

DAYANANDA SAGAR UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING SCHOOL OF
ENGINEERING

DAYANANDA SAGAR UNIVERSITY

KUDLU GATE

BANGALORE – 560068



PROJECT REPORT

ON

Apparel Shopping Website

Cloud Application Development (20CS3603)

6th SEMESTER

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by

P Prithvi Reddy – (ENG20CS0243)

Parag Mandal – (ENG20CS0246)

Pranav Padaki – (ENG20CS0262)

Sai Shreenidhi Satish-(ENG20C0308)

Under the supervision of

Dr. Mouleeswaran S K

Associate Professor Dept. of CSE, DSU

DAYANANDA SAGAR UNIVERSITY

School of Engineering, Kudlu Gate, Bangalore-560068



CERTIFICATE

This is to certify that Mr. P Prithvi Reddy, Mr. Parag Mandal, Mr. Pranav Padaki and Mr. Sai Shreenidhi Satish bearing USN ENG20CS0243, ENG20CS0246, ENG20CS0262 and ENG20CS0308 respectively have satisfactorily completed their Mini Project as prescribed by the University for the SIXTH-semester of B. Tech in Computer Science & Engineering during the year 2022-2023 at the School of Engineering, Dayananda Sagar University, Bangalore.

Date: _____

Signature of the faculty in-charge

Max Marks	Marks Obtained

Signature of Chairman

Department of Computer Science & Engineering

DECLARATION

I hereby declare that the work presented in this project entitled -
“Apparel Shopping Website”, has been carried out by me and it has not been submitted for the award of any degree, diploma or the project of any other college or university.

P Prithvi Reddy – (ENG20CS0243)
Parag Mandal – (ENG20CS0246)
Pranav Padaki – (ENG20CS0262)
Sai Shreenidhi Satish – (ENG20CS0308)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of a task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts with success.

I am very much thankful to our guide **Dr. Mouleeswaran S K**, for providing help and suggestions in the completion of this project successfully.

We have received a great deal of guidance and we wish to thank all who have directly or indirectly helped us complete our project work.

P Prithvi Reddy – (ENG20CS0243)
Parag Mandal – (ENG20CS0246)
Pranav Padaki – (ENG20CS0262)
Sai Shreenidhi Satish – (ENG20CS0308)

TABLE OF CONTENTS

CHAPTER	CONTENT	PAGE
	Abstract	1
1	Introduction	2
2	Software and Hardware Requirements	
2.1	--Software Requirements	3
2.2	--Hardware Requirements	3
3	Design	4
4	Directory Structure	6
5	Code	7
6	Output	26
7	Conclusion	31
8	References	32

ABSTRACT

The Apparel Shopping Website project is a web-based application designed to provide an online platform for clothing items. It is built using HTML, JavaScript, CSS, PHP, and MySQL, and is executed on the XAMPP server. The website offers several features, such as user authentication, product categorization, shopping cart functionality, wishlist management, and order placement.

The project's primary goal is to provide an easy-to-use and accessible platform for shopping, allowing users to browse and purchase clothing items from the comfort of their homes. The website is designed to be responsive and user-friendly, with a clean and modern interface that enables smooth navigation.

Furthermore, the project includes an admin panel that allows administrators to manage users, admins, products, and order statuses. This feature ensures that the website is secure and well-maintained, with administrators having complete control over the platform.

Overall, the Apparel Shopping Website project is a comprehensive and well-executed application that offers an excellent user experience for shopping enthusiasts. It utilizes the XAMPP server environment to provide a stable and secure platform for its users and administrators.

1. INTRODUCTION

The popularity of online shopping has significantly grown in recent years, prompting the development of diverse e-commerce platforms. The Apparel Shopping Website project aims to cater to a specific niche within the e-commerce market by providing a platform dedicated to apparel shopping. Apparel shopping website offers numerous advantages, including affordability, sustainability, and the opportunity to find unique and vintage clothing items.

This project utilizes XAMPP as the server environment for executing the web application. XAMPP is a widely-used software package that provides a local development environment for creating and testing web applications. It includes components such as Apache HTTP Server, MySQL database, PHP, and Perl, simplifying the setup and configuration process. By leveraging XAMPP, the Apparel Shopping Website can be seamlessly executed on a local server, allowing for efficient development and testing.

The primary objective of our Shopping Website project is to create a user-friendly web application that replicates the experience of shopping in a physical apparel store. Users can explore a wide range of clothing items divided into categories such as Men, Women, and Kids. The website offers features like user registration and login to personalize the shopping experience and allow users to maintain their cart and wishlist.

Our Shopping Website project also emphasizes the significance of administrative functionalities. An admin panel is implemented using XAMPP to manage users, admins, products, and order statuses efficiently. Administrators can add or delete users and admins, manage product listings, and track and update order statuses. This enhances the overall management and control of the website.

Throughout this project report, we will delve into the various components and functionalities of the Shopping Website. We will discuss the technologies employed, including HTML, JavaScript, CSS, PHP, and MySQL, as well as the utilization of XAMPP as the server environment. We will outline the project's design principles, development process, challenges faced, and future scope for enhancements.

Overall, our Shopping Website project aims to provide an engaging and convenient online shopping experience for users while offering an efficient management system for administrators, utilizing XAMPP as the server environment. The following sections will provide an in-depth analysis of each aspect of the project, offering insights into its design, implementation, and potential for further development.

2. HARDWARE AND SOFTWARE REQUIREMENTS

2.1. Hardware Requirements:

- **Processor:** A modern processor with multiple cores, such as Intel Core i5 or AMD Ryzen 5, is recommended to handle the server load efficiently.
- **RAM:** At least 4GB of RAM is recommended for smooth operation, but higher capacities may be required depending on the expected user traffic and database size.
- **Storage:** Sufficient storage space is necessary to accommodate the web application files, images, and database. A minimum of 10GB of available disk space is recommended.
- **Network:** A stable internet connection is essential for hosting the website and allowing users to access the apparel shopping platform seamlessly.

