

**EAST WEST UNIVERSITY**

**Project Report**

**Course Code:** CSE411

**Course Title:** Software Engineering and Information System Design

**Section:** 03

**Project Title:** Online Web Chat Application

**Md Mohsin Uddin**

Senior Lecturer

Department of Computer Science and Engineering

Submitted to

Submitted by

1. **Omair Bin Abdur Rahman**

2018-1-60-201

1. **Samaiya Haque**

2017-2-60-008

**TABLE OF CONTENTS**

**Topic Page**

**Introduction**

Description 2

Purpose 2

Motivation and Scope 2

Challenges 2

**Model View Controller (MVC) diagram**

Diagram 2

**Requirements**

Non Functional Requirements 3

Functional Requirements 3

Technologies 4

Operational Environment 4

**Appendix A: Analysis Model**

User Case Diagram 4

Class Diagram 5

Sequence Diagram 5

Activity Diagram 6

ER Diagram 7

**Appendix B: External Interface**

User Interfaces 7-18

Hardware Interfaces 18

Software Interfaces 18

Communications Interfaces 18

**Introduction**

**Description**

In this modern edge technology, the chat system has become one of the most important tools for everyone to share knowledge and materials via network. Chat System is a form of communication that utilizes computer programs that allow conversations between users in real time. Typically the users will connect to a chat server using a chat client in a chat room. Once the users are in the same chat room, they can converse with one another by typing messages into a window where all of the other users in the chat room can see the message.

Users have a default profile. They can edit their own profile and are able create posts. They are able to react to other’s posts. This makes it easy for people to share their thoughts, knowledge and thinking.

**Purpose**

This environment is developed in order to enhance the communication among students and others via a network which will be easy for the discussion related to their studies and other important things. People can easily share their thoughts with each other and also with the whole world.

**Motivation and Scope**

The environment is created to increase communication between students and others through networks so that it becomes easier for them to study and discuss important topics. They can also post important contents and view their own profile.

**Challenges**

* Providing real time conversation.
* Interactive conversation.
* Providing a smart environment for profile and post.
* Sending and displaying the message on both ends at the same time.
* Providing a smart and real chatting environment.
* Managing the security system.

**Model View Controller (MVC) diagram**

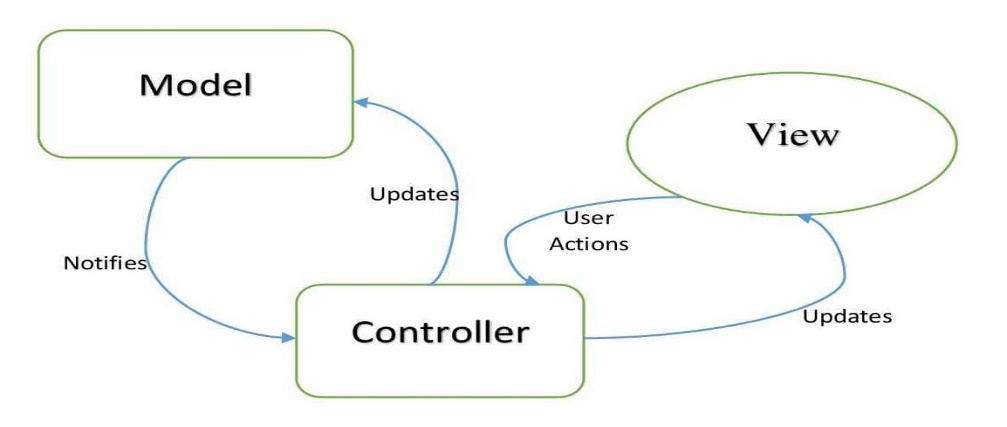
**Diagram**

Figure: Model View Controller (MVC) Diagram.

**Requirements**

**Non-Functional Requirements**

* Ensuring a great user experience is important. Users shall not find any difficulty while using the application.
* The application shall be time efficient so that the user should send and receive messages as soon as possible.
* Ensuring security is must. It shall provide proper security to users.
* It should provide user friendly environments.
* It should ensure login and registration facility.

**Functional Requirements**

* It should have a home page.

**Scenario:** In the home page there will be some contents. Users can choose an option for chat module or post module or profile module.

* Interactive web chat application.

