TABLE OF CONTENTS

5. **Conclusion and Future Scope**

- Summary of the User Management System Implementation
- Future Enhancements
    1. Product Management
    2. Order Processing
    3. Shopping Cart and Wishlist
    4. Enhanced Security Features
    5. User Profile Customization
    6. Admin Analytics and Reporting

INTRODUCTION

In today’s digital era, e-commerce has become a critical business model, allowing consumers to purchase goods and services online. One of the foundational elements of any e-commerce platform is the User Management System (UMS), which facilitates secure user registration, login, and role-based access control.

An effective User Management System ensures that sensitive information, such as passwords and personal details, is securely stored and retrieved, preventing unauthorized access and data breaches. With the rise of global platforms like Amazon, eBay, and Alibaba, the need for well-structured and scalable user management systems has grown.

This project focuses on designing and implementing a User Management System for an Amazon clone using PHP and MySQL. By leveraging password hashing and session management, the system ensures a secure and smooth user experience.

Additionally, this project lays the foundation for further development, such as product management, order processing, and an integrated shopping cart, making it scalable and expandable for future needs.

LITERATURE REVIEW

The concept of user management systems is not new; they have evolved significantly over the years. Various studies and projects have focused on improving the security and functionality of user authentication systems, given their importance in today’s e-commerce platforms.

One of the most critical aspects of user security is password management. According to Chaudhary et al. (2020), password hashing is essential to prevent attackers from retrieving plain-text passwords in case of a database breach. The use of bcrypt and similar algorithms, as discussed in Juels et al. (2019), is considered an industry standard for securely storing passwords. In this project, we use password_hash() to hash the user's password and password_verify() to authenticate users during login, ensuring high-level security.

Session management is another key element, as highlighted in Sandhu et al.'s (1996) work on Role-Based Access Control (RBAC). The ability to manage user sessions securely allows a web application to differentiate between logged-in users and non-logged-in users and restrict access based on user roles (admin, customer). Research from Anderson (2018) emphasizes the importance of protecting session data from vulnerabilities like session hijacking or cross-site scripting (XSS). The implementation of PHP sessions in this project ensures that users can access their accounts securely without exposing session data to potential threats.

Lastly, Juels et al. (2019) discuss the growing importance of differentiating user roles within a system. An admin, for example, needs to have access to more features than a regular customer. This principle is implemented in this project through the account_type field, where users are categorized as either admins or customers, granting them different levels of access to the platform.

METHODOLOGY

The methodology for developing the User Management System involves a combination of frontend and backend development, focusing on database design, secure authentication, and session management. Below is a breakdown of the core phases of this project:

1. Database Design:

The system is built around a MySQL relational database, with the users table serving as the core for managing user information. The table contains fields for user details such as username, email, password (hashed), address, phone, and account_type (to differentiate between admins and customers). The created_at field uses the TIMESTAMP data type to automatically capture the user's registration date and time. The database is designed to be scalable, supporting future expansion to include additional tables such as products and orders.

2. Frontend Development:

The frontend of the system is developed using HTML and CSS. The user registration and login forms collect user input, such as email, password, and personal information. Basic form validation is implemented using HTML5 attributes like required, ensuring that all necessary fields are filled before form submission. The visual layout is kept simple and intuitive to ensure a smooth user experience.

3. Backend Development:

On the backend, PHP is used to handle form submissions and interact with the MySQL database. When a user registers, their password is hashed using PHP's password_hash() function, ensuring secure storage. During login, the password_verify() function is used to check the entered password against the hashed password in the database. If the credentials are correct, a session is created, and the user is redirected to the homepage. The backend also handles error messages, such as invalid login attempts or duplicate email registrations, and displays appropriate feedback to the user.

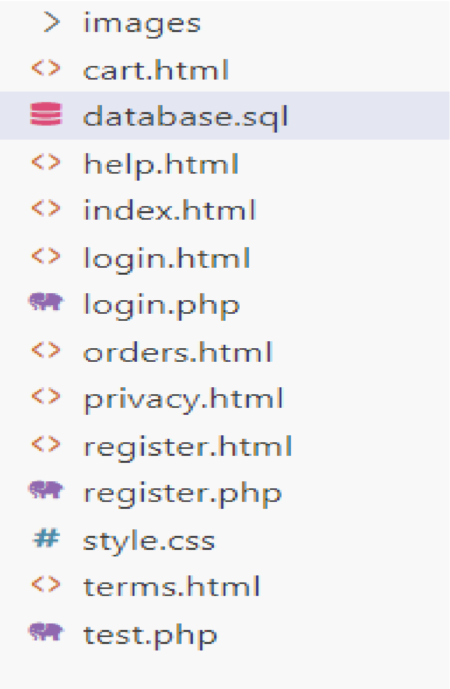
4. Testing and Validation:

The system undergoes rigorous testing to ensure that it meets the required functionality. Unit tests are conducted to verify that users can successfully register and log in. The system is also tested against common vulnerabilities like SQL injection and cross-site scripting (XSS) to ensure that it is secure.

