**PHP INTERNSHIP REPORT**

**PERSONAL DETAILS :**

Name : Sohil Vora

Collage Name : Noble University

Degree : MCA

Semester : 4

GitHub URL : <https://github.com/sohilvora>

**COMPANY DETAILS:**

Company Name : Akash Technolabs

External Guide : Akash Padhiyar

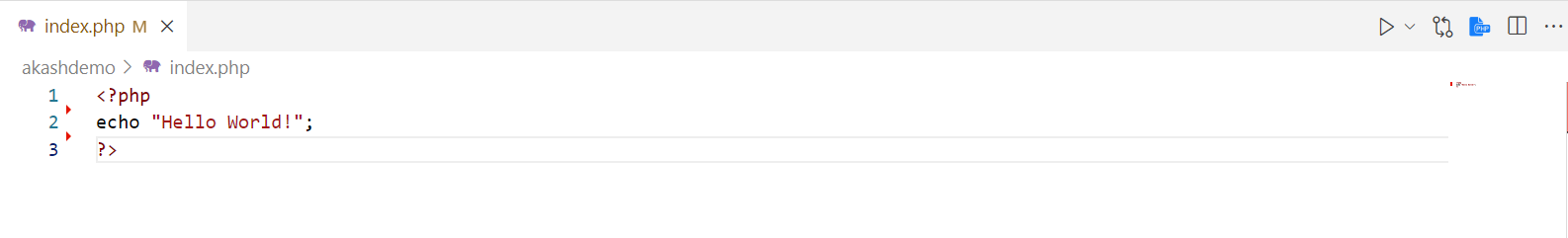
Training Duration : 01 Jan 2024 to 15 May 2024

**INDEX**

|  |  |  |
| --- | --- | --- |
| **No.** | **Task Name** | **Page** |
| **1** | PHP Crash Course | 3 |
| **2** | CRUD operation for employee table | 7 |
| **3** | Bootstrap admin template, CRUD operation for product table | 9 |
| **4** | Login and logout with session | 11 |
| **5** | Login and logout user functionality, after login user can change password & forgot password | 14 |
| **6** | E-commerce website demo, register, login, logout, display product category wise, add to cart, checkout | 17 |
| **7** | E-commerce admin & user bootstrap template register, login, logout, display product category wise, add to cart, checkout | 22 |
| **8** | E-commerce admin and user with PDO statement using MySQL | 27 |
| **9** | Chat app in PHP with MySQL using Ratchet WebSocket | 33 |

* **index.php:**
* This is file simply print a hello wolrd message on web browser.

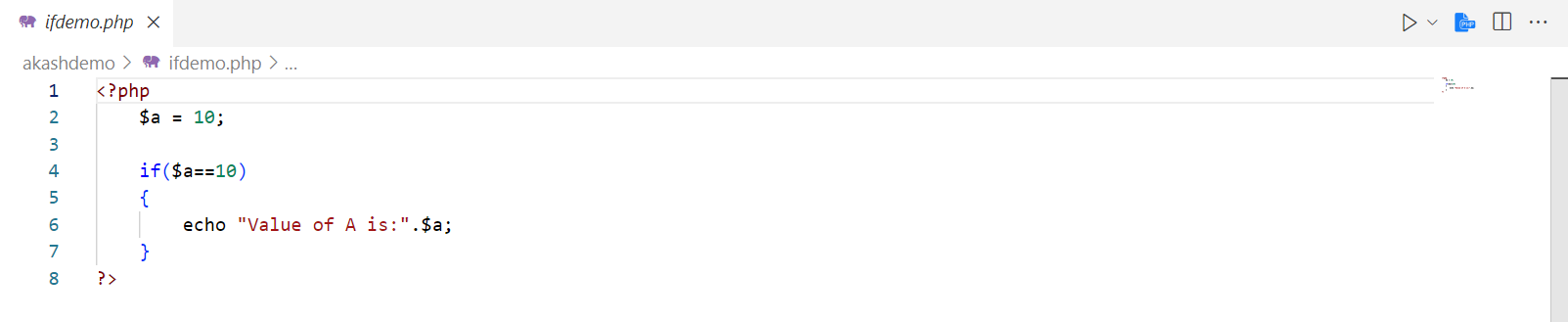
1. **PHP Crash Course**



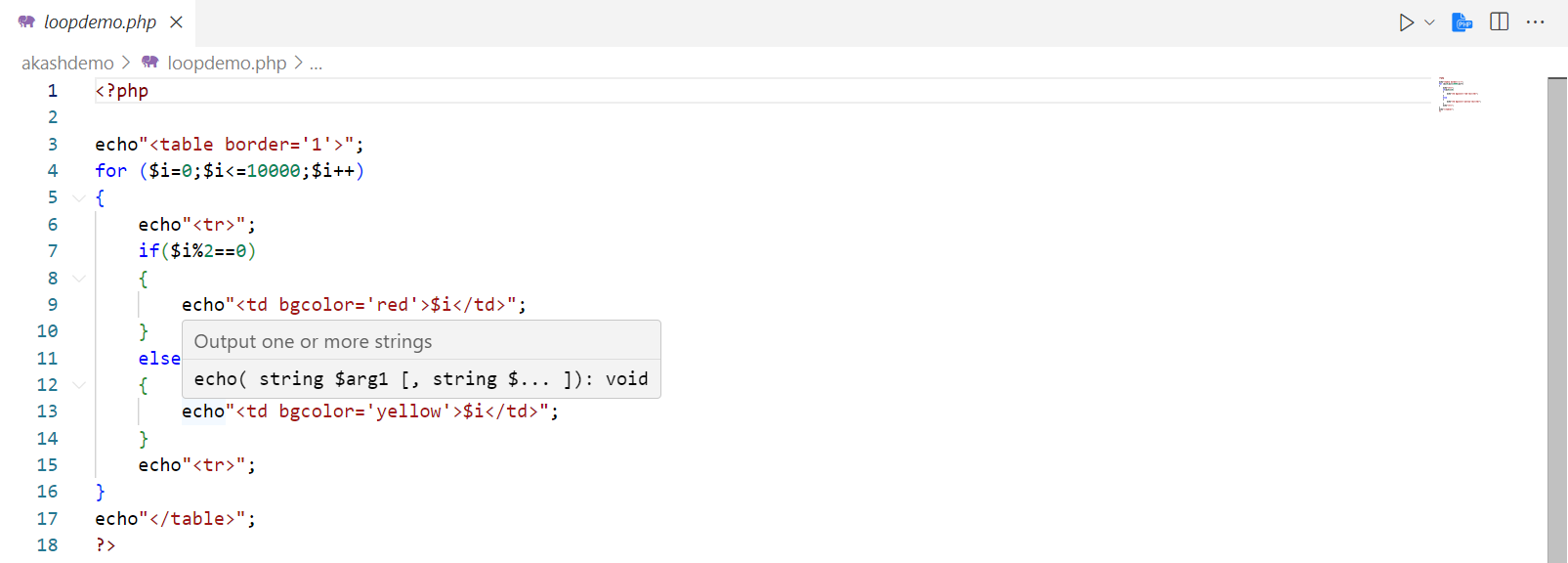
* **variabledemo.php:**
* This is file simply we declare the variable and assign the value.



* **Ifdemo.php:**
* This is file we declare condition if condition satisfied then statement will execute.



* **Loopdemo.php:**
* This is file we use looping statement that is use to print a statement repeatedly while condition goes true.



* **Formdemo.php:**
* This is file we use html form within php file that is use to get data from users.



* **dbdemo.php:**
* This is file we connect to the database to store data that comes from formdemo.php file.

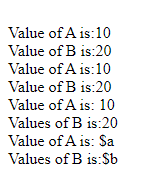


Output

* **Index page:**



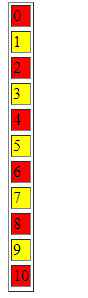
* **Variable page:**



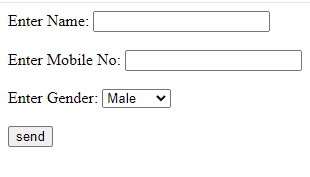
* **Ifdemo page:**



* **Loopdemo page:**

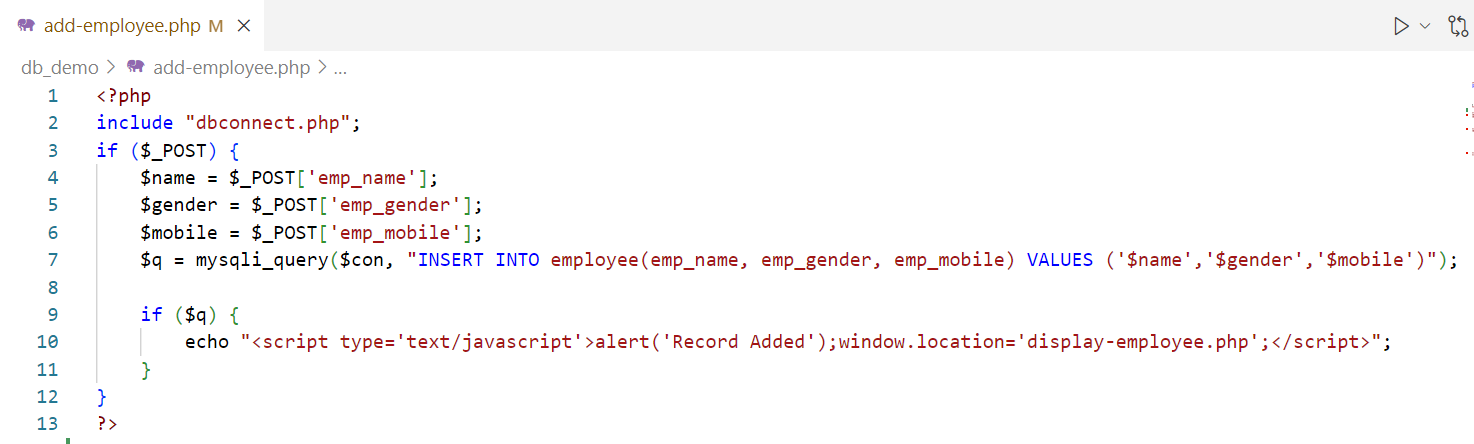


* **Form page:**



1. **Crud Operation for Employee Table**

* **Add\_employee.php:**
* Add new records to the employee table. This involves inserting a new row with the relevant employee information into the table.

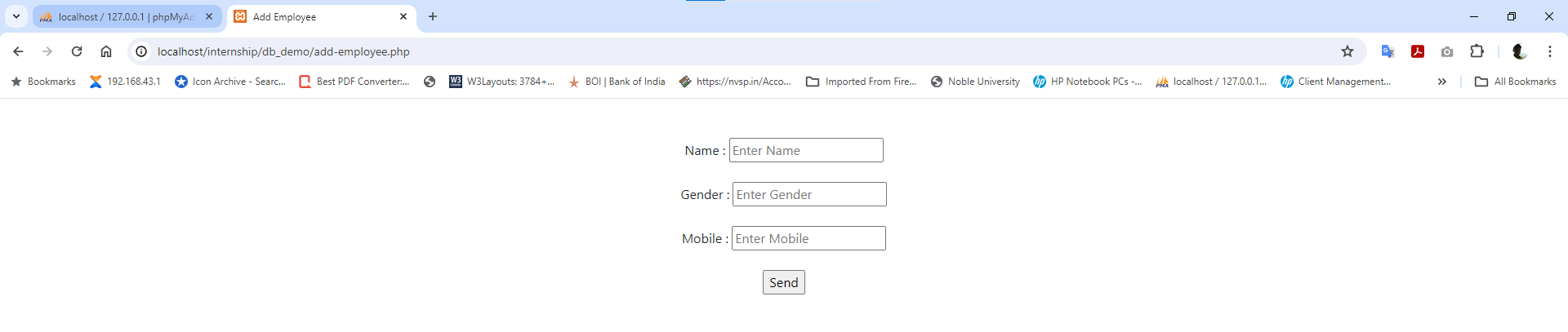


* **Edit\_employee.php:**
* Modify existing records in the employee table. This involves updating the values of one or more columns in a specific row based on certain conditions.
* **Delete\_employee.php:**
* Remove records from the employee table. This file deleting one or more rows from the table based on certain conditions.

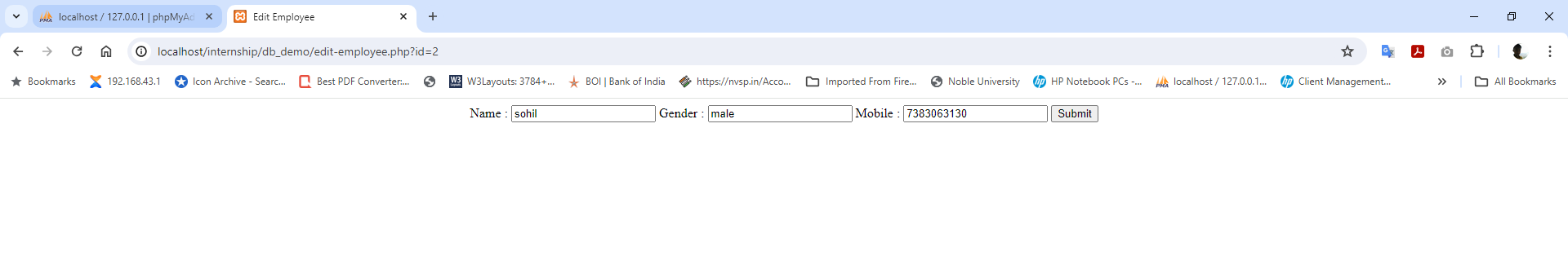


Output

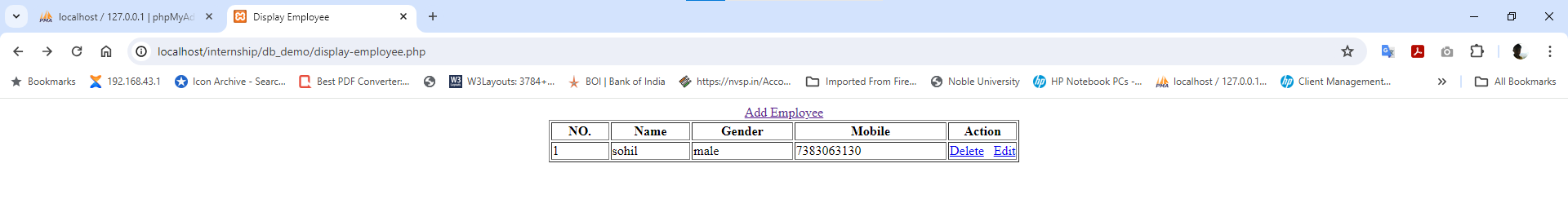
* **Add employee Page:**



* **Edit employee Page:**



* **Display, Delete and Edit employee Page:**

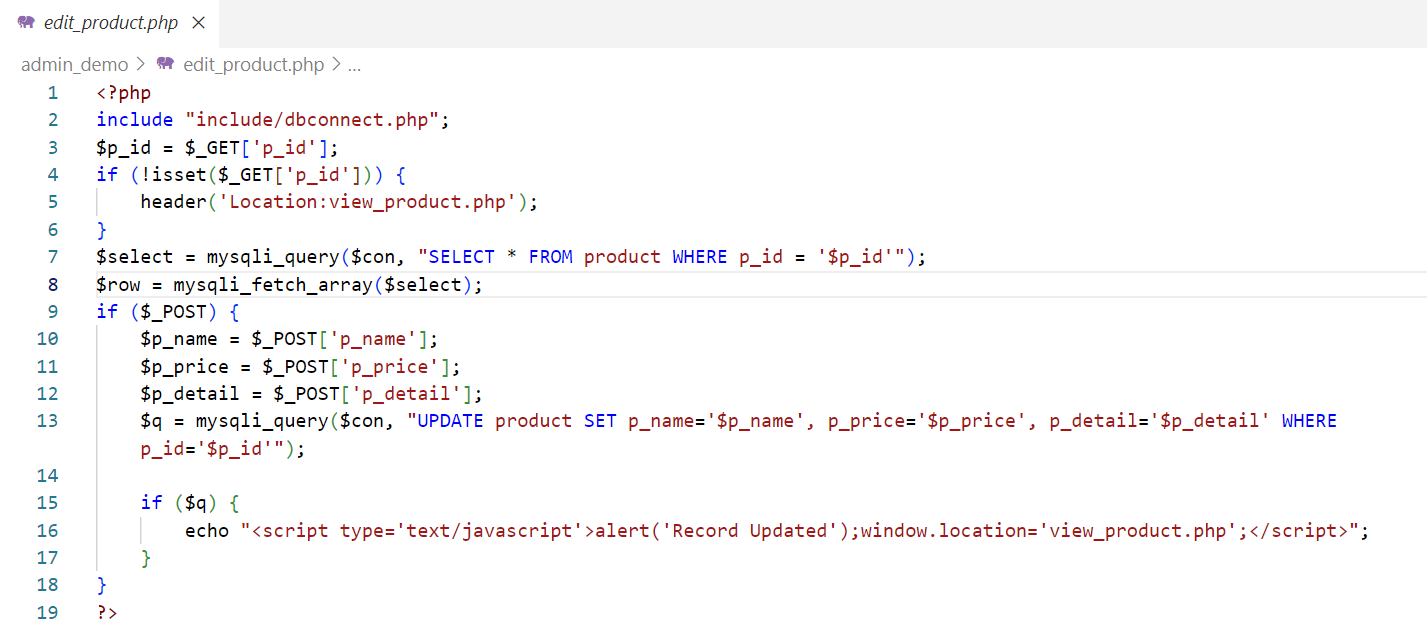


1. **Bootstrap Admin template, Crud Operation for poduct Table**

* **Add\_product.php:**
* Designing a form within the admin template to input product details.
* Processing the form submission to add a new product record to the database.

