

Swinburne University of Technology

ASSIGNMENT PART 2 - GROUP 4

INDIVIDUAL REPORT

Student name: Chi Linh Hau

Student ID: 104177160

Hanoi, April 10th, 2023

I. INTRODUCTION

1. Objective of the report

This report's main goal is to analyse some of this dynamic website's key aspects, including its core functions, technical details and some innovative features relating to server-side embedded scripting in PHP and link up with MySQL server. Some recommendations for future improvements will also be discussed.

2. Outline of the report's structure

2.1. Website introduction: Introduce the website content and the aspects of the website that shall be discussed in this report.

2.2. Main functionalities of the website: Description and screenshots of the website functions that can take the data input from the user, validate, display invalid input stores and retrieve information from the database to show in the receipt page and manager's page.

2.3. Technical details on the website development: Explain the main functionalities listed in the previous part in detail.

2.4. Key innovative features (Enhancements): Highlight the key/innovative features and describe the enhancements.

2.5. My contribution: List and discuss my main contributions.

2.6. Reflection & Discussion

II. THE WEBSITE

1. Website introduction

The website that we built is a comprehensive online platform named TCAMV (first letter of all members' names) for e-learning that provides three different types of courses to users. The website has been designed to offer a wide range of courses to help users gain more skills and knowledge in three main fields: Computer Science, Business and Media & Communication. This website allows users to view all available courses, browse a course by entering a payment form and receive an e-receipt right after submitting the form. It also allows the manager to manager orders submitted from the payment form.

2. Main functionalities of the website

2.1. Use PHP to validate data sent in the “payment” HTML form to a new “process_order.php” page and provide user feedback through a new “fix_order.php” form if any errors

This function involves utilising PHP to validate the information submitted in the "payment" HTML form to a newly created "process_order.php" page. In case of any errors, user feedback is given through a separate "fix_order.php" form. This technique helps ensure that the information entered by users is accurate and complete, reducing the likelihood of errors during processing.

If the input data does not meet the format requirements, errors are returned to "fix_order.php", which displays the form with errors highlighted as shown below:

The screenshot shows the TCAMV website with a navigation bar (Home, Product, Enquire, About, Enhancements) and a search, cart, and user icon. A large pink error message box is displayed, listing various validation errors: First name is required, Lastname is required, You must enter your email, Phone number is required, Address is required, Postcode is required, Please select an option, Quantity is required, Please enter your comment, Please enter your credit card name, Credit Card number is required, Please enter your credit card expiry date, and Please enter your verification value. Below the error messages is an 'Error Feedbacks' section with input fields for First Name and Last Name.

2.2. Receipt after filling payment form

An order receipt.php page is returned to the user after all the inputs are validated with no errors, displaying all the order information, including the order ID and order status.

The screenshot shows the TCAMV website with a navigation bar (Home, Product, Enquire, About, Enhancements) and a search, cart, and user icon. The 'Payment Form' is displayed, containing input fields for First Name (Chi), Last Name (Hau), Email (Touhinhchi188@gmail.com), Phone number (0969251881), Delivery address (80 Day Tan), Postcode (1234), List of Products (Introduction to Programming - 560), Quantity (1), Comment fields (no), Name On Card (KHU LINH CHI), Credit Card Type (Visa selected), Credit Card Number (411222233334444), Credit Card Expiry Date (10/31), and Card Verification Value (123). A 'Check Out' button is located at the bottom of the form.

TCAMV

Home Product Enquire About Enhancements

TCAMV

Thank you for your order!

Your order

Order ID: 12
Order status: PENDING
Order time: 2023-04-09 11:54:23

Course name: Introduction to Programming - \$60
Quantity: 1
Comments: no

Total: **\$60**

Delivery Details
First name: Chi
Last name: Hau
Email: haulinhchi1881@gmail.com
Phone number: 0969251881
Delivery Address: 80 Duy Tan
Postcode: 1234

Payment Details
Name on card: *****
Credit card type: *****
Credit card number: *****
Credit card expiry date: *****
Card verification value: ****

[Back to Payment Form](#)

2.3. Create database and use PHP to store the order data in a server-side MySQL database table "orders"

Some functions are used to set up a database and PHP is utilised to save order data in a MySQL database table called "orders" on the server-side. Storing order information in a database enables easy access and management of data, as well as the ability to perform various operations on the data such as sorting, filtering, and searching.

