

Accommodation Assistance

Crud System using MySQL and PHP



[11/28/23]

[Mariata Homes]

Contents

[Part 1 [Systems Design] [45 Marks] 3](#_Toc152066923)

[Introduction 3](#_Toc152066924)

[Project Overview 3](#_Toc152066925)

[Objectives 3](#_Toc152066926)

[System Design 3](#_Toc152066927)

[Normalization to 3rd Normal Form 3](#_Toc152066928)

[1st Normal Form (1NF): 4](#_Toc152066929)

[2nd Normal Form (2NF): 4](#_Toc152066930)

[3rd Normal Form (3NF): 5](#_Toc152066931)

[Entity Relationship Diagram (ERD) 6](#_Toc152066932)

[Purpose 6](#_Toc152066933)

[Diagram Creation 6](#_Toc152066934)

[UML Class Diagram 6](#_Toc152066935)

[Purpose 6](#_Toc152066936)

[Diagram Creation 7](#_Toc152066937)

[Explanation 8](#_Toc152066938)

[Use Case Diagrams 8](#_Toc152066939)

[Purpose 8](#_Toc152066940)

[Diagram Creation 8](#_Toc152066941)

[Explanation 9](#_Toc152066942)

[UI Front-end Forms 9](#_Toc152066943)

[Purpose 9](#_Toc152066944)

[Explanation: 9](#_Toc152066945)

[Part 2 [Systems Development] [45 Marks] 13](#_Toc152066946)

[Systems Development 13](#_Toc152066947)

[Registration of User and Admin 13](#_Toc152066948)

[Implementation 13](#_Toc152066949)

[Security Measures: 14](#_Toc152066950)

[Input User Details and Information 14](#_Toc152066951)

[Functionality 14](#_Toc152066952)

[Admin User's Login and Dashboard 15](#_Toc152066953)

[CRUD Operations 17](#_Toc152066954)

[Read (View User): 17](#_Toc152066955)

[Update (Edit User): 18](#_Toc152066956)

[Delete (Remove User): 19](#_Toc152066957)

[Additional Functionalities 20](#_Toc152066958)

[Testing 20](#_Toc152066959)

[Conclusion 21](#_Toc152066960)

[Project Summary 21](#_Toc152066961)

[Outcomes 22](#_Toc152066962)

[Future Work 22](#_Toc152066963)

[References: 22](#_Toc152066964)

# Part 1 [Systems Design] [45 Marks]

# Introduction

## Project Overview

With an emphasis on promoting community support and sustainable living alternatives, Mariata Homes is an organization committed to helping those in need find housing. Mariata Homes understands the need for a streamlined and effective digital solution to manage their processes as the need for housing assistance grows.

As things stand, Mariata Homes is having trouble keeping track of user information, organizing and handling requests for accommodation aid, and making sure that both administrators and applicants have an easy-to-use and transparent experience. Mariata Homes is starting a digital transformation journey to improve the overall efficacy, accessibility, and efficiency of their housing aid program in order to address these issues.

## Objectives

The key objectives of the project are:

1. **Digitize the Accommodation Assistance Process:** Put in place a strong digital platform that makes the process of providing accommodations more automated and efficient. This include keeping track of application statuses, managing user data, and enabling communication between Mariata Homes, applicants, and other relevant parties.
2. **Enhance User Experience:** By offering an intuitive and user-friendly interface, administrators and applicants can benefit from an improved overall user experience. This attempts to streamline application procedures, cut down on manual documentation, and enhance lines of communication.
3. **Effective Data Management:** To effectively manage user data, lodging information, and other pertinent data, set up a safe, centralized database system. This guarantees data accessibility and integrity and makes reporting easier for admins.
4. **Transparency and Accountability:** Promote transparency in the process of allocating accommodations by giving applicants up-to-date information on the status of their applications in real time. Provide administrators with tracking tools so they can keep an eye on application status and guarantee decision-making accountability.
5. **Community Engagement:** Make it easier for Mariata Homes to work with local communities, volunteers, and funders by incorporating features that make this possible. This promotes sustainable living practices and builds a supportive ecosystem.
6. **Adaptability and Scalability:** Provide a scalable system that can change to meet Mariata Homes' changing requirements. This include adding new features, supporting a growing number of apps, and making sure it is compatible with developing technology.

# System Design

## Normalization to 3rd Normal Form

In relational databases, normalization is an essential component of architecture that tries to reduce dependencies and redundancies. Improving data organization, lowering redundancy, and enhancing integrity are the main goals of normalization. Normalization guarantees accurate and efficient data storage, which promotes optimal database performance and maintainability.

**Process Description:**

Organizing data in a database to eliminate duplicate entries and establish connections across tables is known as normalization. The goals are to reduce data duplication and enhance data integrity. In the scenario given, the following entities can be identified:

|  |  |
| --- | --- |
| **#** | **Entities** |
| 1 | User |
| 2 | UserInformation |
| 3 | PassportPhotograph |
|  |  |

### **1st Normal Form (1NF):**

1st Normal Form involves eliminating duplicate data and ensuring that each column contains atomic (indivisible) values. Key steps for achieving 1NF include:

* **Atomic Values:** Ensure that each cell in a table contains a single, indivisible value. For example, if a cell has multiple values, break them into separate rows.
* **No Duplicate Rows:** Remove duplicate rows to ensure unique records within the table.

Now, let's go through the normalization process.

|  |  |  |
| --- | --- | --- |
| **User** | **UserInformation** | **Customers** |
| ID(PK)  Username  Password  Role | ID(PK)  UserID(FK)  Name  Dob  Email  Telephone  NextOfKin  Age  Illness  LastResidenceAddress  RecommendedSource  RecommendedSourceAddress | PhotoID(PK)  UserID  Image |

### 2nd Normal Form (2NF):

2nd Normal Form builds upon 1NF by addressing partial dependencies. Key steps include:

* **Functional Dependencies:** Identify and remove partial dependencies, ensuring that non-key attributes depend on the entire primary key.

Now, let's go through the normalization process.

|  |  |  |
| --- | --- | --- |
| **User** | **UserInformation** | **Customers** |
| ID(PK)  Username  Password  Role | ID(PK)  UserID(FK)  Name  Dob  Email  Telephone  NextOfKin  Age  Illness  LastResidenceAddress  RecommendedSource  RecommendedSourceAddress | PhotoID(PK)  UserID  Image |

### 3rd Normal Form (3NF):

3rd Normal Form further eliminates transitive dependencies. Key steps include:

* **Transitive Dependencies:** Identify and remove dependencies where non-key attributes depend on other non-key attributes.

Now, let's go through the normalization process.

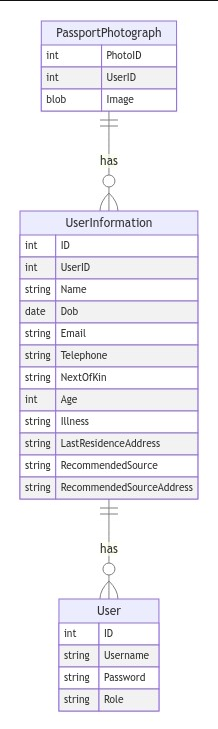
|  |  |  |
| --- | --- | --- |
| **User** | **UserInformation** | **Customers** |
| ID(PK)  Username  Password  Role | ID(PK)  UserID(FK)  Name  Dob  Email  Telephone  NextOfKin  Age  Illness  LastResidenceAddress  RecommendedSource  RecommendedSourceAddress | PhotoID(PK)  UserID  Image |

Custom Data to be stored in our Database:

|  |  |  |
| --- | --- | --- |
| CustomerID | Name | Address |
| 1 | Ali | 88 west end Avenue,E10 6DZ  Tel: +447909942001 |
| 2 | Raza | 88 west end Avenue,E10 6DZ  Tel: +447909942001 |
| 3 | Admin | Admin |

## Entity Relationship Diagram (ERD)

### Purpose

The ERD illustrates the structure of the Mariata Homes database and provides a visual representation of the entities, their attributes and relationships. This ERD serves as a blueprint for organizing and understanding the data model of a digital solution.

### Diagram Creation

* **UserInformation:** Represents detailed information about users of Mariata Homes, including personal details, health information, and contact information.
* **User:** Represents user accounts with login credentials and roles, forming a relationship with UserInformation to establish a comprehensive user profile.
* **PassportPhotograph:** Stores user passport photographs, linked to UserInformation for a one-to-one relationship, ensuring each user has one photograph.

**Explanation:**

**Entities**

1. **User:** Represents users of the system with properties like ID, Username, Password, and Role.
2. **UserInformation:** Contains additional information about users such as Name, Date of Birth, Email, etc.
3. **PassportPhotograph:** Stores user photos with properties PhotoID, UserID, and Image.

**Relationships**

* **UserInformation has User:** Indicates a relationship where each UserInformation entity is associated with one User.
* **PassportPhotograph has UserInformation:** Shows a relationship where each PassportPhotograph is associated with one UserInformation.

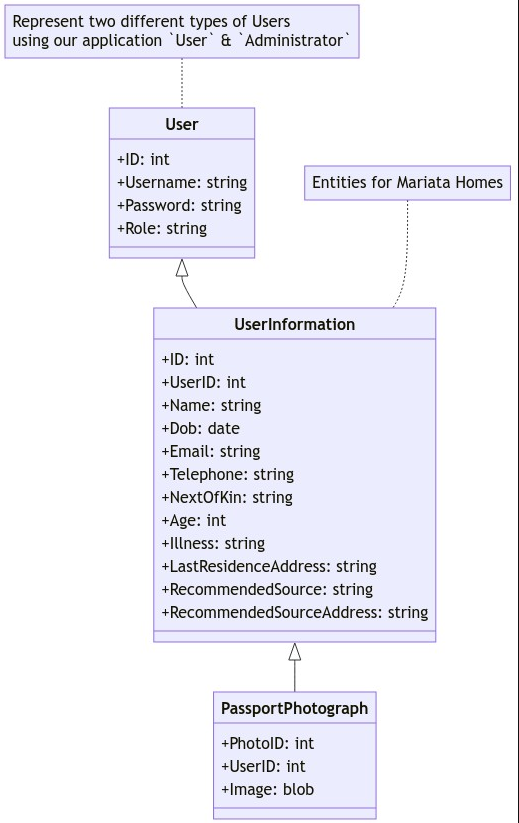
## UML Class Diagram

### ****Purpose****

The UML Class Diagram visually represents the structure of Mariata Homes' system, capturing the relationships between key classes: User, UserInformation, and PassportPhotograph.