RESULT

i) Input:

This is our workflow -



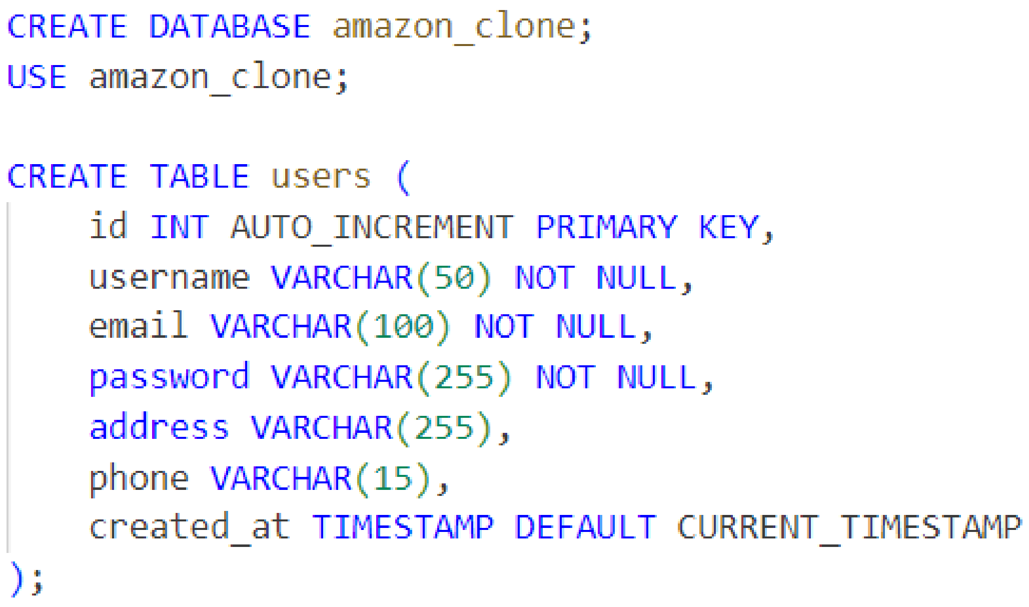
```
> images
cart.html
database.sql
help.html
index.html
login.html
login.php
orders.html
privacy.html
register.html
register.php
style.css
terms.html
test.php
```

This is our main page index.html -

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Amazon Clone</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <header class="navbar">
        <div class="logo"><a href="index.html">Amazon</a></div>
        <div class="search-bar">
            <input type="text" placeholder="Search Amazon">
            <button>Search</button>
        </div>
        <div class="nav-links">
            <a href="login.html">Account</a>
            <a href="orders.html">Orders</a>
            <a href="cart.html">Cart</a>
        </div>
    </header>
    <section class="hero">
        <img src="images/beer.jpg" alt="Promotional Banner">
    </section>

    <section class="product-section">
        <h2>Top Products</h2>
        <div class="product-grid">
            <div class="product-card">
                <img src="images/coke.jpg" alt="Product Image">
                <h3>Coca Cola</h3>
                <p>Rs99</p>
                <img src="images/coke 2.jpg" alt="Product Image">
                <h3>Tin Can</h3>
                <p>Rs999</p>
                <img src="images/coke 3.jpg" alt="Product Image">
                <h3>Mocup</h3>
                <p>Rs9999</p>
            </div>
            <div class="product-card">
                <img src="images/glass.jpg" alt="Product Image">
                <h3>Beer Glass</h3>
                <p>Rs99</p>
                <img src="images/glass 2.jpg" alt="Product Image">
                <h3>Glass</h3>
                <p>Rs999</p>
                <img src="images/glass3.jpg" alt="Product Image">
                <h3>Vine Glass</h3>
                <p>Rs9999</p>
            </div>
            <div class="product-card">
                <img src="images/printer.jpg" alt="Product Image">
                <h3>HP Printer</h3>
                <p>Rs99</p>
                <img src="images/printer 2.jpg" alt="Product Image">
                <h3>Dell Printer</h3>
                <p>Rs999</p>
                <img src="images/printer 3.jpg" alt="Product Image">
                <h3>Lenovo Printer</h3>
                <p>Rs9999</p>
            </div>
            <div class="product-card">
                <img src="images/Cream.jpg" alt="Product Image">
                <h3>Face Cream</h3>
                <p>Rs99</p>
                <img src="images/Cream 2.jpg" alt="Product Image">
                <h3>Eye Cream</h3>
                <p>Rs999</p>
                <img src="images/Cream 3.jpg" alt="Product Image">
                <h3>Night Cream</h3>
```

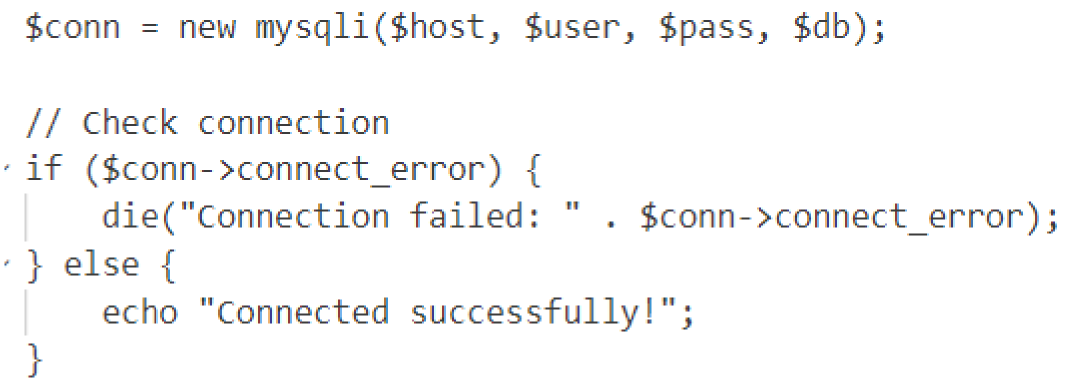
```
                <p>Rs9999</p>
            </div>
            <div class="product-card">
                <img src="images/Perfume.jpg" alt="Product Image">
                <h3>Dior</h3>
                <p>Rs99</p>
                <img src="images/Perfume 2.jpg" alt="Product Image">
                <h3>Axe</h3>
                <p>Rs999</p>
                <img src="images/Perfume 3.jpg" alt="Product Image">
                <h3>Versace</h3>
                <p>Rs9999</p>
            </div>
            <div class="product-card">
                <img src="images/phone.jpg" alt="Product Image">
                <h3>Iphone</h3>
                <p>Rs99</p>
                <img src="images/phone 2.jpg" alt="Product Image">
                <h3>Samsung s24 ultra</h3>
                <p>Rs999</p>
                <img src="images/phone 3.jpg" alt="Product Image">
                <h3>Samsung s20 ultra</h3>
                <p>Rs9999</p>
            </div>