phpMyAdmin

Server: Linux MySQL (MySQL) Database: s104177160_db Table: orders

Showing rows 0 - 7 (8 total. Query took 0.0017 seconds)

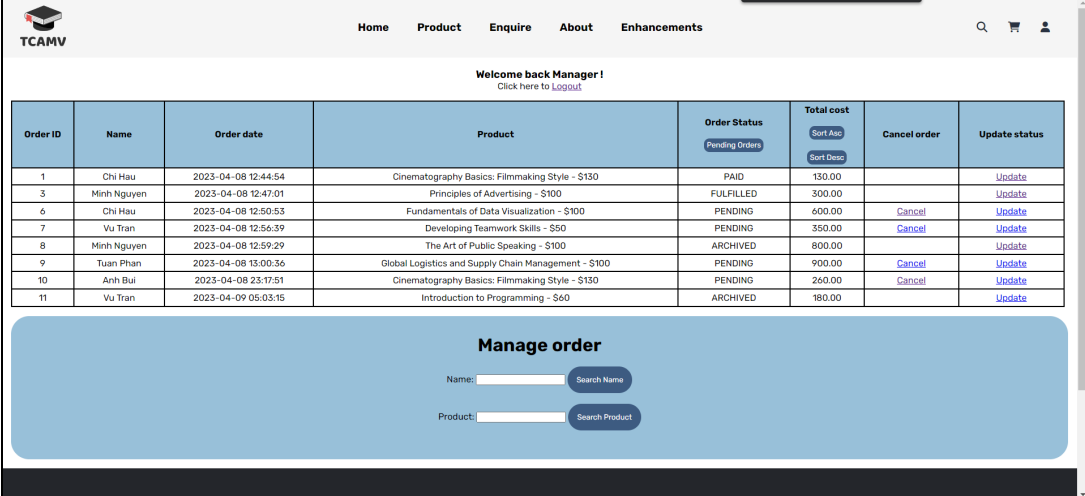
SELECT * FROM "orders"

Extra options

	order_id	first_name	last_name	email	phone_number	delivery_address	postcode	product	quantity	comments	name
<input type="checkbox"/>	1	Chi	Hau	haulinhchi1881@gmail.com	0969251881	80 Duy Tan	1881	Cinematography Basics: Filmmaking Style - \$130	1	no	HAU L
<input type="checkbox"/>	3	Minh	Nguyen	duyminhnguyenvu@gmail.com	0912892992	02 Duong Khue	5678	Principles of Advertising - \$100	3		NGUYI DUY M
<input type="checkbox"/>	6	Chi	Hau	haulinhchi1881@gmail.com	0969251881	80 Duy Tan	8888	Fundamentals of Data Visualization - \$100	6	hello	HAU L
<input type="checkbox"/>	7	Vu	Tran	tranhuuyvu18082004@gmail.com	0843230040	02 Duong Khue	1001	Developing Teamwork Skills - \$50	7		TRAN
<input type="checkbox"/>	8	Minh	Nguyen	duyminhnguyenvu@gmail.com	0122345668	80 Duy Tan	1099	The Art of Public Speaking - \$100	8	hola	NGUYI DUY M
<input type="checkbox"/>	9	Tuan	Phan	phansytuan@gmail.com	0912345678	02 Duong Khue	9999	Global Logistics and Supply Chain	9		PHAN

2.4. Order management

This function involves creating a web page that enables a manager to query orders, display the results in an HTML table, and update the status of an order. Each query presents the order ID, date, product details (including the total cost), customer first and last names and order status. This page also allows the manager to choose to display all orders, orders for a specific customer or product, orders that are pending, or orders sorted by total cost in ascending and descending order. Additionally, the manager can update an order's status by clicking on a link or button beside the order in the table, and can cancel a pending order through the page. This functionality can help the manager to efficiently manage orders and monitor the progress of orders.



The screenshot shows the TCAMV Manager interface. At the top, there is a navigation bar with links: Home, Product, Enquire, About, and Enhancements. Below the navigation bar, a welcome message reads "Welcome back Manager!" with a link to "Click here to Logout". The main content area features a table with the following columns: Order ID, Name, Order date, Product, Order Status, Total cost, Cancel order, and Update status. The table contains 11 rows of order data. Below the table, there is a "Manage order" section with two search filters: "Name:" and "Product:", each with a corresponding search button.

