```

Figure 21: CSS maincontent and all product

HTML footer

```
<div class="footer">
  <button class="facebook"><a href="https://www.facebook.com/linh.huong.0808">Facebook</a>
</button>
  <button class="instagram"><a href="https://www.instagram.com/linh.huong.0808/">Instagram</a>
</button>
  <button class="zalo"><a href="https://zalo.me/0972208243">Zalo</a>
</button>
  <button class="gmail"><a href="https://mail.google.com/mail/u/1/#inbox">Gmail</a>
</button>
</div>
```

Figure 22: HTML footer

Show how to contact the store.

CSS footer

```
<style>
  .footer{
    position: fixed;
    bottom: 40px;
    left: 20px;
    z-index: 999;
  }
  .facebook{
    background: burlywood;
    border-radius: 20px;
    padding: 0 18px;
    color: white;
    display: block;
    margin-bottom: 6px;
    text-align: center;
  }
  .instagram{
    background: pink;
    border-radius: 30px;
    padding: 0 18px;
    color: white;
    display: block;
    margin-bottom: 6px;
    text-align: left;
  }
}
```

```
  .zalo{
    background: orchid;
    border-radius: 20px;
    padding: 0 18px;
    color: white;
    display: block;
    margin-bottom: 6px;
    text-align: right;
  }
  .gmail{
    background: blueviolet;
    border-radius: 20px;
    padding: 0 18px;
    color: white;
    display: block;
    margin-bottom: 6px;
    text-align: justify;
  }
</style>
```

Figure 23: CSS footer

2. Product detail

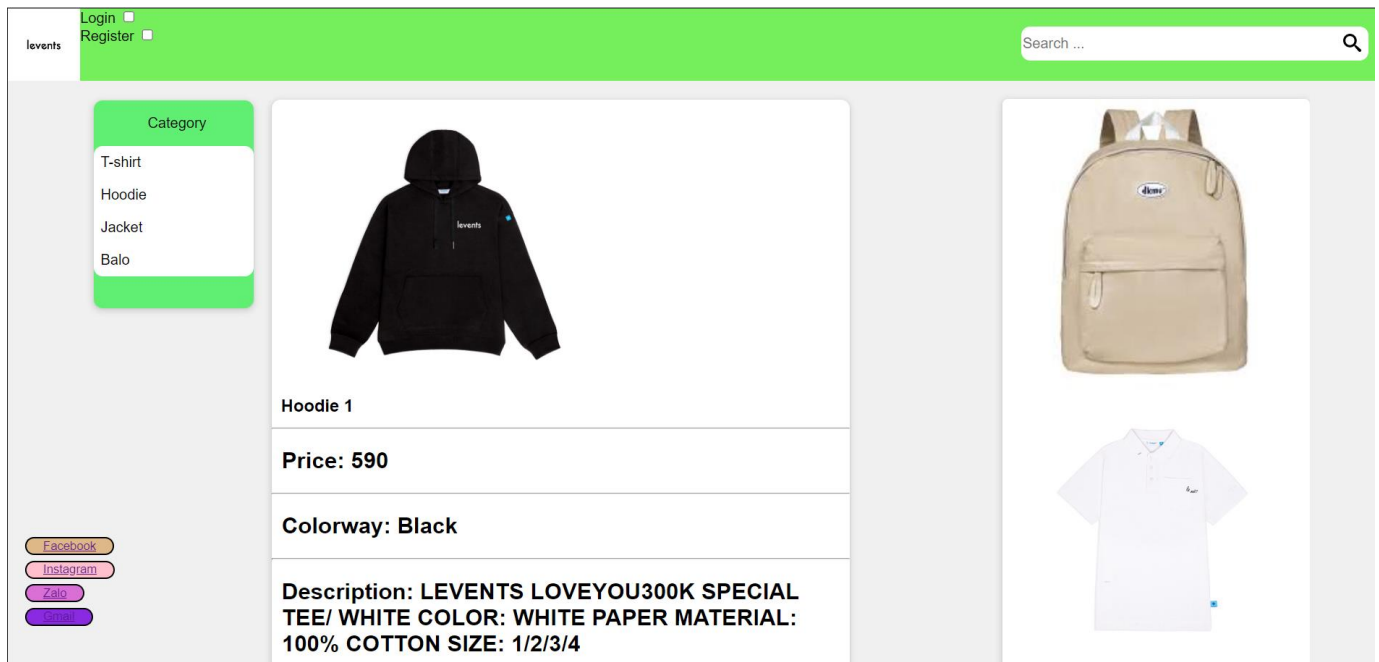


Figure 24: Product detail

When users click on any product on the homepage, they will see product details like.

PHP and HTML product detail