        </div>
    </section>

    <footer>
        <div class="footer-links">
            <a href="help.html">Help</a>
            <a href="privacy.html">Privacy</a>
            <a href="terms.html">Terms</a>
        </div>
    </footer>
</body>
</html>
```

This is database sql code -

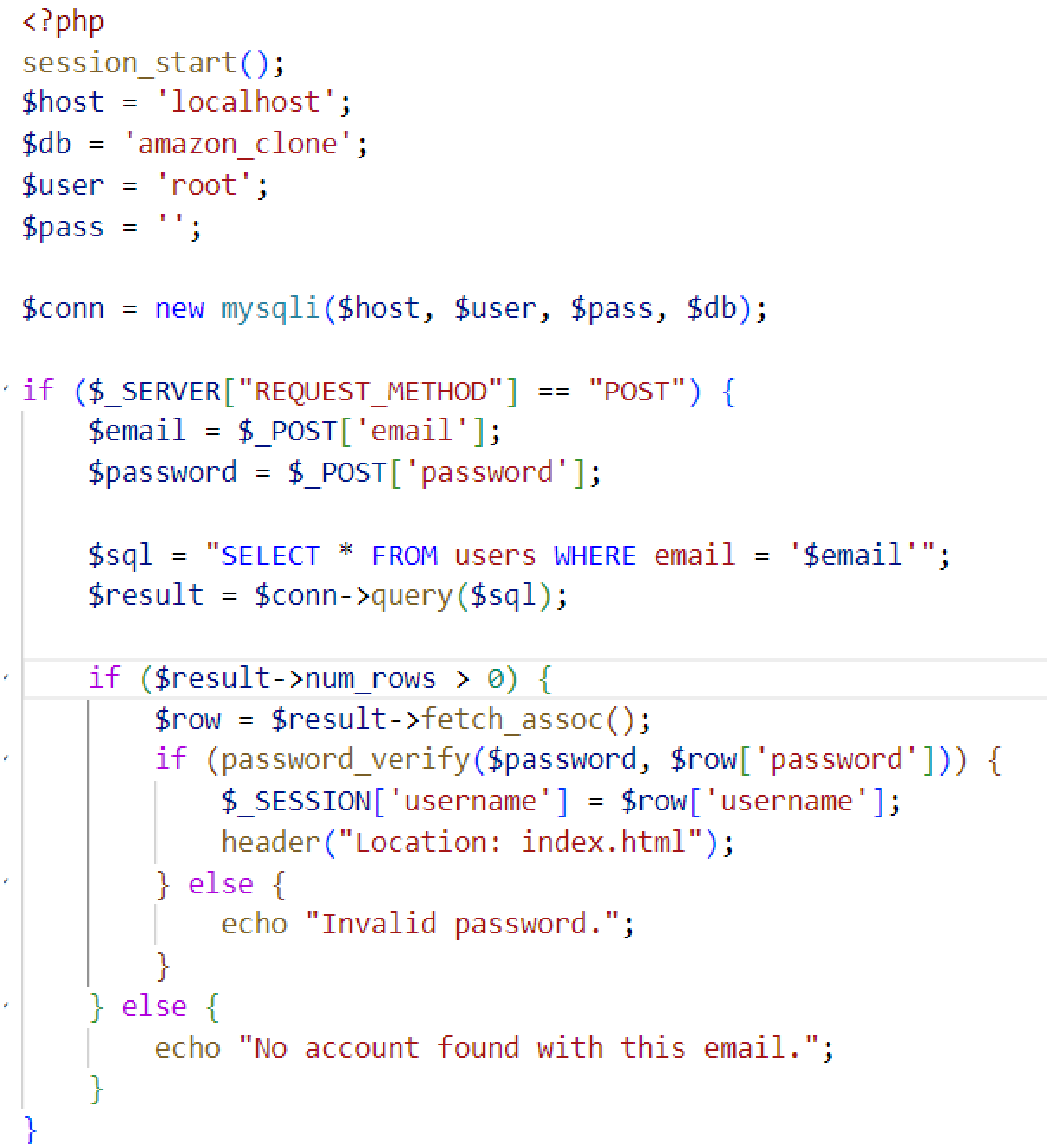


```
CREATE DATABASE amazon_clone;
USE amazon_clone;

CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL,
    email VARCHAR(100) NOT NULL,
    password VARCHAR(255) NOT NULL,
    address VARCHAR(255),
    phone VARCHAR(15),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