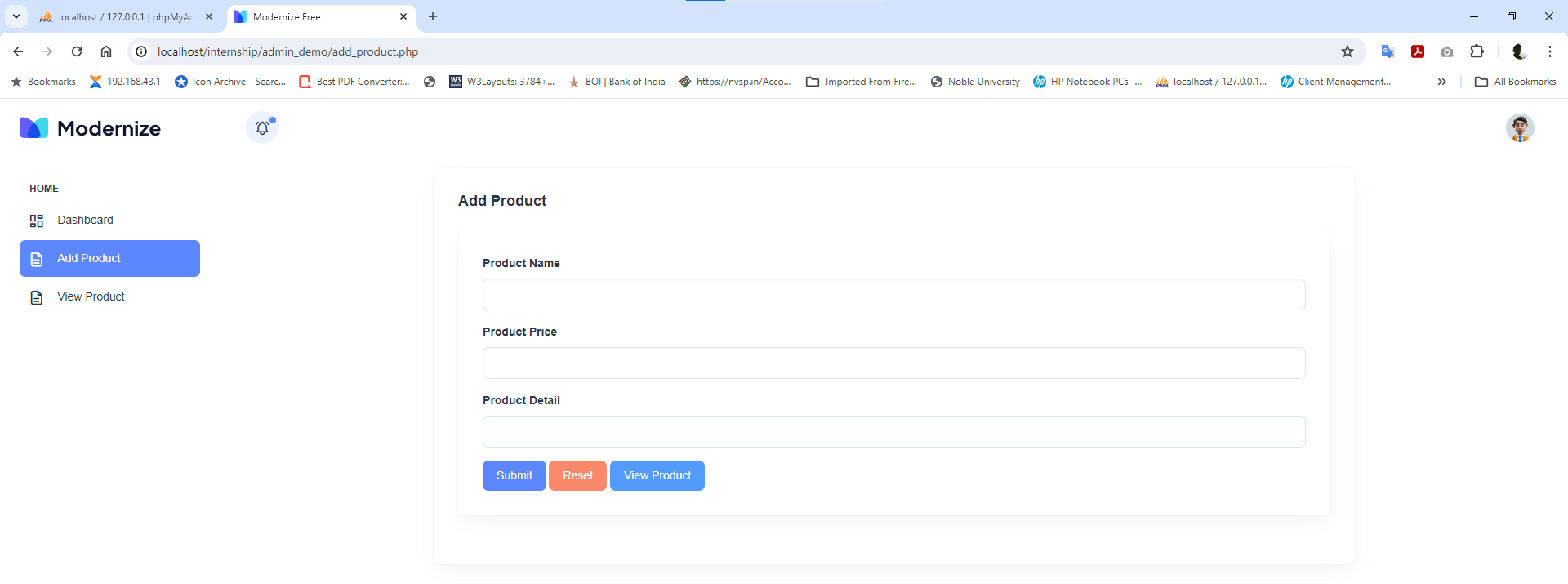
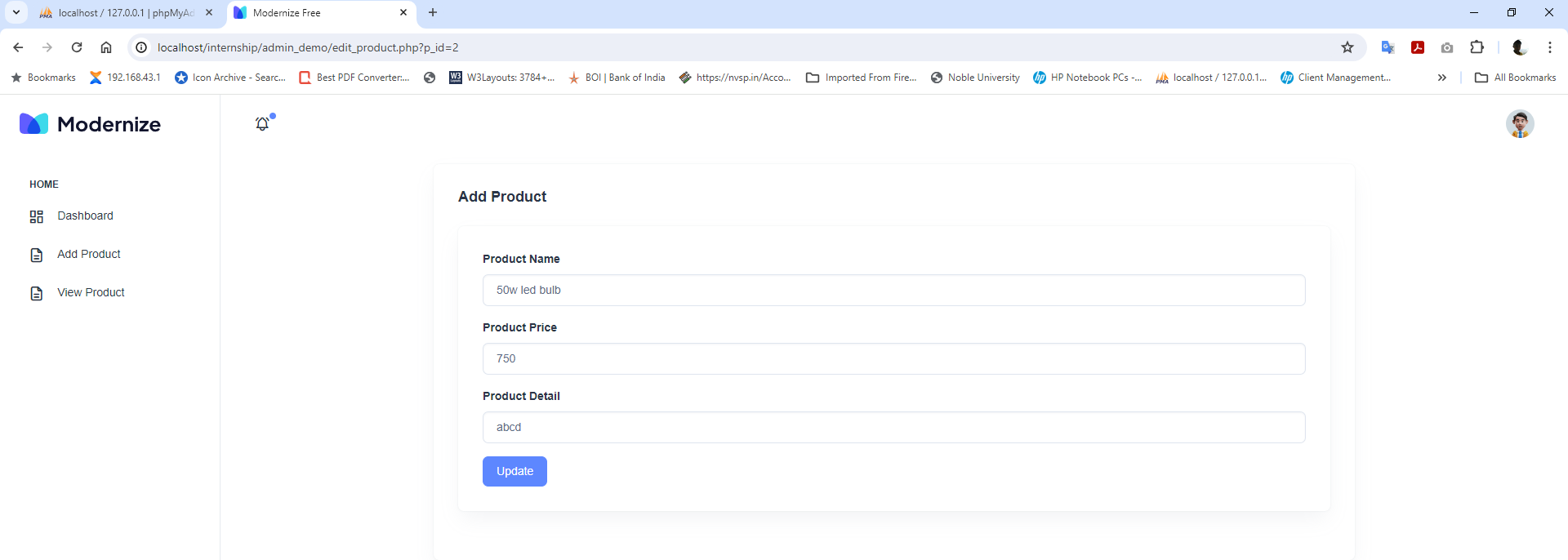


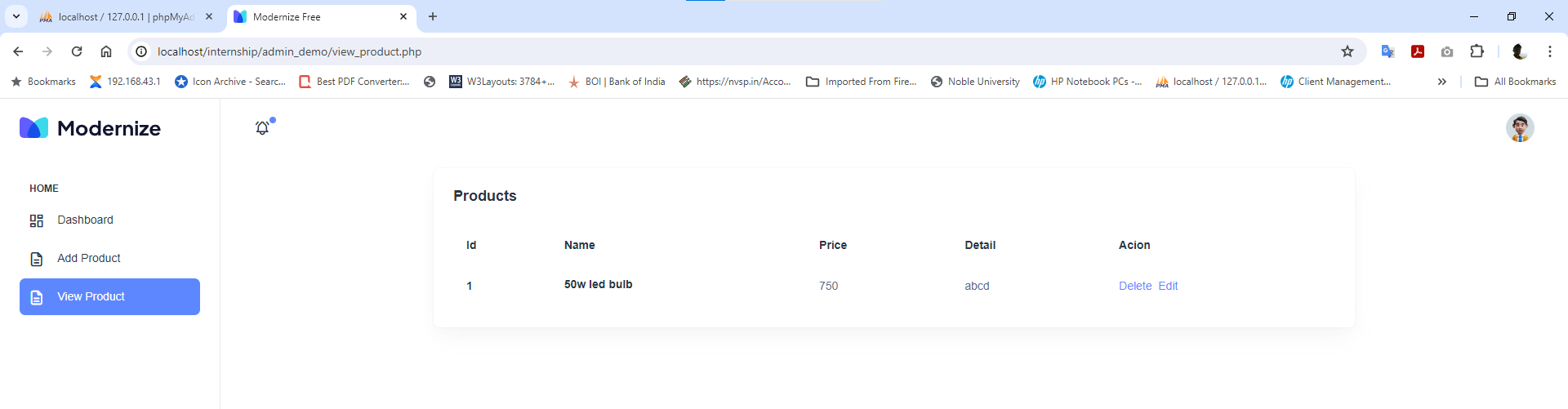
* **Edit\_product.php:**
* We create editable form in the admin template to modify product details.
* Then we got existing data which we want to alter
* Then we update existing data into form submission to update the corresponding product record in the database.



* **Delete\_product.php:**
* Processing the deletion request to remove the selected product(s) from the database.

Output

* **Add product Page:**
* **Edit product Page:**
* **Display and Delete product Page:**

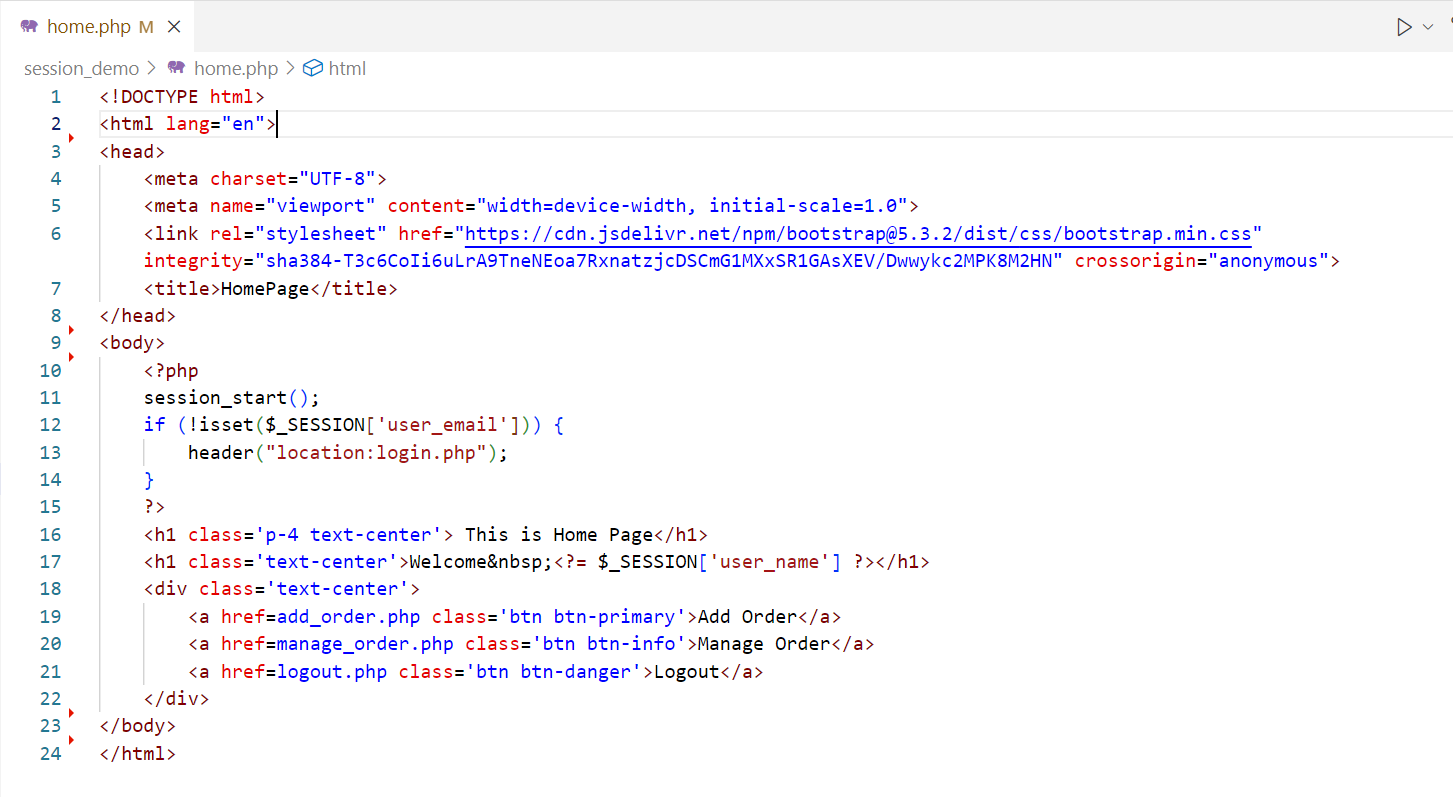


**4. Login & Logout with Session**

* **Login.php:**
* We create login form after login, a session is created to maintain the user's authentication state throughout their interaction with the system.



* **index.php:**
* After login successfully user will redirected to the index page.

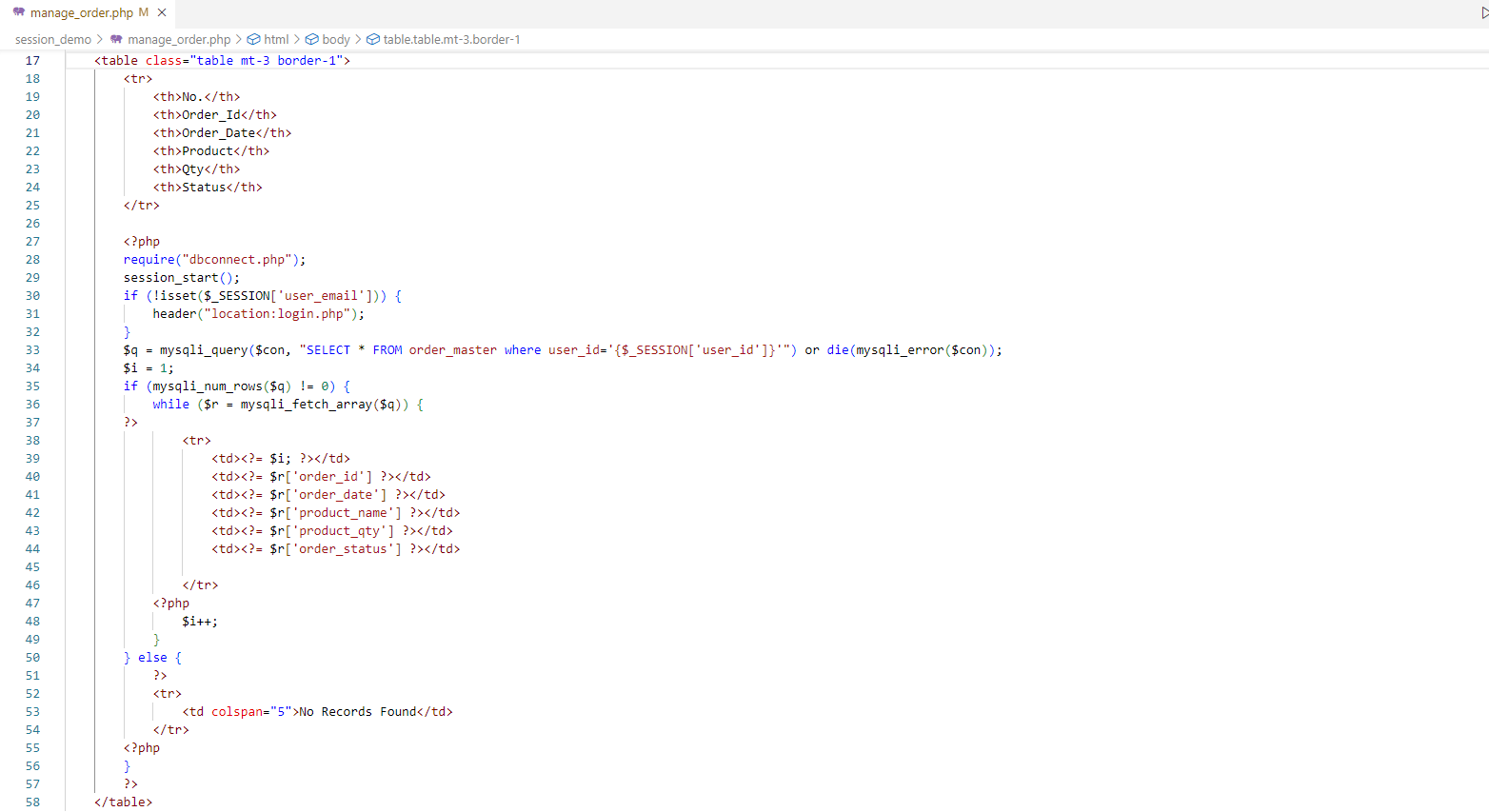


* **add\_order.php:**
* In this file user add order after log’s in .

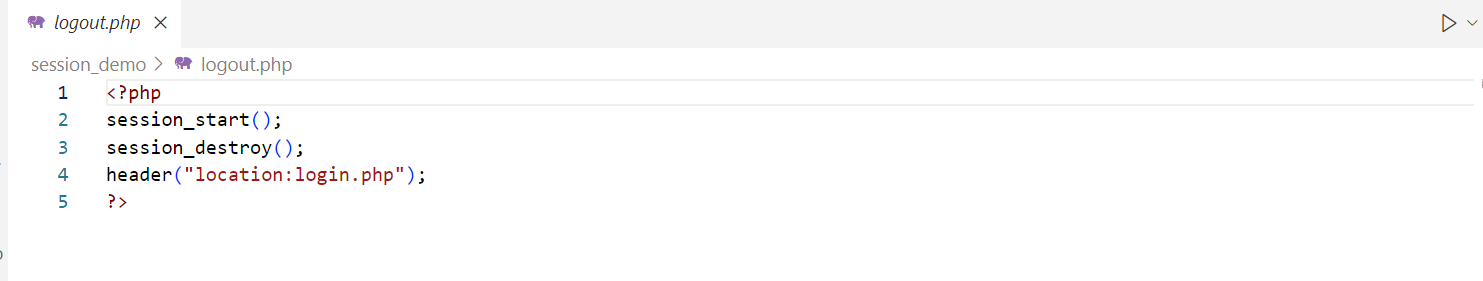


* **manage\_order.php:**

• In this file user Manage the order after log’s in .

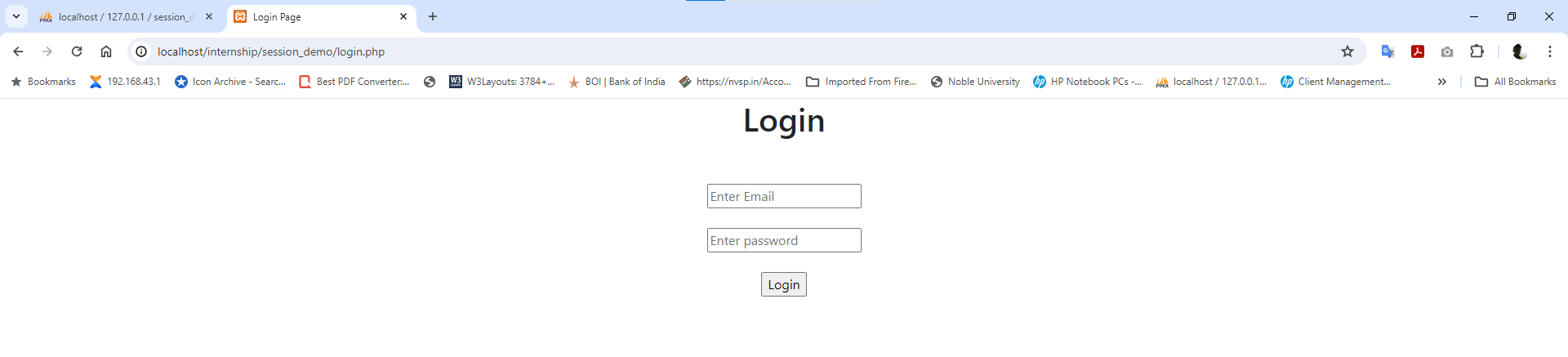


* **logout.php**
* When the user chooses to log out, the session is ended, thereby revoking their access.