**Scenario:** When a user sends a message then the user’s messages should be on the left side and the receiver’s messages should be on the right and vice versa.

* Interactive profile module.

**Scenario:** Users may have a default profile. They can update their profile any time they need.

* Interactive post module.

**Scenario:** Users can view all the posts which have been posted. On the other hand, users can upload posts. They can also update or delete their own post. Users can react such as ‘like’, ‘love’, ‘haha’ on the post as well.

* Image uploading.

**Scenario:** Users can upload profile pictures.

* Attractive look.

**Scenario:** The application should be attractive so that users can feel pleased while using the application. Like the background, content, text and alignment should have a professional look.

**Technologies**

* Language: Python.
* Framework: Django.
* Template: HTML, CSS and JavaScript.
* Database: MongoDB.

**Operational Environment**

For accessing the system users need a modern browser of a PC, Tab, or Mobile phone with any operating system.

**Appendix A: Analysis Model**

**User Case Diagram**

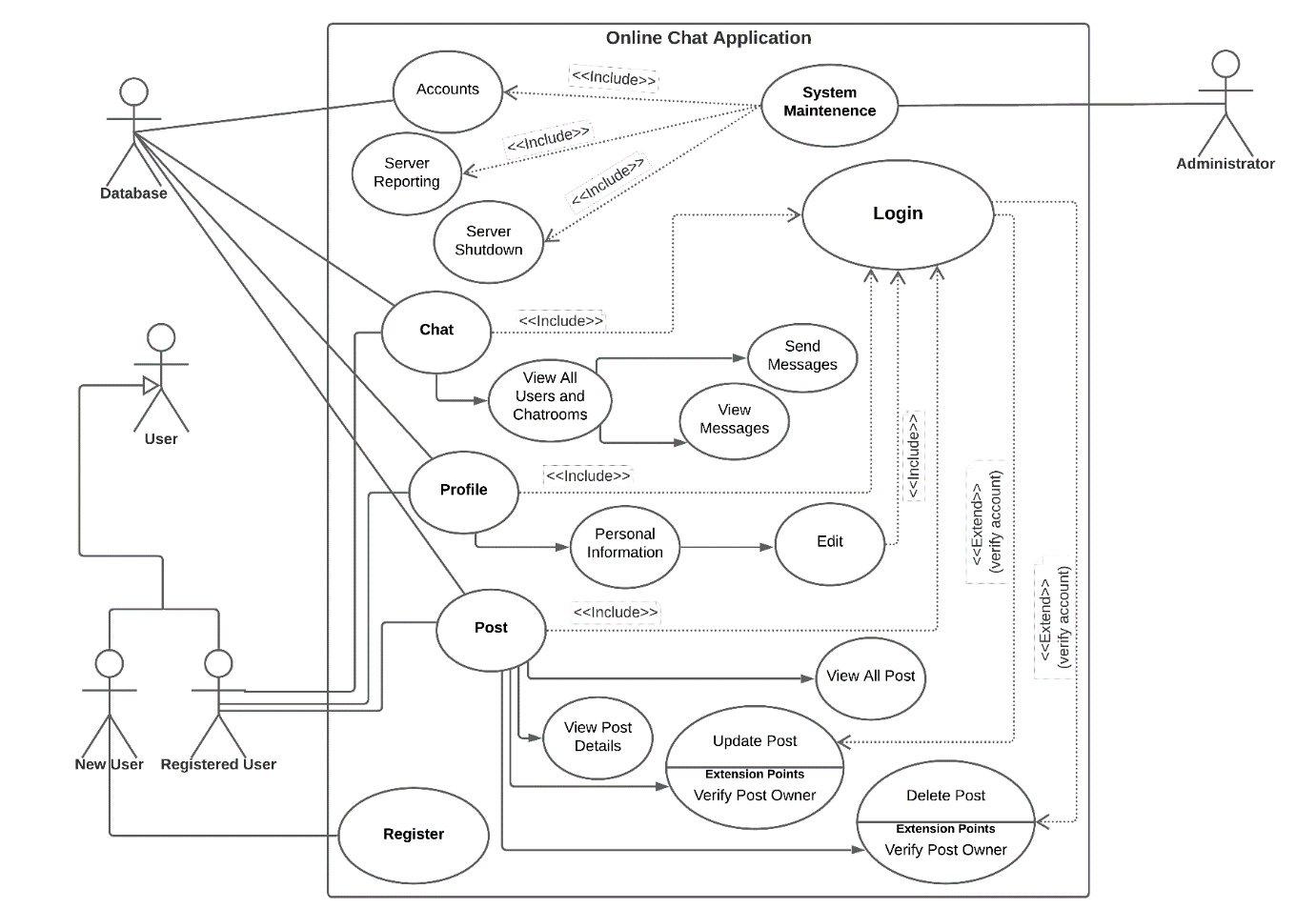


Figure: Use case diagram

**Class Diagram**

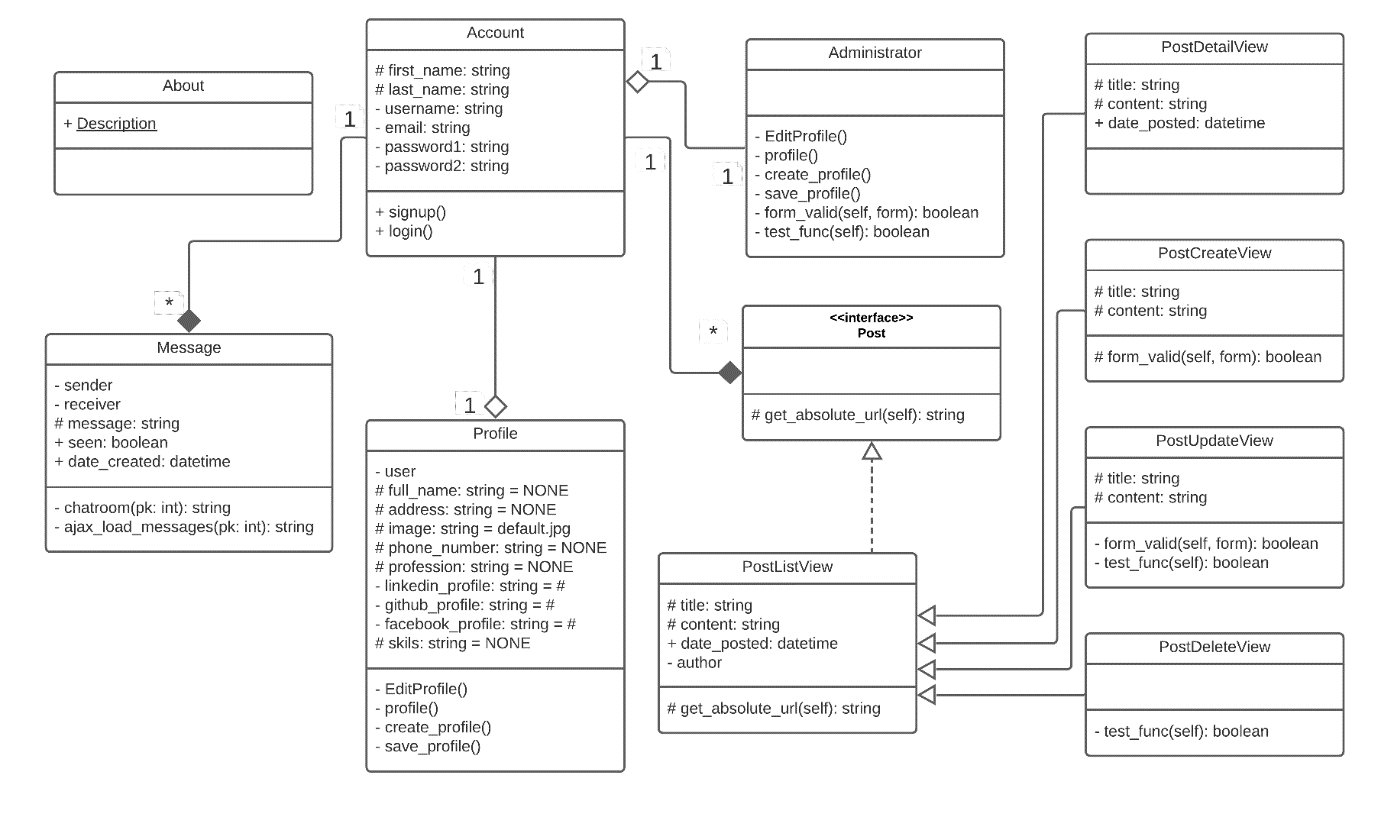