This is our login.php code -

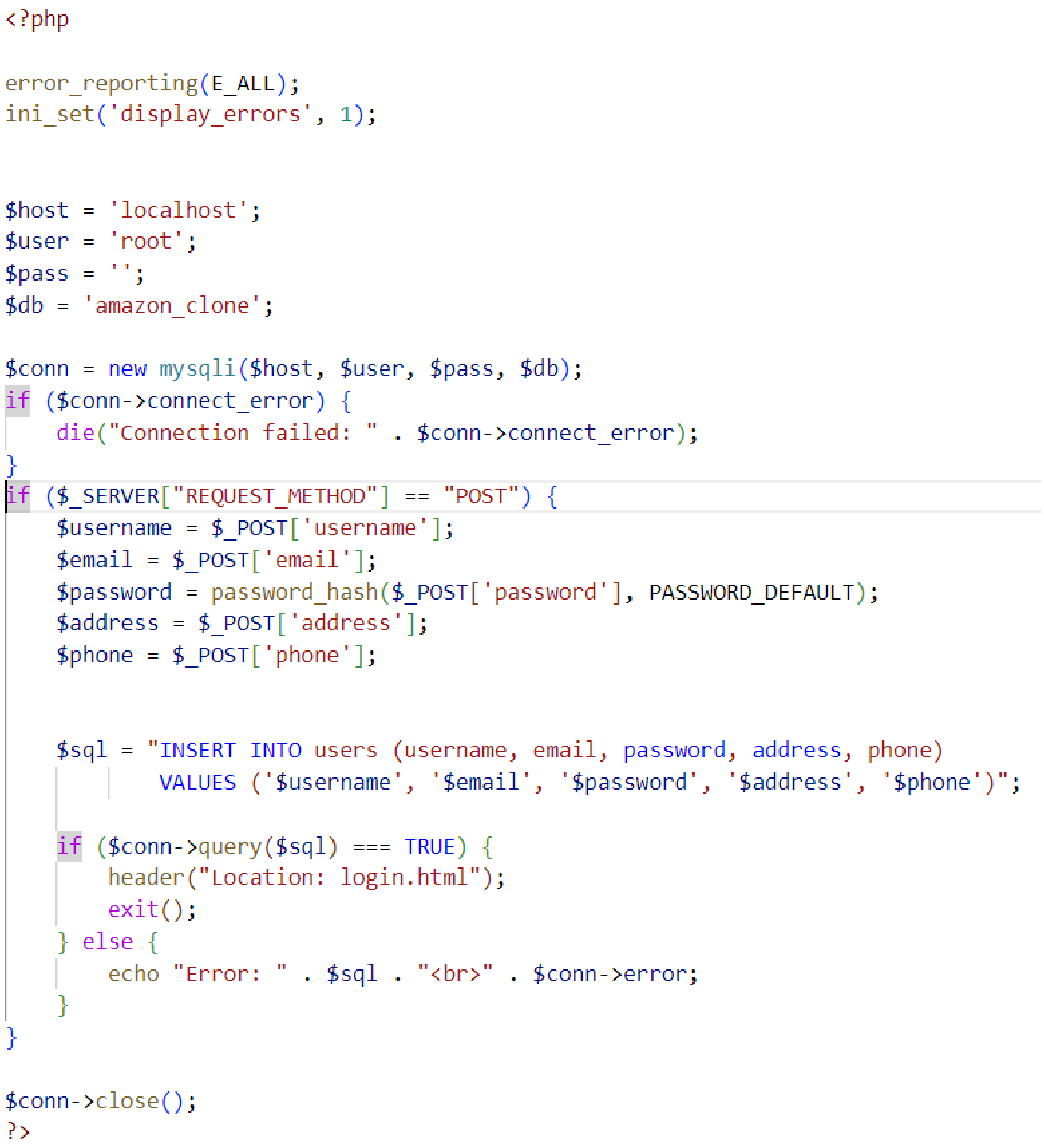


```php
$conn = new mysqli($host, $user, $pass, $db);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
} else {
    echo "Connected successfully!";
}
```