2.2. Software Requirements:

- **Operating System:** The Apparel Shopping Website can be deployed on various operating systems supported by XAMPP, including Windows, macOS, or Linux. Choose an OS based on your familiarity and compatibility with XAMPP.
- **XAMPP:** XAMPP is a software package that includes the Apache HTTP Server, MySQL database, PHP, and Perl. Make sure to download and install the latest version of XAMPP suitable for your operating system.
- **Web Browser:** The website should be compatible with popular web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari. Ensure that you have the latest versions of these browsers installed for testing and compatibility checks.
- **Text Editor:** You will need a text editor or an integrated development environment (IDE) to edit and manage the website's source code. Popular options include Visual Studio Code, Sublime Text, or PHPStorm.

3. DESIGN

Our Shopping Website offers a range of features and functionalities designed to enhance the overall user experience and streamline the apparel shopping process. This section outlines the key features of the platform and what users can expect when using it.

3.1. User Registration and Authentication:

- Users can create an account by registering with their email address and setting a password.
- Secure authentication mechanisms are implemented to ensure the privacy and security of user information.
- Registered users have access to personalized profiles, allowing them to manage their information and track their orders.

3.2. Product Categorization and Browsing:

- Clothing items are categorized into sections such as Men, Women, and Kids for easy navigation.
- Users can browse through the available products within each category, view detailed product information, and see product images.

3.3. Shopping Cart Management:

- Users can add desired items to their shopping cart, which allows them to keep track of selected products before proceeding to checkout.
- The shopping cart displays a summary of the added items, including their prices and quantities, providing a convenient overview.

3.4. Wishlist Functionality:

- Users can create and manage a wishlist of desired clothing items.
- Wishlist functionality enables users to save items for future reference, making it easier to revisit and purchase them later.

3.5. Order Placement and Checkout:

- When ready to make a purchase, users can proceed to the checkout process.
- Users provide their address, phone number, and recipient's name for accurate delivery.
- Payment options are available, allowing users to choose their preferred mode of payment, such as credit card, PayPal, or other supported methods.

3.6. Order Tracking and History:

- Users can track the status of their placed orders, from pending to completed.
- Order history is maintained, allowing users to review past purchases and order details.

3.7. Admin Panel:

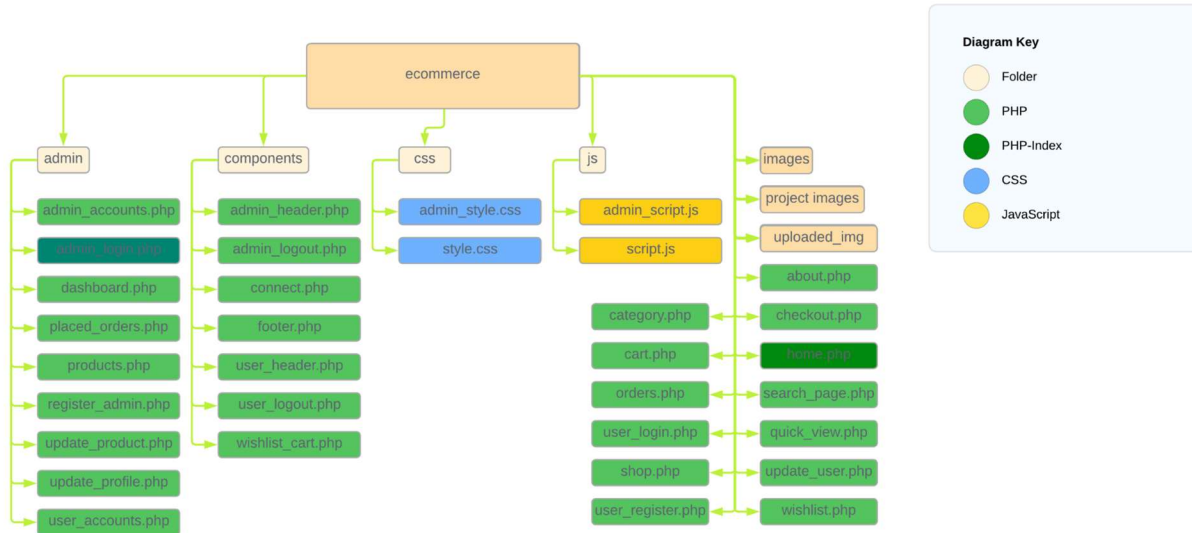
- Administrators have access to a dedicated admin panel to manage various aspects of the website.
- Admins can add or delete users and admins, ensuring smooth user management.
- Product management features enable admins to add, edit, or remove products from the website.
- Order management functionalities allow admins to update order statuses, marking them as pending or completed.

3.8. Responsive and User-Friendly Interface:

- The platform is designed with a responsive layout, ensuring compatibility with various screen sizes and devices.
- The user interface is intuitive and visually appealing, offering a seamless and enjoyable shopping experience.

The Shopping Website encompasses these features and functionalities to provide users with a comprehensive and user-friendly platform for apparel shopping. Whether it's browsing through categories, managing shopping carts and wishlists, placing orders, or tracking order statuses, the platform strives to deliver an engaging and convenient apparel shopping experience.

4. DIRECTORY STRUCTURE



5. CODE

5.1. admin_accounts.php:

```
<?php
include '../components/connect.php';
session_start();
$admin_id = $_SESSION['admin_id'];
if(!isset($admin_id)){
    header('location:admin_login.php');
}

if(isset($_GET['delete'])){
    $delete_id = $_GET['delete'];
    $delete_admins = $conn->prepare("DELETE FROM `admins` WHERE id = ?");
    $delete_admins->execute([$delete_id]);
    header('location:admin_accounts.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>admin accounts</title>

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">

    <link rel="stylesheet" href="../css/admin_style.css">
</head>
<body>

<?php include '../components/admin_header.php'; ?>

<section class="accounts">

    <h1 class="heading">admin accounts</h1>
    <div class="box-container">

        <div class="box">
```

```

        <a href="register_admin.php" class="option-btn">Register New Admin</a>
    </div>

    <?php
        $select_accounts = $conn->prepare("SELECT * FROM `admins`");
        $select_accounts->execute();
        if($select_accounts->rowCount() > 0){
            while($fetch_accounts = $select_accounts->fetch(PDO::FETCH_ASSOC)){
                ?>
                <div class="box">
                    <p> Admin ID : <span><?= $fetch_accounts['id']; ?></span> </p>
                    <p> Admin Username : <span><?= $fetch_accounts['name']; ?></span> </p>
                    <div class="flex-btn">
                        <a href="admin_accounts.php?delete=<?= $fetch_accounts['id']; ?>"
                        onclick="return confirm('delete this account?')" class="delete-btn">delete</a>
                        <?php
                            if($fetch_accounts['id'] == $admin_id){
                                echo '<a href="update_profile.php" class="option-btn">update</a>';
                            }
                        ?>
                    </div>
                </div>
            }
        }
        <?php
        }
        }else{
            echo '<p class="empty">no accounts available!</p>';
        }
    ?>

    </div>

</section>
<script src="../../js/admin_script.js"></script>

</body>
</html>