Figure: Class diagram

**Sequence Diagram**

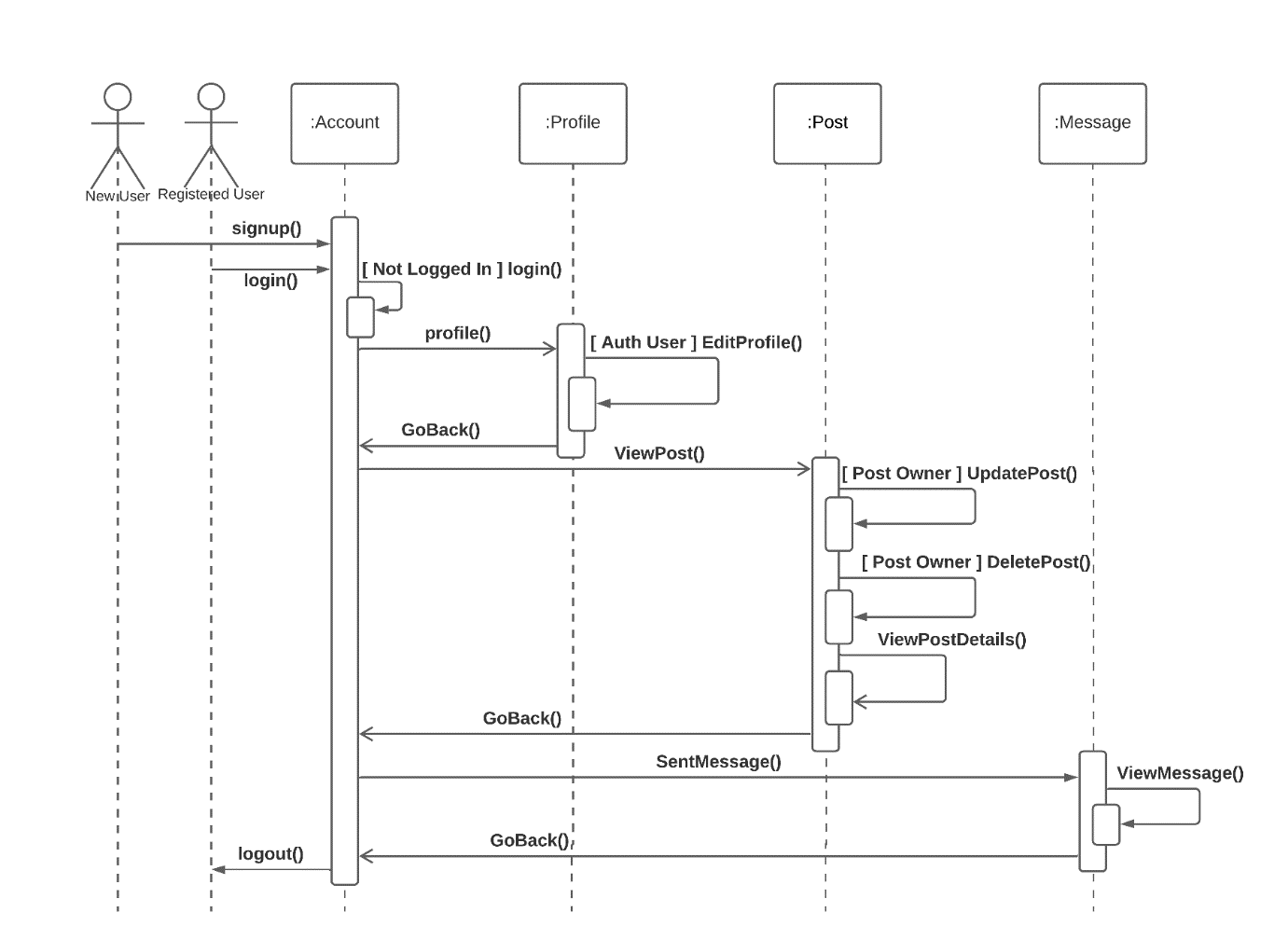


Figure: Sequence diagram

**Activity Diagram**

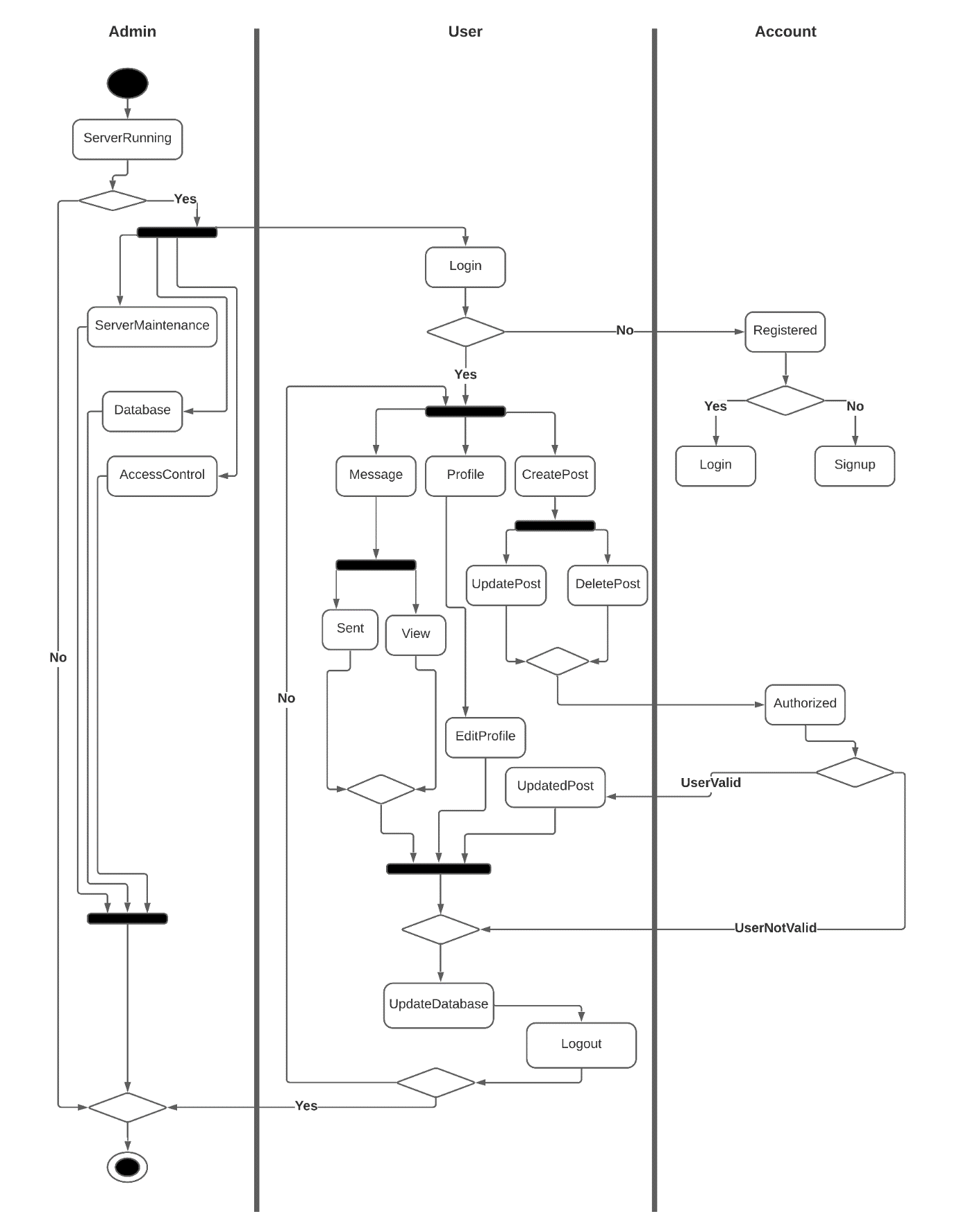


Figure: Activity diagram

**ER Diagram**

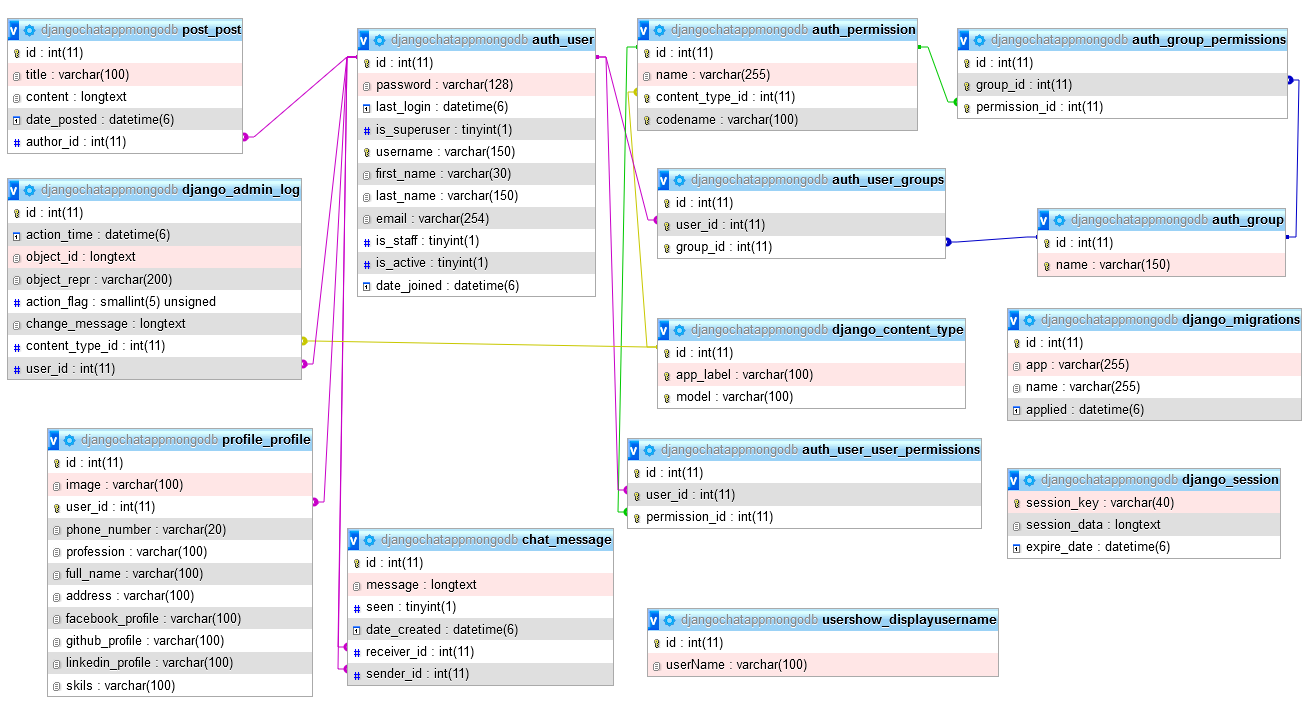


Figure: ER diagram