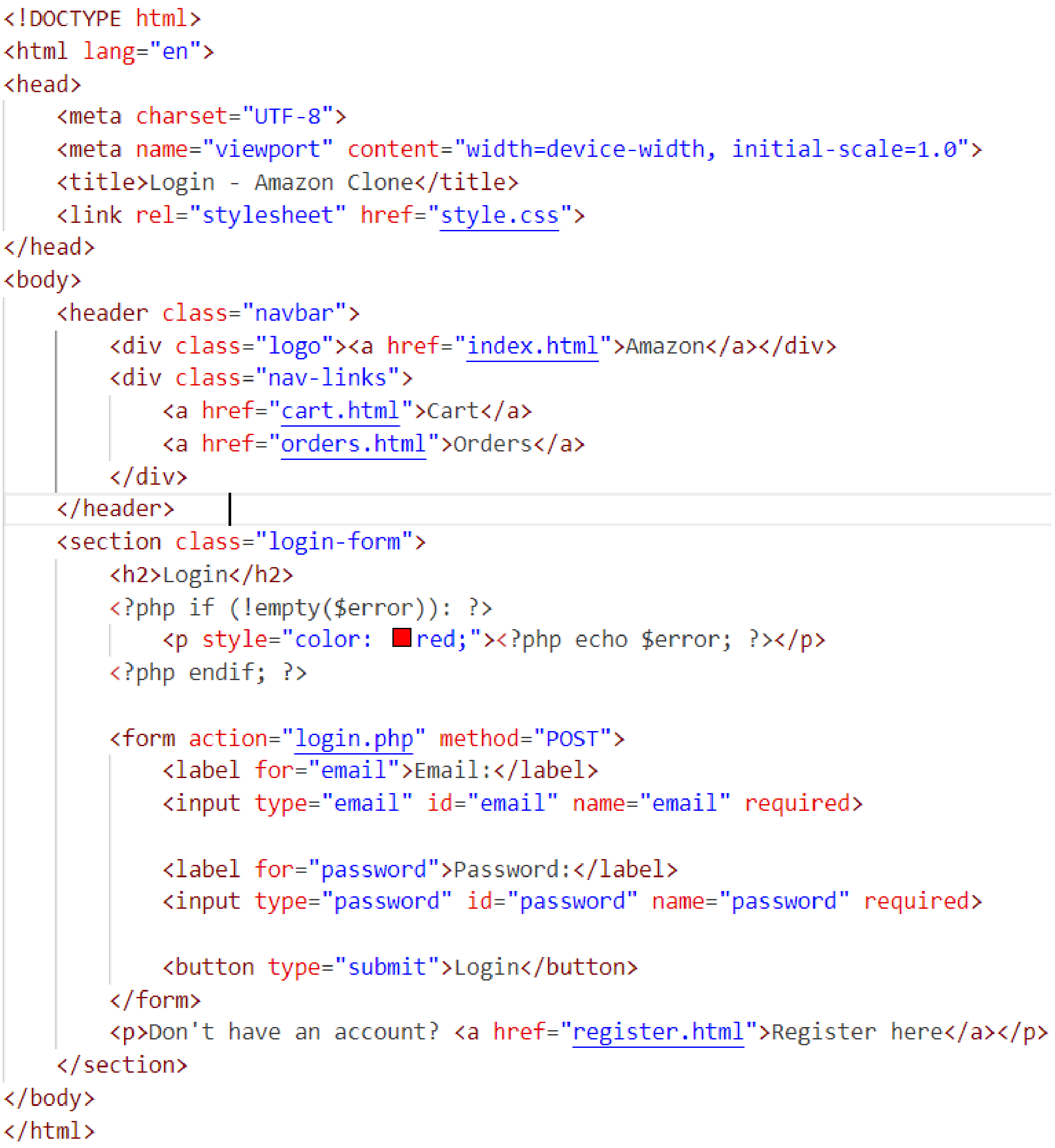
```php
<?php
session_start();
$host = 'localhost';
$db = 'amazon_clone';
$user = 'root';
$pass = '';

$conn = new mysqli($host, $user, $pass, $db);

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = $_POST['email'];
    $password = $_POST['password'];

    $sql = "SELECT * FROM users WHERE email = '$email'";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        $row = $result->fetch_assoc();
        if (password_verify($password, $row['password'])) {
            $_SESSION['username'] = $row['username'];
            header("Location: index.html");
        } else {
            echo "Invalid password.";
        }
    } else {
        echo "No account found with this email.";
    }
}
```

This is our register.php -



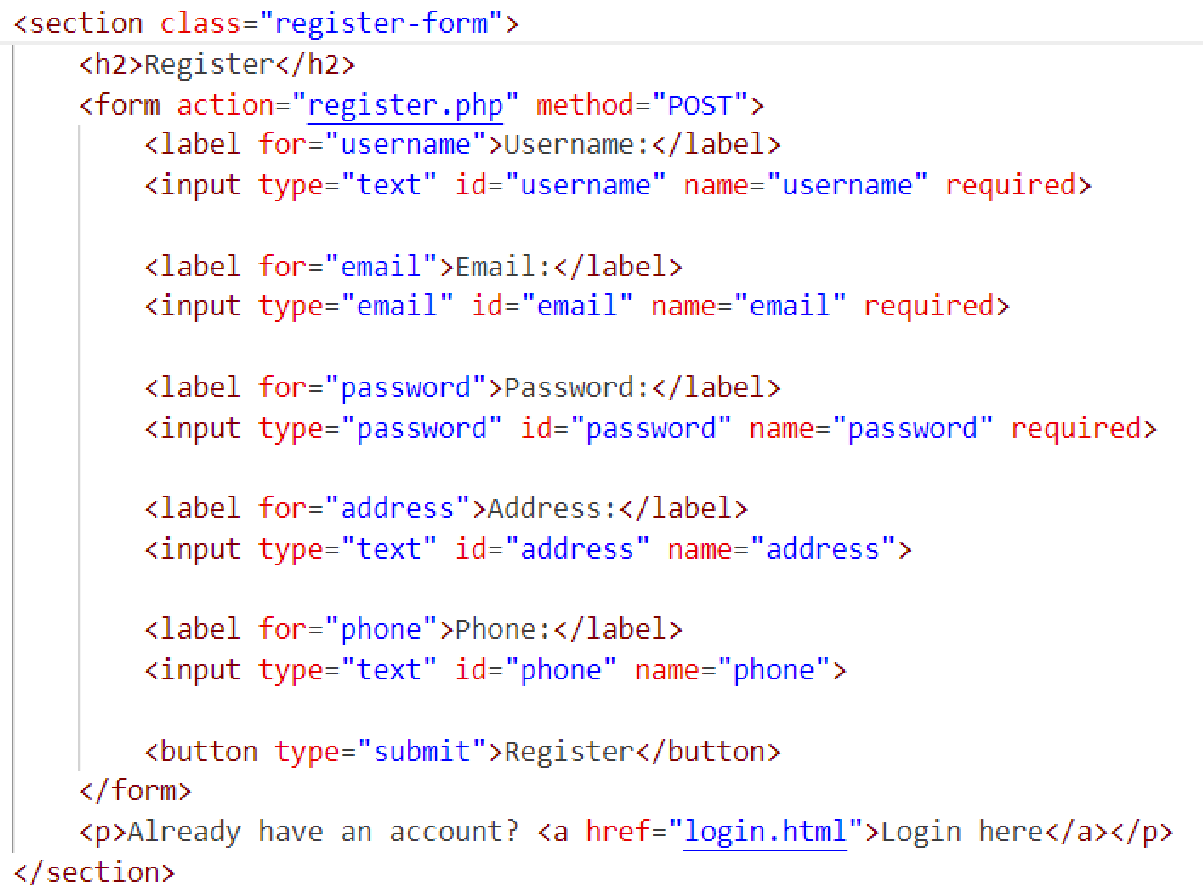
```php
<?php

error_reporting(E_ALL);
ini_set('display_errors', 1);


$host = 'localhost';
$user = 'root';
$pass = '';
$db = 'amazon_clone';

$conn = new mysqli($host, $user, $pass, $db);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST['username'];
    $email = $_POST['email'];
    $password = password_hash($_POST['password'], PASSWORD_DEFAULT);
    $address = $_POST['address'];
    $phone = $_POST['phone'];


    $sql = "INSERT INTO users (username, email, password, address, phone)
            VALUES ('$username', '$email', '$password', '$address', '$phone')";

    if ($conn->query($sql) === TRUE) {
        header("Location: login.html");
        exit();
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
}

$conn->close();
?>
```