```
<?php
include_once('db.php');
if (isset($_GET['productid']))
    $sql = "select * from Product where ProductId = " . $_GET['productid'] . " ";
elseif (isset($_GET['keyword']))
    $sql = "select * from product where productname like '%" . $_GET['keyword'] . "%' or productid like '%" . $_GET['keyword'] . "%' ";
else
    $sql = "select * from Product";
$rows = query($sql);
for ($i = 0; $i < count($rows); $i++) {
    ?>
    <div class="detail">
        " ?>
        <h1><?=$rows[$i][1]></h1><hr>
        <h2>Price: <?=$rows[$i][3]>
        </h2><hr>
        <h2>Colorway: <?=$rows[$i][4]>
        </h2><hr>
        <h2>Description: <?=$rows[$i][5]>
        </h2>
    </div>
    <?php
} ?>
```

Figure 25: PHP and HTML product detail

Show product details.

CSS product detail

```
.detail h1 {
  font-size: 19px;
  margin-left: 10px;
}

.detail {
  background-color: white;
  height: auto;
  width: 55%;
  float: left;
  margin: 1.5%;
  box-shadow: 0px 1px 2px 0px, 0px 2px 6px 2px;
  border-radius: 10px;
  padding: 0px;
}

.detail img {
  width: 90%;
  margin-left: 41px;
  margin-top: 10px;
}

.detail p {
  text-indent: 4%;
  margin-left: 10px;
  margin-right: 10px;
}

h2 {
  font-size: 25px;
  margin-left: 11px;
}
```

Figure 26: CSS product detail

3. Register

HTML and PHP register

```

<div class="register" method="POST">
  <label for="dangki">Register</label>
  <input type="checkbox" id="dangki" >

  <div class="formregister" id="id02">
    <?php
      require("admin/register.php");
    ?>
  </div>
</div>

```

```

<form method="POST">
  <fieldset>
    <legend>Register</legend>
    <table>
      <tr>
        <td>Username</td>
        <td><input type="text" name="username" size="30"></td>
      </tr>
      <tr>
        <td>Password</td>
        <td><input type="password" name="password" size="30"></td>
      </tr>
      <tr>
        <td>Re-Password</td>
        <td><input type="repassword" name="repassword" size="30"></td>
      </tr>
      <tr>
        <td>
          <button type="submit" class="btn" name="btn_submit">Register</button>
        </td>
      </tr>
    </table>
  </fieldset>
</form>

```

```

<?php
if (isset($_POST["btn_submit"])) {
    $username = $_POST["username"];
    $password = $_POST["password"];
    $repassword = $_POST["repassword"];

    if ($username == "" || $password == "" || $repassword == "") {
        echo '<p style="text-align: center;"> Please enter the full information </p>';
    } else {
        $sql = "INSERT INTO customer(
            CusId,
            username,
            password,
            repassword
        ) VALUES (
            (1, 'adminn', 123, 123),
            (2, 'link', 123, 123),
            (3, 'linh1', 123, 123),
            (4, 'lyn', 123, 123),
        )";

        mysqli_query($conn, $sql);
        echo "Congratulations, you have successfully registered";
        header('Location: index.php');
    }
}
?>

```

Figure 27: HTML and PHP register

The customer must create an account and the information is saved in the database.

CSS register

```
.formregister {
display: none;
position: absolute;
background-color: #88eb64;
min-width: 160px;
box-shadow: 0px 8px 16px 0px rgb(0 0 0 / 20%);
padding: 12px 16px;
z-index: 1;
color: white;
top: 81px;
left: 1026px;
border-radius: 4px;
}