### ****Diagram Creation****

* **UserInformation Class:** Represents detailed information about Mariata Homes users, including personal details, health information, and contact information.
* **PassportPhotograph Class:** Stores user passport photographs, linked to UserInformation, forming a composition relationship.
* **User Class:** Represents user accounts with login credentials and roles, associated with UserInformation, forming a composition relationship.



### Explanation

**Classes:**

1. **User:** Represents basic user information with properties ID, Username, Password, and Role.
2. **UserInformation:** Contains detailed user information with properties like Name, Date of Birth, Email, etc.
3. **PassportPhotograph:** Represents user photos with properties PhotoID, UserID, and Image.

**Relationships:**

* **User inherits UserInformation:** Indicates that User is a specialized form of UserInformation.
* **UserInformation inherits PassportPhotograph:** Shows that UserInformation is a specialized form of PassportPhotograph.

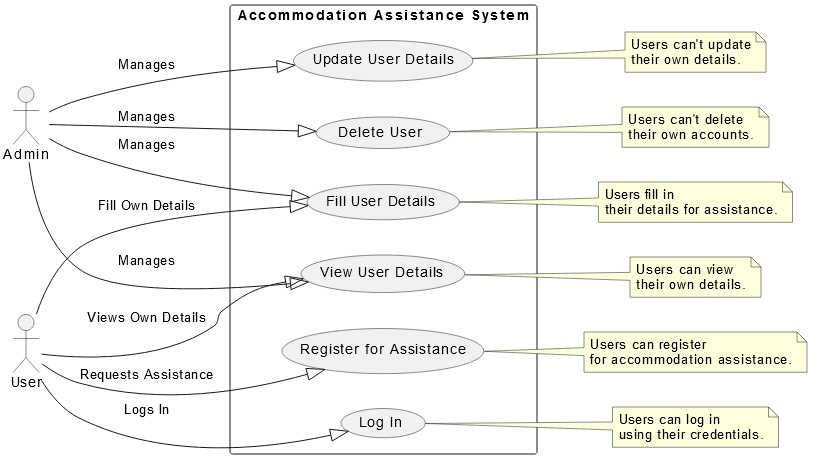
## Use Case Diagrams

### Purpose

The Mariata Homes Accommodation Assistance System's functioning is depicted in the use case diagram from the viewpoints of the admin and user. It describes important tasks including signing up, logging in, managing user information, and administrative duties.

### Diagram Creation

* **User Actions:**
  + Users can register, log in, view their details, and fill in their own details.
  + Users can't update or delete their accounts.
* **Admin Actions:**
  + Admin manages viewing, updating, and deleting user accounts.
  + Admin can also fill in details on behalf of users.



### Explanation

**Actors**

* **Admin:** Manages user details, including viewing, updating, and deleting user information.
* **User:** Interacts with the system by registering, logging in, filling details, viewing own details, and logging out.

**Use Cases**

1. **Register for Assistance:** Users can register for accommodation assistance.
2. **Log In:** Users can log into the system using their credentials.
3. **Fill User Details:** Users can provide their details for assistance.
4. **View User Details:** Users can view their own details.
5. **Update User Details:** Admin manages updating user details.
6. **Delete User:** Admin manages deleting user accounts.

**Relationships**

* **Admin Manages Use Cases:** Admin can manage (perform) view, update, and delete operations.
* **User Requests Assistance:** Users can request accommodation assistance by registering.
* **User Logs In:** Users can log into the system.
* **User Fills Details:** Users can provide their information.
* **User Views Own Details:** Users can view their own details.
* **Admin Manages Update/Delete:** Admin manages updating and deleting user accounts.

## UI Front-end Forms

### Purpose

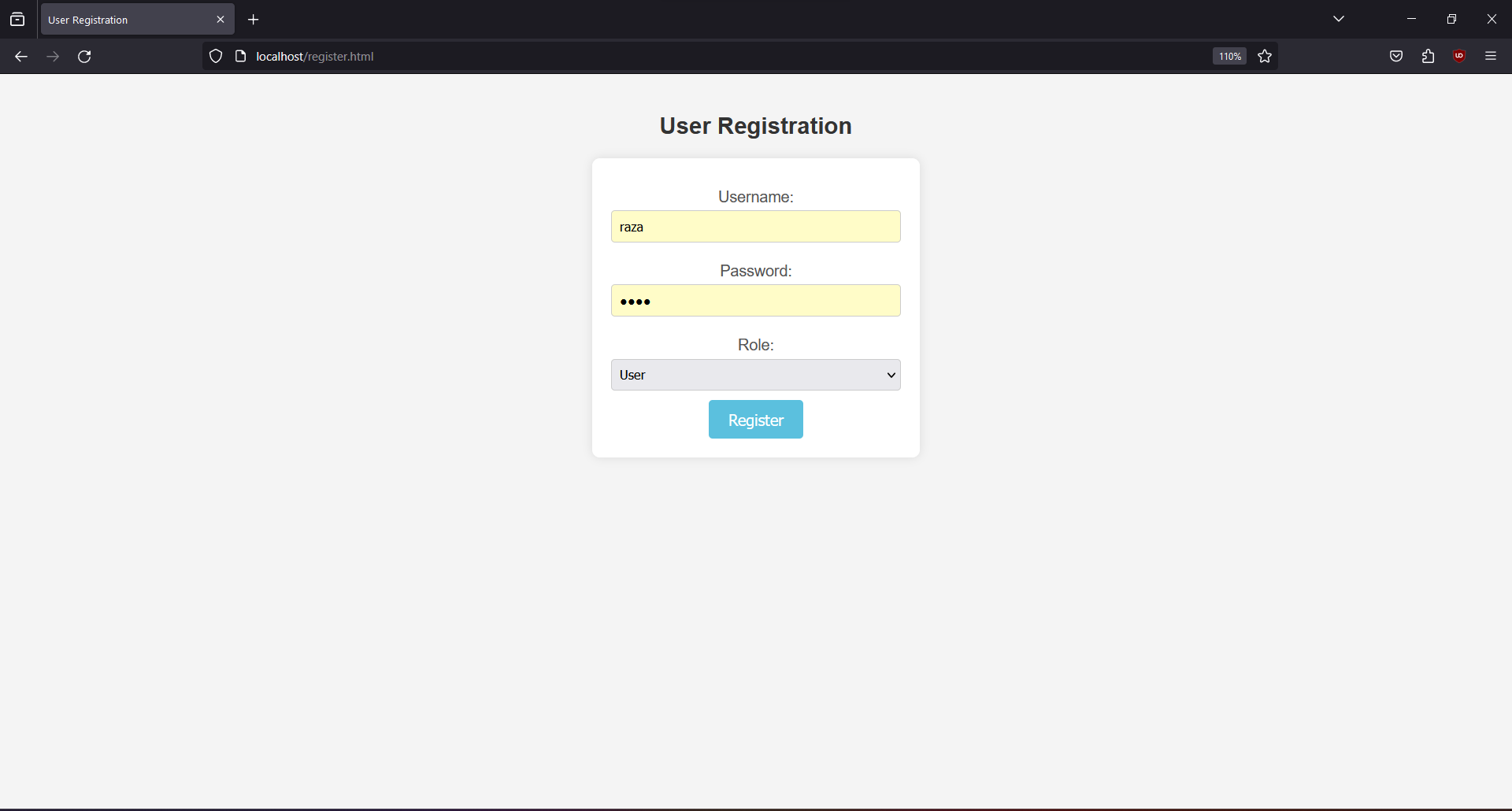
### The UI Front-end Structures act as the visual door for clients to collaborate with the Mariata Homes Convenience Help Framework. The design is to give a natural and easy to understand interface that works with consistent correspondence among clients and the framework.

### Explanation:

1. **User-Centric Design:**
   * The design places a high priority on user-centric principles, making the forms simple to use and understand for users with different degrees of technical proficiency.
2. **Intuitive Layout:**
   * Users are guided through the necessary procedures for operations like registration, login, and amending user details by the forms' logical and straightforward layout. This encourages an easy-to-use interface.