This is login.html –



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login - Amazon Clone</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <header class="navbar">
        <div class="logo"><a href="index.html">Amazon</a></div>
        <div class="nav-links">
            <a href="cart.html">Cart</a>
            <a href="orders.html">Orders</a>
        </div>
    </header>
    <section class="login-form">
        <h2>Login</h2>
        <?php if (!empty($error)): ?>
            <p style="color: 🟥red;"><?php echo $error; ?></p>
        <?php endif; ?>

        <form action="login.php" method="POST">
            <label for="email">Email:</label>
            <input type="email" id="email" name="email" required>

            <label for="password">Password:</label>
            <input type="password" id="password" name="password" required>

            <button type="submit">Login</button>
        </form>
        <p>Don't have an account? <a href="register.html">Register here</a></p>
    </section>
</body>
</html>
```

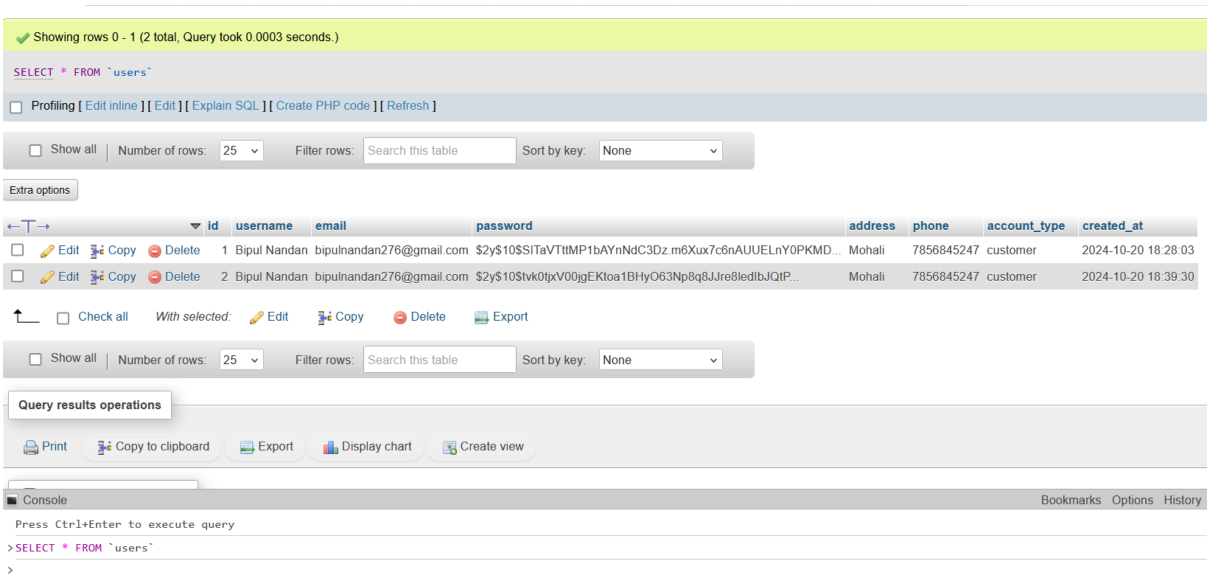
This is code for our registration form –



```
<section class="register-form">
    <h2>Register</h2>
    <form action="register.php" method="POST">
        <label for="username">Username:</label>
        <input type="text" id="username" name="username" required>

        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>

        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required>

        <label for="address">Address:</label>
        <input type="text" id="address" name="address">

        <label for="phone">Phone:</label>
        <input type="text" id="phone" name="phone">

        <button type="submit">Register</button>
    </form>
    <p>Already have an account? <a href="login.html">Login here</a></p>
</section>
```

ii) Output:

This is where our data is being in our database -



✓ Showing rows 0 - 1 (2 total, Query took 0.0003 seconds.)