.register input[type=checkbox]:checked ~ .formregister{
display: block;
}
```

Figure 28: CSS register

4. Login HTML and PHP login

```
<div class="login" method="POST">
  <label for="dangnhap">Login</label>
  <input type="checkbox" id="dangnhap" >

  <div class="formlogin" id="id01">
    <?php
      require("admin/login.php");
    ?>
  </div>
</div>
```

```
<form method="POST">
  <fieldset>
    <legend>Log in</legend>
    <table>
      <tr>
        <td>Username</td>
        <td><input type="text" name="username" size="30"></td>
      </tr>
      <tr>
        <td>Password</td>
        <td><input type="password" name="password" size="30"></td>
      </tr>
      <tr>
        <td colspan="2"><input name="btn_submit" type="submit" value="Login"></td>
      </tr>
    </table>
  </fieldset>
</form>
```

```
<?php
session_start();
// Kiểm tra nếu người dùng đã ấn nút đăng nhập thì mới xử lý
if (isset($_POST["btn_submit"])) {
    // lấy thông tin người dùng
    $username = $_POST["username"];
    $password = $_POST["password"];

    if ($username == "" || $password == "") {
        echo "Username or password you can't leave blank!";
    } else {
        $sql1 = "select * from account where username = '$username' and password = '$password' ";
        $query1 = mysqli_query($conn, $sql1);
        $num_rows1 = mysqli_num_rows($query1);
        if ($num_rows1 == 0) {
            $sql2 = "select * from customer where username = '$username' and password = '$password' ";
            $query2 = mysqli_query($conn, $sql2);
            $num_rows2 = mysqli_num_rows($query2);
            if ($num_rows2 == 0) {
                echo "Username or password is incorrect !";
            } else {
                $_SESSION['username'] = $username;
                header('Location: cus-index.php');
            }
        } else {
            $_SESSION['username'] = $username;
            header('Location: ad-index.php');
        }
    }
}
?>
```

Figure 29: HTML and PHP login

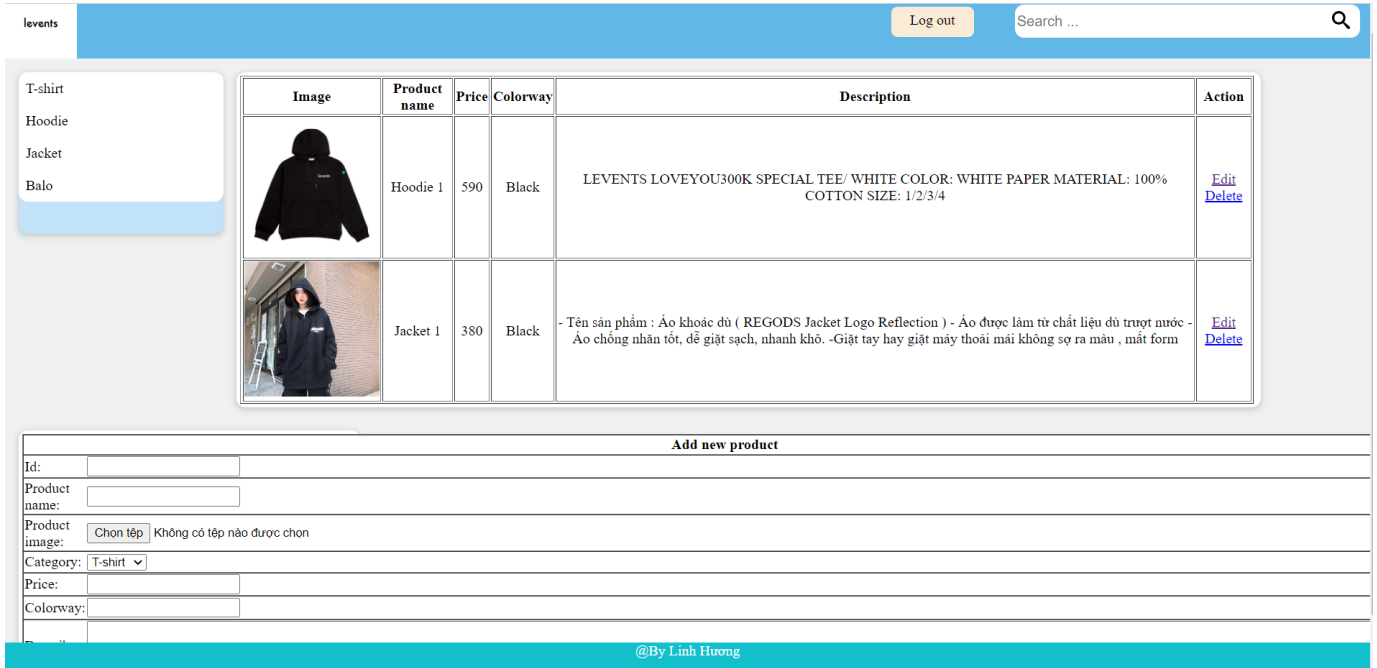
When a user clicks the login, the browser displays the login panel, and if the user is logged in with an admin account, it takes the user to the admin page.

CSS Login