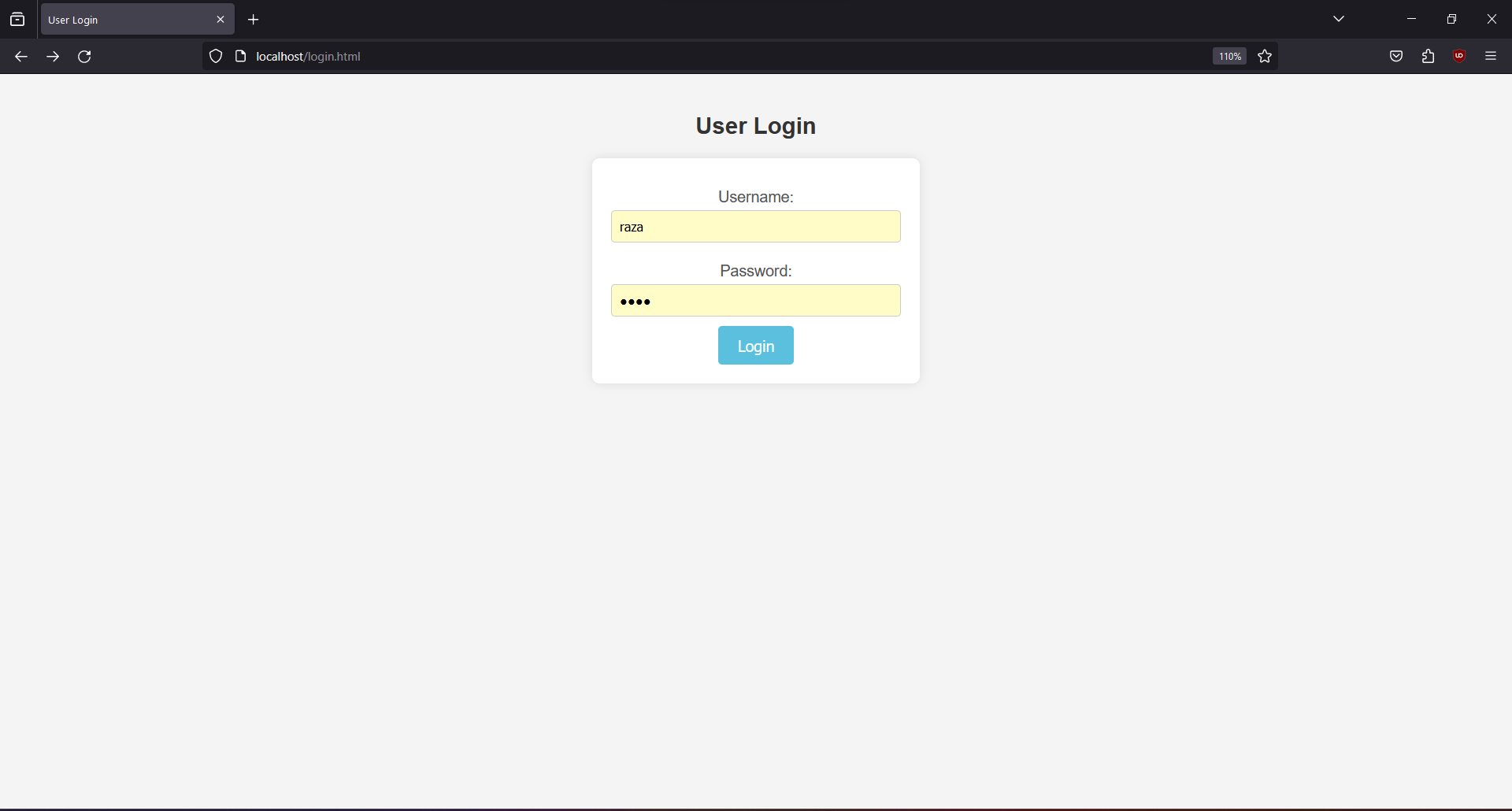
**Registration Form:**

* **Purpose:** The Mariata Homes Accommodation Assistance System's Registration Form is the first stage in the user onboarding process. During the registration procedure, users must choose a secure password, a unique username, and a role inside the system.
* **Process:**
  + **Input Fields:** The form includes fields for Username, Password, and Role selection.
  + **Role Assignment:** Users define their Role, determining the level of access and functionalities available to them.
  + **Database Validation:** Entered details undergo validation against the database to ensure uniqueness.
* **User Feedback:**
  + Users receive feedback messages for successful registration or are prompted to revise their input in case of errors.
  + A successful registration initiates the creation of a user account in the system.



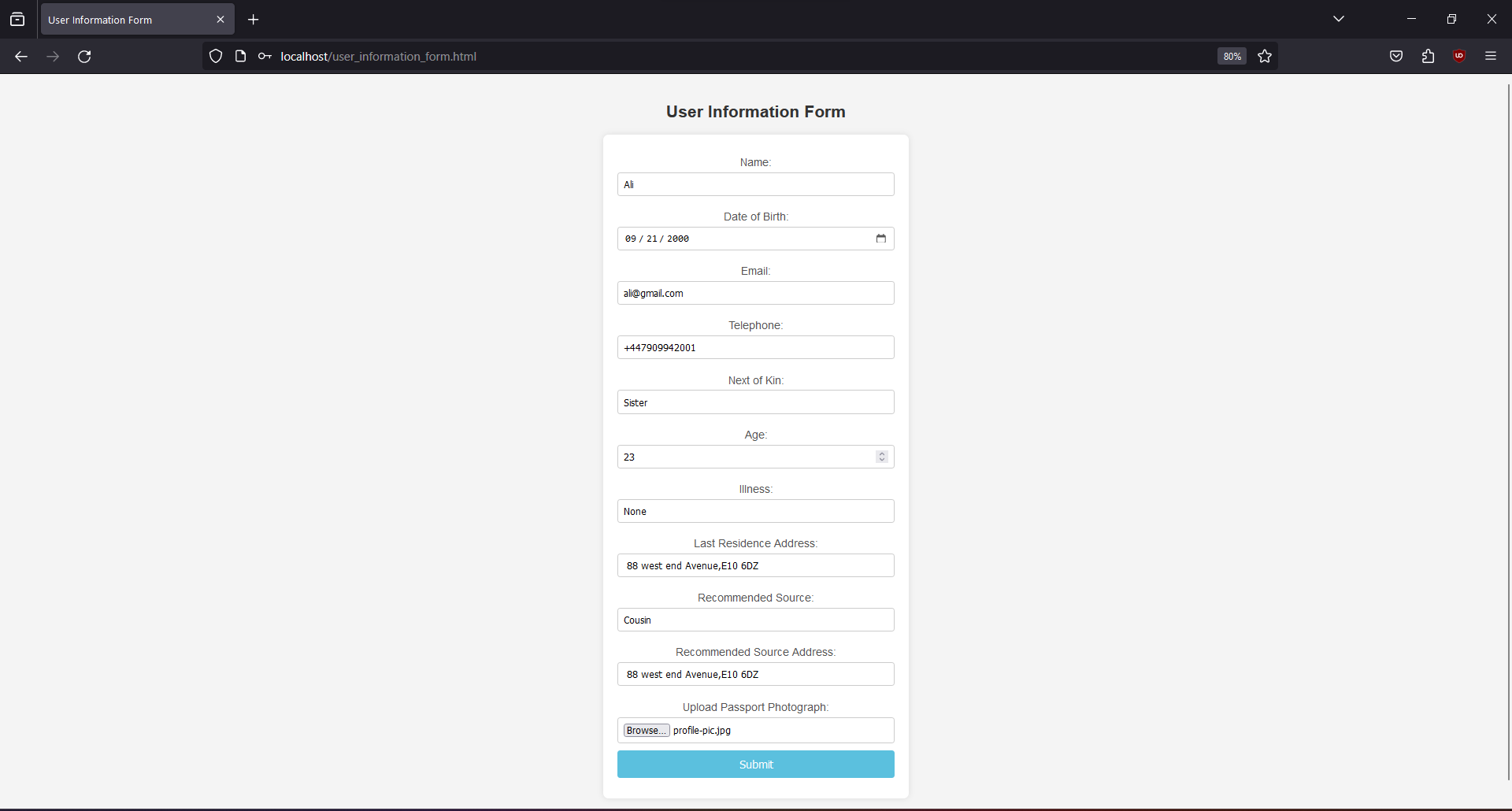
**Login Form:**

* **Purpose:** The Login Form provides a secure gateway for users to access their accounts. It requires the input of the registered email and password, enabling verification against the database.
* **Process:**
  + **Input Fields:** Users input their registered email and password during login.
  + **Database Verification:** Entered credentials are validated against stored records in the database.
  + **Session Creation:** A successful login results in the creation of a session, facilitating user tracking and seamless application operation.
* **User Feedback:**
  + Users receive login success feedback, granting access to the system.
  + In case of login failures, users are informed to reattempt or recover their credentials.



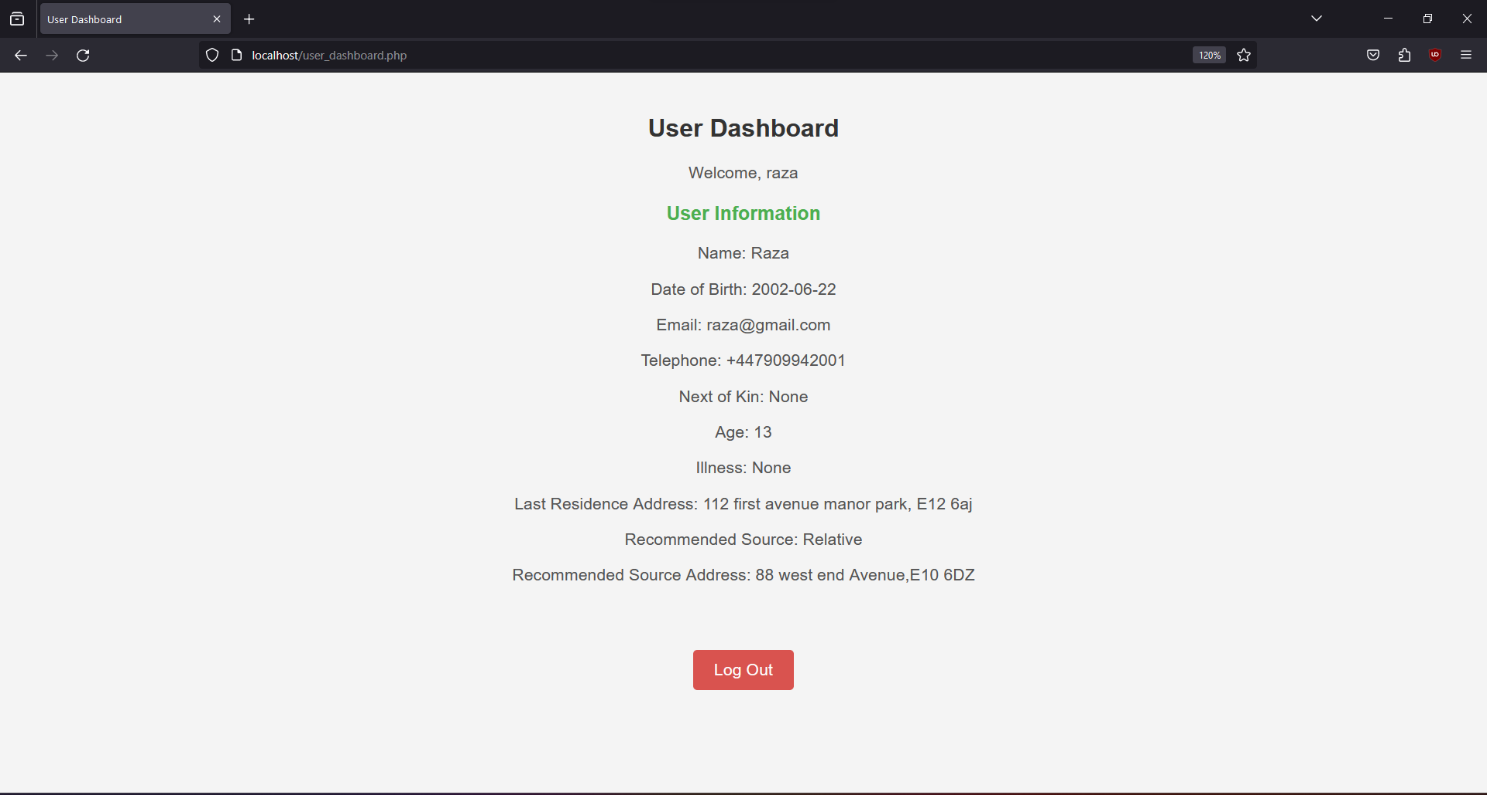
**User Information Form:**

* **Purpose:** Upon successful login, users are directed to the User Information Form. Here, users are prompted to input personal details, contributing to a comprehensive user profile.
* **Process:**
  + **Input Fields:** The form includes fields for various user details, such as name, date of birth, contact information, and more.
  + **Submission:** Users submit their information, triggering data storage in the system's User Information table.
* **User Feedback:**
  + Users receive confirmation of successful information submission.
  + The entered details contribute to the establishment of a user profile for personalized assistance.