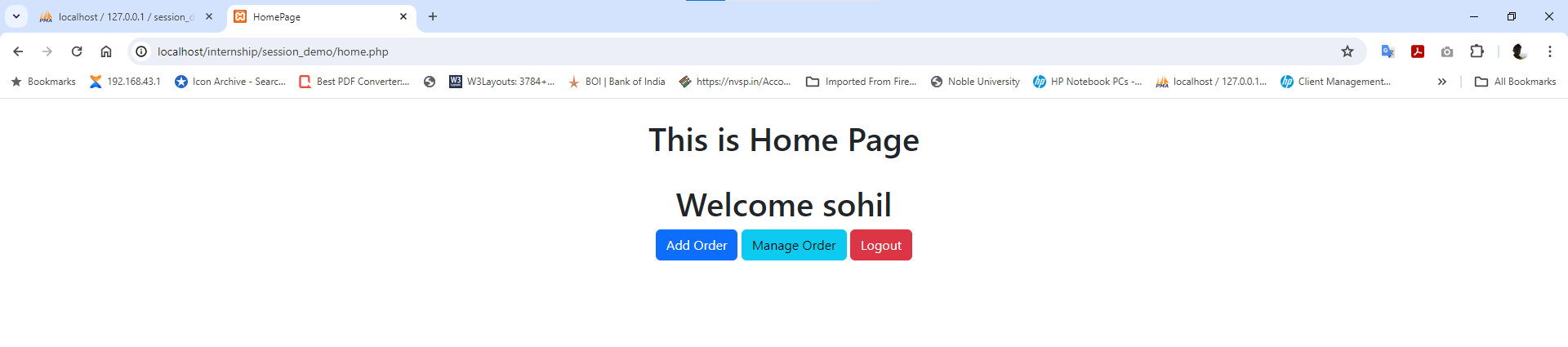


Output

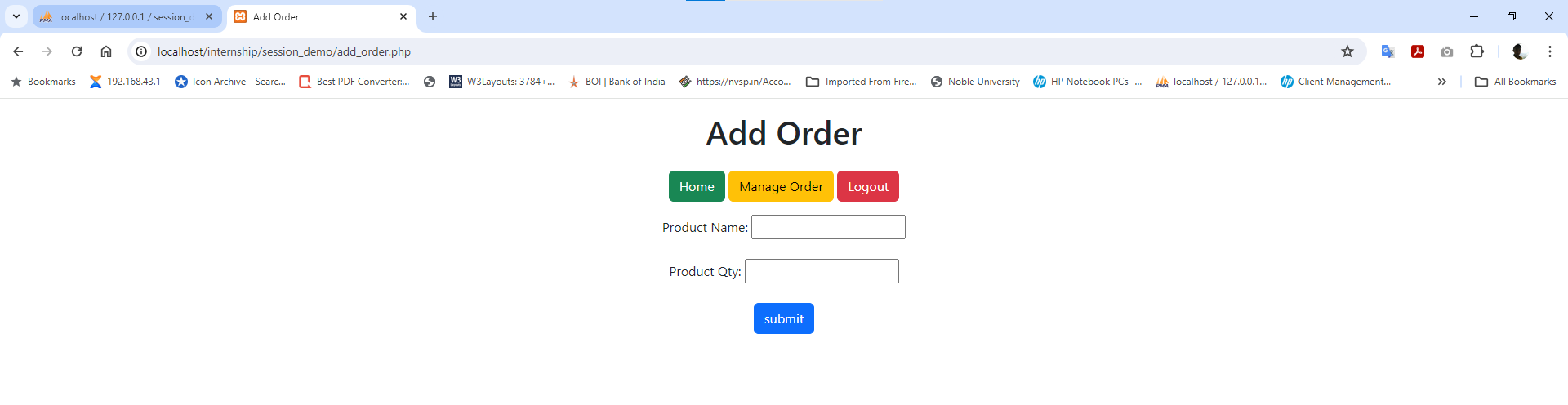
* **Login Page:**



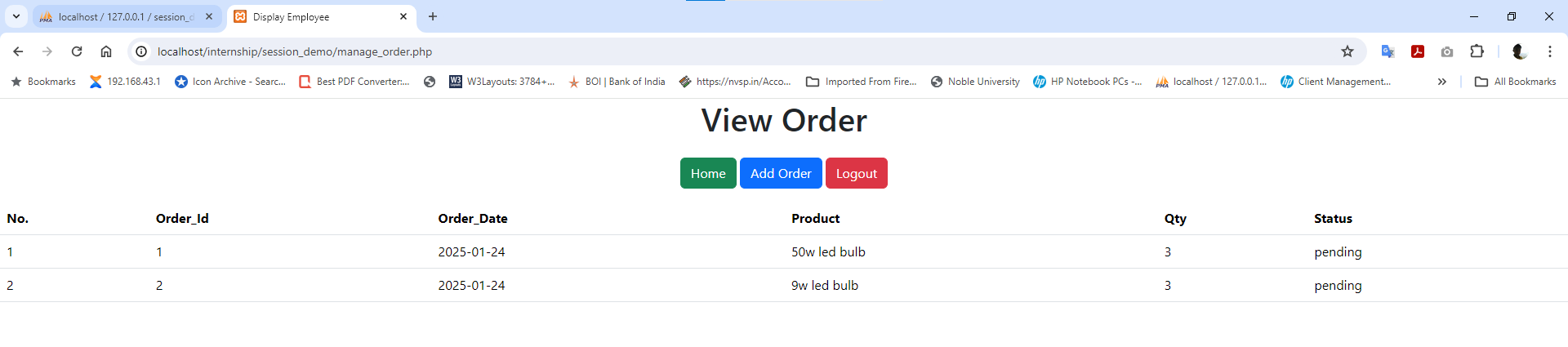
* **Index Page:**



* **Add Order Page:**



* **Manage Order Page:**



**5. Login & Logout User, Change Password and Forgot Password**

* **Login.php:**

• Once registered, users can access their accounts by logging in with their credentials, typically a combination of username/email and password.

• The login process verifies the provided information against the stored data in the database.

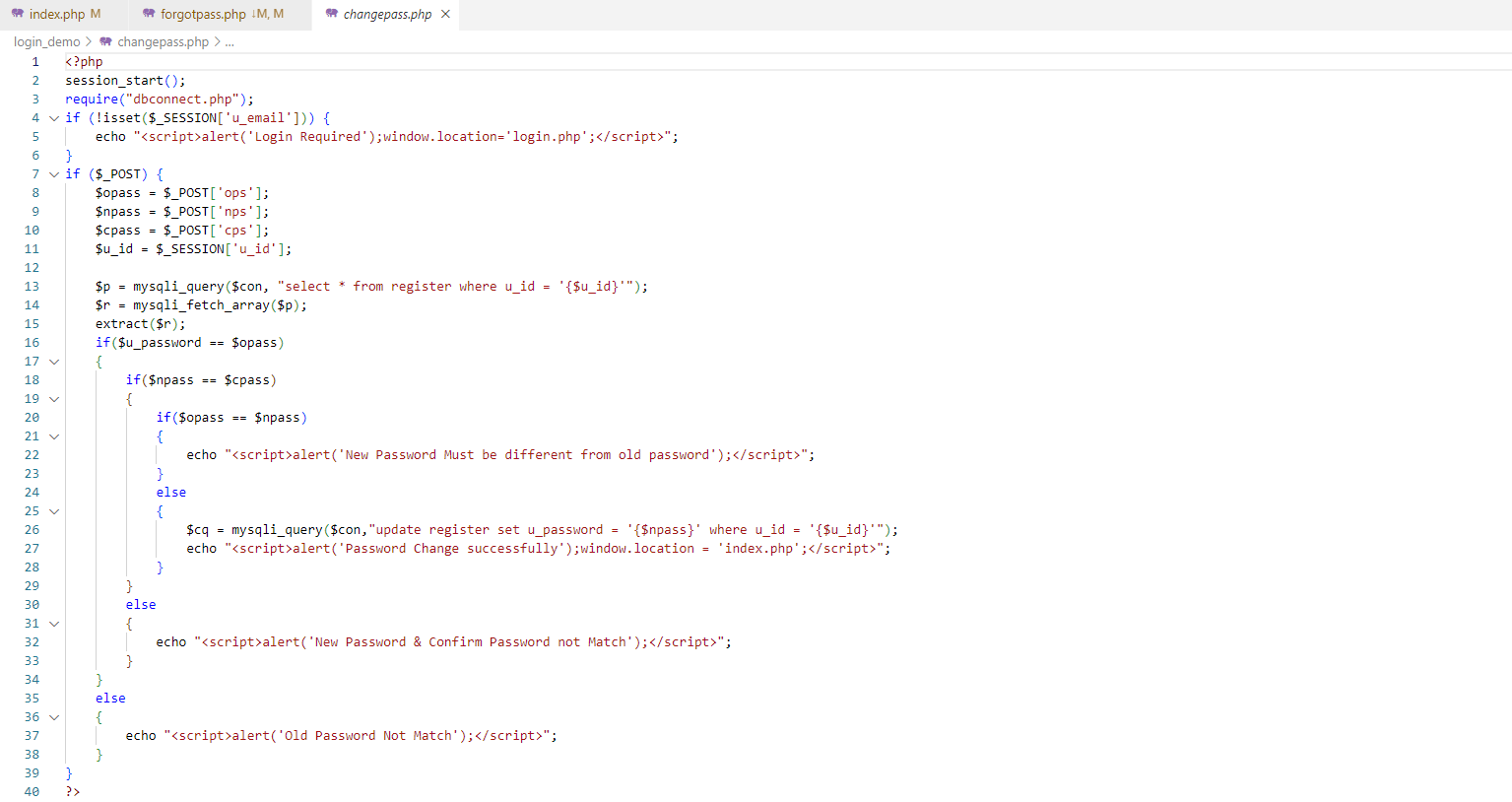
• Upon successful authentication, users can access to change password.



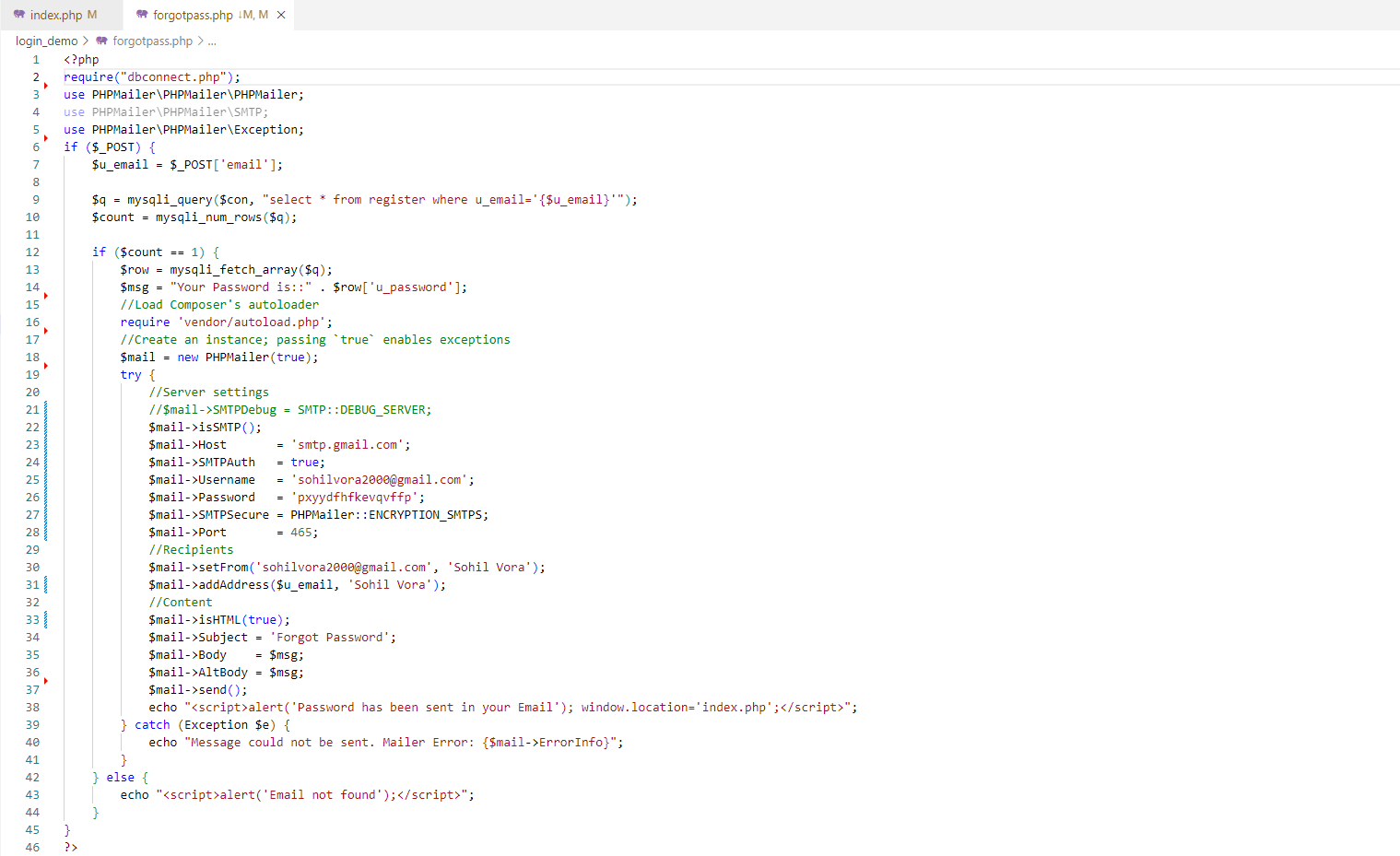
* **index.php:**



* **changepass.php:**
* Users have the option to change their account password for security reasons or if they forget their current password.
* This functionality typically requires users to provide their current password for verification before allowing them to set a new one.