```

5.2. admin_login.php:

```
<?php
include '../components/connect.php';
session_start();
if(isset($_POST['submit'])){
    $name = $_POST['name'];
    $name = filter_var($name, FILTER_SANITIZE_STRING);
    $pass = sha1($_POST['pass']);
    $pass = filter_var($pass, FILTER_SANITIZE_STRING);
    $select_admin = $conn->prepare("SELECT * FROM `admins` WHERE name = ?
AND password = ?");
    $select_admin->execute([$name, $pass]);
    $row = $select_admin->fetch(PDO::FETCH_ASSOC);
    if($select_admin->rowCount() > 0){
        $_SESSION['admin_id'] = $row['id'];
        header('location:dashboard.php');
    }else{
        $message[] = 'Incorrect username or password!';
    }
}
?>
```

```
<!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="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.1.1/css/all.min.css">
```

```

    <link rel="stylesheet" href="../css/admin_style.css">
```

```
</head>
```

```
<body>
```

```
<?php
if(isset($message)){
    foreach($message as $message){
        echo '
        <div class="message">
            <span>'.$message.'</span>
            <i class="fas fa-times" onclick="this.parentElement.remove();"></i>
```

```
        </div>
    };
}
?>
```

```
<section class="form-container">
```

```
    <form action="" method="post">
        <h3>login now</h3>
        <input type="text" name="name" required placeholder="Username" maxlength="20"
class="box" oninput="this.value = this.value.replace(/\s/g, "")">
        <input type="password" name="pass" required placeholder="Password"
maxlength="20" class="box" oninput="this.value = this.value.replace(/\s/g, "")">
        <input type="submit" value="login now" class="btn" name="submit">
    </form>
```

```
</section>
```

```
</body>
```

```
</html>
```

5.3. placed_orders.php:

```
<?php
include '../components/connect.php';
session_start();
$admin_id = $_SESSION['admin_id'];
if(!isset($admin_id)){
    header('location:admin_login.php');
}

if(isset($_POST['update_payment'])){
    $order_id = $_POST['order_id'];
    $payment_status = $_POST['payment_status'];
    $payment_status = filter_var($payment_status, FILTER_SANITIZE_STRING);
    $update_payment = $conn->prepare("UPDATE `orders` SET payment_status = ?
    WHERE id = ?");
    $update_payment->execute([$payment_status, $order_id]);
    $message[] = 'payment status updated!';
}

if(isset($_GET['delete'])){
    $delete_id = $_GET['delete'];
    $delete_order = $conn->prepare("DELETE FROM `orders` WHERE id = ?");
    $delete_order->execute([$delete_id]);
    header('location:placed_orders.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>placed orders</title>

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
    awesome/6.1.1/css/all.min.css">

    <link rel="stylesheet" href="../css/admin_style.css">

</head>
<body>

<?php include '../components/admin_header.php'; ?>
```



```

<section class="orders">
<h1 class="heading">placed orders</h1>
<div class="box-container">

    <?php
        $select_orders = $conn->prepare("SELECT * FROM `orders`");
        $select_orders->execute();
        if($select_orders->rowCount() > 0){
            while($fetch_orders = $select_orders->fetch(PDO::FETCH_ASSOC)){
                ?>
                <div class="box">
                    <p> Date : <span><?= $fetch_orders['placed_on']; ?></span> </p>
                    <p> Name : <span><?= $fetch_orders['name']; ?></span> </p>
                    <p> Mobile Number : <span><?= $fetch_orders['number']; ?></span> </p>
                    <p> Address : <span><?= $fetch_orders['address']; ?></span> </p>
                    <p> Total products : <span><?= $fetch_orders['total_products']; ?></span> </p>
                    <p> Total price : <span>$<?= $fetch_orders['total_price']; ?></span> </p>
                    <p> Payment method : <span><?= $fetch_orders['method']; ?></span> </p>
                    <form action="" method="post">
                        <input type="hidden" name="order_id" value="<?= $fetch_orders['id']; ?>">
                        <select name="payment_status" class="select">
                            <option selected disabled><?= $fetch_orders['payment_status']; ?></option>
                            <option value="pending">Pending</option>
                            <option value="completed">Completed</option>
                        </select>
                        <div class="flex-btn">
                            <input type="submit" value="update" class="option-btn"
name="update_payment">
                            <a href="placed_orders.php?delete=<?= $fetch_orders['id']; ?>" class="delete-btn"
onclick="return confirm('delete this order?');">Delete</a>
                        </div>
                    </form>
                </div>
            }
        }
        else{
            echo '<p class="empty">no orders placed yet!</p>';
        }
    ?>

</div>
</section>
</section>
<script src="../js/admin_script.js"></script>
</body>
</html>