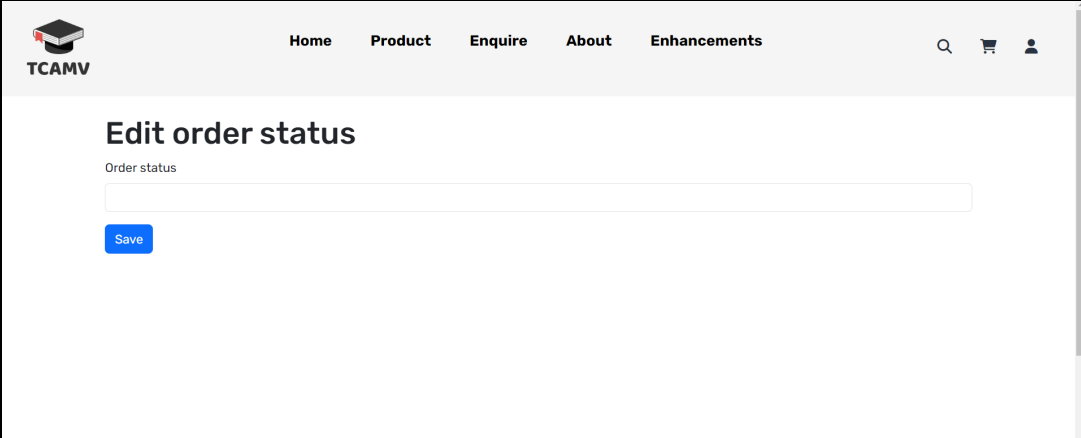
Order ID	Name	Order date	Product	Order Status	Total cost	Cancel order	Update status
1	Chi Hau	2023-04-08 12:44:54	Cinematography Basics: Filmmaking Style - \$130	PAID	130.00		Update
3	Minh Nguyen	2023-04-08 12:47:01	Principles of Advertising - \$100	FULFILLED	300.00		Update
6	Chi Hau	2023-04-08 12:50:53	Fundamentals of Data Visualization - \$100	PENDING	600.00	Cancel	Update
7	Vu Tran	2023-04-08 12:56:39	Developing Teamwork Skills - \$50	PENDING	350.00	Cancel	Update
8	Minh Nguyen	2023-04-08 12:59:29	The Art of Public Speaking - \$100	ARCHIVED	800.00		Update
9	Tuan Phan	2023-04-08 13:00:36	Global Logistics and Supply Chain Management - \$100	PENDING	900.00	Cancel	Update
10	Anh Bui	2023-04-08 23:17:51	Cinematography Basics: Filmmaking Style - \$130	PENDING	260.00	Cancel	Update
11	Vu Tran	2023-04-09 05:03:15	Introduction to Programming - \$60	ARCHIVED	180.00		Update

Manage order

Name: [Search Name](#)

Product: [Search Product](#)

Update an order's status:



TCAMV

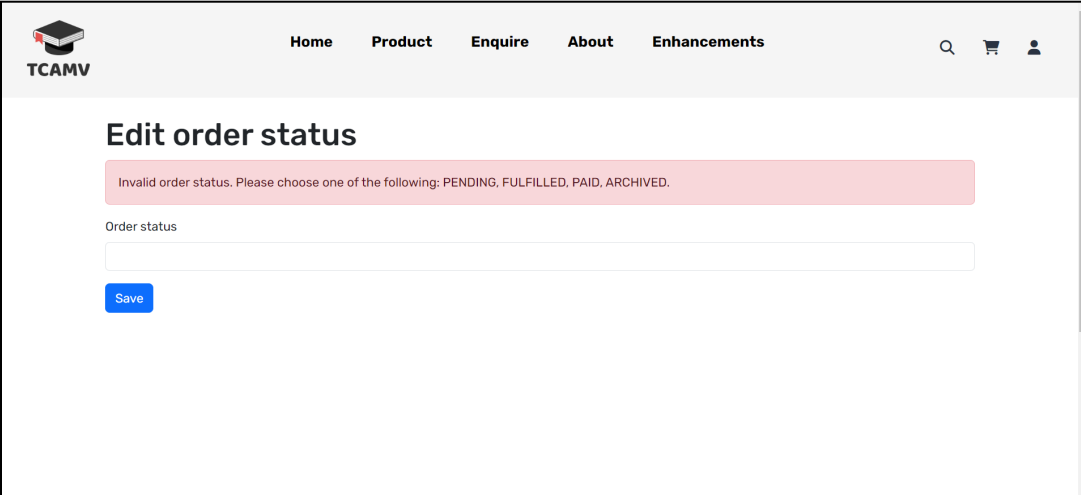
Home Product Enquire About Enhancements

Search Cart User

Edit order status

Order status

Save



TCAMV

Home Product Enquire About Enhancements

Search Cart User

Edit order status

Invalid order status. Please choose one of the following: PENDING, FULFILLED, PAID, ARCHIVED.

Order status

Save

3. Technical details on the website development

3.1. Use PHP to include common webpage code

PHP offers us methods to organise and reuse our web application code. In Assignment Part 2, the web pages are modified in such a way that the general static HTML elements like the menu, header, and footer are written in common text files. These text files, named with an .inc extension (menu.inc, header.inc, footer.inc), are then inserted back into

the web pages using "include" statements. This replaces the corresponding sections of HTML in the main pages.

3.2. Use PHP to validate data sent in the “payment” HTML form to a new “process_order.php” page and provide user feedback through a new “fix_order.php” form if any errors

This function involves implementing server-side data format checking for a payment form in PHP. The form action is set to "process_order.php", and HTML5 form data validation is disabled. All values received by "process_order.php" are sanitised to remove leading/trailing spaces, backslashes, and HTML control characters.

```
function sanitise_input($data){  
    $data = trim($data);  
    $data = stripslashes($data);  
    $data = htmlspecialchars($data);  
    return $data;  
}
```

The data format rules, including those for customer details and credit cards, are checked before an order is written to the orders table. The credit card type must be Visa, Mastercard, or American Express.

```
if (isset($_POST["type"])) {  
    $type = sanitise_input($_POST["type"]);  
    $type = $_POST["type"];  
    $_SESSION['type'] = $type;  
} else {  
    $errors["type"] = "Please enter your credit card type";  
}
```

The credit card number must be exactly 15 or 16 digits, match the selected card type and must match the requirements: Visa card start

number is 4, Mastercard start number is between the range of 51 to 55 and American Express start number is 34 or 37. To validate these rules, the "isset" and "!preg_match" functions is used.

```
if (isset($_POST["cre"])) {  
    $cre = sanitise_input($_POST["cre"]);  
    $cre = $_POST["cre"];  
    $_SESSION['cre'] = $cre;  
  
    if ($cre == "") {  
        $errors['cre'] = "Credit Card number is required.";  
    } else if (!preg_match("/^[0-9]*$/", $cre)) {  
        $errors['cre'] = "Credit Card number only accepts integers.";  
    } else if ($type == "Visa") {  
        // Visa should start with 4 and be 16 digits long  
        if (!preg_match("/^4\d{15}$/", $cre)) {  
            $errors['cre'] = "Invalid card number (must start with 4 and contains 16 digits).";  
        }  
    } else if ($type == "Mastercard") {  
        // Mastercard should start with 51-55 and be 16 digits long  
        if (!preg_match("/^5[1-5]\d{14}$/", $cre)) {  
            $errors['cre'] = "Invalid card number (must start with 51 to 55 and contains 16 digits).";  
        }  
    } else if ($type == "American Express") {  
        // American express should start with 34/37 and be 15 digits long  
        if (!preg_match("/^3[4-7]\d{13}$/", $cre)) {  
            $errors['cre'] = "Invalid card number (must start with 34 to 37 and contains 15 digits).";  
        }  
    }  
} else {  
    $errors["cre"] = "Please enter your credit card number";  
}
```

If the input data does not meet the format requirements, errors are returned to "fix_order.php", which displays the form with errors highlighted.