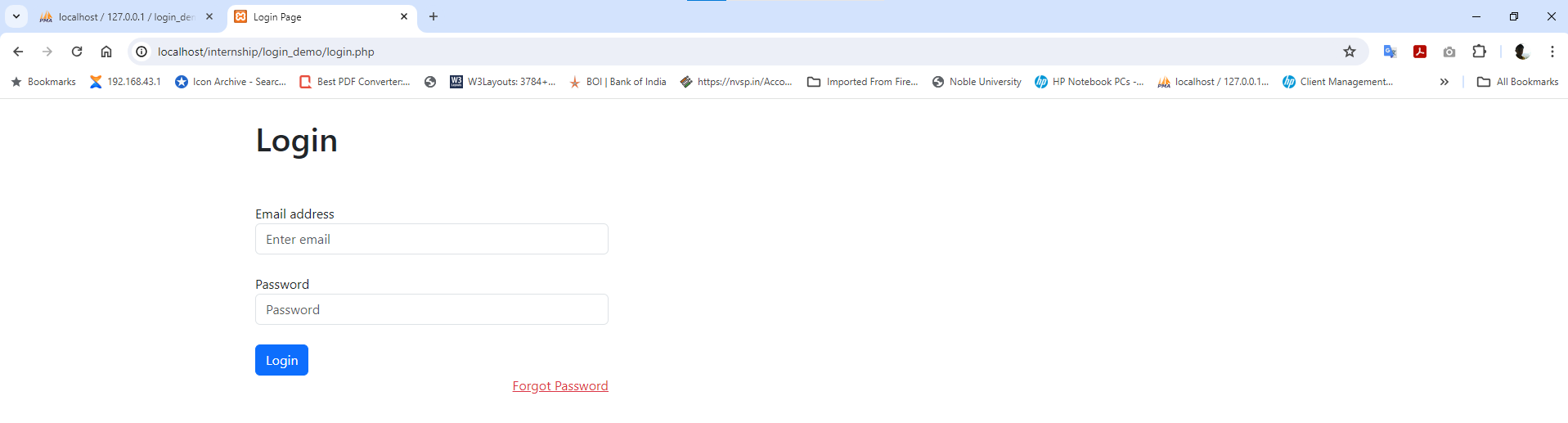


* **Forgotpass.php:**

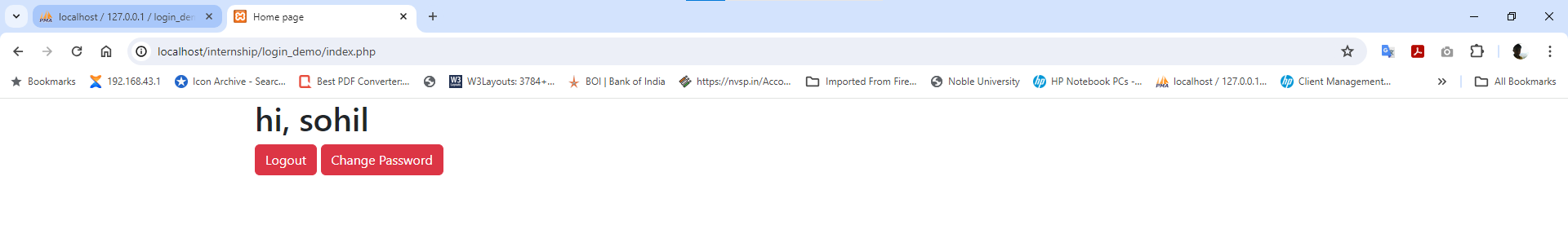


Output

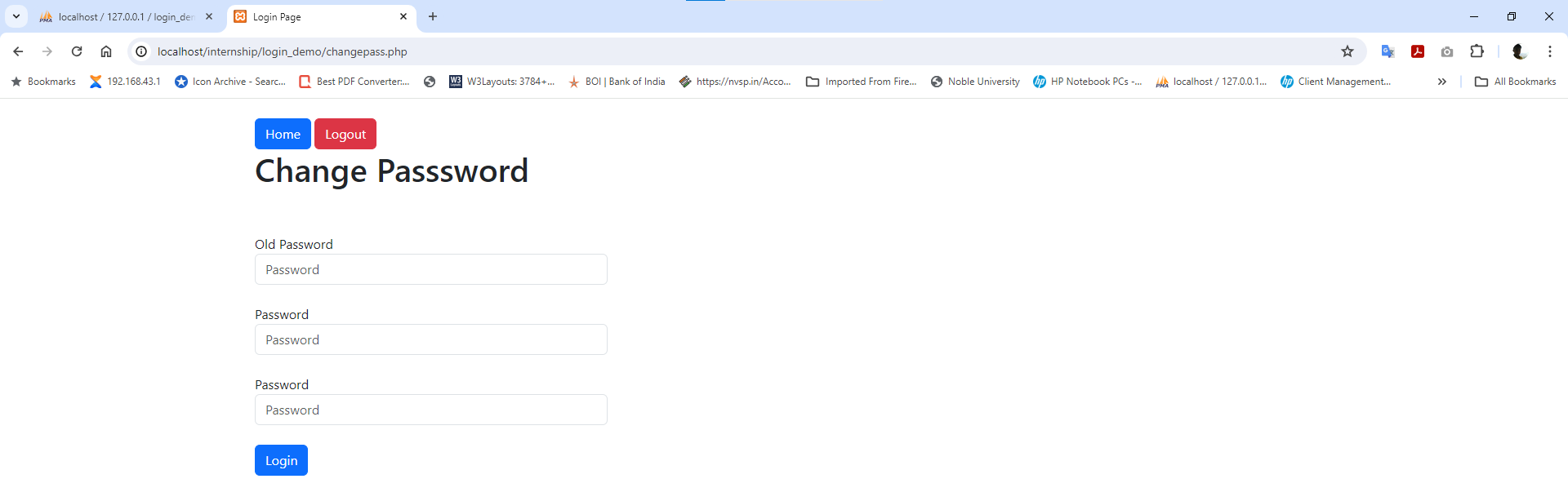
* **Login Page:**



* **Index Page:**



* **ChangePass Page:**



* **Forgotpass Page:**

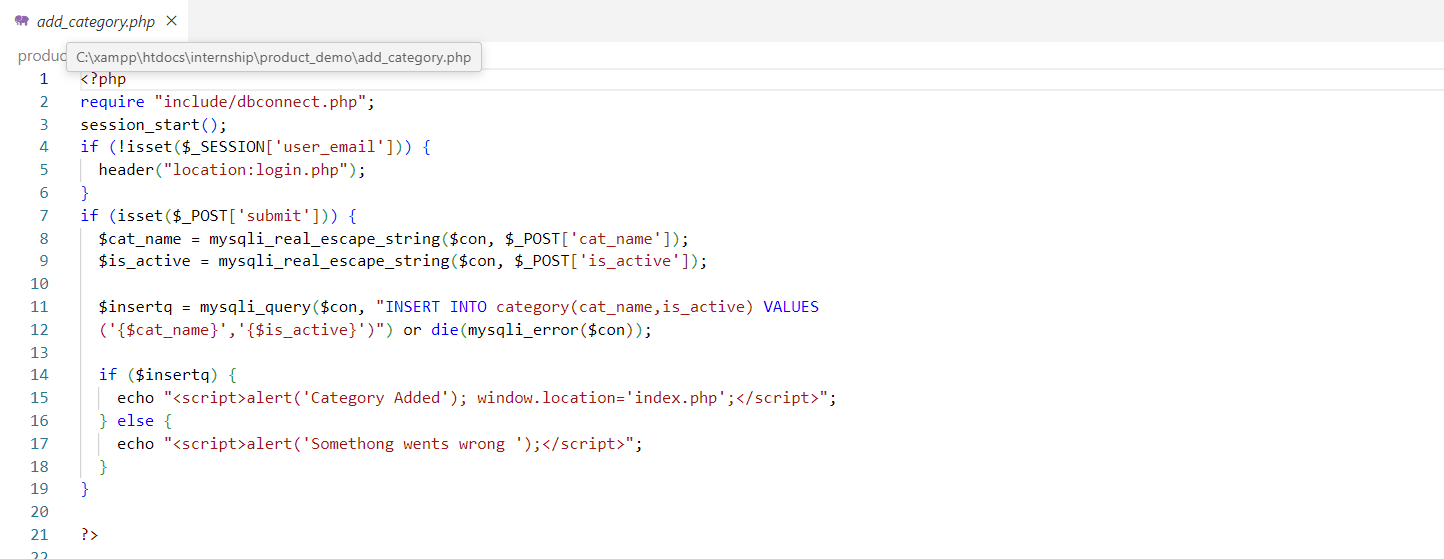


**6. E-commerce website demo, register, login, logout, display product category wise, add to cart, checkout**

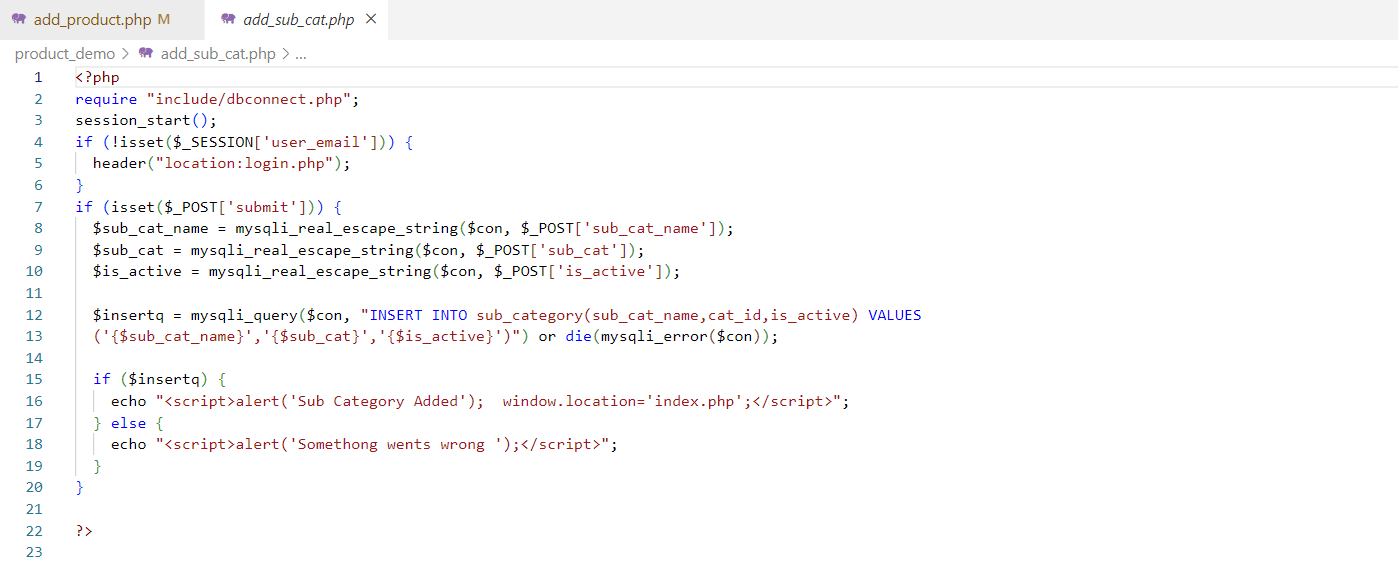
* **Login.php:**
* We create ecommerce website demo with bootstrap
* This is login file of ecommerce website demo project.



* **Add\_catgeory.php:**
* In this file user can add category after login.



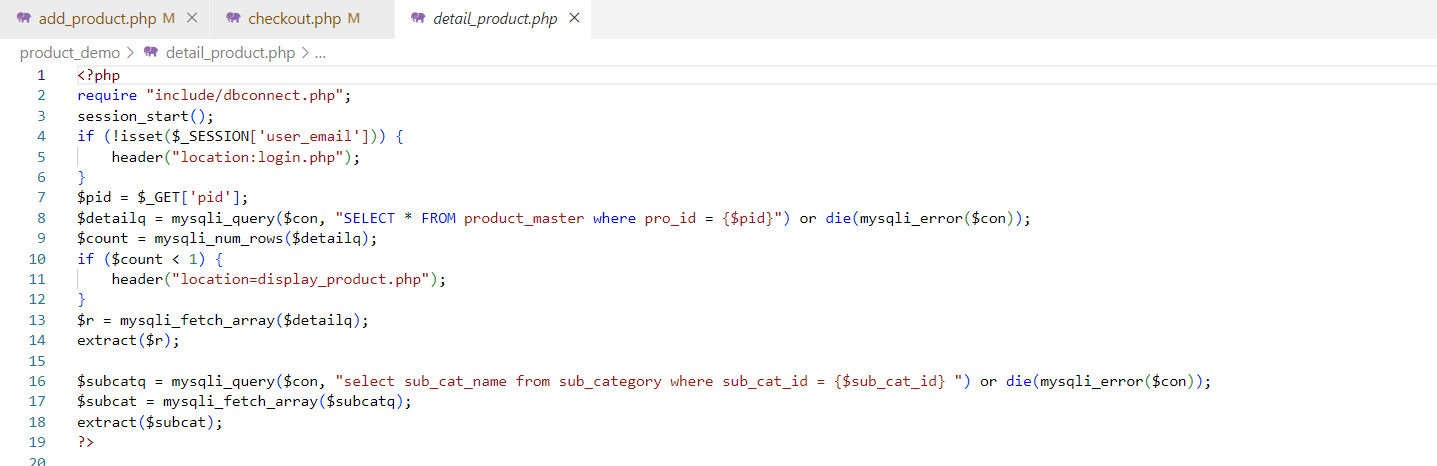
* **add\_sub\_category.php:**
* In this file user can add sub category after login.



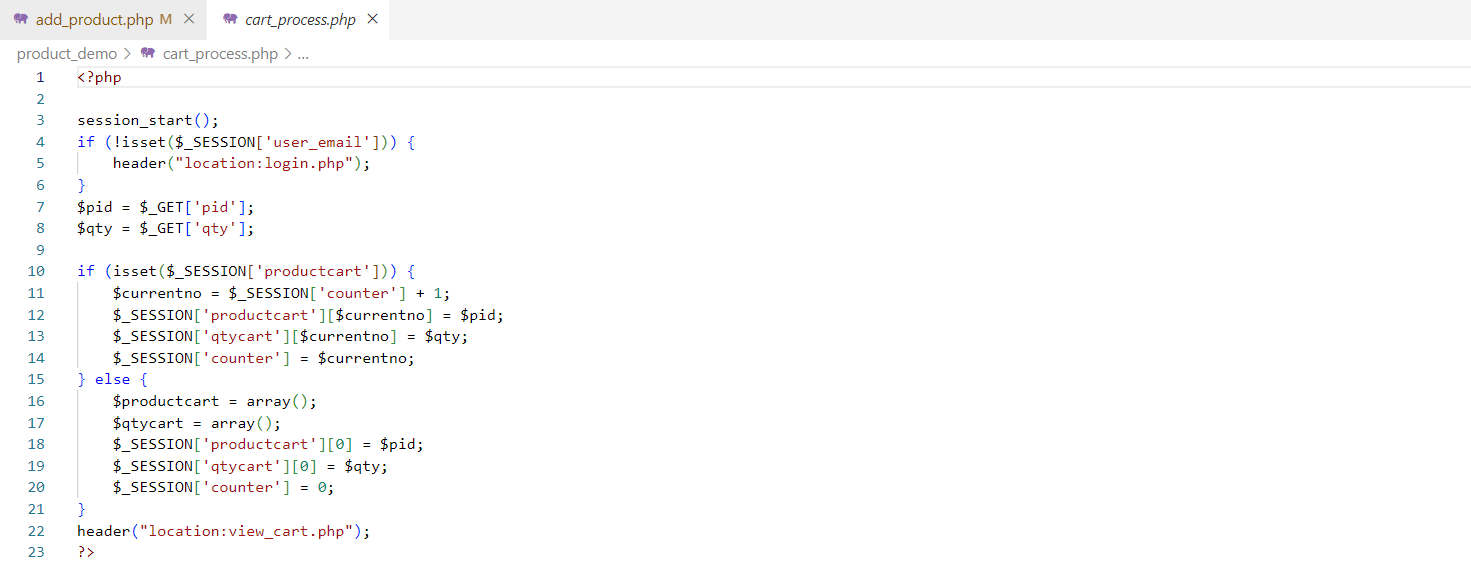
* **add\_product.php:**
* In this file user can add product after login.



* **detail\_product.php:**



* **cart\_process.php:**
* The platform features a user-friendly cart system where customers can add desired items for purchase.
* The cart provides a convenient overview of selected products, allowing users to modify quantities or remove items as needed.



* **checkout.php:**
* During the checkout process, the demo ensures a secure transaction experience.
* Users can enter shipping and payment details with ease, with options for various payment methods.
* The checkout process is designed to be intuitive, guiding users step-bystep to complete their purchase efficiently.