```

5.4. register_admin.php:

```
<?php
include '../components/connect.php';
session_start();
$admin_id = $_SESSION['admin_id'];
if(!isset($admin_id)){
    header('location:admin_login.php');
}

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

    $name = $_POST['name'];
    $name = filter_var($name, FILTER_SANITIZE_STRING);
    $pass = sha1($_POST['pass']);
    $pass = filter_var($pass, FILTER_SANITIZE_STRING);
    $cpass = sha1($_POST['cpass']);
    $cpass = filter_var($cpass, FILTER_SANITIZE_STRING);

    $select_admin = $conn->prepare("SELECT * FROM `admins` WHERE name = ?");
    $select_admin->execute([$name]);

    if($select_admin->rowCount() > 0){
        $message[] = 'username already exist!';
    }else{
        if($pass != $cpass){
            $message[] = 'confirm password not matched!';
        }else{
            $insert_admin = $conn->prepare("INSERT INTO `admins`(name, password)
VALUES(?,?)");
            $insert_admin->execute([$name, $cpass]);
            $message[] = 'new admin registered successfully!';
        }
    }
}

?>

<!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>register admin</title>
```

```

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.1.1/css/all.min.css">

<link rel="stylesheet" href="../css/admin_style.css">

</head>
<body>

<?php include '../components/admin_header.php'; ?>

<section class="form-container">

    <form action="" method="post">
        <h3>register now</h3>
        <input type="text" name="name" required placeholder="enter your username"
maxlength="20" class="box" oninput="this.value = this.value.replace(/s/g, "")">
        <input type="password" name="pass" required placeholder="enter your password"
maxlength="20" class="box" oninput="this.value = this.value.replace(/s/g, "")">
        <input type="password" name="cpass" required placeholder="confirm your
password" maxlength="20" class="box" oninput="this.value = this.value.replace(/s/g,
)">
        <input type="submit" value="register now" class="btn" name="submit">
    </form>

</section>
<script src="../js/admin_script.js"></script>

</body>
</html>

```

5.5. connect.php:

```

<?php

$db_name = 'mysql:host=localhost;dbname=shop_db';
$user_name = 'root';
$user_password = '';

$conn = new PDO($db_name, $user_name, $user_password);

?>

```

5.6. cart.php:

```
<?php include 'components/connect.php';
session_start();
if(isset($_SESSION['user_id'])){
    $user_id = $_SESSION['user_id'];
}else{
    $user_id = "";
    header('location:user_login.php');
};

if(isset($_POST['delete'])){
    $cart_id = $_POST['cart_id'];
    $delete_cart_item = $conn->prepare("DELETE FROM `cart` WHERE id = ?");
    $delete_cart_item->execute([$cart_id]);
}

if(isset($_GET['delete_all'])){
    $delete_cart_item = $conn->prepare("DELETE FROM `cart` WHERE user_id = ?");
    $delete_cart_item->execute([$user_id]);
    header('location:cart.php');
}

if(isset($_POST['update_qty'])){
    $cart_id = $_POST['cart_id'];
    $qty = $_POST['qty'];
    $qty = filter_var($qty, FILTER_SANITIZE_STRING);
    $update_qty = $conn->prepare("UPDATE `cart` SET quantity = ? WHERE id = ?");
    $update_qty->execute([$qty, $cart_id]);
    $message[] = 'cart quantity updated';
}

?>

<!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>Cart</title>
    <style>
        body {
            background-image: url("images/orangeBG.avif");
            background-size: cover;
            background-repeat: no-repeat;
```

```

        background-position: center center;
    }
</style>

<!-- font awesome cdn link -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.1.1/css/all.min.css">

<!-- custom css file link -->
<link rel="stylesheet" href="css/style.css">

</head>
<body>

<?php include 'components/user_header.php'; ?>

<section class="products shopping-cart">

    <h3 class="heading">shopping cart</h3>

    <div class="box-container">

        <?php
            $grand_total = 0;
            $select_cart = $conn->prepare("SELECT * FROM `cart` WHERE user_id = ?");
            $select_cart->execute([$user_id]);
            if($select_cart->rowCount() > 0){
                while($fetch_cart = $select_cart->fetch(PDO::FETCH_ASSOC)){
                    ?>
                    <form action="" method="post" class="box">
                        <input type="hidden" name="cart_id" value="<?= $fetch_cart['id']; ?>">
                        <a href="quick_view.php?pid=<?= $fetch_cart['pid']; ?>" class="fas fa-eye"></a>
                        
                        <div class="name"><?= $fetch_cart['name']; ?></div>
                        <div class="flex">
                            <div class="price"><?= $fetch_cart['price']; ?></div>
                            <input type="number" name="qty" class="qty" min="1" max="99"
onkeypress="if(this.value.length == 2) return false;" value="<?= $fetch_cart['quantity'];
?>">
                            <button type="submit" class="fas fa-edit" name="update_qty"></button>
                        </div>
                        <div class="sub-total"> sub total : <span><?= $sub_total = ($fetch_cart['price'] *
$fetch_cart['quantity']); ?></span> </div>
                        <input type="submit" value="delete item" onclick="return confirm('delete this from
cart?');" class="delete-btn" name="delete">
                    </form>
                }
            }
        ?>
    </div>
</section>

```

```

<?php
$grand_total += $sub_total;
}
}else{
    echo '<p class="empty">your cart is empty</p>';
}
?>
</div>

<div class="cart-total">
    <p>grand total : <span>$<?= $grand_total; ?>/-</span></p>
    <a href="shop.php" class="option-btn">continue shopping</a>
    <a href="cart.php?delete_all" class="delete-btn <?= ($grand_total > 1)?":'disabled';
?>" onclick="return confirm('delete all from cart?');">delete all item</a>
    <a href="checkout.php" class="btn <?= ($grand_total > 1)?":'disabled'; ?>">proceed
to checkout</a>
</div>

</section>

<?php include 'components/footer.php'; ?>

<script src="js/script.js"></script>

</body>
</html>