If the input data is validated correctly, the total cost of the order is calculated, and the order is stored in the "order"s table using a mysql query.

```
if (!isset($processed)) {  
    $processed = true; // Set flag variable  
    $cost = preg_replace('/^[^0-9]/', '', $product);  
    $total_cost = $cost * $quan;  
}
```

```
require_once("settings.php");
$conn = @mysqli_connect($host, $user, $pwd, $sql_db);
if (!$conn) {
    echo "<p>Database connection failure</p>";
} else {
    $sql_table = "orders";
    $query = "INSERT INTO $sql_table (first_name, last_name, email, phone_number,
    delivery_address, postcode, product,
    quantity, comments, name_on_card, credit_card_type, credit_card_number,
    credit_card_expiry, cvv, order_cost, order_time)
    values ('$fname', '$lname', '$email', '$phone', '$address', '$postcode',
    '$product', '$quan', '$comment', '$name',
    '$type', '$cre', '$cry', '$ver', '$total_cost', '$date_time')";
```

3.3. Receipt after filling payment form

All submitted details are stored in the \$_SESSION['values'] array (A session is a way to store information, in variables, to be used across multiple pages).

```
$_SESSION['values'] = array(
    'product' => $product,
    'fname' => $fname,
    'lname' => $lname,
    'email' => $email,
    'phone' => $phone,
    'address' => $address,
    'postcode' => $postcode,
    'type' => $type,
    'quan' => $quan,
    'name' => $name,
    'comment' => $comment,
    'total_cost' => $total_cost,
);
```

The last inserted ID are stored as the order ID. After that, the order status and order time of the order ID are retrieved from the table using a query. All the information stored in the record is then printed out in the “receipt.php” page.

```
if ($conn->query($query) === TRUE) {  
    // Get the last inserted ID as the order ID  
    $order_id = $conn->insert_id;  
    $_SESSION['order_id'] = $order_id;  
    $_SESSION['order_time'] = $date_time;  
    // Retrieve the order status from the orders table  
    $query = "SELECT order_status FROM orders WHERE order_id = $order_id";  
    $result = $conn->query($query);  
    if ($result->num_rows > 0) {  
        $row = $result->fetch_assoc();  
        $order_status = $row['order_status'];  
        $_SESSION['order_status'] = $order_status;  
    }  
}
```

3.4. Create database and use PHP to store the order data in a server-side MySQL database table “orders”

A table named “orders” is created on the MySQL server. This table will record the data sent from the “payment.php” form. This data contains information on the customer, product and payment details as specified in the Assignment Part 1 and some additional information such as order ID, cost, time and status. The table is created by using the “CREATE TABLE” command in the orders.sql file and managed on phpMyAdmin.

```
CREATE TABLE orders (  
  order_id INT AUTO_INCREMENT PRIMARY KEY,  
  first_name VARCHAR(50) NOT NULL,  
  last_name VARCHAR(50) NOT NULL,  
  email VARCHAR(100) NOT NULL,  
  phone_number VARCHAR(20) NOT NULL,  
  delivery_address VARCHAR(200) NOT NULL,  
  postcode VARCHAR(20) NOT NULL,  
  product VARCHAR(100) NOT NULL,  
  quantity INT NOT NULL,  
  comments VARCHAR(500),  
  name_on_card VARCHAR(100) NOT NULL,  
  credit_card_type VARCHAR(20) NOT NULL,  
  credit_card_number VARCHAR(16) NOT NULL,  
  credit_card_expiry VARCHAR(5) NOT NULL,  
  cvv VARCHAR(5) NOT NULL,  
  order_cost DECIMAL(10,2) NOT NULL,  
  order_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,  
  order_status ENUM('PENDING', 'FULFILLED', 'PAID', 'ARCHIVED') NOT NULL DEFAULT 'PENDING'  
);
```

The “payment.php” page form action being set to “process_order.php” to ensure all the data submitted are validated and checked using PHP before storing them to the database. If all the input data meet the format requirements, they will be store in the “orders” table by using the “INSERT INTO” command:

```
$query = "INSERT INTO $sql_table (first_name, last_name, email, phone_number, delivery_address,  
  postcode, product,  
  quantity, comments, name_on_card, credit_card_type, credit_card_number, credit_card_expiry, cvv,  
  order_cost, order_time)  
  values ('$fname', '$lname', '$email', '$phone', '$address', '$postcode', '$product', '$quan',  
  '$comment', '$name',  
  '$type', '$cre', '$cry', '$ver', '$total_cost', '$date_time')";
```