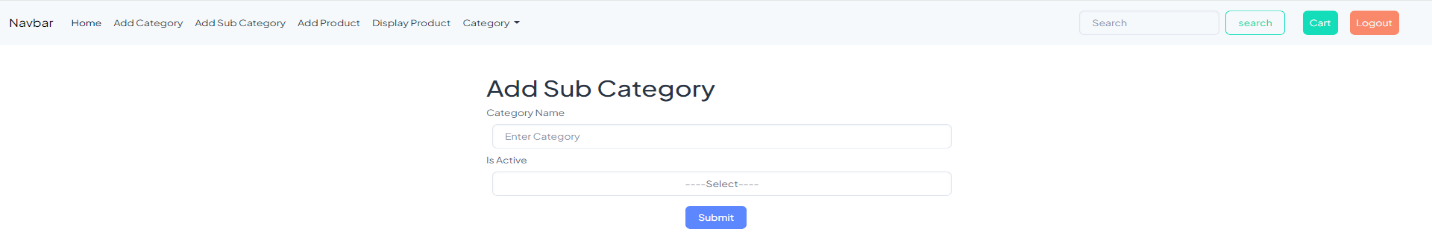


Output

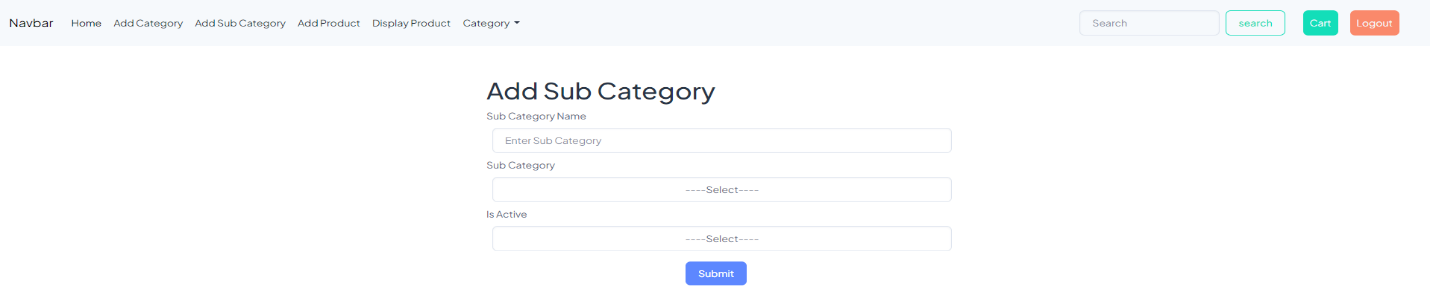
* **Login Page:**



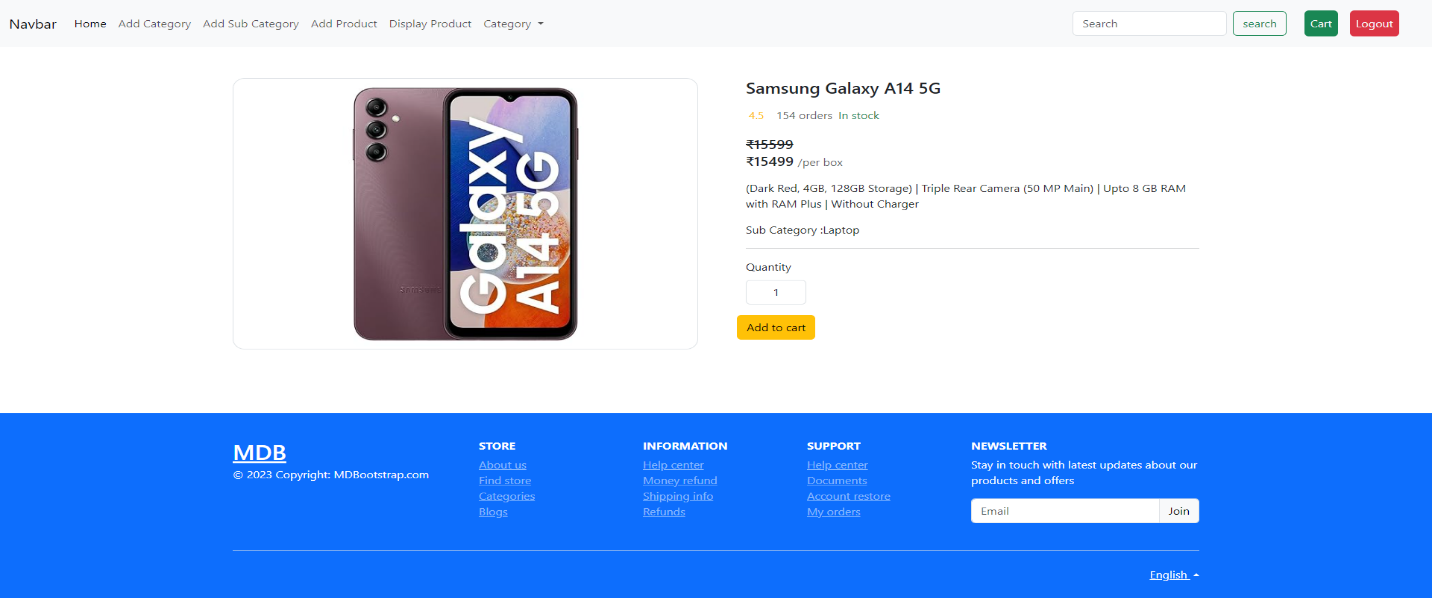
* **Add Category Page:**



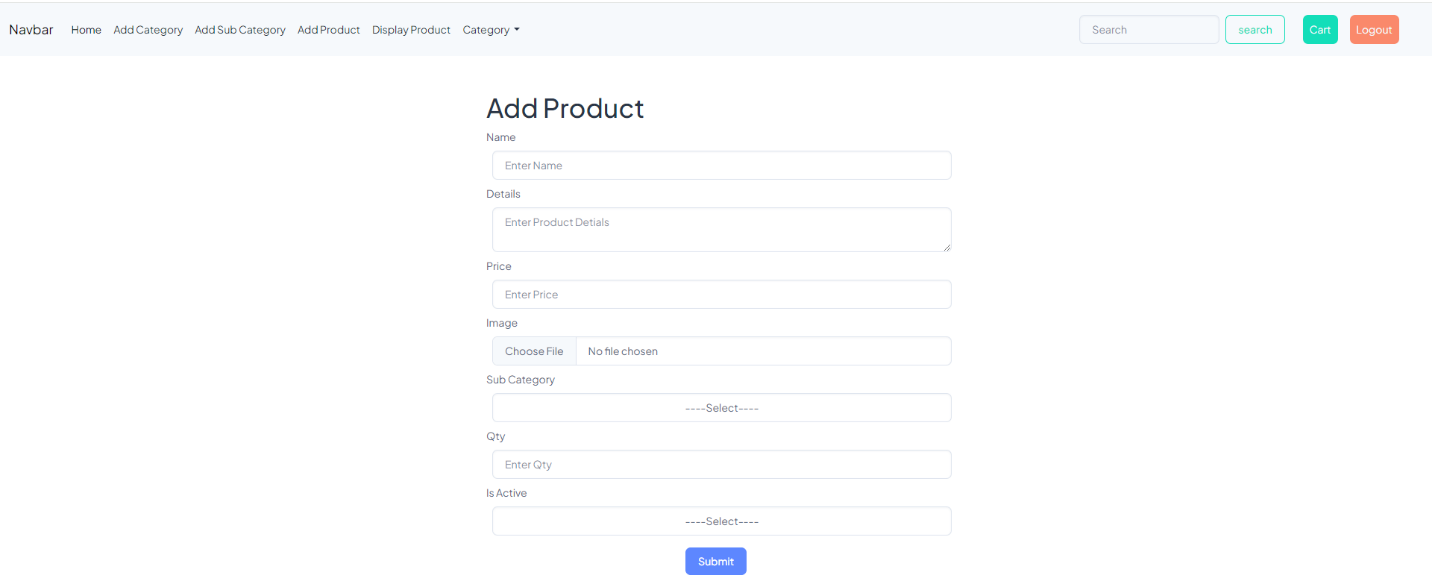
* **Add Sub Category Page:**



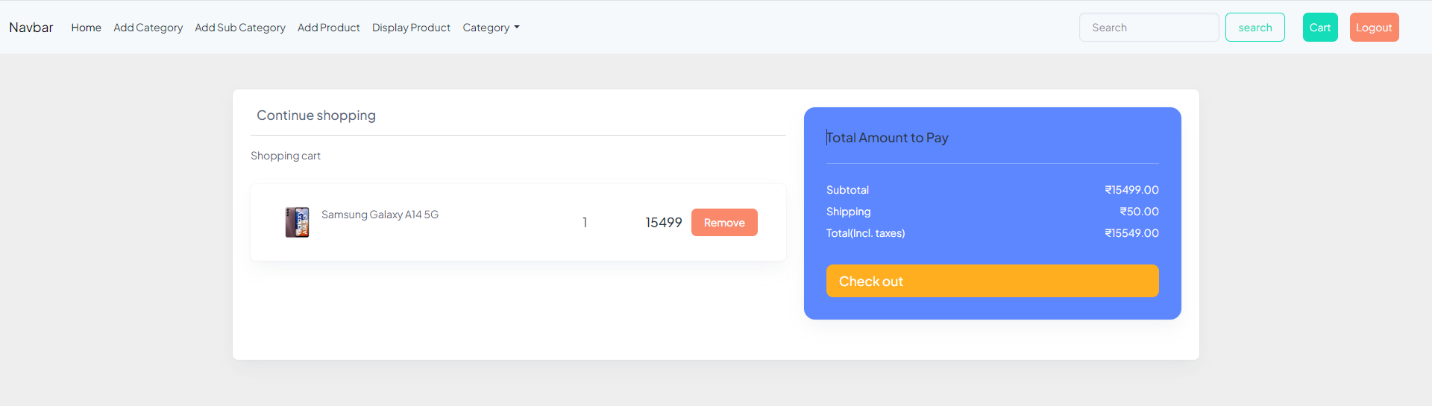
* **Detail Product Page:**



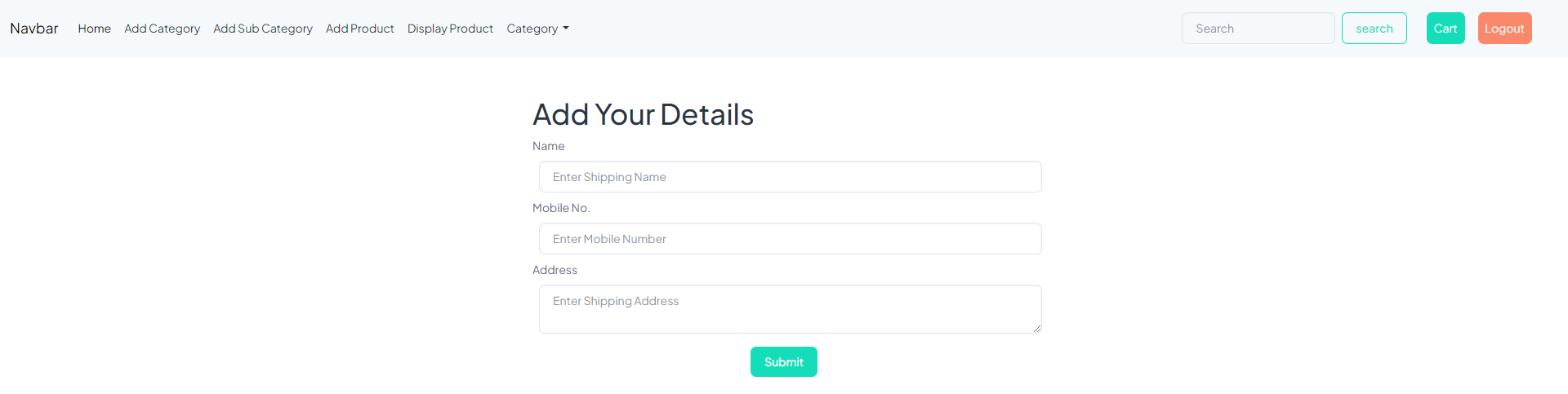
* **Add Product Page:**



* **Cart Page:**

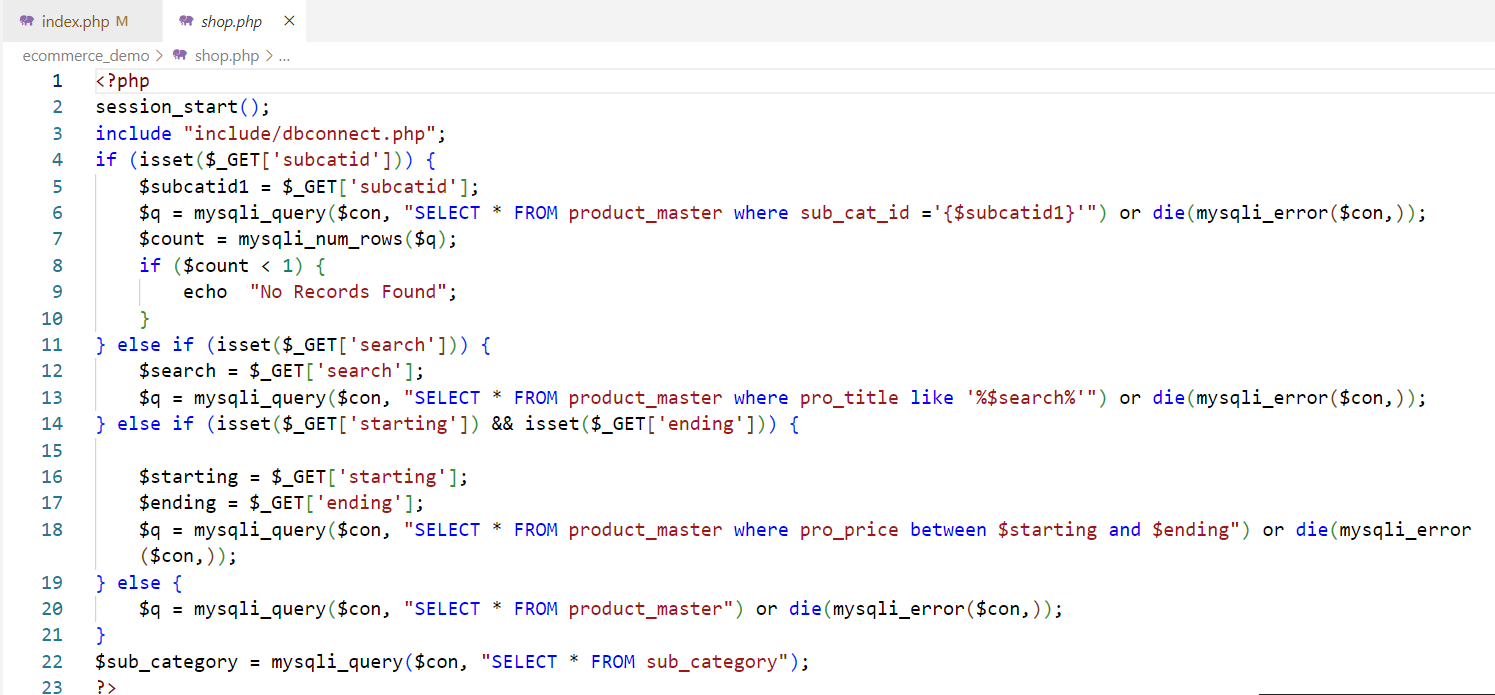


* **Checkout Page:**

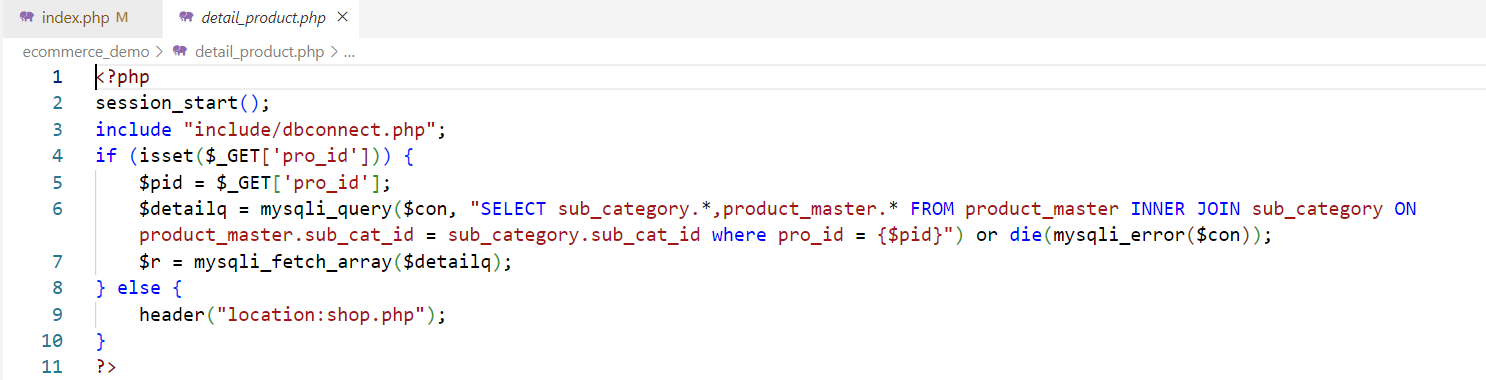


**7. Ecommerce admin and user bootstrap template, registater, login, logout, display product, category, add to cart checkout**

* An E-commerce Bootstrap template streamlines the design and functionality of online stores for both administrators and users.
* Users can register accounts, log in securely, and log out as needed.
* The template organizes products into categories for easy navigation and display, facilitating a seamless shopping experience.
* Users can conveniently add items to their carts, which are dynamically updated.
* At checkout, they can review their orders and proceed to payment.
* Admins have access to backend functionalities for managing products, categories, orders, and user accounts, ensuring smooth operation of the online store.
* **shop.php:**