**Second Scenario:**

* **Context:** If a user has previously completed the User Information Form, subsequent logins redirect them to their personalized homepage.
* **Process:**
  + **Homepage Display:** The system detects previous form submissions, displaying the user's information on their homepage.
  + **Navigation:** Users can navigate to other functionalities from their homepage, streamlining their interaction with the system.
* **User Experience:**
  + Returning users experience a personalized and efficient interaction with the system.
  + Previously submitted information is readily available for review or update as needed.



# Part 2 [Systems Development] [45 Marks]

# Systems Development

## Registration of User and Admin

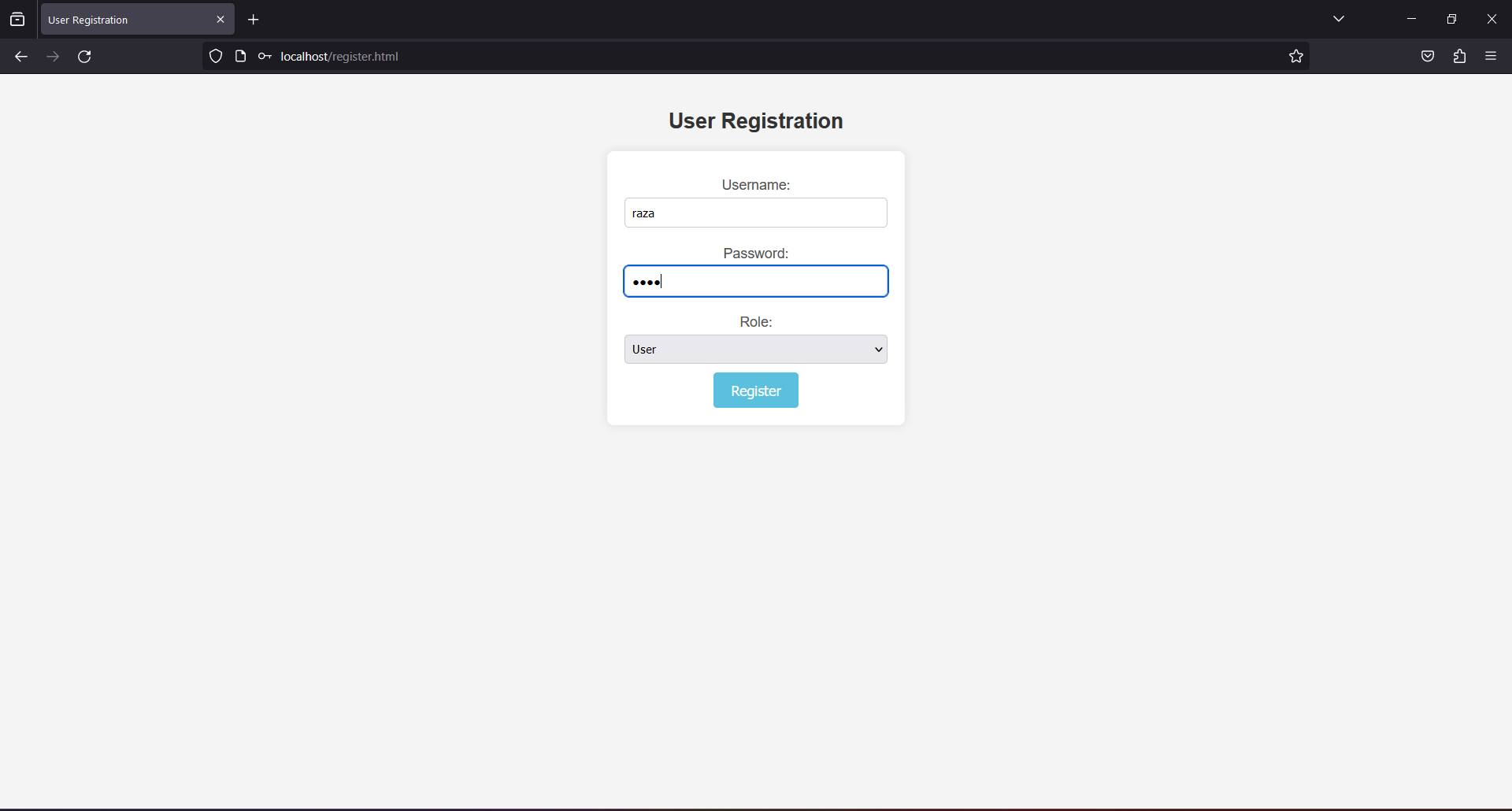
### Implementation

Specialized registration forms are used to carry out the registration procedure for users and administrators of the Mariata Homes Accommodation Assistance System. Below is a summary of how it was done:

**User Registration:**

* **Form Design:** An intuitive registration form with areas for entering a secure password, a unique username, and the user's role (such as "user") is created.
* **Backend Validation:** To guarantee the uniqueness of the selected username and compliance with password security guidelines, input data is validated on the backend.
* **Database Interaction:** database interaction involves storing validated data in the database's 'User' table, which creates a new user account.

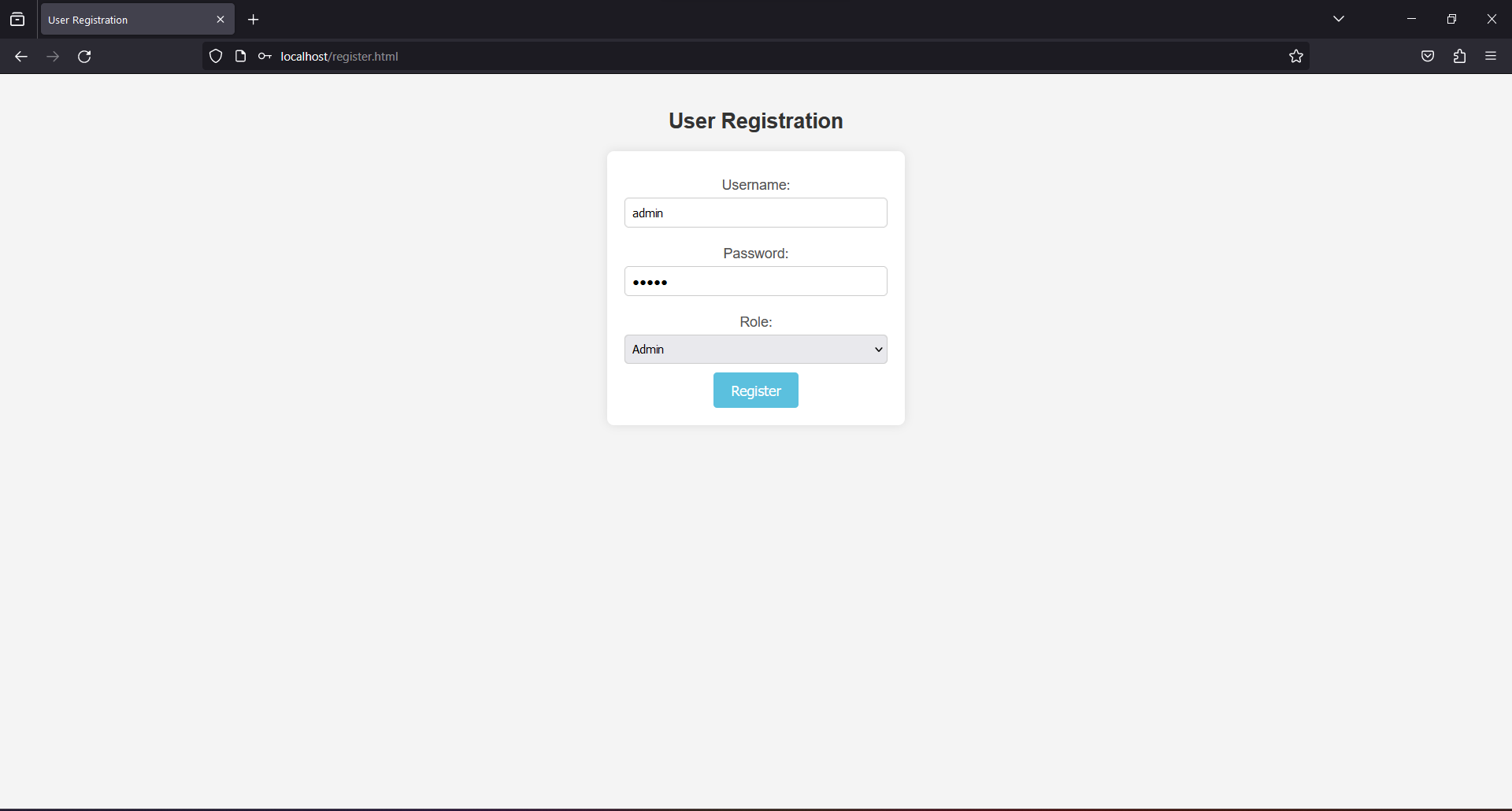




**Admin Registration:**

* **Admin-Specific Form:** Admins register through a similar form, but with additional fields specific to admin roles, such as extended permissions or responsibilities.
* **Validation and Storage:** Admin registration follows a similar process to user registration, with validation and storage in the 'User' table.