```

5.7. checkout.php:

```
<?php include 'components/connect.php';
session_start();
if(isset($_SESSION['user_id'])){
    $user_id = $_SESSION['user_id'];
}else{
    $user_id = "";
    header('location:user_login.php');
};

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

    $name = $_POST['name'];
    $name = filter_var($name, FILTER_SANITIZE_STRING);
    $number = $_POST['number'];
    $number = filter_var($number, FILTER_SANITIZE_STRING);
    $email = $_POST['email'];
    $email = filter_var($email, FILTER_SANITIZE_STRING);
    $method = $_POST['method'];
    $method = filter_var($method, FILTER_SANITIZE_STRING);
    $address = 'flat no. ' . $_POST['flat'] . ', ' . $_POST['street'] . ', ' . $_POST['city'] . ', ' .
    $_POST['state'] . ', ' . $_POST['country'] . ' - ' . $_POST['pin_code'];
    $address = filter_var($address, FILTER_SANITIZE_STRING);
    $total_products = $_POST['total_products'];
    $total_price = $_POST['total_price'];

    $check_cart = $conn->prepare("SELECT * FROM `cart` WHERE user_id = ?");
    $check_cart->execute([$user_id]);

    if($check_cart->rowCount() > 0){

        $insert_order = $conn->prepare("INSERT INTO `orders`(user_id, name, number,
        email, method, address, total_products, total_price) VALUES(?,?,?,?,?,?,?)");
        $insert_order->execute([$user_id, $name, $number, $email, $method, $address,
        $total_products, $total_price]);

        $delete_cart = $conn->prepare("DELETE FROM `cart` WHERE user_id = ?");
        $delete_cart->execute([$user_id]);

        $message[] = 'order placed successfully!';
    }else{
        $message[] = 'your cart is empty';
    }
}
?>
```

```

<!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>Checkout</title>
  <style>
    body {
      background-image: url("images/orangeBG.avif");
      background-size: cover;
      background-repeat: no-repeat;
      background-position: center center;
    }
  </style>

  <!-- font awesome cdn link -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.1.1/css/all.min.css">

  <!-- custom css file link -->
  <link rel="stylesheet" href="css/style.css">

</head>
<body>

<?php include 'components/user_header.php'; ?>

<section class="checkout-orders">

  <form action="" method="POST">

    <h3>your orders</h3>

    <div class="display-orders">
      <?php
        $grand_total = 0;
        $cart_items[] = "";
        $select_cart = $conn->prepare("SELECT * FROM `cart` WHERE user_id = ?");
        $select_cart->execute([$user_id]);
        if($select_cart->rowCount() > 0){
          while($fetch_cart = $select_cart->fetch(PDO::FETCH_ASSOC)){
            $cart_items[] = $fetch_cart['name'].' ( '.$fetch_cart['price'].' x ' .
$fetch_cart['quantity'].' ) - ' ;
            $total_products = implode($cart_items);
            $grand_total += ($fetch_cart['price'] * $fetch_cart['quantity']);

```



```

?>
    <p> <?= $fetch_cart['name']; ?> <span>(<?= '$'.$fetch_cart['price'].'- x '.
$fetch_cart['quantity']; ?>)</span> </p>
    <?php
    }
    }else{
        echo '<p class="empty">your cart is empty!</p>';
    }
?>
    <input type="hidden" name="total_products" value="<?= $total_products; ?>">
    <input type="hidden" name="total_price" value="<?= $grand_total; ?>" value="">
    <div class="grand-total">grand total : <span>$<?= $grand_total; ?>/-
</span></div>
</div>

<h3>place your orders</h3>

<div class="flex">
    <div class="inputBox">
        <span>your name :</span>
        <input type="text" name="name" placeholder="enter your name" class="box"
maxlength="20" required>
    </div>
    <div class="inputBox">
        <span>your number :</span>
        <input type="number" name="number" placeholder="enter your number"
class="box" min="0" max="9999999999" onkeypress="if(this.value.length == 10) return
false;" required>
    </div>
    <div class="inputBox">
        <span>your email :</span>
        <input type="email" name="email" placeholder="enter your email" class="box"
maxlength="50" required>
    </div>
    <div class="inputBox">
        <span>payment method :</span>
        <select name="method" class="box" required>
            <option value="cash on delivery">cash on delivery</option>
            <option value="credit card">credit card</option>
            <option value="paytm">paytm</option>
            <option value="paypal">paypal</option>
        </select>
    </div>
    <div class="inputBox">
        <span>address line 01 :</span>
        <input type="text" name="flat" placeholder="e.g. flat number" class="box"

```

```

maxlength="50" required>
    </div>
    <div class="inputBox">
        <span>address line 02 :</span>
        <input type="text" name="street" placeholder="e.g. street name" class="box"
maxlength="50" required>
    </div>
    <div class="inputBox">
        <span>city :</span>
        <input type="text" name="city" placeholder="e.g. mumbai" class="box"
maxlength="50" required>
    </div>
    <div class="inputBox">
        <span>state :</span>
        <input type="text" name="state" placeholder="e.g. maharashtra" class="box"
maxlength="50" required>
    </div>
    <div class="inputBox">
        <span>country :</span>
        <input type="text" name="country" placeholder="e.g. India" class="box"
maxlength="50" required>
    </div>
    <div class="inputBox">
        <span>pin code :</span>
        <input type="number" min="0" name="pin_code" placeholder="e.g. 123456"
min="0" max="999999" onkeypress="if(this.value.length == 6) return false;"
class="box" required>
    </div>
</div>

    <input type="submit" name="order" class="btn <?= ($grand_total > 1)?":'disabled';
?>" value="place order">

</form>

</section>

<?php include 'components/footer.php'; ?>

<script src="js/script.js"></script>

</body>
</html>

```

5.8. orders.php:

```
<?php include 'components/connect.php';
session_start();
if(isset($_SESSION['user_id'])){
    $user_id = $_SESSION['user_id'];
}else{
    $user_id = "";
};
?>

<!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>Orders</title>
    <style>
        body {
            background-image: url("images/orangeBG.avif");
            background-size: cover;
            background-repeat: no-repeat;
            background-position: center center;
        }
    </style>

    <!-- font awesome cdn link -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.1.1/css/all.min.css">

    <!-- custom css file link -->
    <link rel="stylesheet" href="css/style.css">

</head>
<body>

<?php include 'components/user_header.php'; ?>

<section class="orders">

    <h1 class="heading">placed orders</h1>

    <div class="box-container">

        <?php
```

```

        if($user_id == ""){
            echo '<p class="empty">please login to see your orders</p>';
        }else{
            $select_orders = $conn->prepare("SELECT * FROM `orders` WHERE user_id =
?");
            $select_orders->execute([$user_id]);
            if($select_orders->rowCount() > 0){
                while($fetch_orders = $select_orders->fetch(PDO::FETCH_ASSOC)){
                    ?>
                    <div class="box">
                        <p>placed on : <span><?=$fetch_orders['placed_on']; ?></span></p>
                        <p>name : <span><?=$fetch_orders['name']; ?></span></p>
                        <p>email : <span><?=$fetch_orders['email']; ?></span></p>
                        <p>number : <span><?=$fetch_orders['number']; ?></span></p>
                        <p>address : <span><?=$fetch_orders['address']; ?></span></p>
                        <p>payment method : <span><?=$fetch_orders['method']; ?></span></p>
                        <p>your orders : <span><?=$fetch_orders['total_products']; ?></span></p>
                        <p>total price : <span>$<?=$fetch_orders['total_price']; ?>/</span></p>
                        <p> payment status : <span style="color:<?php if($fetch_orders['payment_status'] ==
'pending'){ echo 'red'; }else{ echo 'green'; }; ?>"><?=$fetch_orders['payment_status'];
?></span> </p>
                    </div>
                    <?php
                    }
                    }else{
                        echo '<p class="empty">no orders placed yet!</p>';
                    }
                }
            }
        }
    }
    ?>

</div>

</section>

<?php include 'components/footer.php'; ?>

<script src="js/script.js"></script>

</body>
</html>