```
.login input[type=checkbox]:checked ~ .formLogin{
    display: block;
}
.formLogin {
    display: none;
    position: absolute;
    background-color: #afcaef;
    min-width: 160px;
    box-shadow: 0px 8px 16px 0px rgb(0 0 0 / 20%);
    padding: 12px 16px;
    z-index: 1;
    color: white;
    top: 70px;
    right: 1137px;
    border-radius: 4px;
}
```

Figure 30: CSS Login

5. Admin page



The screenshot shows the Admin page interface. At the top, there's a navigation bar with 'levents' and a 'Log out' button. A search bar is also present. On the left, a sidebar lists categories: T-shirt, Hoodie, Jacket, and Balo. The main content area displays a table of products:

Image	Product name	Price	Colorway	Description	Action
	Hoodie 1	590	Black	LEVENTS LOVEYOU300K SPECIAL TEE/ WHITE COLOR: WHITE PAPER MATERIAL: 100% COTTON SIZE: 1/2/3/4	Edit Delete
	Jacket 1	380	Black	- Tên sản phẩm : Áo khoác dù (REGODS Jacket Logo Reflection) - Áo được làm từ chất liệu dù trượt nước - Áo chống nhăn tốt, dễ giặt sạch, nhanh khô. -Giặt tay hay giặt máy thoải mái không sợ ra màu , mất form	Edit Delete

Below the table is a form titled 'Add new product' with fields for Id, Product name, Product image (with a 'Chọn tệp' button), Category (dropdown), Price, and Colorway. The footer shows '@By Linh Hương'.

Figure 31: Admin page

Admin can add, edit, delete products when out of stock or have a new model.

HTML header, logo, search, logout

```

<div class="header">
<a class="logo" href="admin-index.php"></a>
<div class="search">
  <form action="" method="get" id="ex_fsearch">
    <input type="text" name="keyword" placeholder="Search ..."/>
    <button type="submit" name="btn" onclick="window.location.href = 'index.php'" id="ex_sbm" >
  </form>
</div>
<a class="label" href="admin/session.php">Log out</a>
</div>

```

Figure 32: HTML header, logo, search, logout

When the user hovers over the logo icon, the browser takes them back to the home page.

The browser will display the corresponding products when the user clicks on the search box for the product they want to find.

When the user clicks the Logout button, the browser logs out of the account and returns to the website's homepage.

CSS header, logo, search, logout

```
.header{
  border: 0 solid #000;
  width: 100%;
  height: 80px;
  background-color: #97, 184, 230;
}
.search{
  float: right;
  width: 380px;
  padding: 0 15px 0 10px;
}
.search input[type=text]{
  padding: 5px -1px 5px 12px;
  width: 100%;
  border: 0px;
  font-size: 16px;
  height: 35px;
  border-radius: 10px;
}
```