### ****Security Measures:****

To ensure the security of user data during the registration process, the system incorporates the following measures:

* **Password Hashing:** bcrypt and other robust one-way encryption algorithms are used to safely hash passwords. This guarantees that passwords are secure even in the event that the database is compromised.

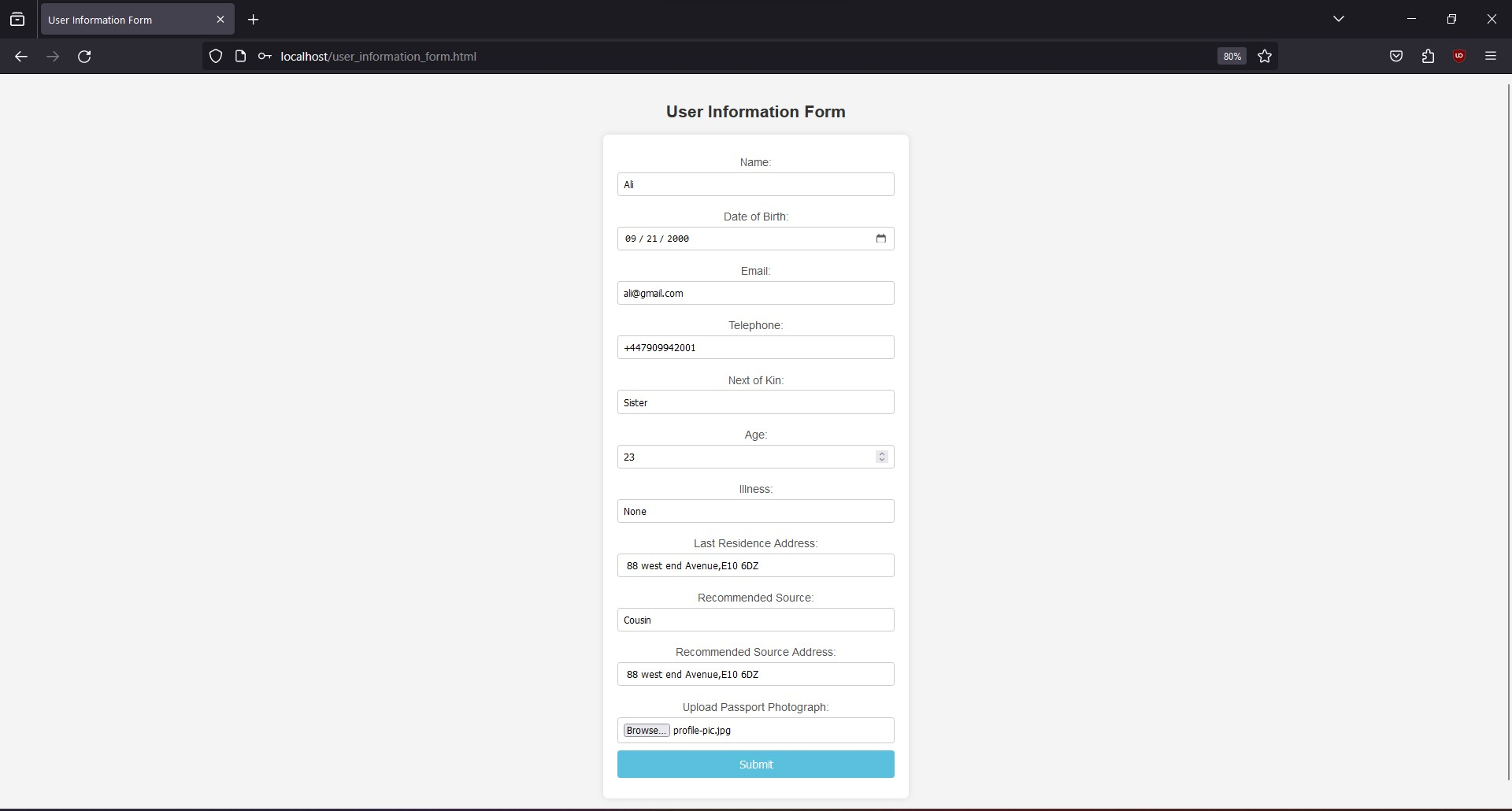
****

## Input User Details and Information

### Functionality

**User Information Form:**

* **Purpose:** After successful login, users are directed to the User Information Form to input and manage their details.
* **Input Fields:**
  + *Name, Date of Birth, Email, Telephone:* Personal details.
  + *Next of Kin:* Emergency contact information.
  + *Age, Illness, Last Residence, Recommended Source:* Additional details.



**Data Validation:**

* To guarantee the correctness and completeness of user inputs, validation procedures are in place both client-side (using PHP) and server-side (using JavaScript).

**Submission:**

* When users submit their completed forms, the relevant server-side scripts are activated for data processing and database storage.

**Feedback:**

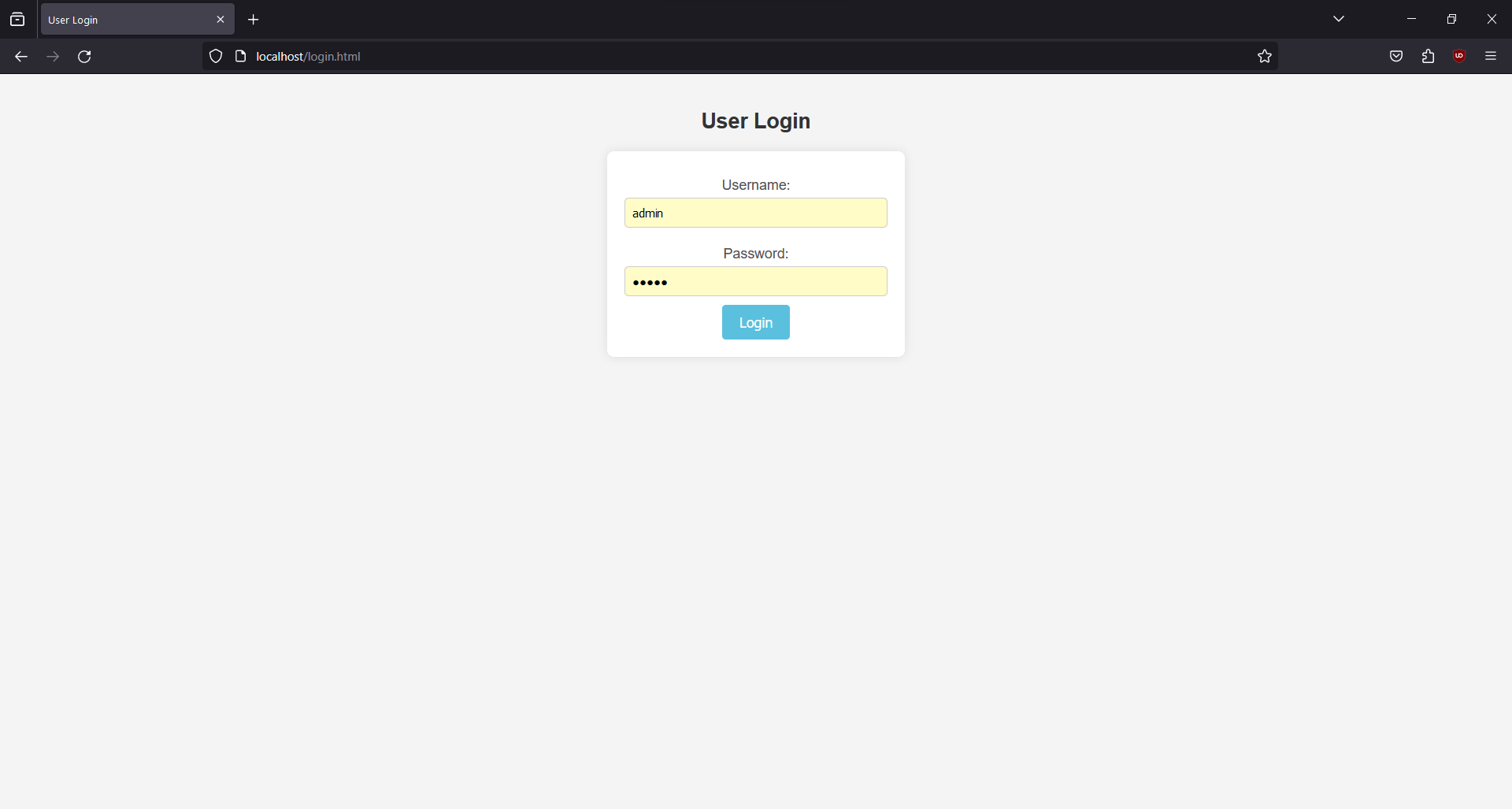
* Feedback messages that clearly explain the system's reaction are sent to users regarding successful registrations, logins, or validation failures.

## Admin User's Login and Dashboard

**Login Process:**

1. **Purpose:** The admin login process is designed to authenticate administrators and grant access to the administrative dashboard.
2. **Mechanism:**
   1. Admins use a same login form where they provide their username and password.
   2. Upon submission, the system verifies the credentials against the stored data in the database.
   3. If the authentication is successful, a session is initiated, allowing the admin access to the administrative features.