3.5. Create a settings.php file to store database connection variables

Using a “settings.php” script can help enable the website to be ported from a development environment to a testing environment.

```
settings.php
1  <?php
2      $host = "feenix-mariadb.swin.edu.au";
3      $user = "s104177160";
4      $pwd = "191004";
5      $sql_db = "s104177160_db";
6  ?>
```

3.6. Order management

The function described in the report is a web page that provides a user interface for a manager to query, display and update orders. The web page allows the manager to search for orders based on different criteria such as customer name, product, order pending status and sort them by total cost.

```
$sql = "SELECT * FROM orders";
if (isset($_POST["sname"])) {
    $sname = $_POST["sname"];
    $sql .= " WHERE first_name LIKE '%$sname%' OR last_name LIKE '%$sname%'";
}
if (isset($_POST["sproduct"])) {
    $sproduct = $_POST["sproduct"];
    $sql .= " WHERE product LIKE '%$sproduct%' ";
}

if (isset($_POST["pending"])) {
    $pending = $_POST["pending"];
    $sql .= " WHERE order_status='$pending'";
}
if (isset($_POST["sortascending"])) {
    $sortascending = $_POST["sortascending"];
    $sql .= " ORDER BY order_cost";
}
```

The web page presents the search results in an HTML table that displays order number, order date, product details including cost, customer's first and last names, and order status. The web page also allows the manager to update the order status by clicking on a link or button next to the order in the table. The status can be changed from pending, fulfilled, paid or

archived. Additionally, the manager can cancel pending orders via the web page.

```
<tbody>
  <?php foreach ($orders as $order): ?>
    <tr>
      <td>
        <?php echo $order['order_id']; ?>
      </td>
      <td>
        <?php echo $order['first_name'] . " " . $order['last_name']; ?>
      </td>
      <td>
        <?php echo $order['order_time']; ?>
      </td>
      <td>
        <?php echo $order['product']; ?>
      </td>
      <td>
        <?php echo $order['order_status']; ?>
      </td>
      <td>
        <?php echo $order['order_cost']; ?>
      </td>
```

```
      <td>
        <?php
          if ($order['order_status'] == "PENDING") {
            echo "<a href='cancel.php?id=" . $order['order_id'] . "'>Cancel</a>";
          }
        ?>
      </td>
      <td>
        <?php
          echo "<a href='update.php?id=" . $order['order_id'] . "'>Update</a>";
        ?>
      </td>
    </tr>
  <?php endforeach; ?>
</tbody>
```

4. Key innovative features (Enhancements)

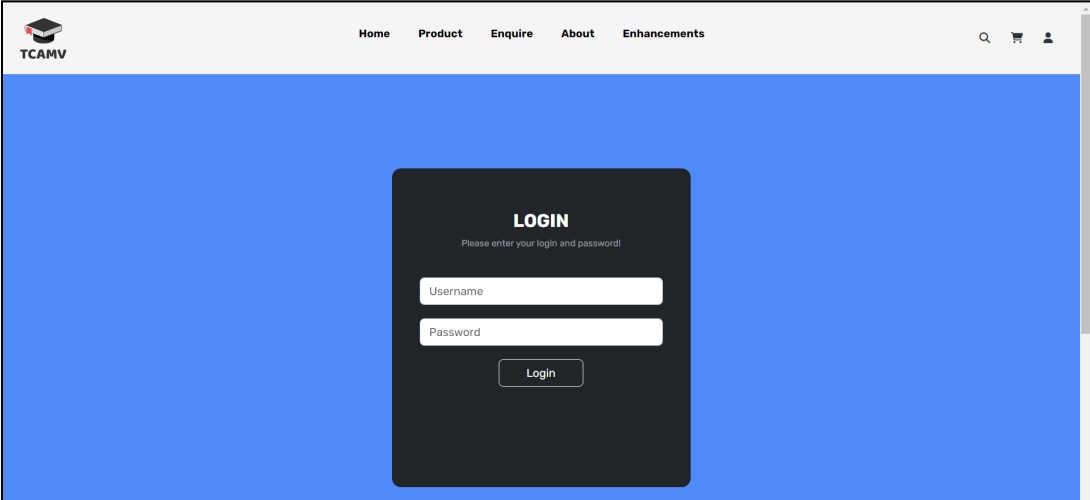
4.1. Login before accessing manager.php page and logout

A database is created to store the username and password information of the manager. Anyone attempting to access the manager page must enter

the proper username and password in order to do so. The corresponding username and password values are saved in the session variables using `$_SESSION["username"] = $row ['username']` and `$_SESSION["pwd"] = $row ['pwd']` after the form is submitted and the user's credentials have been verified. Using `isset($_SESSION["username"])` function, the session variables are checked on subsequent pages. The `header("location:manager.php")` is utilised to redirect the user to the “manager.php” page if the “username” session variable is set. When the user logs out, all session data associated with the current session ID will be removed by calling the `session_destroy()` function. Finally, the user is redirected to the login page using the `header()` function.