```
.Label {
  display: inline-block;
  border-radius: 7px;
  background-color: #antiquewhite;
  width: 80px;
  height: 22px;
  text-align: center;
  padding: 6px;
  margin-top: 22px;
  margin-right: 36px;
  text-decoration: none;
  color: #black;
  float: right;
}
.Logo img {
  width: auto;
  height: 80px;
  float: left;
}
```

Figure 33: CSS header, logo, search, logout

HTML and PHP maincontent

```
<div class="right">

    <table class="tableproduct" border="1" align="center">
        <tr>
            <th width="150">Image</th>
            <th>Product name</th>
            <th>Price</th>
            <th>Colorway</th>
            <th>Description</th>
            <th>Action</th>
        </tr>
        <?php
        if (isset($_GET['catid']))
            $sql = "select * from Product where CatID ='" . $_GET['catid'] . "'";
        elseif (isset($_GET['keyword']) && !empty($_GET['keyword']))
            $sql = "select * from product where productname like '%" . $_GET['keyword'] . "%' or productid like '%" . $_GET['keyword'] . "%'";
        else
            $sql = "select * from Product";
        $rows = query($sql);
        for($i=0; $i<count($rows); $i++)
        {
            ?>
            <tr align="center">
                <td></td>
                <td><?=$rows[$i][1]></td>
                <td><?=$rows[$i][3]></td>
                <td><?=$rows[$i][4]></td>
                <td><?=$rows[$i][5]></td>
                <td>
                    <a href="ad-index.php?updateid=<?=$rows[$i][0]>">Edit</a>
                    <a href="ad-index.php?deleteid=<?=$rows[$i][0]>">Delete</a>
                </td>
            </tr>
            <?php
        }
        ?>
    </table>
</div>
<?php
if(isset($_GET['updateid']))
    require_once('update.php');
else
    require_once('add.php');
?>
```

Figure 34: HTML and PHP maincontent

Display a list of products that are currently available for editing by users.

CSS maincontent

```
.right {
  width: 75%;
  float: left;
  background-color: white;
  padding: 0 ;
  box-sizing: border-box;
  margin-top: 15px;
  margin-left: 15px;
  border-radius: 10px;
  box-shadow: rgb(60 64 67 / 10%) 0px 1px 2px 0px, white 0px 0px 0px 1px;
  margin-bottom: 10px;
}

.table {
  background-color: #ffffff;
  margin: 4px;
}

table.tableproduct {
  float: left;
  width: 99%;
  margin: 4px;
}
```

Figure 35: CSS maincontent

HTML and PHP add new product, update product

```
<form action="" method="post" enctype="multipart/form-data" class="right2">
  <table border="1" cellspacing="0" class="table">
    <tr>
      <th colspan="2">Add new product</th>
    </tr>
    <tr>
      <td width="20%" class="col1">Id: </td>
      <td width="80%" class="col2"><input type="text" id="id" value="" name="id">
    </td>
    </tr>
    <tr>
      <td class="col1">Product name: </td>
      <td class="col2"><input type="text" name="name" id="fname" value=""></td>
    </tr>
    <tr>
      <td class="col1">Product image: </td>
      <td class="col2"><input type="file" name="image" id="image"></td>
    </tr>
    <tr>
      <td class="col1">Category:</td>
      <td class="col2">
        <select name="Cat" id="Cat">
          <?php
            $sql = "Select * From Category";
            $rows = query($sql);
            for($i=0; $i<count($rows); $i++)
            {
              <?
              <option value="<?=$rows[$i][0]?>"><?=$rows[$i][1]?></option>
              <?php
            }
            <?
          </select>
        </td>
    </tr>
    <tr>
      <td class="col1">Price: </td>
      <td class="col2"><input type="text" name="price" id="price" value=""></td>
    </tr>
    <tr>
      <td class="col1">Colorway: </td>
      <td class="col2"><input type="text" name="colorway" id="colorway" value=""></td>
    </tr>
    <tr>
      <td class="col1">Describe: </td>
      <td class="col2"><textarea rows="3" cols="200" name="description" id="description" value=""></textarea></td>
    </tr>
    <tr>
      <td class="col1"></td>
      <td class="col2"><input type="submit" value="Save" name="add"></td>
    </tr>
  </table>
</form>
```