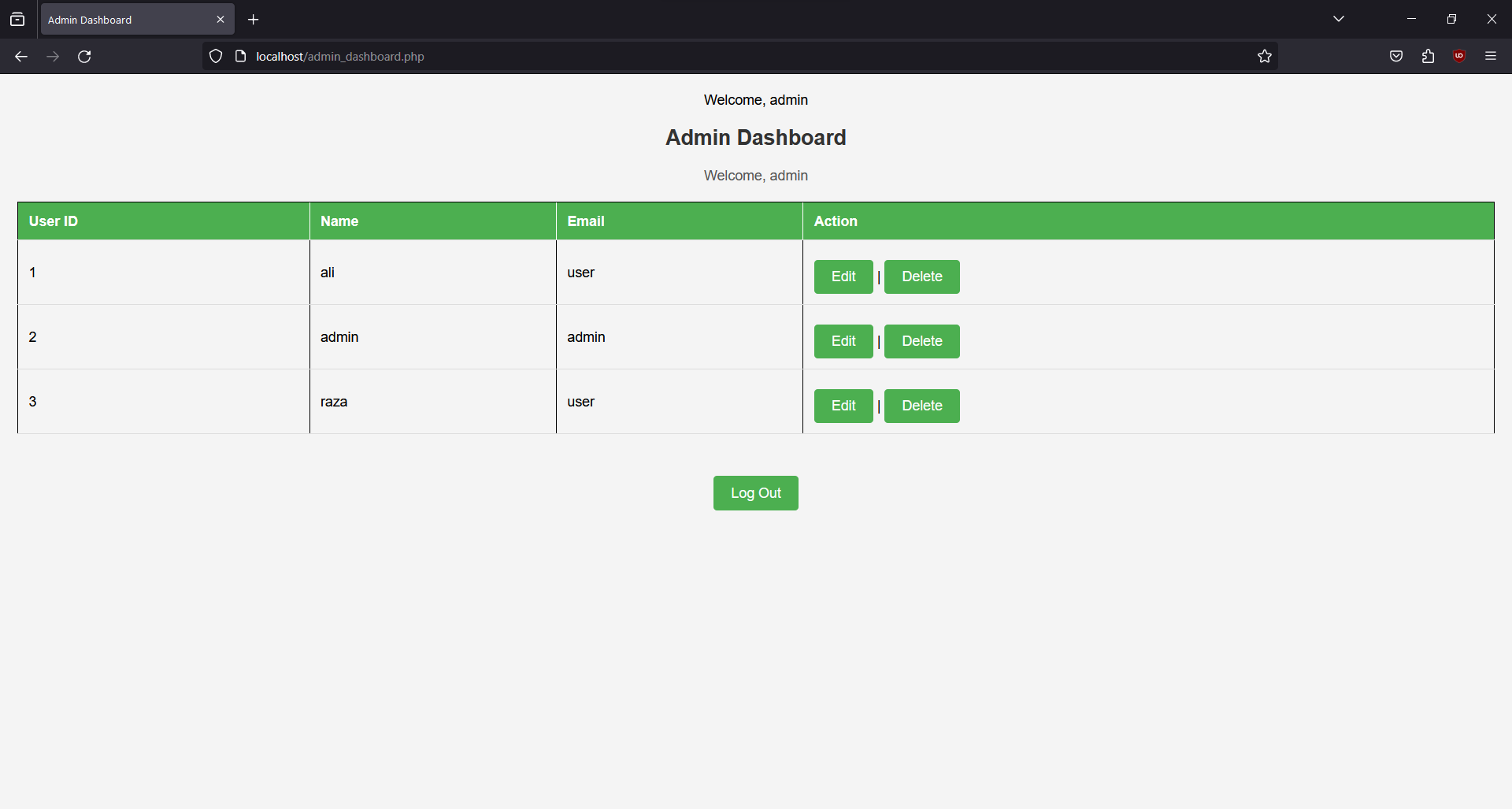


**Dashboard Functionality:**

**User Management:**

* Admins can view a list of users, including their details.
* They can edit user information and update records.
* Deletion of user accounts is also within the admin's purview.





**Security Measures:**

* Admin login credentials are securely stored using hashing algorithms.
* Access to admin functionalities is restricted to authenticated sessions, ensuring data integrity and confidentiality.

**Feedback:**

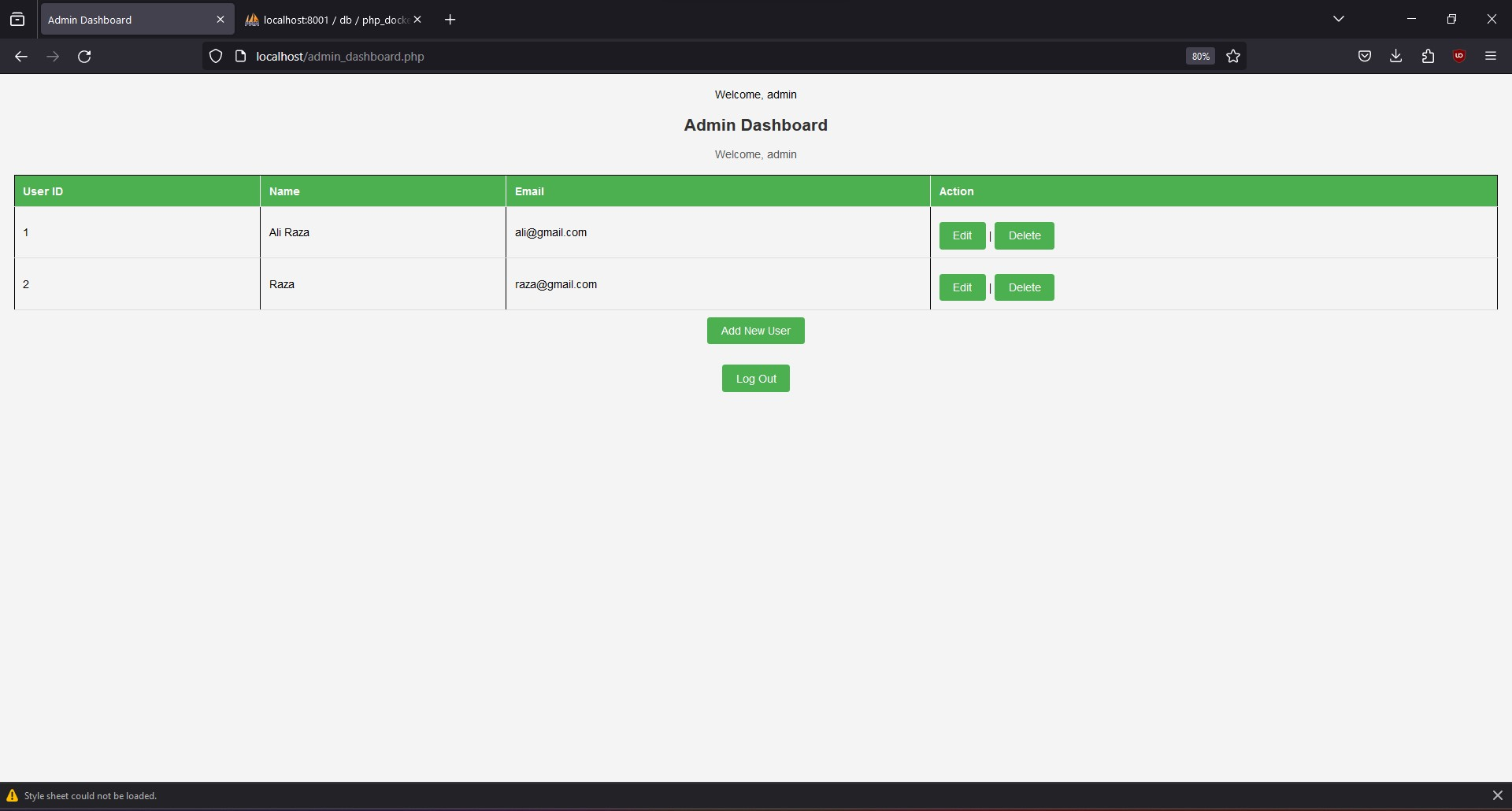
* Admins receive clear feedback messages upon login, indicating success or failure.
* Error messages are informative and help troubleshoot any login issues.

## ****CRUD Operations****

### Read (View User):

* Admins have access to a user list, displaying essential user details.
* Clicking on a user ID or username allows admins to view detailed information.