```

5.9. user_login.php:

```
<?php include 'components/connect.php';
session_start();
if(isset($_SESSION['user_id'])){
    $user_id = $_SESSION['user_id'];
}else{
    $user_id = "";
};

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

    $email = $_POST['email'];
    $email = filter_var($email, FILTER_SANITIZE_STRING);
    $pass = sha1($_POST['pass']);
    $pass = filter_var($pass, FILTER_SANITIZE_STRING);

    $select_user = $conn->prepare("SELECT * FROM `users` WHERE email = ? AND
password = ?");
    $select_user->execute([$email, $pass]);
    $row = $select_user->fetch(PDO::FETCH_ASSOC);

    if($select_user->rowCount() > 0){
        $_SESSION['user_id'] = $row['id'];
        header('location:home.php');
    }else{
        $message[] = 'incorrect username or password!';
    }
}
?>

<!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>
    <style>
        body {
            background-image: url("images/orangeBG.avif");
            background-size: cover;
            background-repeat: no-repeat;
            background-position: center center;
        }
    </style>
```

```
<!-- font awesome cdn link -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.1.1/css/all.min.css">

<!-- custom css file link -->
<link rel="stylesheet" href="css/style.css">

</head>
<body>

<?php include 'components/user_header.php'; ?>

<section class="form-container">

    <form action="" method="post">
        <h3>login now</h3>
        <input type="email" name="email" required placeholder="enter your email"
maxlength="50" class="box" oninput="this.value = this.value.replace(/s/g, "")">
        <input type="password" name="pass" required placeholder="enter your password"
maxlength="20" class="box" oninput="this.value = this.value.replace(/s/g, "")">
        <input type="submit" value="login now" class="btn" name="submit">
        <p>don't have an account?</p>
        <a href="user_register.php" class="option-btn">register now</a>
    </form>

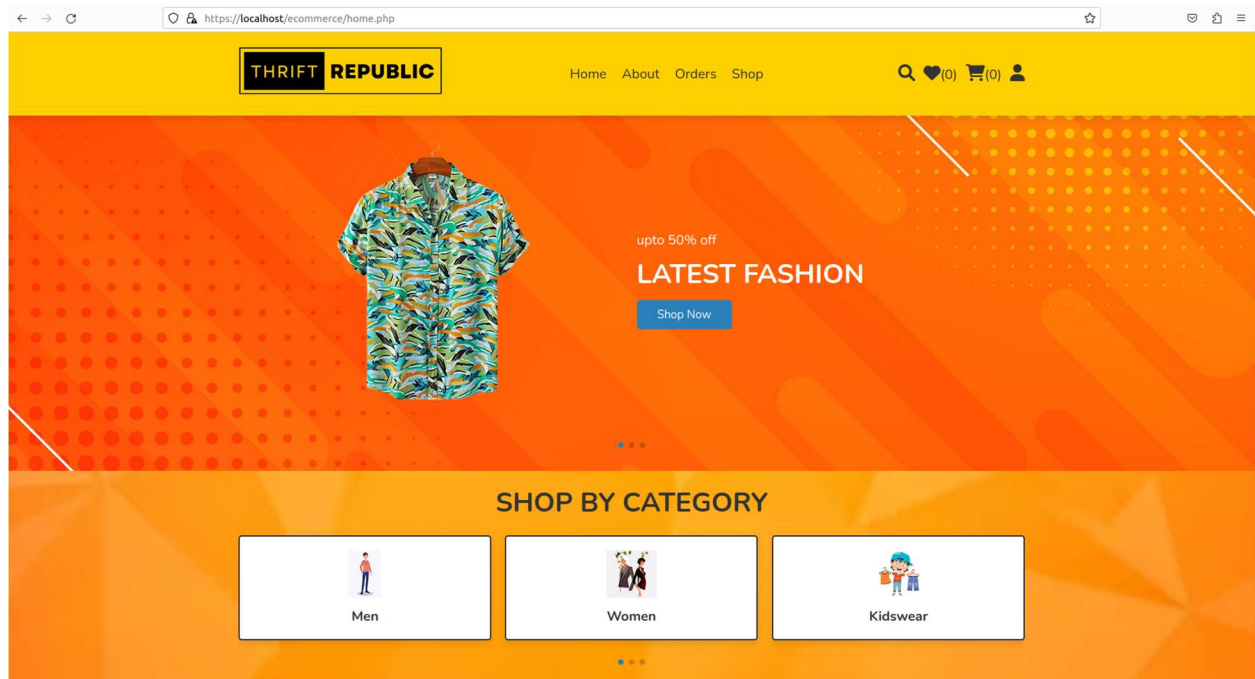
</section>
<?php include 'components/footer.php'; ?>

<script src="js/script.js"></script>

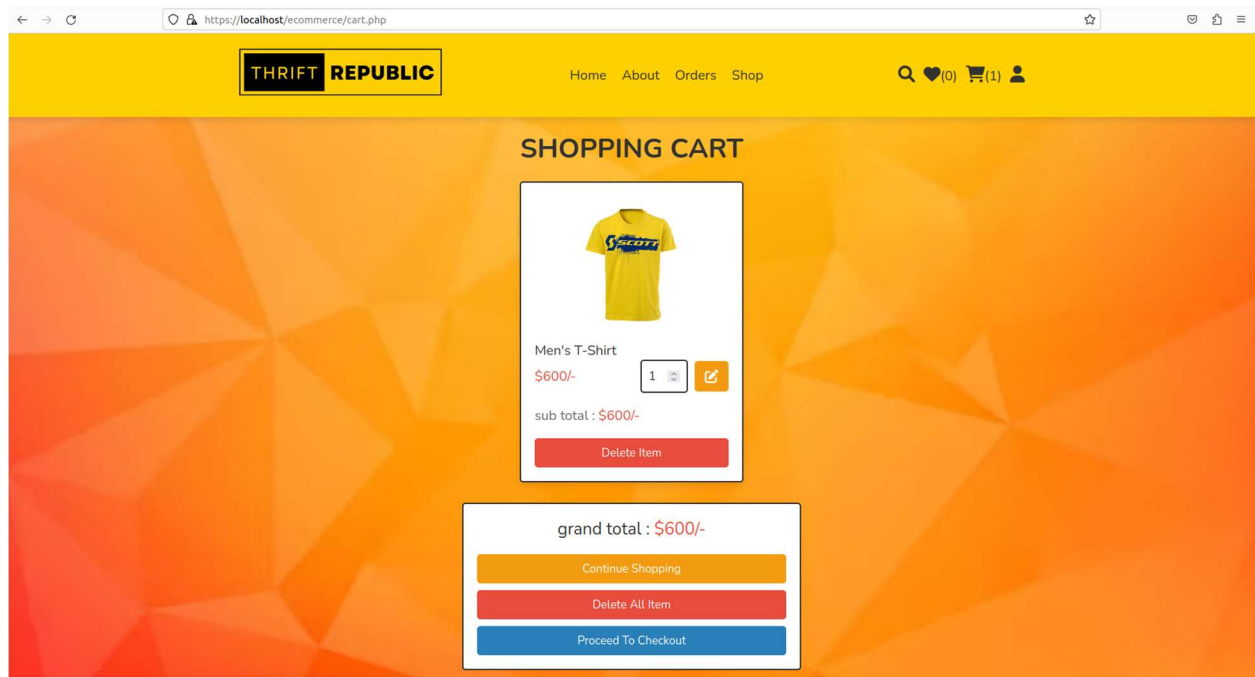
</body>
</html>
```