* **detail\_product.php:**



* **cart.php:**

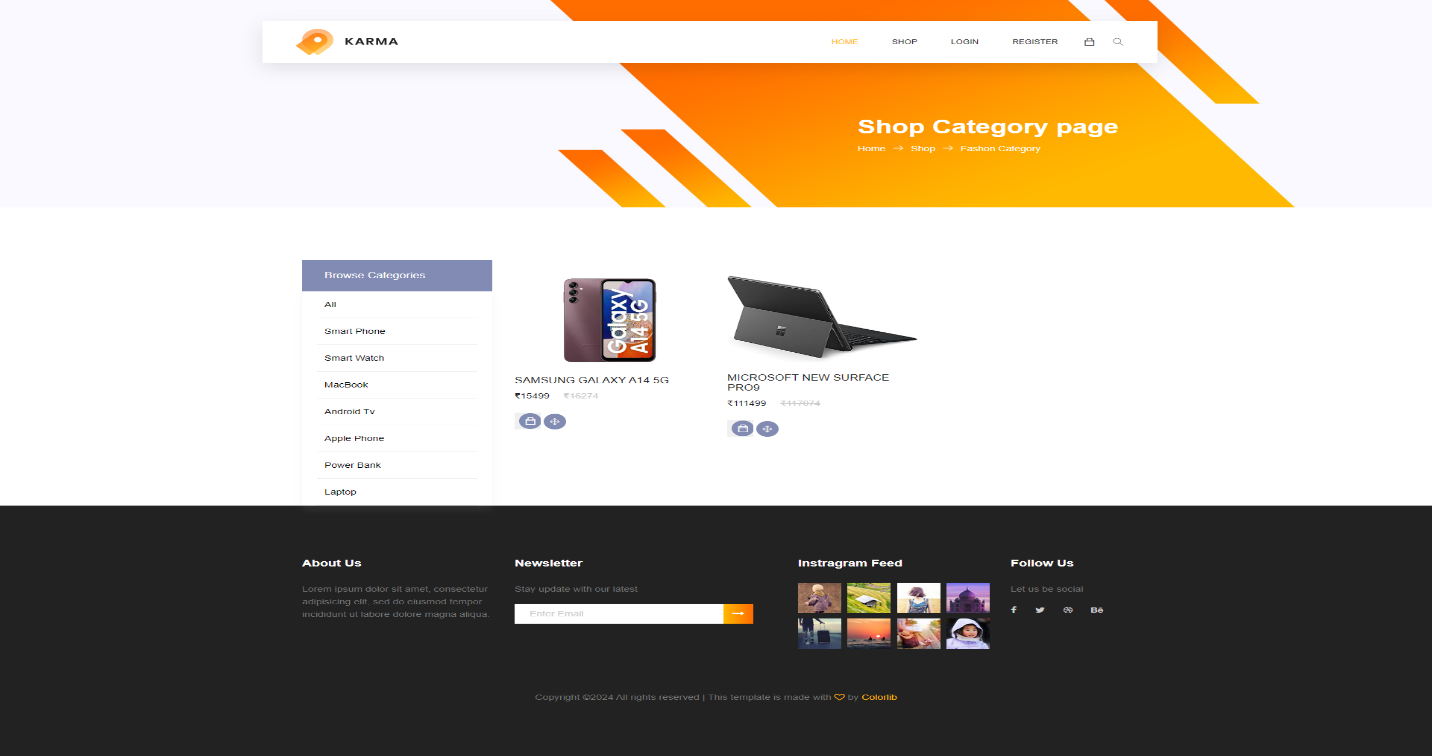


* **checkout.php:**

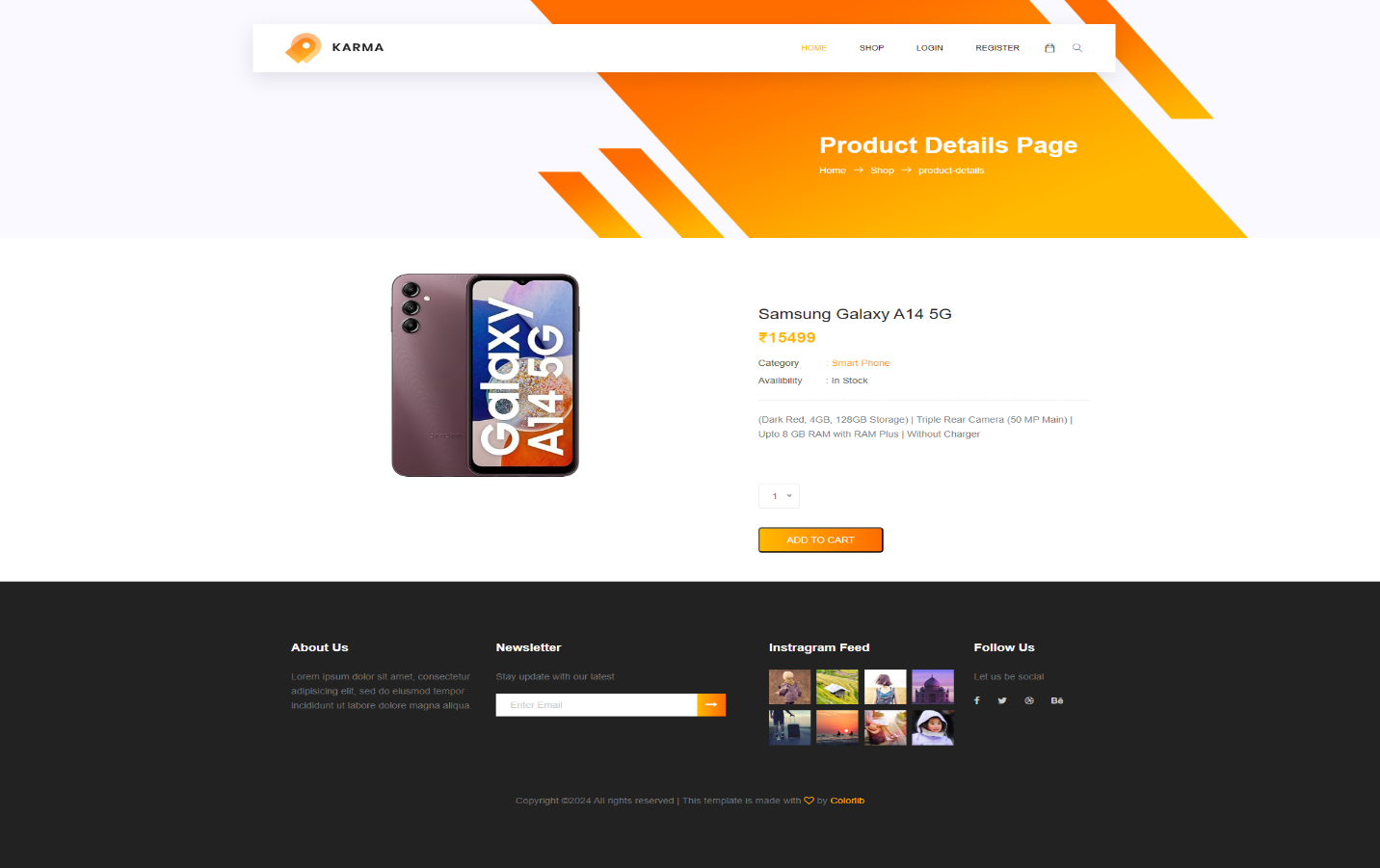


Output

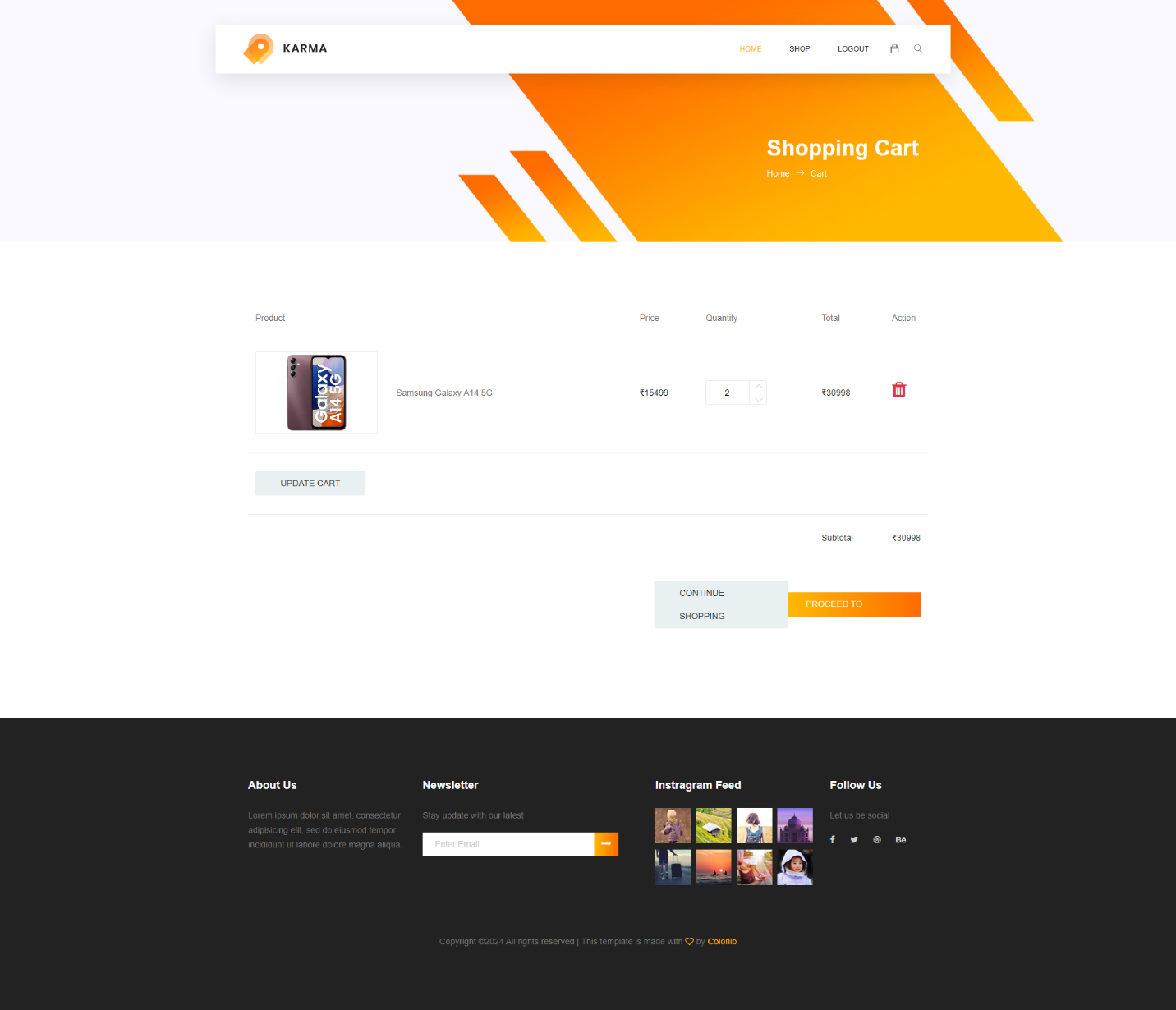
* **Shop Page:**



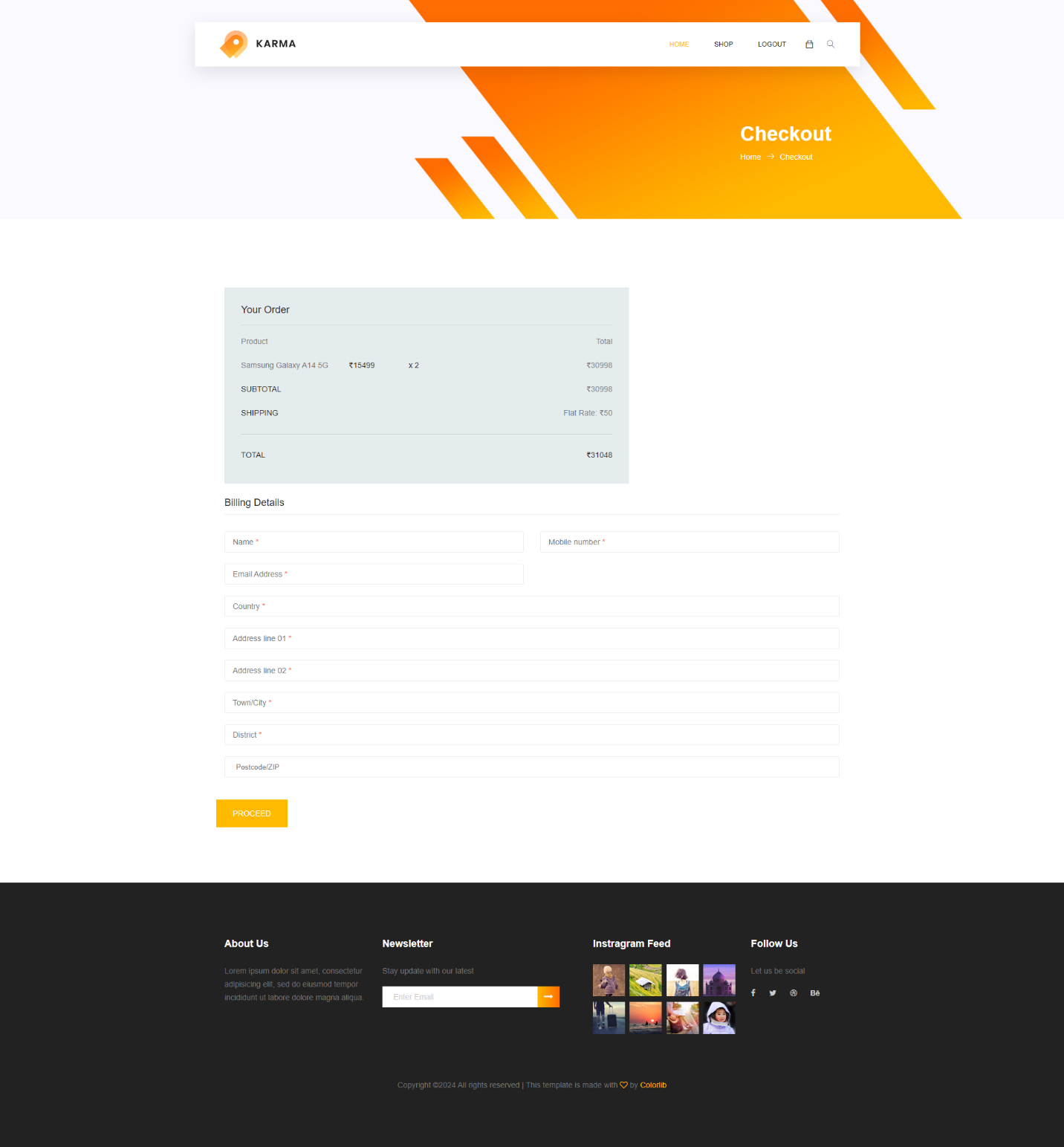
* **Product Detail Page:**



* **Cart Page:**



* **Checkout Page:**



**8. Ecommerce admin and user with PDO statement using MySQL**

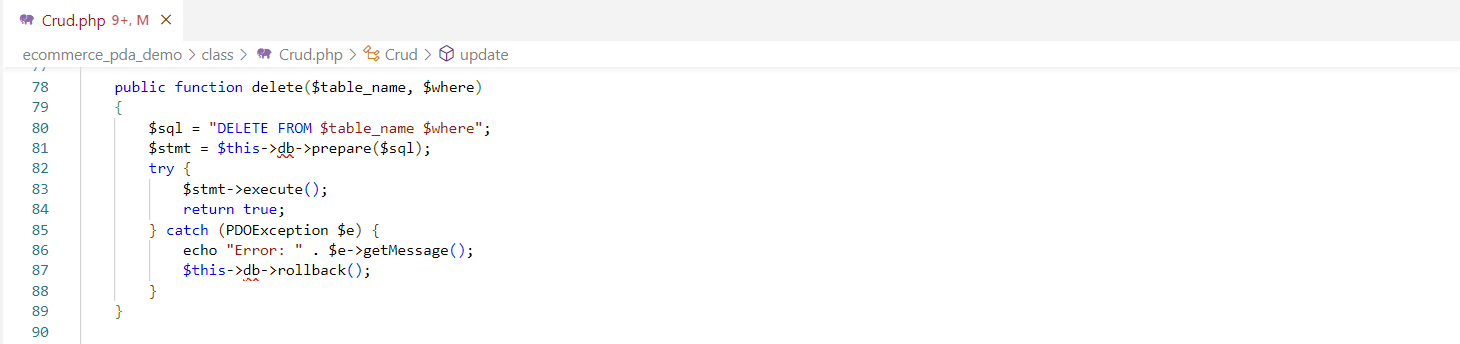
* **User Interface**: A frontend where users can view products, add them to cart, make purchases, and manage their accounts.
* **Admin Interface**: A backend dashboard accessible only to admins, allowing them to add, edit, and remove products, manage user accounts, handle orders, and monitor site activity.
* **Database Structure**: Utilizing MySQL to store data such as user details, product information, orders, and administrative records.
* **PHP with PDO**: PHP scripts are employed to handle server-side logic, including authentication, data validation, and communication with the database using PDO (PHP Data Objects) to ensure secure and efficient database operations, guarding against SQL injection attacks.
* **Add Category ,Brand & Product Crud.php:**
* **Fetch Category ,Brand & Product Crud.php:**



* **Update Category ,Brand & Product Crud.php:**



* **Delete Category ,Brand & Product Crud.php:**



* **Pagination & Slugify for Category, Brand & Product Crud.php:**

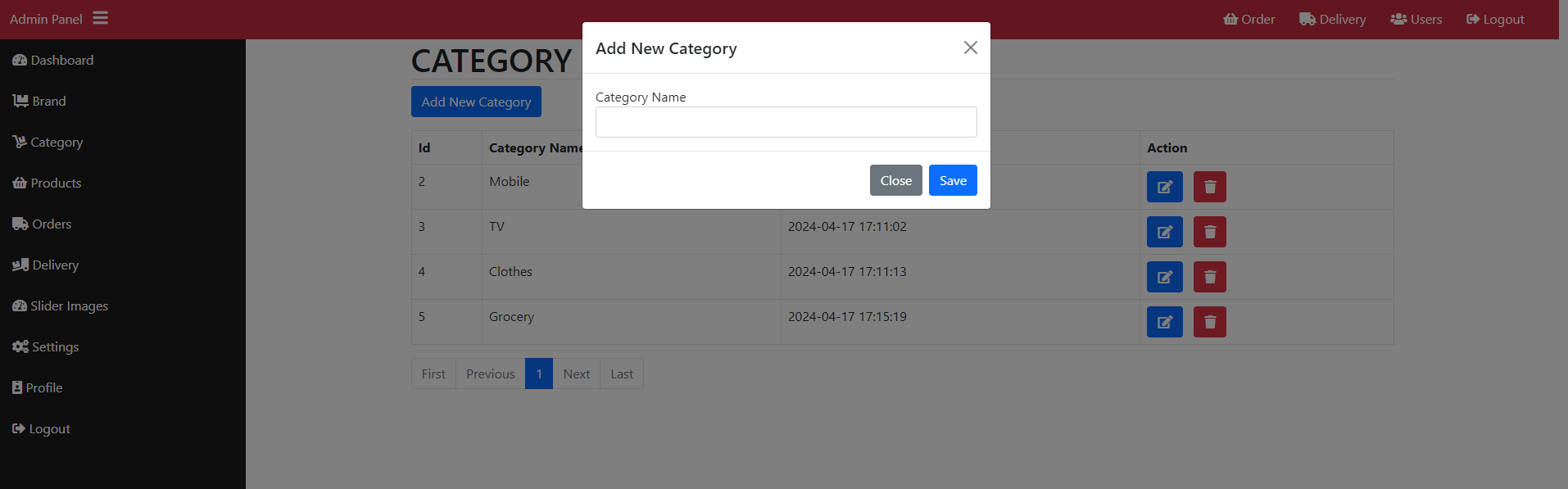


* **Database Db.php:**

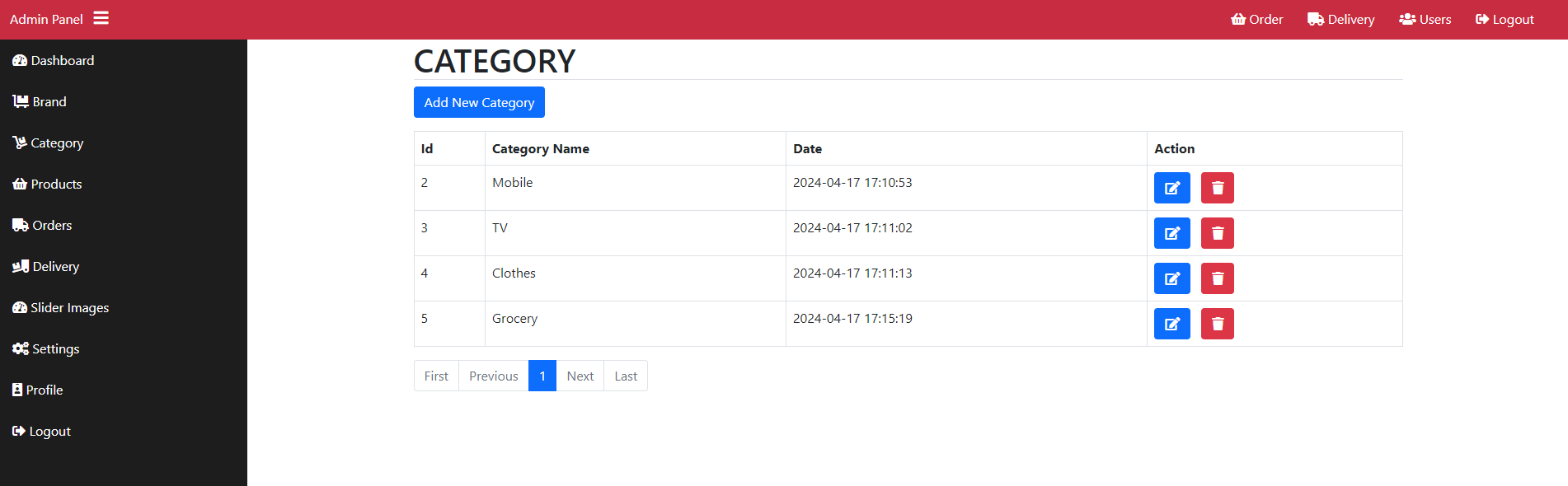


Output:

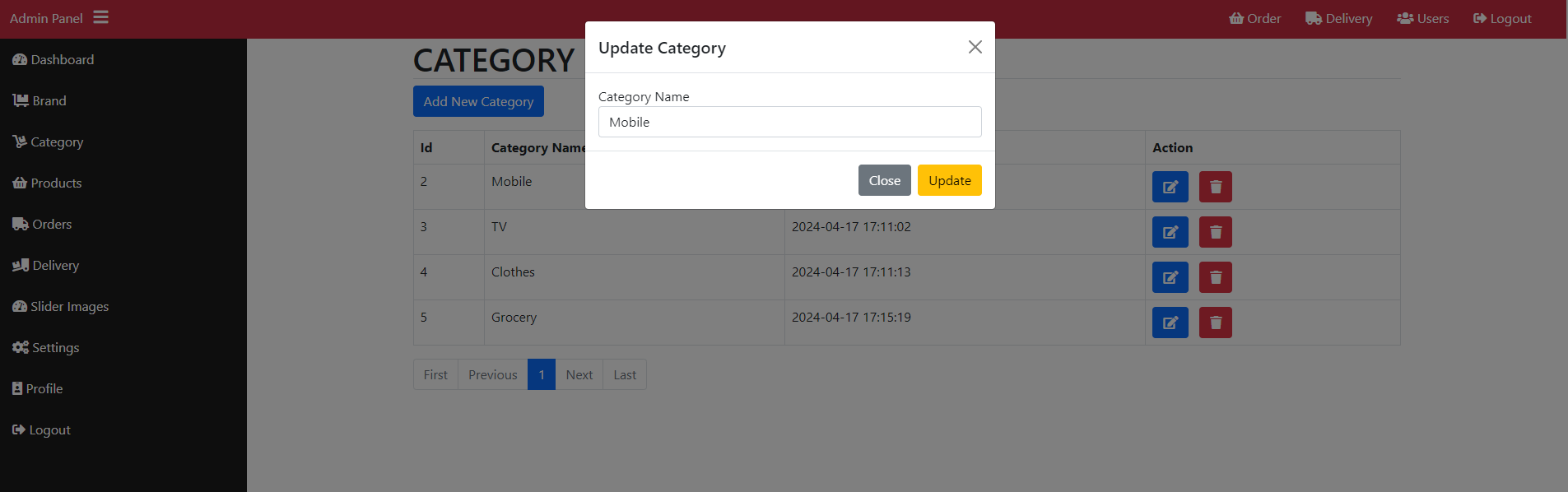
* **Add Category: Admin Panel:**



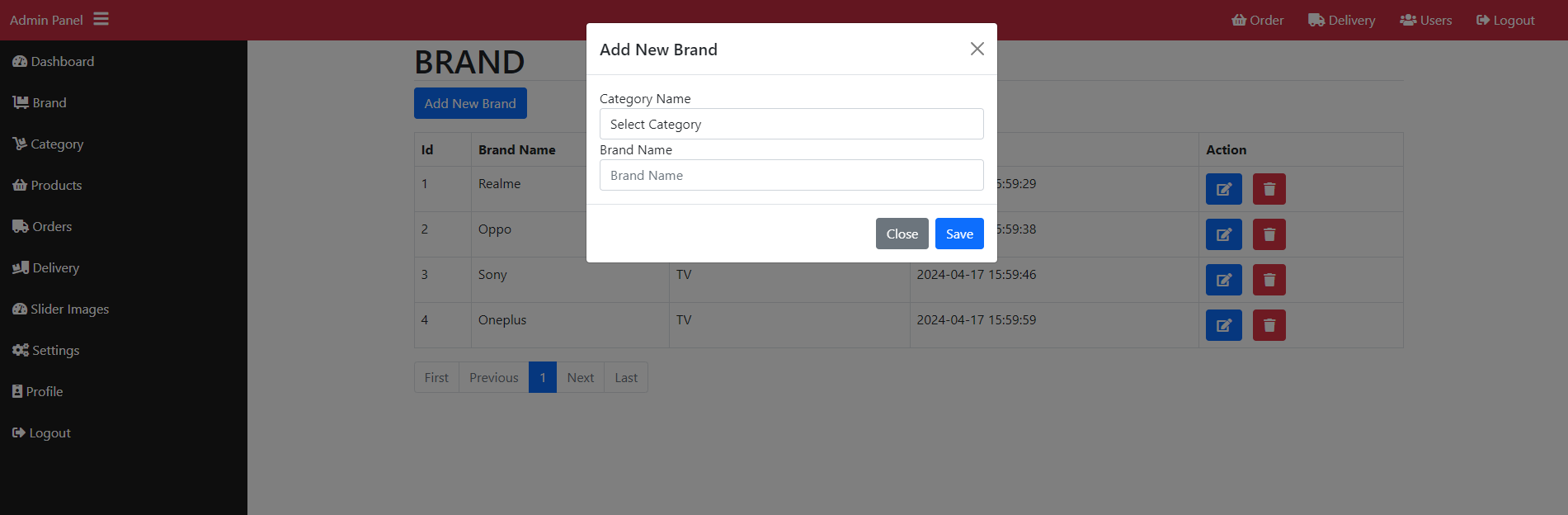
* **Display & Delete Category: Admin Panel:**