```

<?php
$id = $_GET['updateid'];
$sql = "Select * From Product Where ProductId = '" . $id . "'";

$rows = query($sql);
?>
<form action="" method="post" enctype="multipart/form-data" class="right2">
<table border="1" cellspacing="0" class="table">
<tr>
<th colspan="2">Update product</th>
</tr>
<tr>
<td width="20%" class="col1">Id: </td>
<td width="80%" class="col2"><input type="text" id="idshow" value="{?-$rows[0][0]}" disabled>
<input type="hidden" id="id" value="{?-$rows[0][0]}" name="id">
</td>
</tr>
<tr>
<td class="col1">Product name: </td>
<td class="col2"><input type="text" name="name" id="fname" value="{?-$rows[0][1]}"></td>
</tr>
<tr>
<td class="col1">Product image: </td>
<td class="col2">
<input type="text" name="imageold" id="imageold" value="{?-$rows[0][2]}">
<input type="file" name="image" id="image">
</td>
</tr>
<tr>
<td class="col1">Category: </td>
<td class="col2">
<select name="cat" id="cat">
<?php
$sql = "Select * From Category";
$rowcats = query($sql);
for($i=0; $i<count($rowcats); $i++)
{
if($rows[0][4]==$rowcats[$i][0])
{
?>
<option value="{?-$rowcats[$i][0]}" selected="{?-$rowcats[$i][1]}"></option>
<?php
}
else
{
?>
<option value="{?-$rowcats[$i][0]}">{?-$rowcats[$i][1]}</option>
<?php
}
}
}
</select>
</td>
</tr>
<tr>
<td class="col1">Price: </td>
<td class="col2"><input type="text" name="price" id="price" value="{?-$rows[0][3]}"></td>
</tr>
<tr>
<td class="col1">Colorway: </td>
<td class="col2"><input type="text" name="colorway" id="colorway" value=""></td>
</tr>
<tr>
<td class="col1">Description: </td>
<td class="col2"><textarea rows="3" cols="200" id="description" name="description" id="description" value=""></textarea></td>
</tr>
<tr>
<td class="col1"></td>
<td class="col2"><input type="submit" value="Update" name="update"></td>
</tr>
</table>
</form>

```

Figure 36: HTML and PHP add new product, update product

Used to add new product and update product.

CSS add new product and update product

```
form.right2 {
  margin-top: 15px;
  margin-left: 15px;
  width: 25%;
  float: left;
  padding: 0;
  box-sizing: border-box;
  border-radius: 10px;
  box-shadow: 0px 0px 0px 0px, 0px 0px 0px 0px, 0px 0px 0px 0px;
  background-color: white;
  margin-bottom: 10px;
}
```

Figure 37: CSS add new product and update product

HTML and PHP delete product

```
if(isset($_GET['deleteid']))
{
    $id = $_GET['deleteid'];
    $sql = "delete from product where ProductId = '" . $id . "'";
    excecsql($sql);
}
```

Figure 38: HTML and PHP delete product

To delete the product's, click "Delete" and that product will be deleted from the database.

Function session_start

```
<?php session_start();
if (isset($_SESSION['username'] )){
    unset($_SESSION['username']);
    header("location: ../index.php");// xóa session login
}
?>
```

Figure 39: Function session_start

Used to log out the account and return the user to the website's homepage.

My SQL database

```

<?php
$server_username = "root"; // thông tin đăng nhập host
$server_password = ""; // mật khẩu, trong trường hợp này là trống
$server_host = "localhost"; // host là localhost
$database = 'storeman'; // database là website
$port='3306';
// Tạo kết nối đến database dùng mysqli_connect()
$conn = mysqli_connect($server_host,$server_username,$server_password,$database,$port) or die("không thể kết nối tới database");
// Thiết lập kết nối ủa chúng ta khi truy vấn là dạng UTF8 trong trường hợp dữ liệu là tiếng việt có dấu
mysqli_query($conn,"SET NAMES 'UTF8'");
function query($sql)
{
    //sử dụng biến toàn cục đã khai báo ở trên
    global $server_host;
    global $server_username;
    global $server_password;
    global $database;
    global $port;
    //mở kết nối tới cơ sở dữ liệu dùng mysqli với 5 tham số ở trên
    $conn = new mysqli($server_host, $server_username, $server_password, $database, $port);
    if($conn->connect_error)
    {
        //nếu như kết nối không thành công thì dừng chương trình
        echo "Connection fail<br>";
        //dừng chương trình
        die($conn->connect_error);
    }