6. OUTPUT

6.1. User's Home Page



6.2. User Cart



6.3. User Checkout Page

THRIFT REPUBLIC

Home About Orders Shop

Men's T-Shirt (\$600/- x 1)

grand total : \$600/-

PLACE YOUR ORDERS

your name : Prithvi

your number : 1234567891

your email : lordprithvi@testmail.net

payment method : paytm

address line 01 : ABCD

address line 02 : XYZ

city : Bengaluru

state : Karnataka

country : India

pin code : 560060

Place Order

Team-X Quick Links Extra Links

6.4. Database

phpMyAdmin

Recent Favorites

Database: shop_db

Structure SQL Search Query Export Import Operations Privileges Routines Events

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> admins	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> cart	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> orders	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> products	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> users	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> wishlist	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
6 tables	Sum	1	InnoDB	latin1_swedish_ci	96.0 KiB	0 B

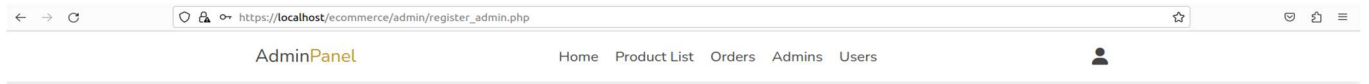
Check all With selected:

Print Data dictionary

Create new table

Table name: Number of columns: 4 Create

6.5. Admin Registration



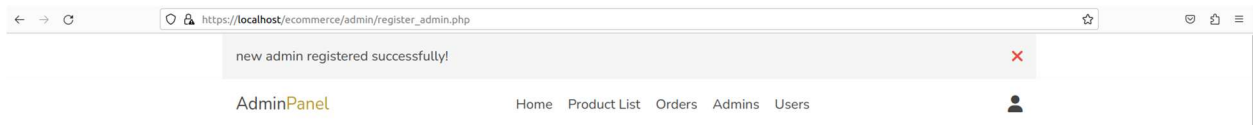
REGISTER NOW

parag

.....

.....

Register Now



REGISTER NOW

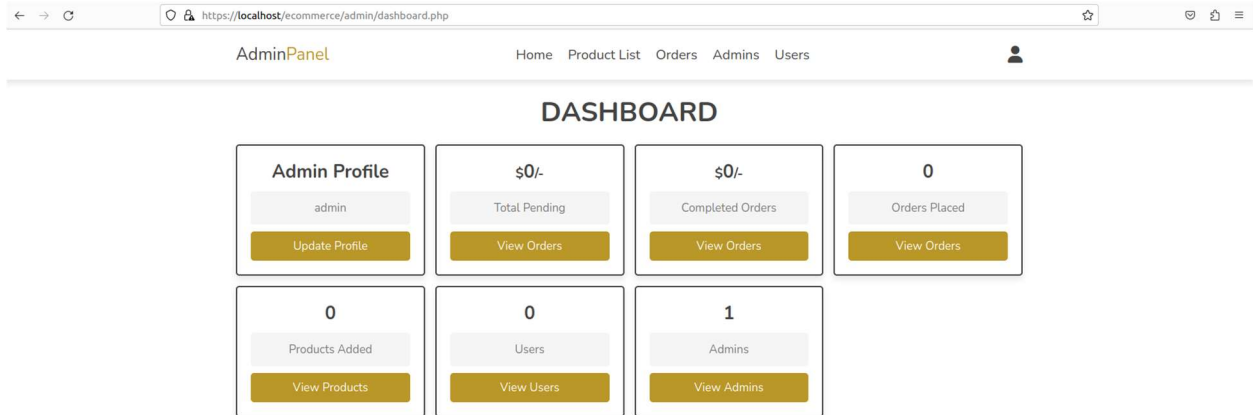
parag

.....