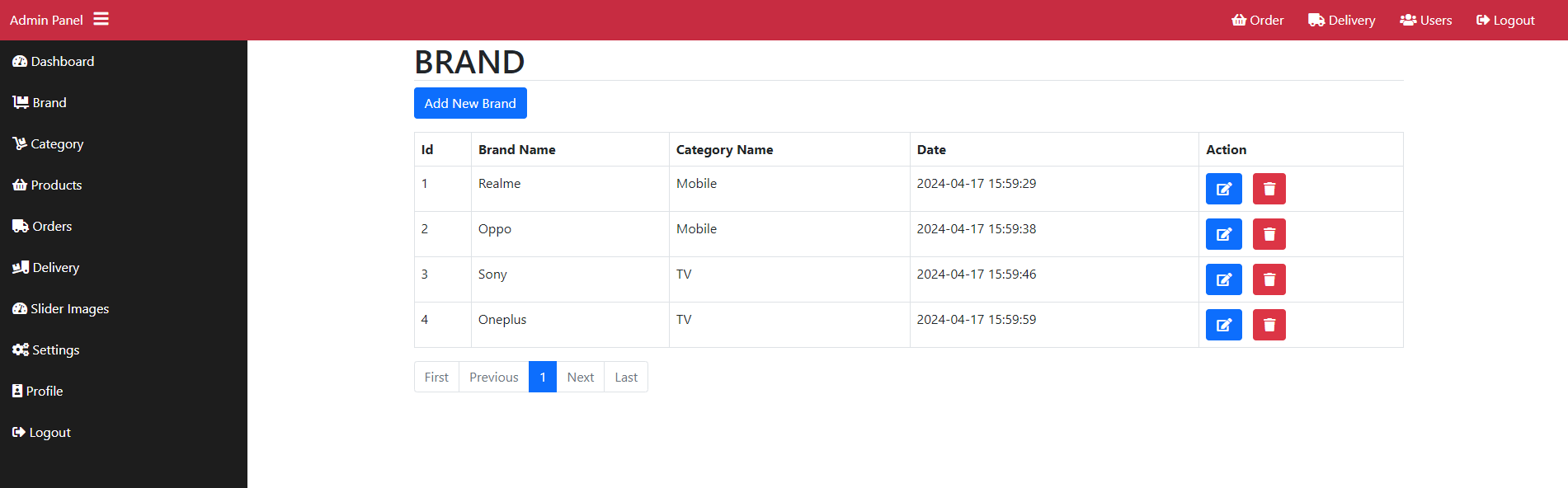
* **Update Category: Admin Panel:**



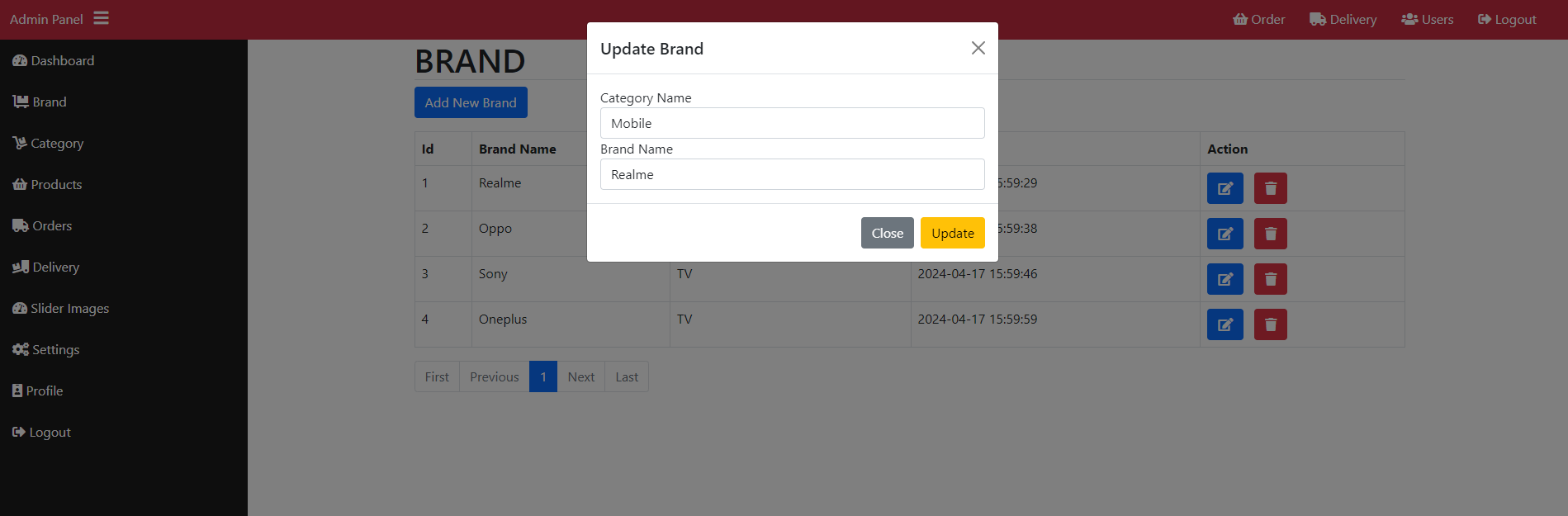
* **Add Category: Admin Panel:**



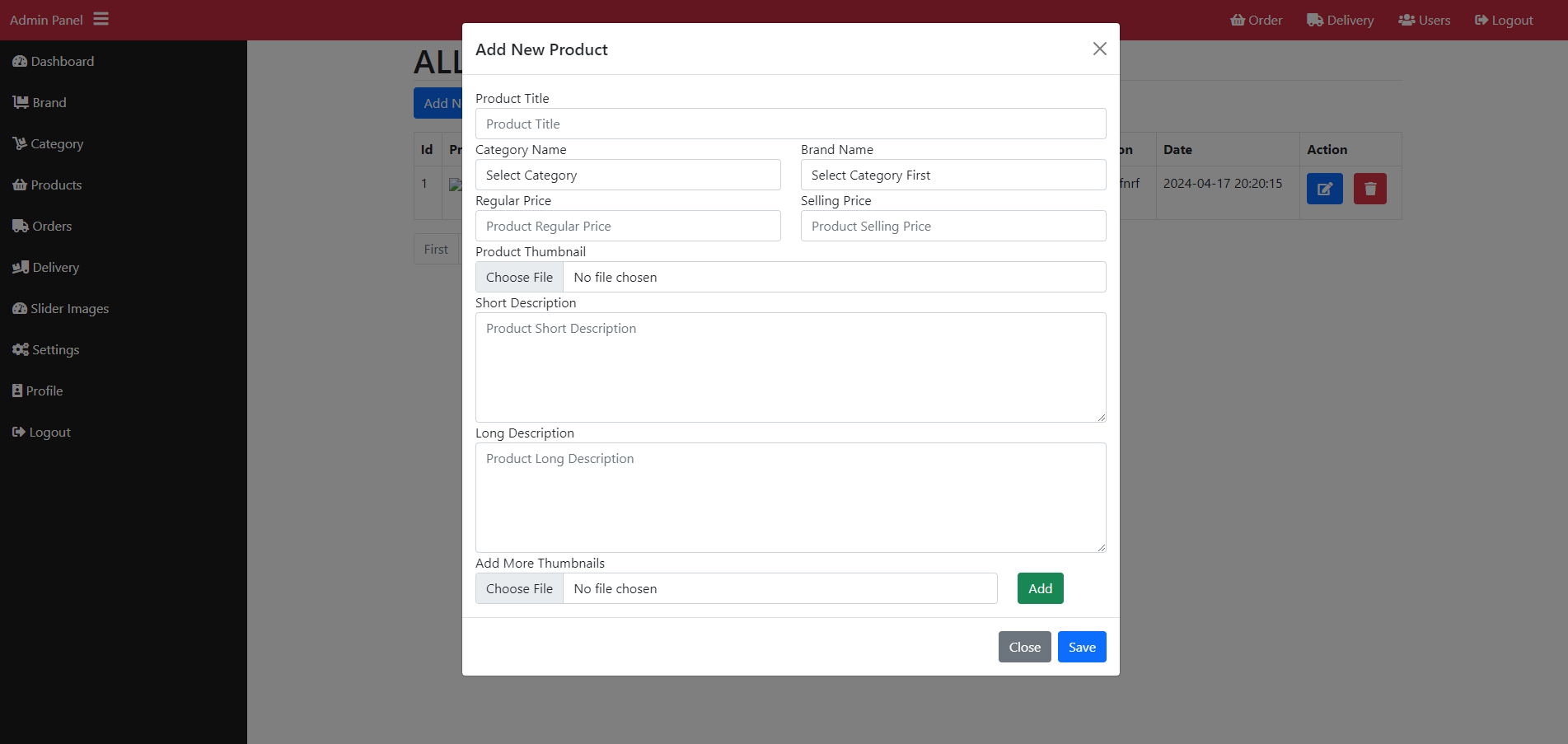
* **Display & Delete Category: Admin Panel:**



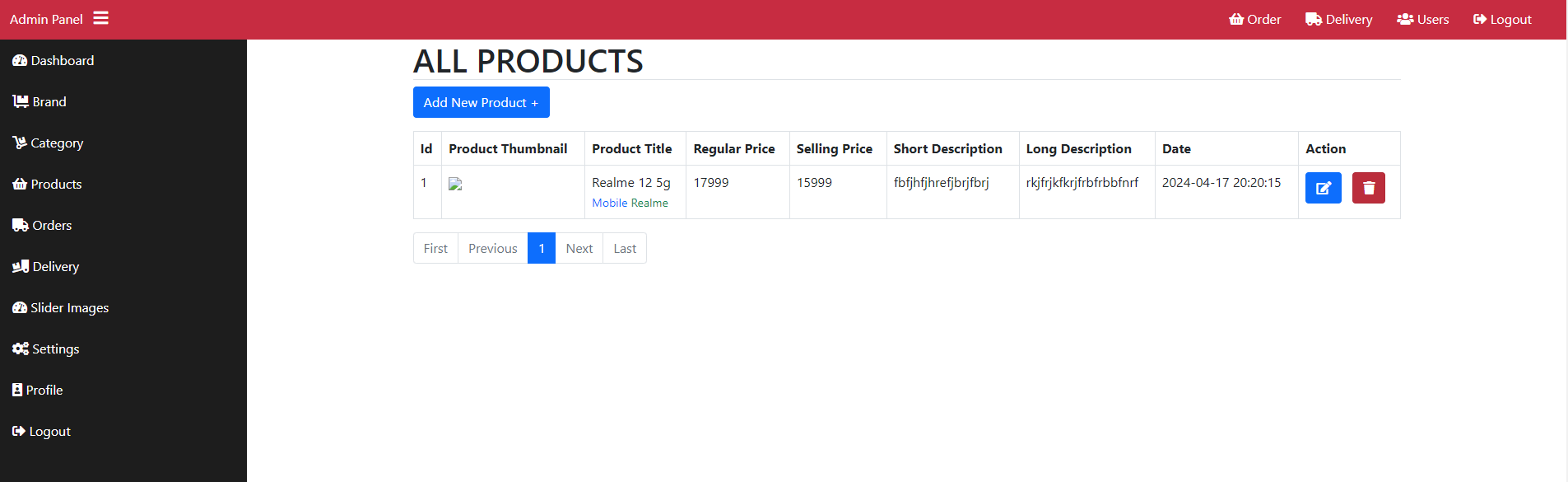
* **Update Category: Admin Panel:**



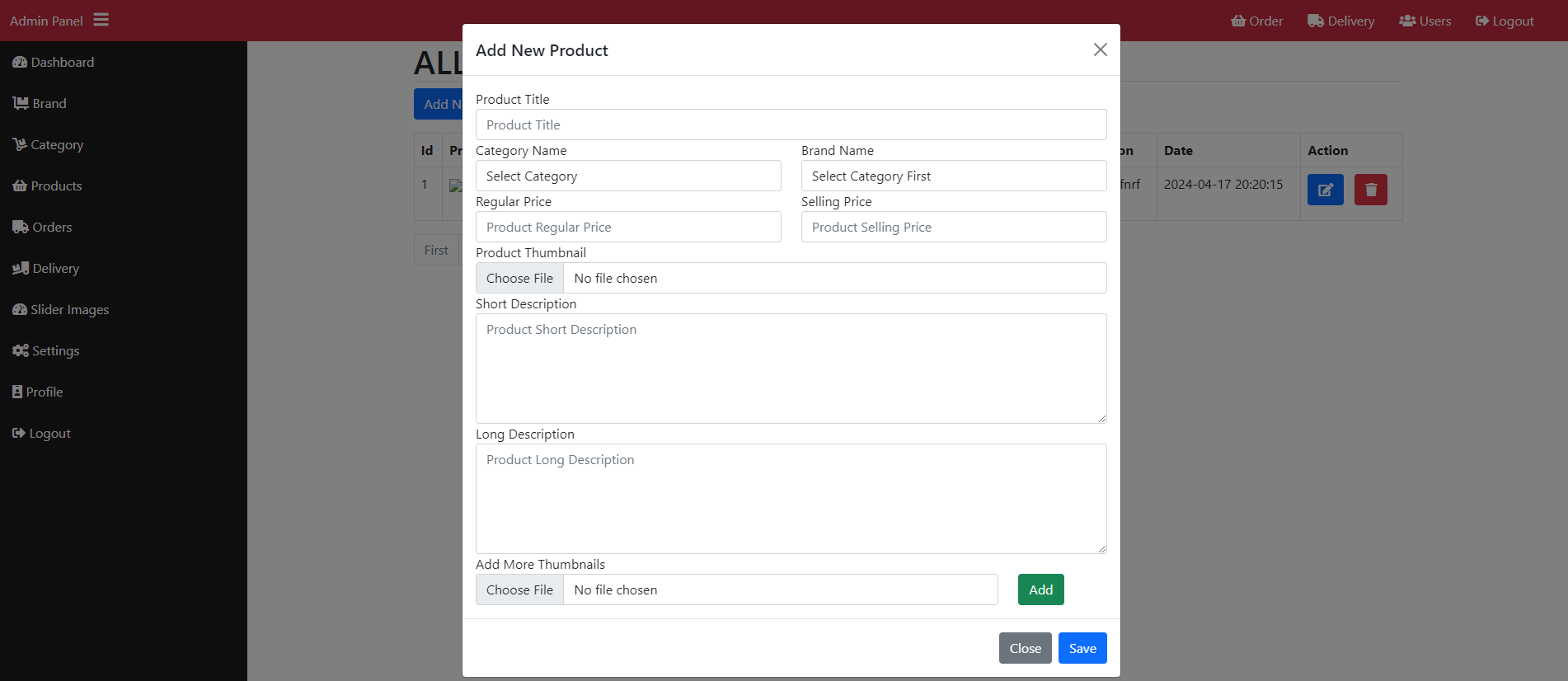
* **Add Product: Admin Panel:**



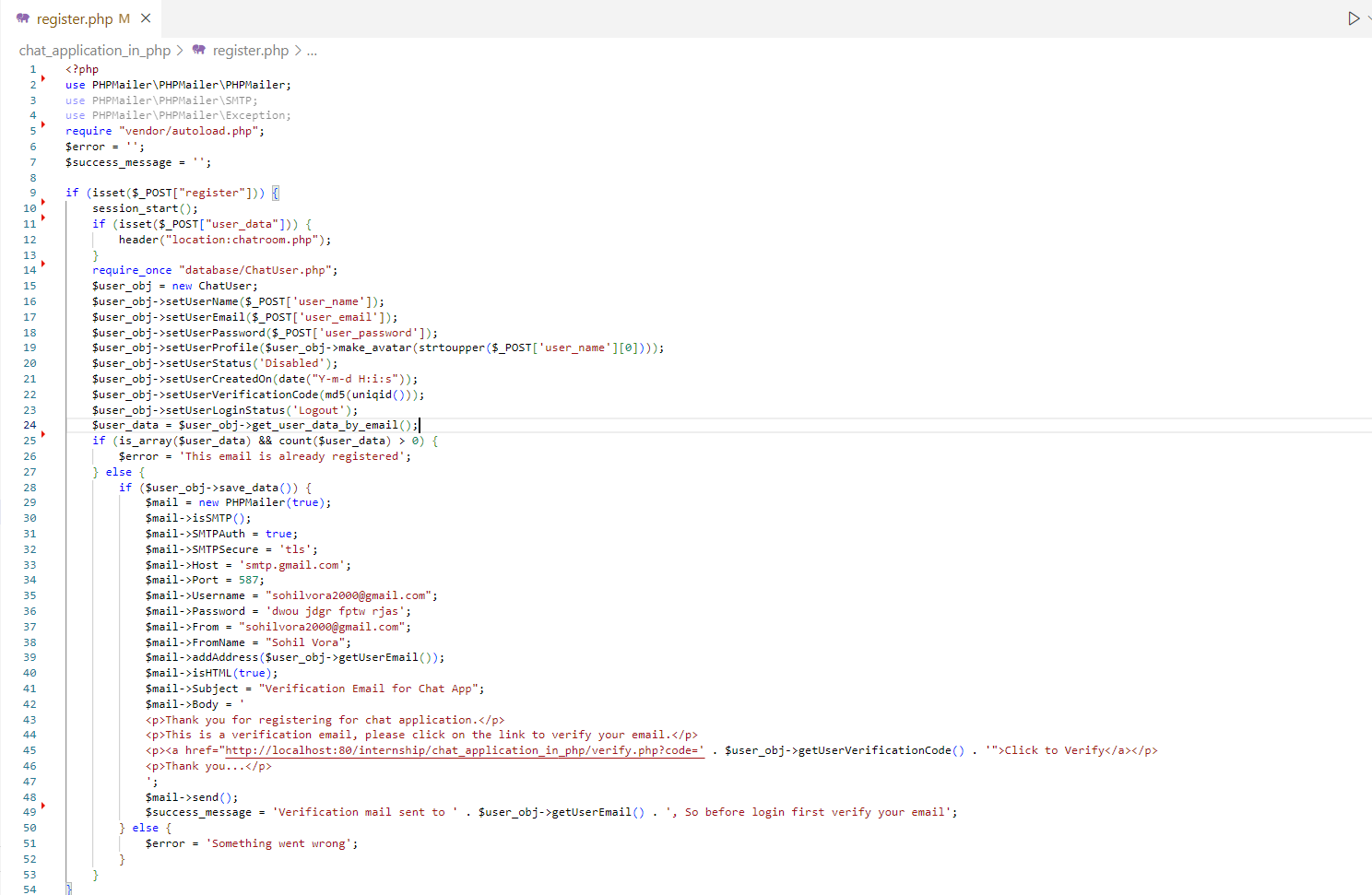
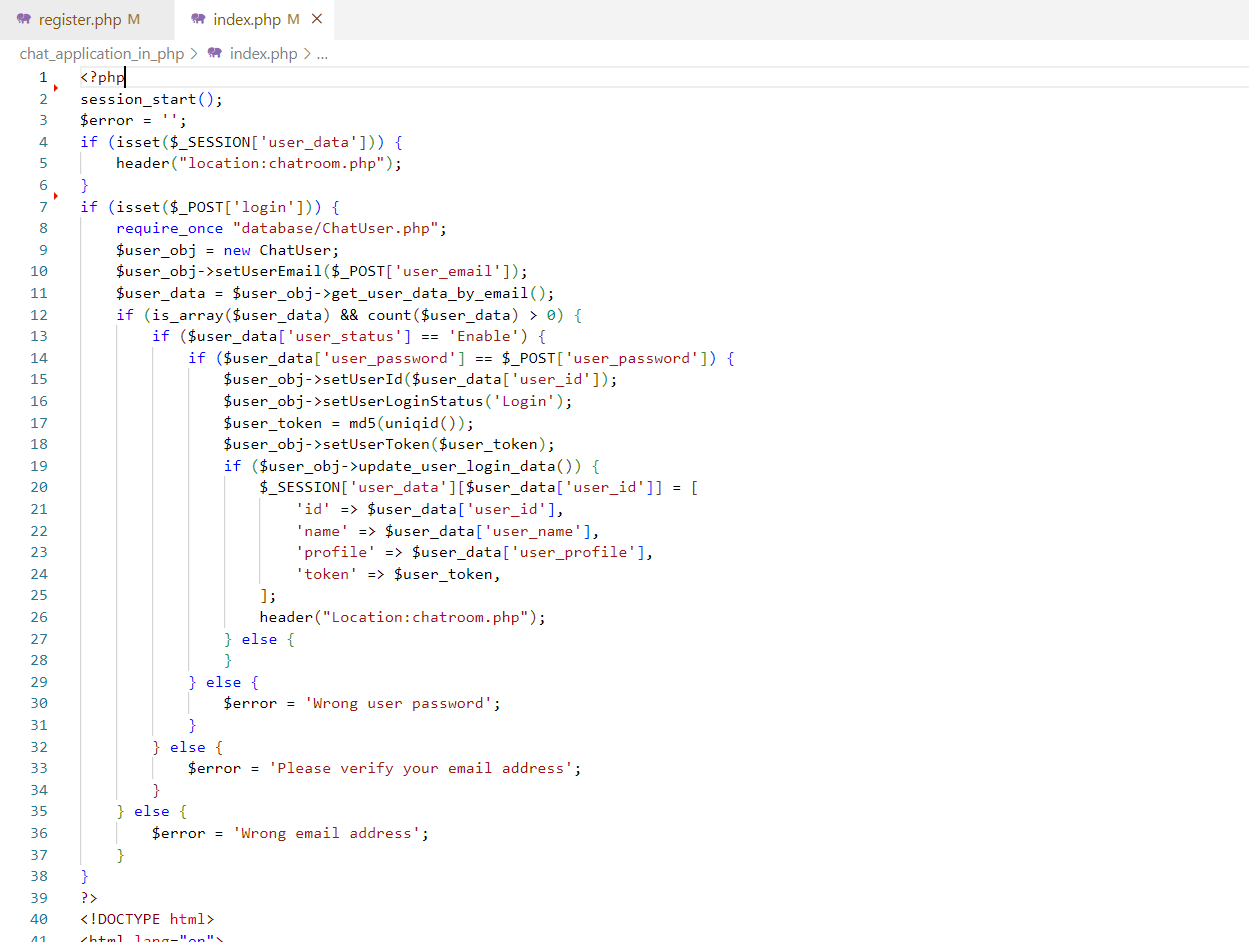
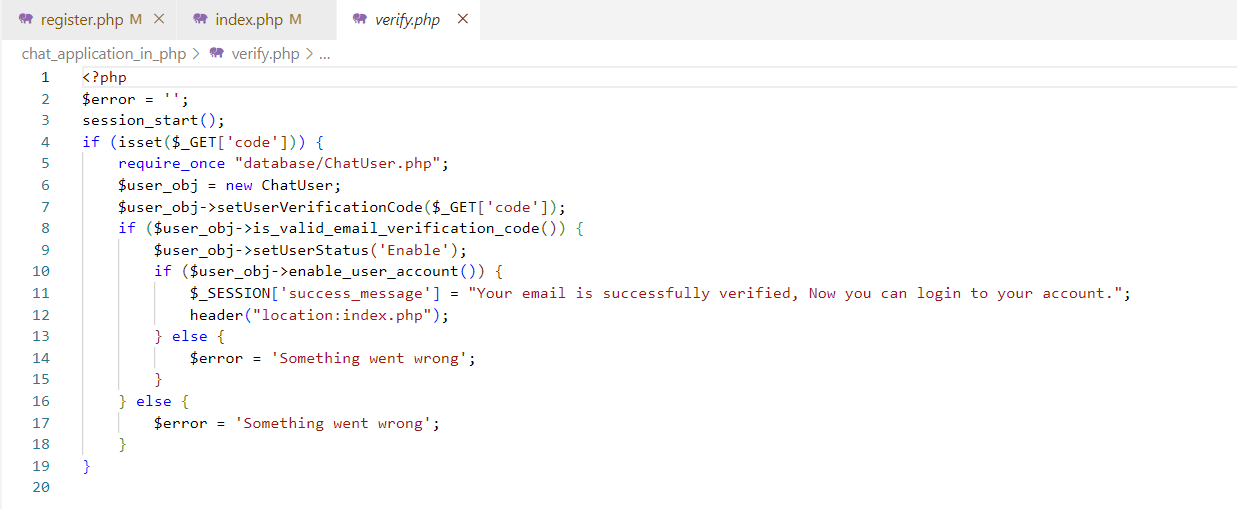
* **Display & Delete Product: Admin Panel:**