**Appendix B: External Interface**

**User Interfaces**

1. Users have to login in their account to use the application. Users can be logged in by using username and password.

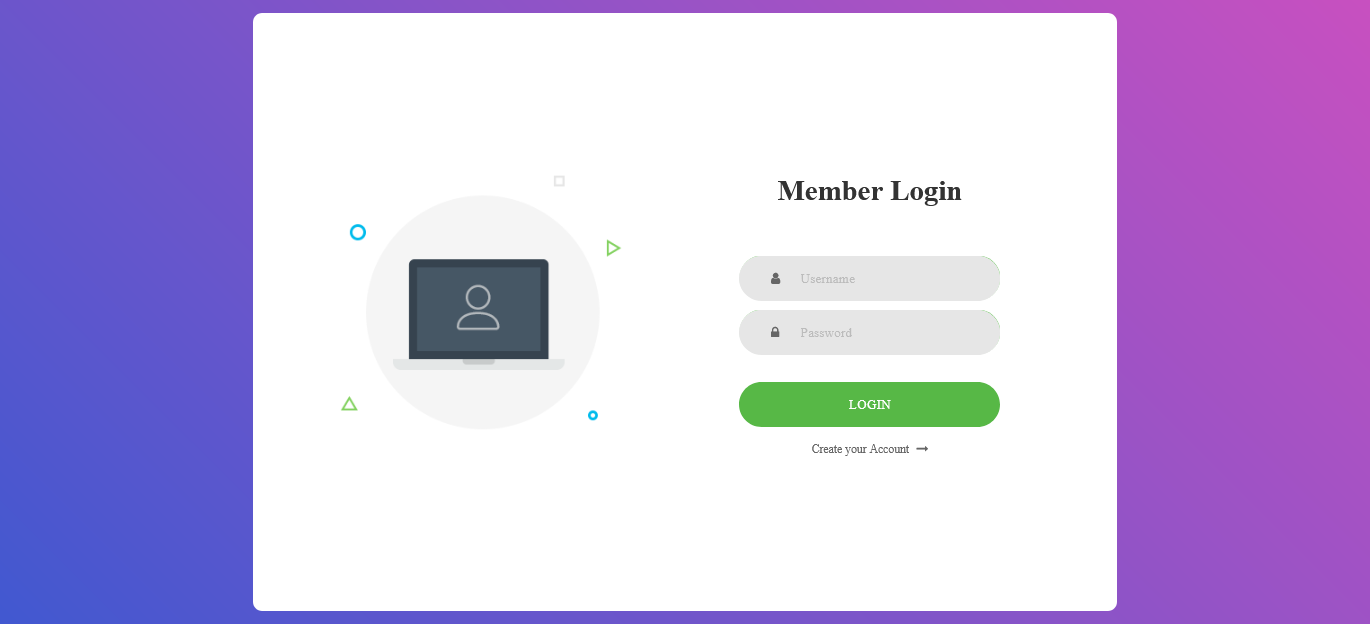


Figure: Login Form

1. New users can register an account for their own by providing first name, last name, username, email and passwords.