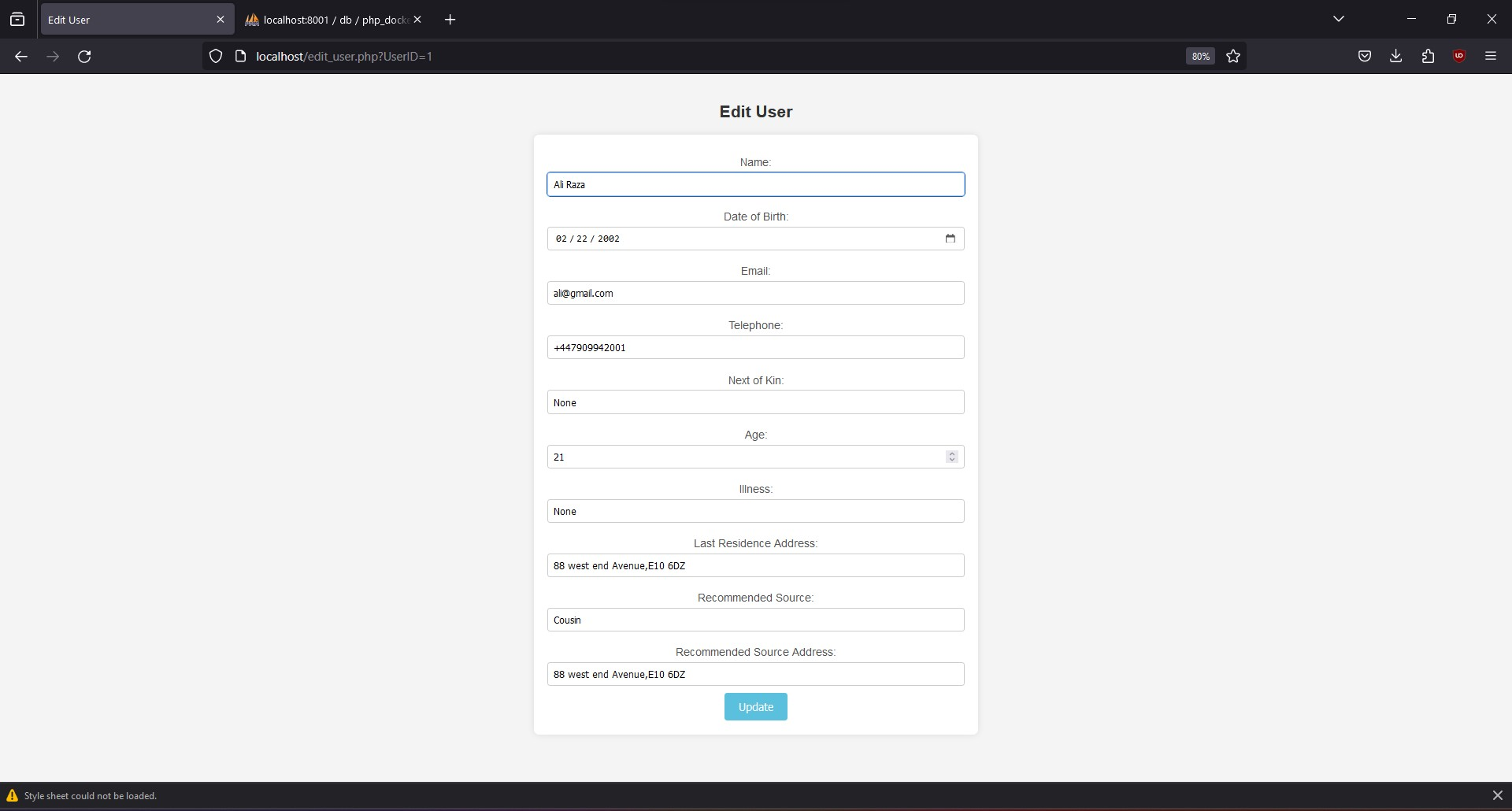
****



### Update (Edit User):

* Admins can initiate updates by clicking the "Edit" button next to a user.
* This redirects admins to a form pre-filled with the selected user's information.
* Upon submission, the updated data is processed and stored in the database.

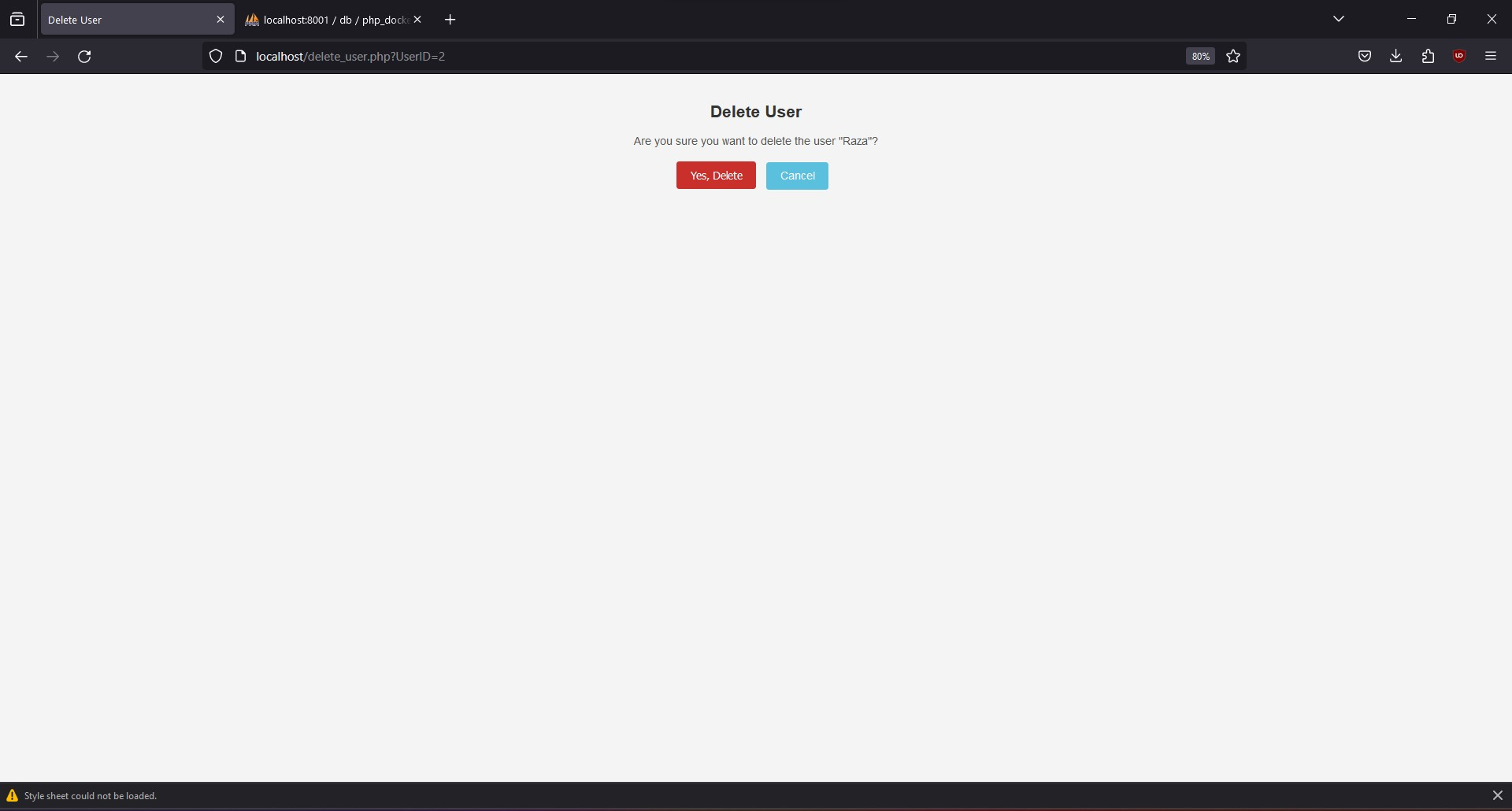
****



### Delete (Remove User):

* Admins can delete users by clicking the "Delete" button next to a user.
* Confirmation is typically required to prevent accidental deletions.