    //chạy câu truy vấn lấy kết quả
    $result = $conn->query($sql);
    if(!$result)
    {
        //Nếu không có kết quả ($result=null) thì dừng chương trình
        echo "SQL execution fail <br>";
        die($conn->error);
    }

    //lấy tất cả các bản ghi từ kết quả
    $rows = mysqli_fetch_all($result);
    return $rows;
}
function execsql($sql)
{
    global $server_host;
    global $server_username;
    global $server_password;
    global $database;
    global $port;
    $conn = new mysqli($server_host, $server_username, $server_password, $database, $port);
    if($conn->connect_error)
    {
        //nếu như kết nối không thành công thì dừng chương trình
        echo "Connection fail<br>";
        //dừng chương trình
        die($conn->connect_error);
    }

    //chạy câu truy vấn lấy kết quả
    $result = $conn->query($sql);
    return $result;
}
?>

```

Figure 40: My SQL database

Chapter 3 – Testing

Testing case ID	Test case description	Test steps	Test data	Expected result	Actual result	Status
1	Choosing category	- Enter the homepage - Choose Balo		Show balo product	As expected	Pass
2	Register	- Enter the homepage - Click register - Enter username, password, re-password	- Username: adminn - Password: 808 - Re-Password:	Please enter the full information	As expected	Pass
3	Register	- Enter the homepage - Click register - Enter username, password, re-password	- Username: adminn - Password: 123 - Re-Password: 123	Congratulations, you have successfully registered	As expected	Pass
4	Login	- Enter the homepage - Click login - Enter username, password	- Username: linhhuong - Password: 123	Incorrect username or password	As expected	Pass
5	Login	- Enter the homepage - Click login - Enter username, password	- Username: linhhuong - Password: 882002	Login successfully	As expected	Pass
6	Search for product name	- Enter the homepage - Enter search	Hoodie	Show all product “Hoodie”	As expected	Pass

7	Search for product name	- Enter the homepage - Enter search	Pan	Show nothing	As expected	Pass
8	Add new product	- Enter the homepage - Click login - Enter username, password - Login - Information	- Id: 8 - Product name: Hoodie 8 - Product image: images/12.jfif - Category: Hoodie - Price: 560 - Colorway: Black - Description: LEVENTS LOVEYOU300K SPECIAL TEE/ WHITE COLOR: WHITE PAPER MATERIAL: 100% COTTON SIZE: 1/2/3/4	Successfully	As expected	Pas
9	Edit	- Enter admin - Choose edit action		Showing all product on information product.	As expected	Pass

Table 1: Test case

The vast majority of data interaction scenarios have been tested. In addition to the standard scenario, the validation instances were tested. The results were exactly what we had hoped for. The Delete, Import, and Update functions are all error-free and work within the parameters. They will not allow the data to be imported into the database as planned when importing or changing authentication data. Because much of the data is currently untestable, the author intends to test it in the near future. At this time, there are no errors. If users encounter any errors or problems while using the site, the author will fix them as soon as possible.

Conclusion

Because of this report, the author was able to learn the rules of website design. This is the first time she has built her own website, and she had to enlist the help of friends due to some web development errors. This is a very useful report that has been very helpful to the author.

Evaluation

Pros:

- Users can easily use data that has been fully transmitted.
- Interface that is both simple.
- Has basic features like add, edit, ...
- Simple, consistent layout.

Coins:

- No shopping cart
- No form of payment
- css is not eye-catching.
- The website sketchy.
- Mobile devices are not yet supported by the website.

References

He, Y., 2020. *Web Accessibility Standards: an overview for designers*. [Online]

Available at: <https://uxdesign.cc/web-accessibility-standards-an-overview-for-designers-1a4d39f2fe5e>