```
SELECT * FROM `users`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

←T→ ▼	id	username	email	password	address	phone	account_type	created_at
□ Edit Copy Delete	1	Bipul Nandan	bipulnandan276@gmail.com	$2y$10$SITaVTttMP1bAYnNdC3Dz.m6Xux7c6nAUUELnY0PKMD...	Mohali	7856845247	customer	2024-10-20 18:28:03
□ Edit Copy Delete	2	Bipul Nandan	bipulnandan276@gmail.com	$2y$10$tvk0tjxV00jgEKtoa1BHyO63Np8q8JJre8ledIbJQtP...	Mohali	7856845247	customer	2024-10-20 18:39:30

□ Check all *With selected:* Edit Copy Delete Export

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

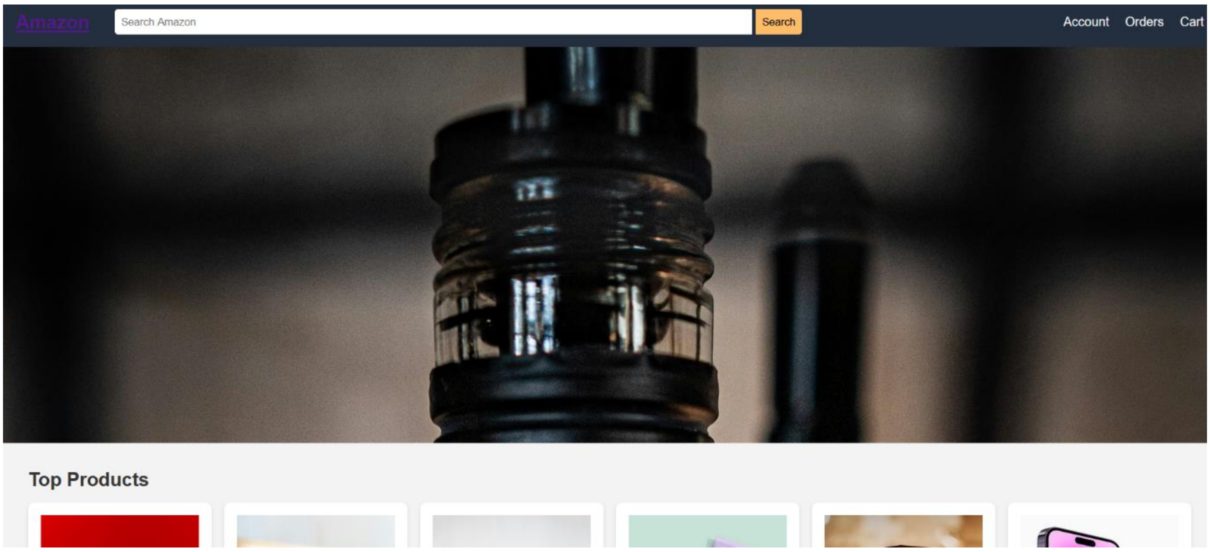
Query results operations

Print Copy to clipboard Export Display chart Create view

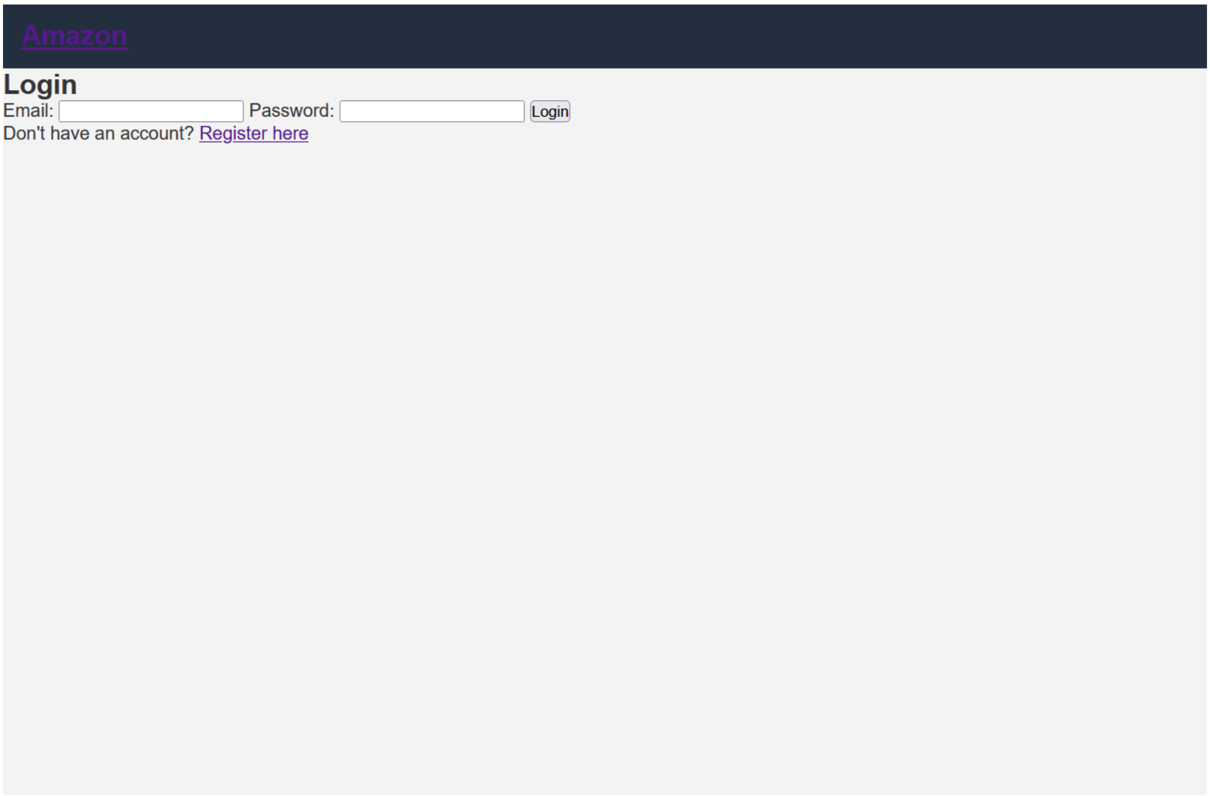
Console Bookmarks Options History

```
Press Ctrl+Enter to execute query
>SELECT * FROM `users`
>
```