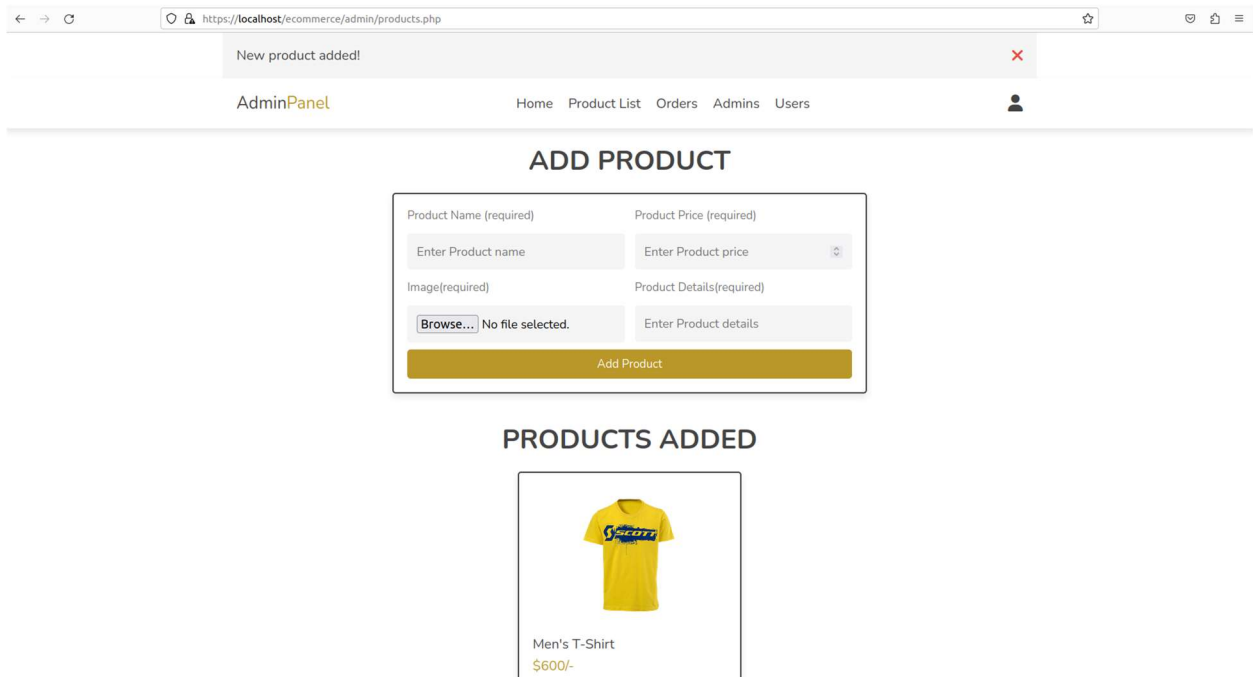
confirm your password

Register Now

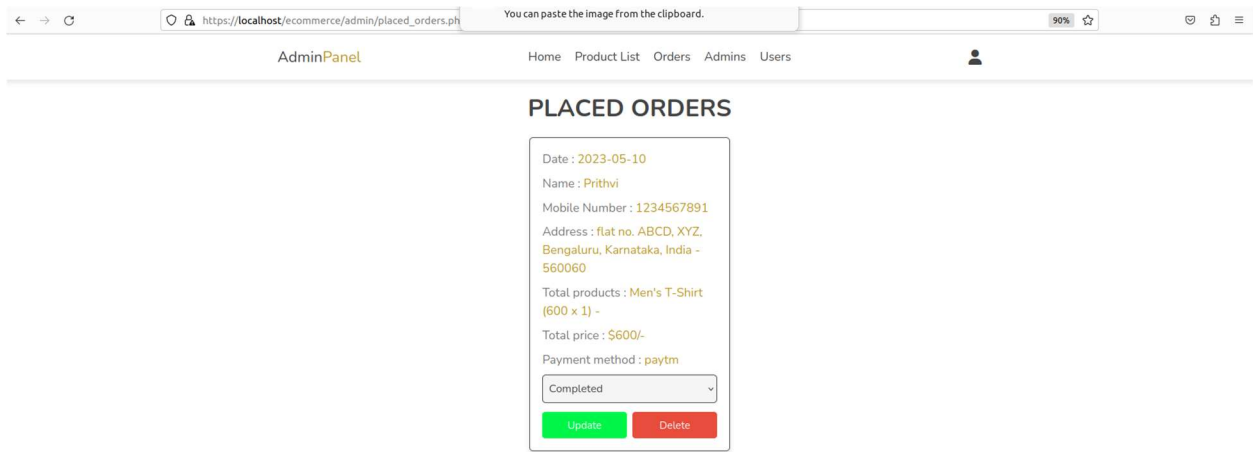
6.6. Admin Dashboard



6.7. Adding Products from the Admin dashboard



6.8. Admin Order Approval Page



7. CONCLUSION

In conclusion, the development of the Apparel Website project has showcased the successful integration of various technologies to create a robust and user-friendly online platform for apparel shopping. Through the use of HTML, CSS, JavaScript, PHP, and MySQL, we have achieved a dynamic and interactive website that offers a seamless and enjoyable shopping experience for users.

The implementation of user registration and authentication features ensures secure access to personalized profiles, allowing users to manage their information and track their orders. The categorization and browsing functionalities make it easy for users to navigate through different apparel categories, view detailed product information, and browse product images.

The shopping cart management feature allows users to add desired items to their cart and keep track of selected products before proceeding to checkout. Additionally, the wishlist functionality enables users to create and manage their wishlist of desired apparel items, making it easier to revisit and purchase them later.

The order placement and checkout process streamline the purchasing journey, with users providing their address, phone number, and recipient's name for accurate delivery. The incorporation of order tracking and history functionalities allows users to track the status of their placed orders, from pending to completed.

From an administrative perspective, the inclusion of an admin panel empowers administrators to manage users, admins, and products efficiently. They can add or delete users and admins, manage product listings, and update order statuses, ensuring smooth website operations.

Overall, the Apparel Website project successfully integrates technology to deliver a user-friendly interface, efficient user management, and an enhanced apparel shopping experience. By leveraging these technologies, the project aims to provide a seamless platform for users to explore, purchase, and manage apparel items while catering to the administrative needs of managing the website effectively.

8. REFERENCES

1. **GitHub Link:** <https://github.com/paragmraw/apparel-shopping-website-on-xampp>