****

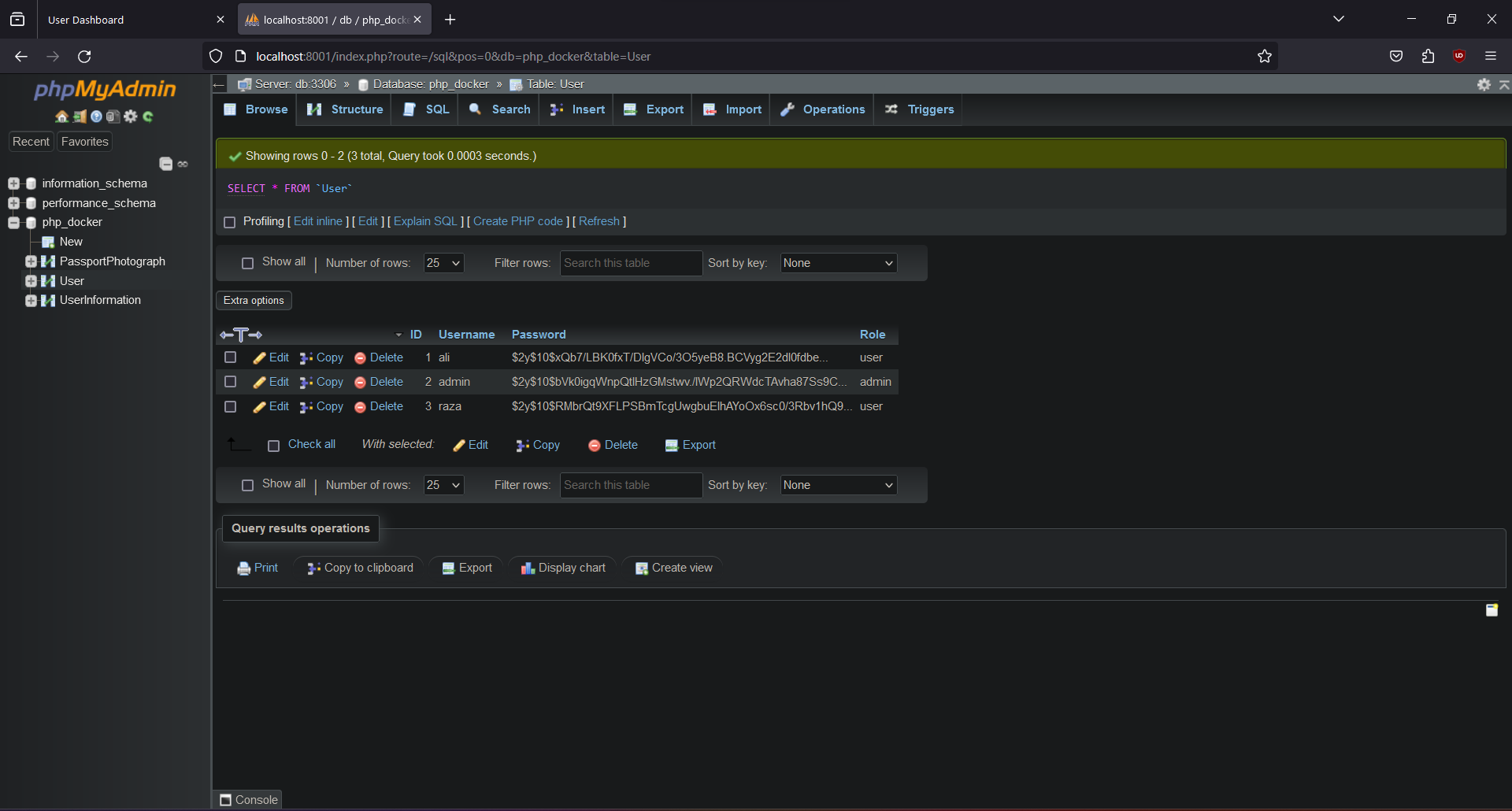


## Additional Functionalities

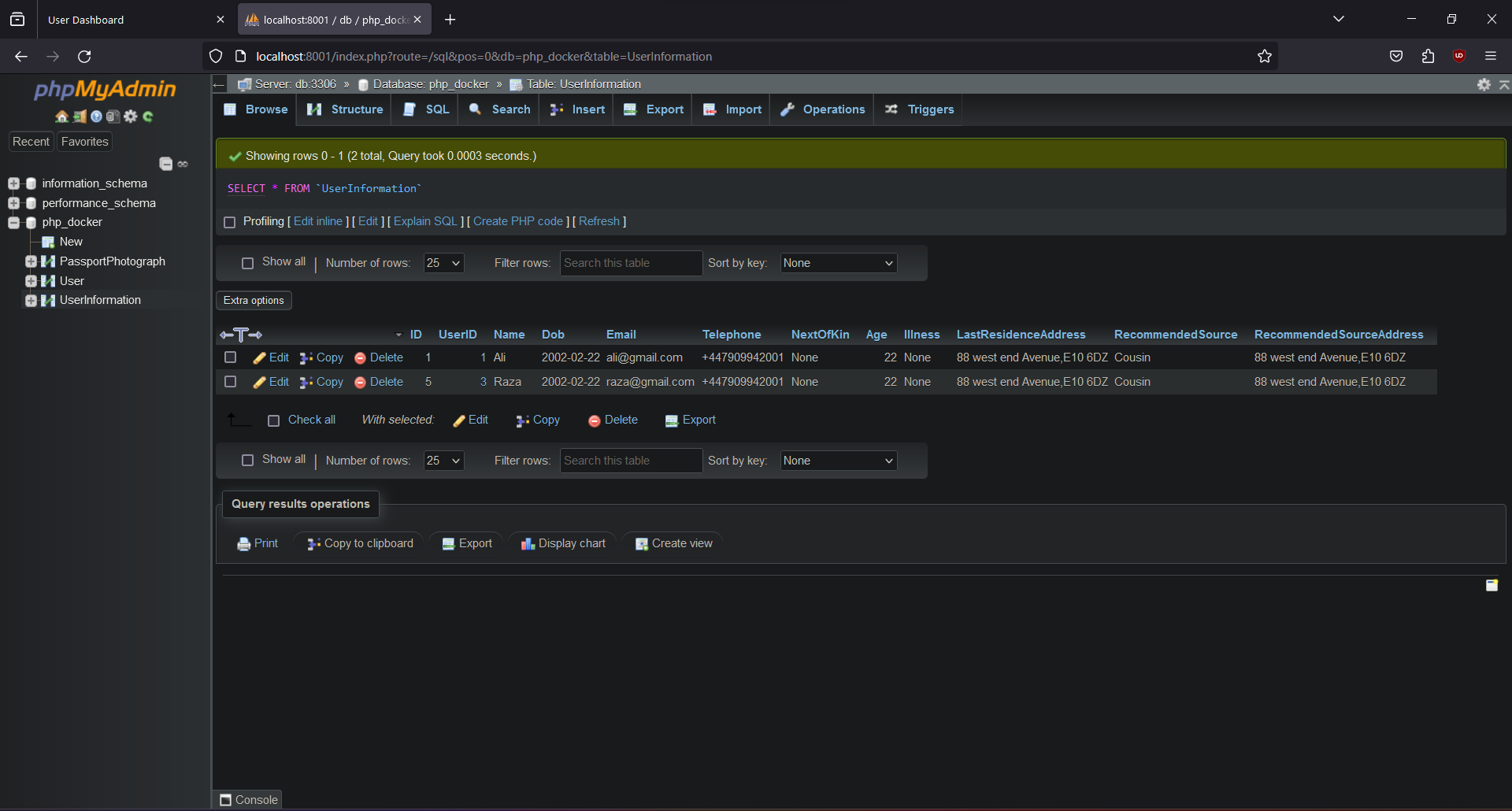
* **User-Friendly Dashboard:**
  + The admin dashboard has a clear menu for CRUD operations and is made for easy navigation.
  + Each operation has a clear label to make sure administrators can find and complete jobs with ease.
* **Responsive User Forms:**
  + CRUD forms have input fields and labeling that are easy to understand.
  + Pre-filled forms during editing enhance efficiency.
* **Visual Feedback:**
  + Administrators receive visual feedback, such as success messages or updated data displays, once CRUD operations are executed successfully.
* **Security Measures:**
  + Administrator authentication is required to access CRUD functionalities.
  + SSL encryption ensures secure data transmission during CRUD operations.
* **Error Handling:**
  + To assist administrators in the event of data entry or processing problems, the interface includes instructive error messages.

## Testing

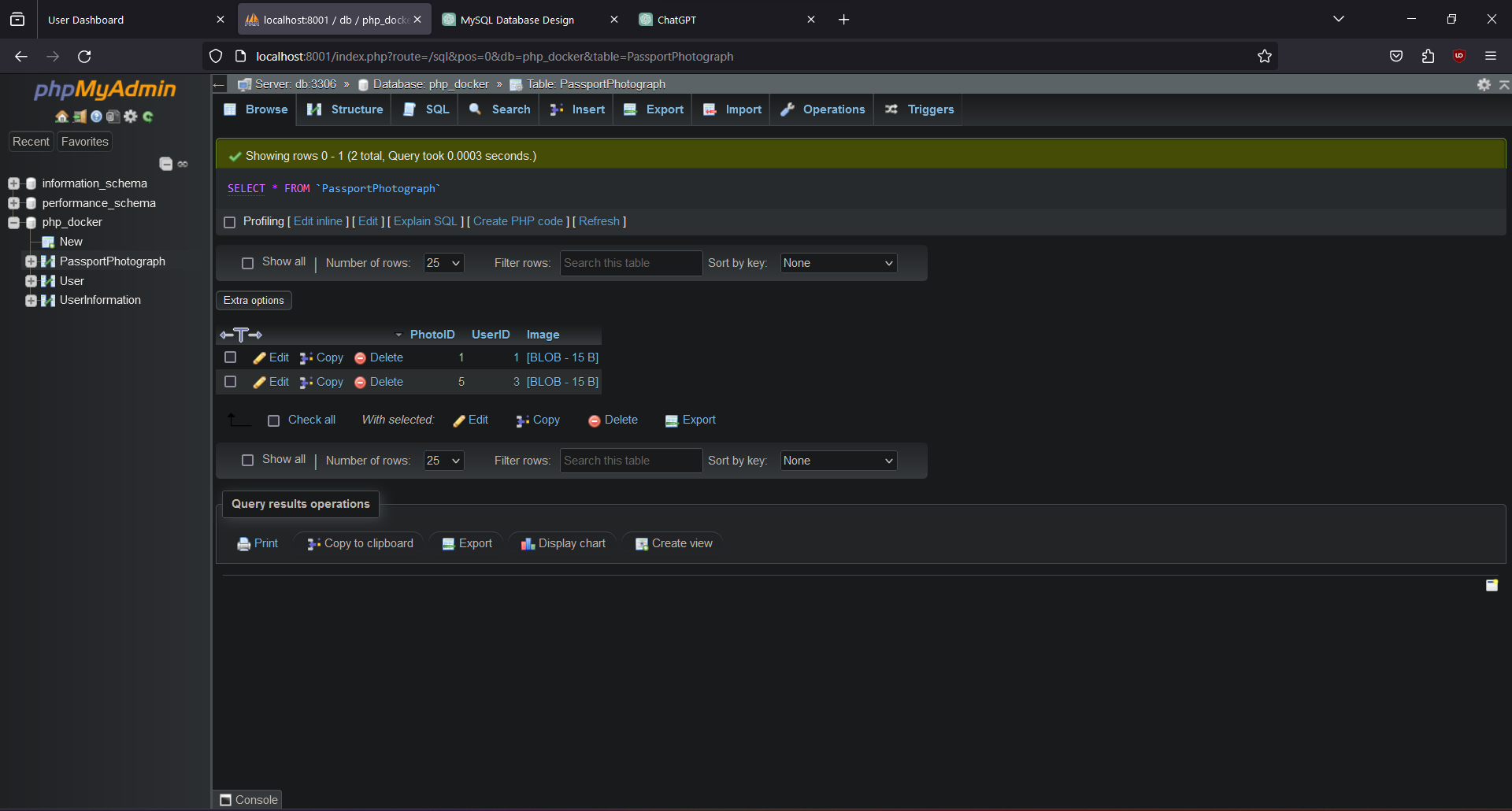
After the registration of two users with the role "User" and one administrator with the role "Admin," the database exhibits a structured organization to store user and administrator information efficiently.



Upon users providing their details through the User Information Form, the database structure accommodates the recorded information for each user. This includes essential personal details such as name, date of birth, email, telephone, next of kin, age, illness, last residence address, recommended source, and recommended source address.



In conjunction with the User Information, users have the option to attach their passport size photographs. The database stores this visual information, associating each image with the respective user. This enhances user identification and provides a comprehensive profile.



## Conclusion

### Project Summary

### 

The digitization of the housing help process was the main goal that the Mariata Homes housing help System effectively accomplished. Through user-friendly interfaces, the initiative sought to expedite user registration, data input, and management. The system's management of user data is made more efficient by the addition of CRUD functions for administrators. Secure interactions inside the system are ensured by the role-based access control and login systems.

### Outcomes

* **User-Focused Design:** Easy registration and data entry are made possible by the UI front-end forms' priority of the user experience.
* **Role-Based Access:** Different features that cater to the responsibilities of administrators and users offer a customized experience.
* **CRUD Operations:** Using Create, Read, Update, and Delete activities, administrators may effectively manage user data.
* **Secure Authentication:** To safeguard user accounts, the login procedure uses secure authentication techniques.

### Future Work

Future enhancements to the Mariata Homes Accommodation Assistance System could include:

* **Enhanced User Dashboard:** Enlarge user dashboards to offer more individualized data and insights pertaining to lodging.
* **Notification System:** Put in place a notification system to inform users of any announcements pertaining to the system or the status of their requests for accommodations.
* **Integration with External Services:** To improve user data, integrate the system with outside services like location-based apps or public databases.
* **Mobile Application:** Create a mobile application to improve user accessibility and convenience.

## ****References:****

[1] W3Schools. "PHP File Handling." <https://www.w3schools.com/php/php_file.asp>  
[2] W3Schools. "PHP MySQLi Introduction." <https://www.w3schools.com/php/php_mysql_intro.asp>  
[3] W3Schools. "PHP Sessions." <https://www.w3schools.com/php/php_sessions.asp>  
[4] W3Schools. "PHP Form Handling." <https://www.w3schools.com/php/php_forms.asp>  
[5] W3Schools. "PHP CRUD Operations." <https://www.w3schools.com/php/php_mysql_crud.asp>

**The End:**