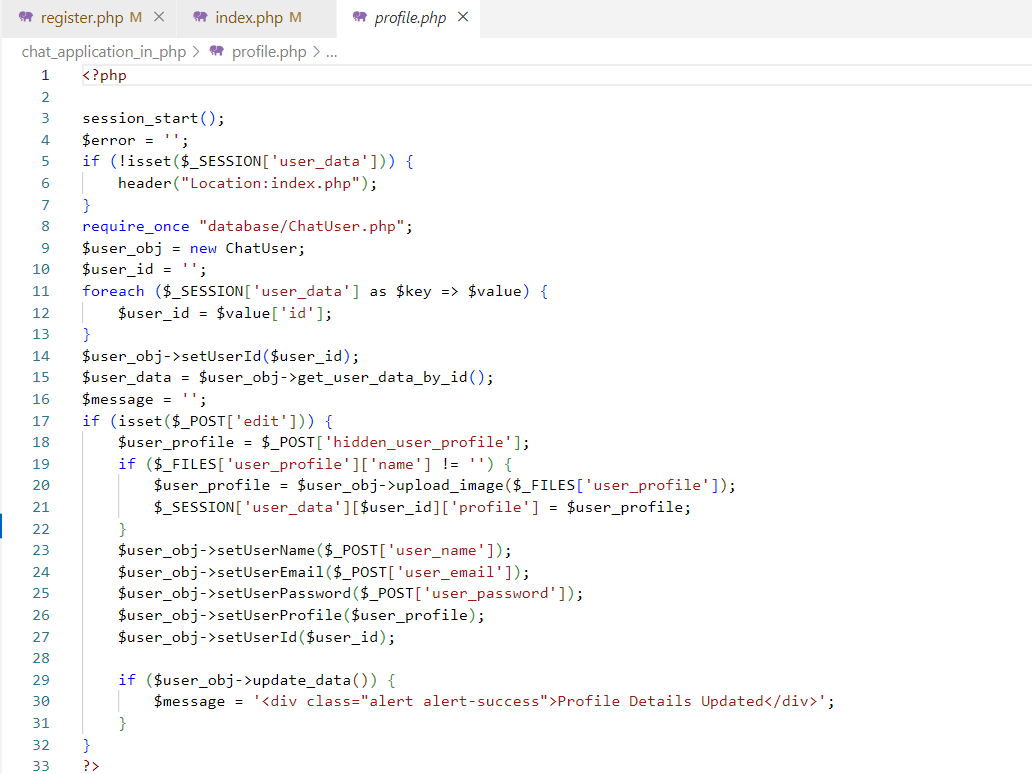


* **Update Product: Admin Panel:**



**9. Chat App in Php with MySQL using Ratchet WebSocket**

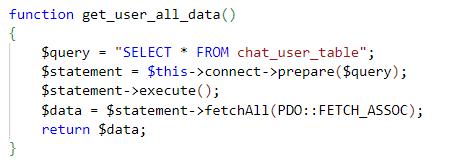
* A Realtime chat app built in PHP with MySQL using Ratchet WebSocket allows users to communicate instantly, resembling the functionality of popular messaging platforms.
* Ratchet WebSocket facilitates bidirectional communication between the server and clients in real-time.
* MySQL serves as the database to store messages, user information, and other relevant data.
* Users can send messages, view chat history, and engage in conversations seamlessly, with updates reflecting instantaneously across all connected clients.
* This setup ensures efficient and responsive communication, enhancing the user experience in the chat application.
* **Register.php:**
* **Login index.php:**
* **Verify.php:**
* **Profile.php:**



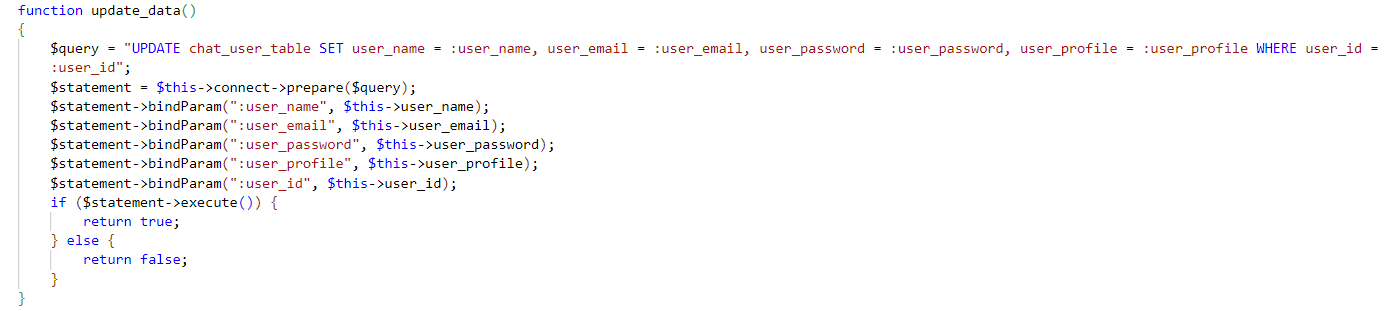
* **Insert User ChatUser.php:**



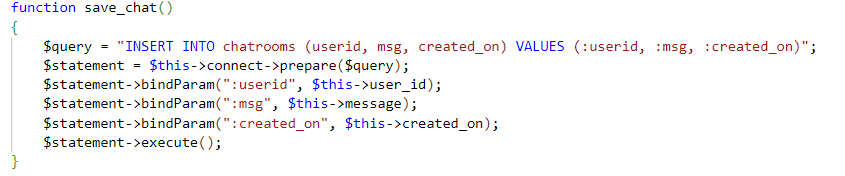
* **Fetch User ChatUser.php:**



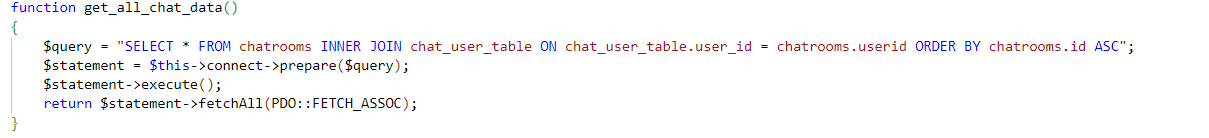
* **Update User ChatUser.php:**



* **Save Chat ChatRooms.php:**

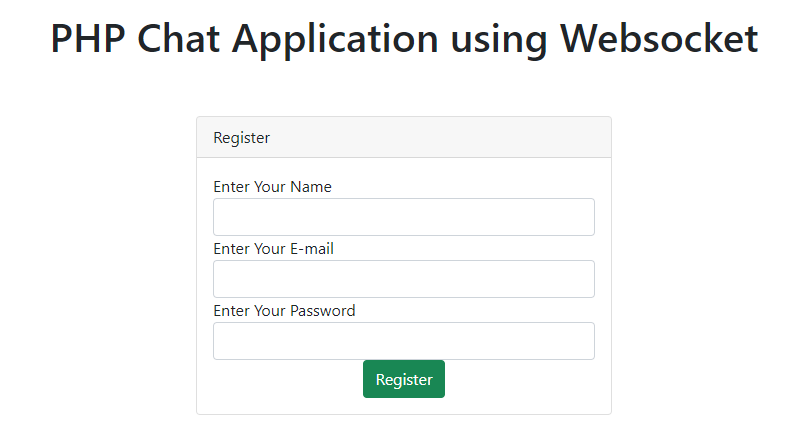


* **Fetch Chat ChatRooms.php:**

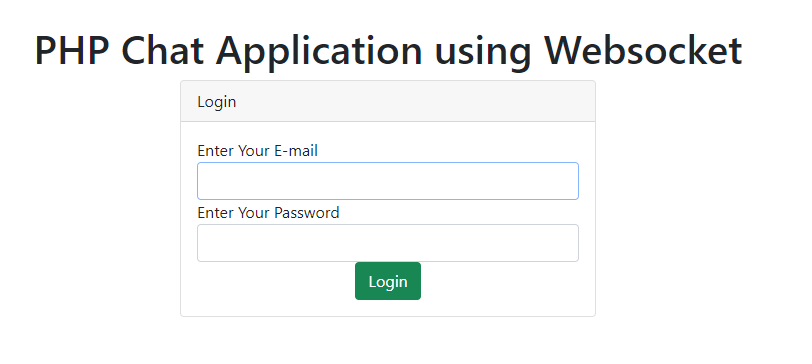


Output

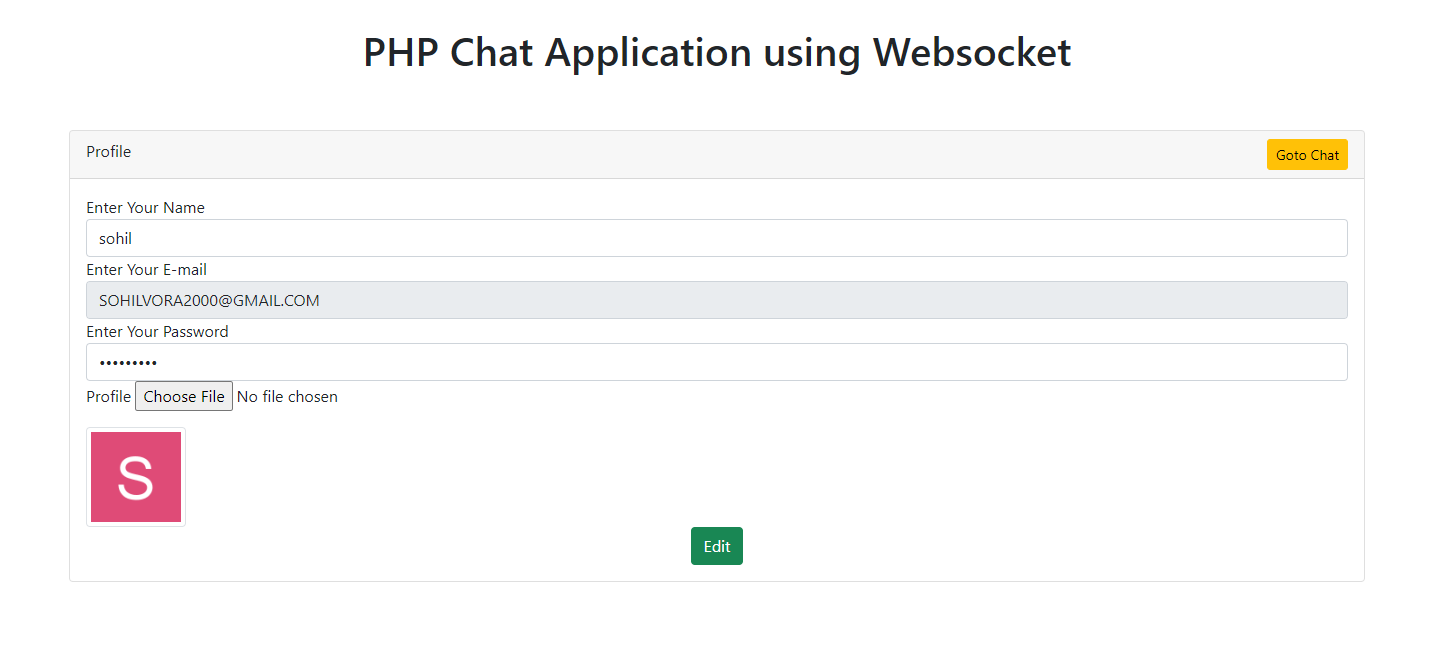
* **Register Page:**



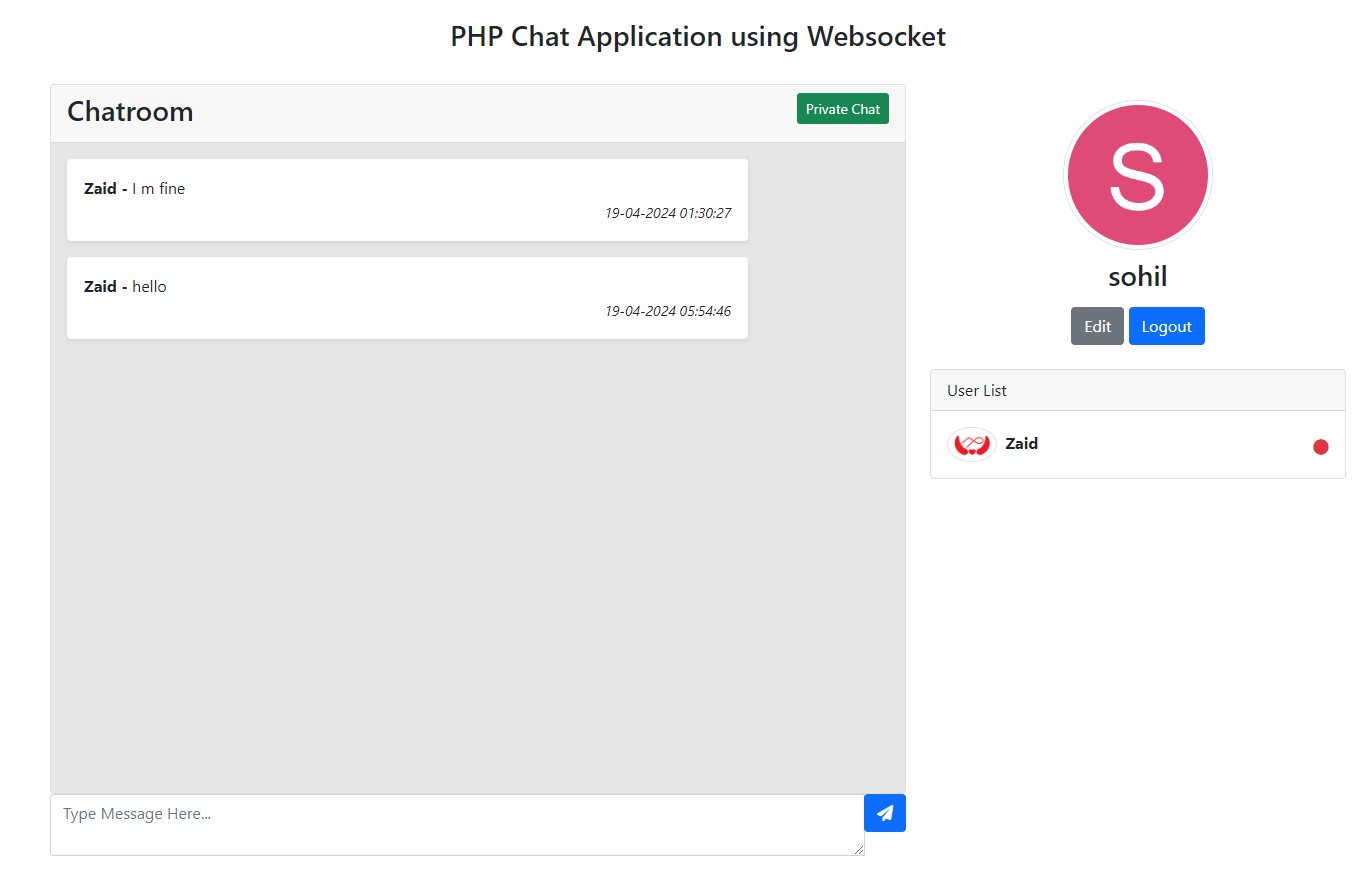
* **Login Page:**



* **Profile Page:**



* **Group Chat Page:**



* **Private Chat Page:**