Furthermore, a layer of protection to the page to prevent anyone from entering via the URL is added. If someone tries to enter the page in this way, they will be immediately redirected to the login page to validate their identity. Additionally, the password stored in the database is encrypted using MD5, which adds an extra layer of security against hackers trying to bypass the password value.

Login:



The screenshot shows a web application interface. The top navigation bar includes the TCAMV logo and links for Home, Product, Enquire, About, and Enhancements. The main content area has a blue background. In the center, there is a dark gray login form with the title "LOGIN" and the instruction "Please enter your login and password!". The form contains two input fields labeled "Username" and "Password", and a "Login" button.

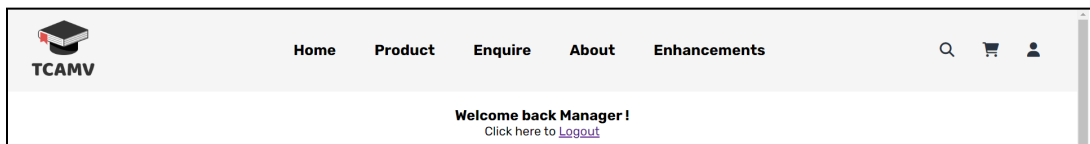
```
<?php
session_start();
require_once("settings.php");
$conn = @mysqli_connect($host, $user, $pwd, $sql_db);
?>
```

```
<?php
if (isset($_POST['login'])) {
    $username = $_POST['username'];
    $pwd = $_POST['pwd'];

    $select = mysqli_query($conn, "SELECT * FROM manager WHERE username =
    '$username' AND pwd = '$pwd'");
    $row = mysqli_fetch_array($select);

    if (is_array($row)) {
        $_SESSION["username"] = $row['username'];
        $_SESSION["pwd"] = $row['pwd'];
    } else {
        echo "<div class='alert alert-danger' role='alert'>Invalid info</div>";
    }
}
if (isset($_SESSION["username"])) {
    header("location:manager.php");
}
?>
```

Logout:



```
<?php
session_start();
if (isset($_SESSION["username"])) {
    echo "<br><h3 style='text-align:center;'>Welcome back Manager !</h3>";
    echo "<p style='text-align:center;'>Click here to <a href='logout.php'>Logout</a></p>";
} else {
    header("location:login.php");
}
?>
```



```
<?php
    session_start();
    if(session_destroy()){
        header("location: login.php");
    }
?>
```

4.2. Sort total cost in descending order

The enhancement involves adding the ability to select and sort the table in descending order based on the total cost column. This feature would enable the manager to quickly identify and prioritise orders with higher costs. It would also enhance the user experience by providing an intuitive and user-friendly way to sort and organise the data.

III. MY CONTRIBUTION

My contributions to the project were coding the manager page and implementing the "sort total cost in descending order" function. Additionally, I also created and managed the database, which is a critical component of the website's functionality.

In terms of the "manager.php" page, it is connected to the database using the mysqli query. After that, there are four queries being used for searching and managing orders, which are searching first or last name, product name, pending status and sorting total cost in descending or ascending order. They can operate using isset(\$_POST[""]) function, which can check whether a variable is empty and whether the variable is declared. Then, the declared data is assigned to a particular variable so it can be called in the \$sql query.

```
$sql = "SELECT * FROM orders";
if (isset($_POST["sname"])) {
    $sname = $_POST["sname"];
    $sql .= " WHERE first_name LIKE '%$sname%' OR last_name LIKE '%$sname%'";
}
if (isset($_POST["sproduct"])) {
    $sproduct = $_POST["sproduct"];
    $sql .= " WHERE product LIKE '%$sproduct%' ";
}

if (isset($_POST["pending"])) {
    $pending = $_POST["pending"];
    $sql .= " WHERE order_status='$pending'";
}
if (isset($_POST["sortascending"])) {
    $sortascending = $_POST["sortascending"];
    $sql .= " ORDER BY order_cost";
}
```