Our website homepage -



Our website login page -

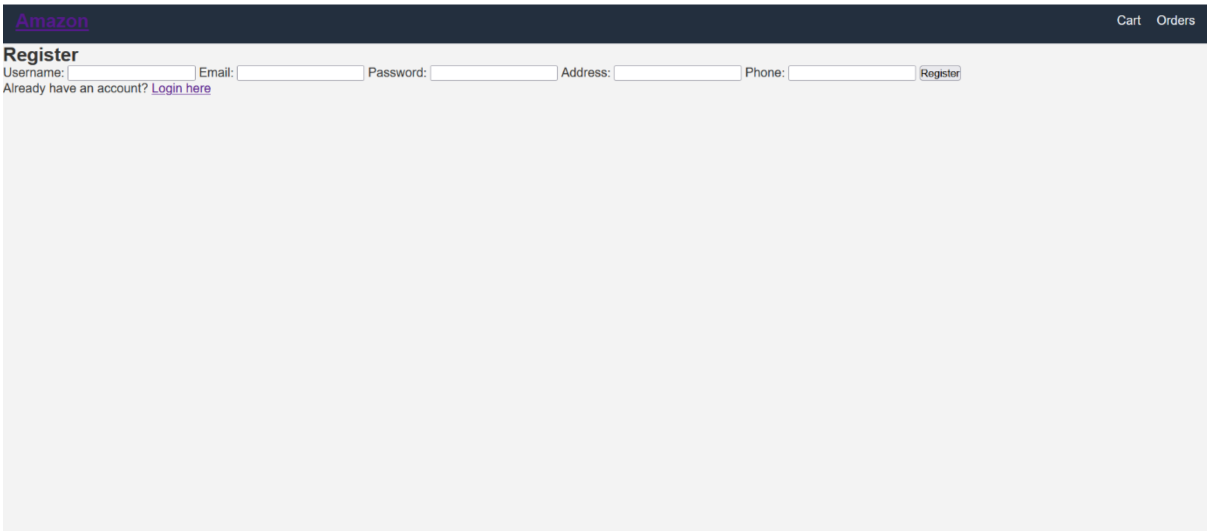


Login

Email: Password: Login

Don't have an account? Register here

Our website registration form –



Register

Username: Email: Password: Address: Phone:

Already have an account? Login here

CONCLUSION/ FUTURE SCOPE

i) Conclusion:

The User Management System for the Amazon Clone e-commerce platform successfully provides secure and efficient user registration, login. By utilizing PHP for backend processing and MySQL for database management, the system ensures that user data is stored securely and that unauthorized access is prevented. The implementation of password hashing and session management further strengthens the security of the platform.

Through this project, key skills such as database design, form validation, and user authentication were developed. The system's scalability ensures that it can be expanded to include additional features like product management and order processing. The use of role-based access control also demonstrates how different user types can be managed effectively within a single system.

ii) Future Scope:

While the current project focuses on user management, there are several areas where the system can be expanded to provide a fully functional e-commerce platform. Some potential future developments include:

1. Product Management:

Admins should be given the ability to manage products in the system, including adding, updating, or deleting products. Product categories can be introduced to help organize items, and inventory management can be added to track stock levels. This will allow the platform to serve as a marketplace where users can browse and purchase items.

2. Order Processing:

Integrating an order processing system will allow users to place orders for products. This system can include features such as payment processing, shipping management, and order tracking. Admins can oversee order statuses and ensure that products are delivered in a timely manner.

3. Shopping Cart and Wishlist:

A shopping cart system should be added, allowing users to add products to their cart before checkout. Additionally, a wishlist feature can be introduced, enabling users to save products for future purchases.

4. Enhanced Security Features:

To further enhance the system's security, multi-factor authentication (MFA) can be implemented during the login process. This would provide an additional layer of security, requiring users to verify their identity through a second factor, such as a mobile device. Additionally, using HTTPS for secure data transmission will protect against man-in-the-middle attacks.

5. User Profile Customization:

Allow users to personalize their profiles with features such as profile pictures, delivery preferences, and payment methods. This will enhance the overall user experience and make the platform more interactive.

6. Admin Analytics and Reporting:

For admin users, detailed analytics and reporting tools can be introduced. These tools can provide insights into product sales, user activity, and order trends, allowing admins to make data-driven decisions to improve the business.