Moreover, in order to display all the orders and the specific search results, a table with the foreach loop is applied. Each time the loop iterates, the value of the current array element is assigned to \$order and the array pointer is moved by one, until it reaches the last array element.

```
<tbody>
  <?php foreach ($orders as $order): ?>
    <tr>
      <td>
        <?php echo $order['order_id']; ?>
      </td>
      <td>
        <?php echo $order['first_name'] . "&nbsp;" . $order['last_name']; ?>
      </td>
      <td>
        <?php echo $order['order_time']; ?>
      </td>
      <td>
        <?php echo $order['product']; ?>
      </td>
      <td>
        <?php echo $order['order_status']; ?>
      </td>
      <td>
        <?php echo $order['order_cost']; ?>
      </td>
    </tr>
  </?php ?>
</tbody>
```

```
<td>
    <?php
        if ($order['order_status'] == "PENDING") {
            echo "<a href='cancel.php?id=" . $order['order_id'] . "'>Cancel</a>";
        }
    ?>
</td>
<td>
    <?php
        echo "<a href='update.php?id=" . $order['order_id'] . "'>Update</a>";
    ?>
</td>
</tr>
<?php endforeach; ?>
</tbody>
```

The manager can cancel the order by taking the ID from the table row where it is clicked and delete it from the database. The manager will remain in the same manager page with the result after deleting the record, without being redirected to any other page.

```
<?php
if (!isset($_GET['id'])) {
    header('Location: manager.php');
}
$id = $_GET['id'];
require_once("settings.php");
$conn = @mysqli_connect($host, $user, $pwd, $sql_db) or die('Unable to connect');
$sql = "DELETE FROM orders WHERE order_id = '$id'";
$result = mysqli_query($conn, $sql);
var_dump($result);
if ($result) {
    header('Location: manager.php');
    exit;
} else {
    echo "<p>Error</p>";
}

mysqli_close($conn);
?>
```

The manager can also update the order by taking the ID from the table row where it is clicked. The order status need to be entered in the input field and match valid order status before it can be saved and updated to the server.

```
if (isset($_POST['order_status'])) {  
    $order_status = sanitise_input($_POST['order_status']);  
    if (!$order_status) {  
        $errors .= 'New order status is required.';  
    } else {  
        // Define a regular expression pattern to match valid order statuses  
        $pattern = '/^(PENDING|FULFILLED|PAID|ARCHIVED)$/i';  
  
        // Check if the input matches the pattern  
        if (!preg_match($pattern, $order_status)) {  
            $errors .= 'Invalid order status. Please choose one of the following: PENDING, FULFILLED, PAID, ARCHIVED.';  
        }  
    }  
}
```

IV. REFLECTION & DISCUSSION

1. Reflection

One of our team's strengths was maintaining effective communication throughout the project. We made sure that we were all on the same page by having regular meetings and updating each other on our progress using Github. This allowed us to quickly identify and address any issues that arose, leading to a more efficient and streamlined project.

Another strength of our team was our ability to collaborate effectively. We all had different skills and strengths, and we leveraged them to complete different parts of the project. This allowed us to make sure that the final product was of high quality, and met all the necessary requirements.

Regarding our team's weakness, we did encounter some technical challenges during the development phase. We discovered that we should be more proactive in seeking help and support from our colleagues and instructors when faced with technical difficulties.

2. Discussion

In any web development project, privacy and security are crucial issues that must be considered. In this online course website project, privacy and security were important considerations throughout the development process. One of the ways to ensure privacy was by implementing a secure login system that allowed users to access their account information and purchase courses securely.

Another aspect of privacy and security was in the handling of sensitive data, such as credit card information. To ensure that this data was protected, the website utilised encryption to encrypt all data transmissions between the website and the user's browser. Additionally, the website only stored the last four digits of the user's credit card number, thereby minimising the risk of data breaches.

V. CONCLUSION

In conclusion, this report has provided an in-depth analysis of a dynamic website's core functions and technical details, particularly related to its server-side embedded scripting in PHP and link up with MySQL server. The report has also highlighted some innovative features of the website and discussed potential areas for future improvement.