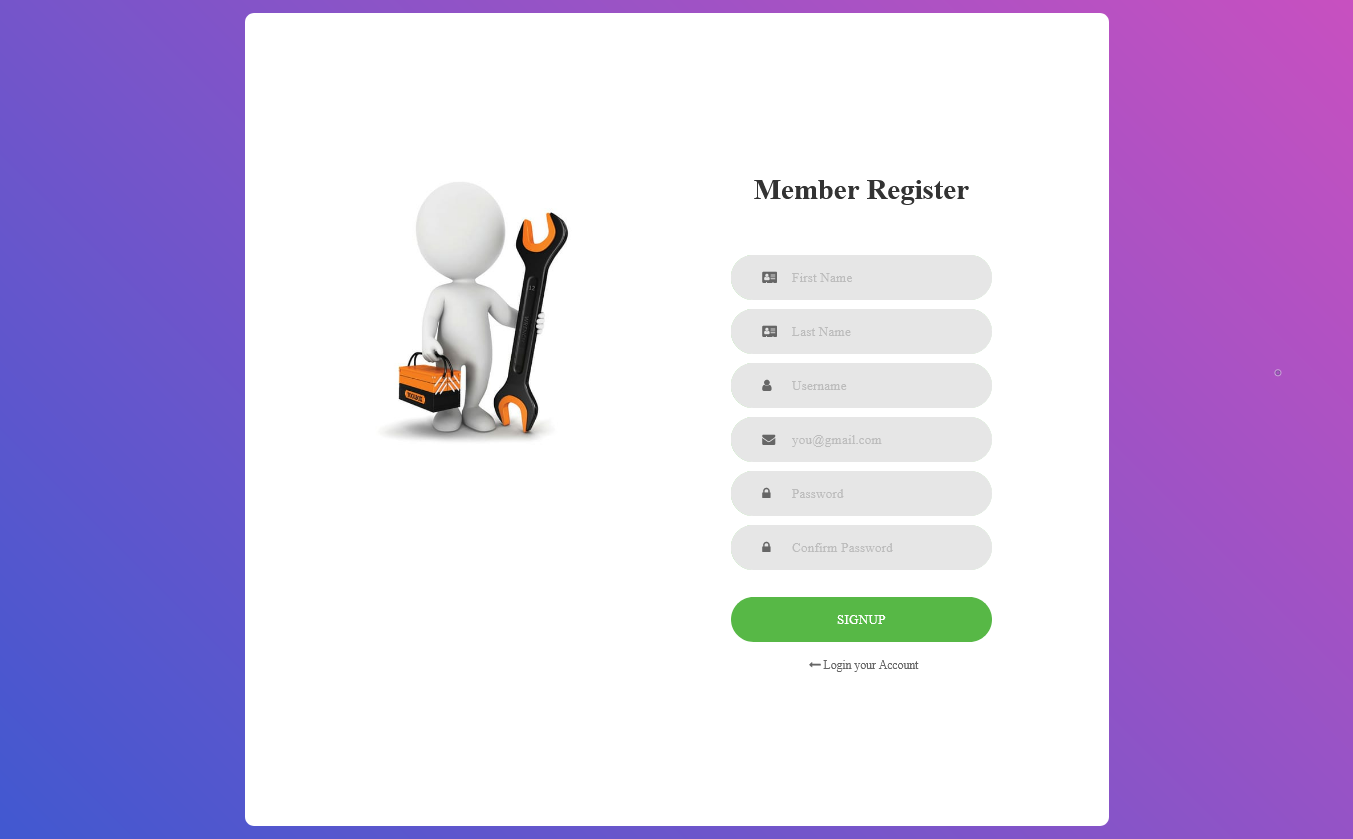


Figure: Registration Form

1. There is a welcome message on the home page. Also some dynamic and attractive design for user friendly environments.

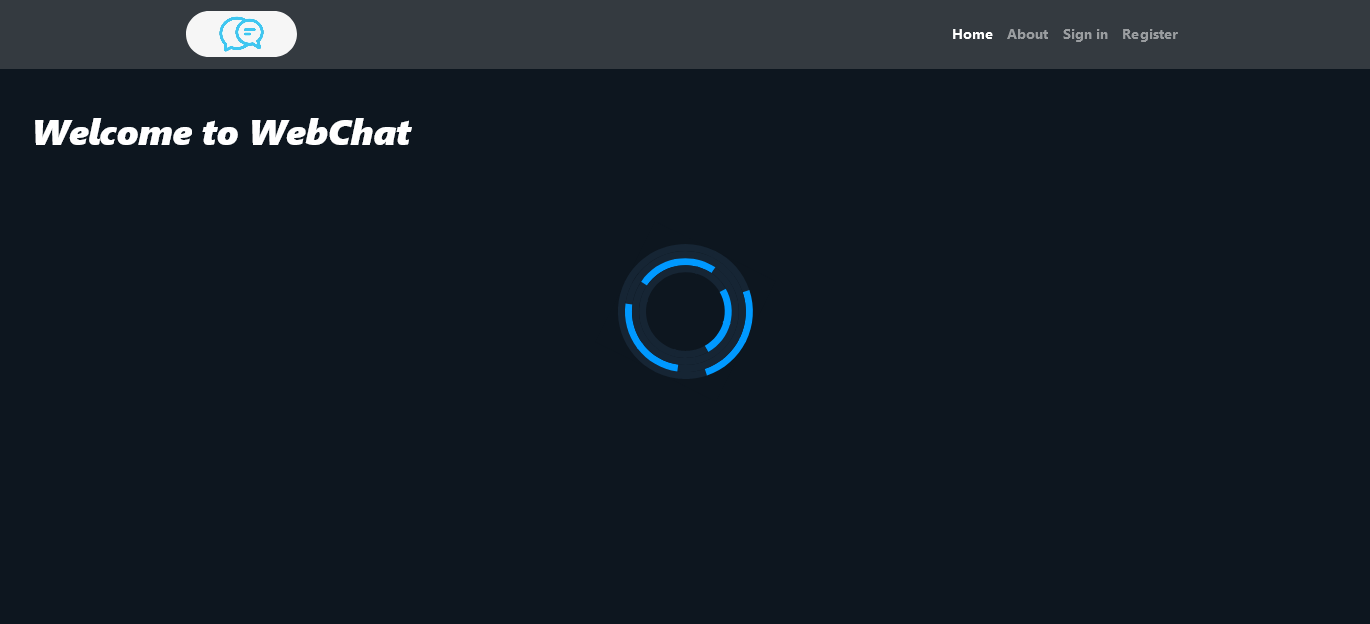


Figure: Home Page

1. From the dashboard, users can make their choice what they want to do.

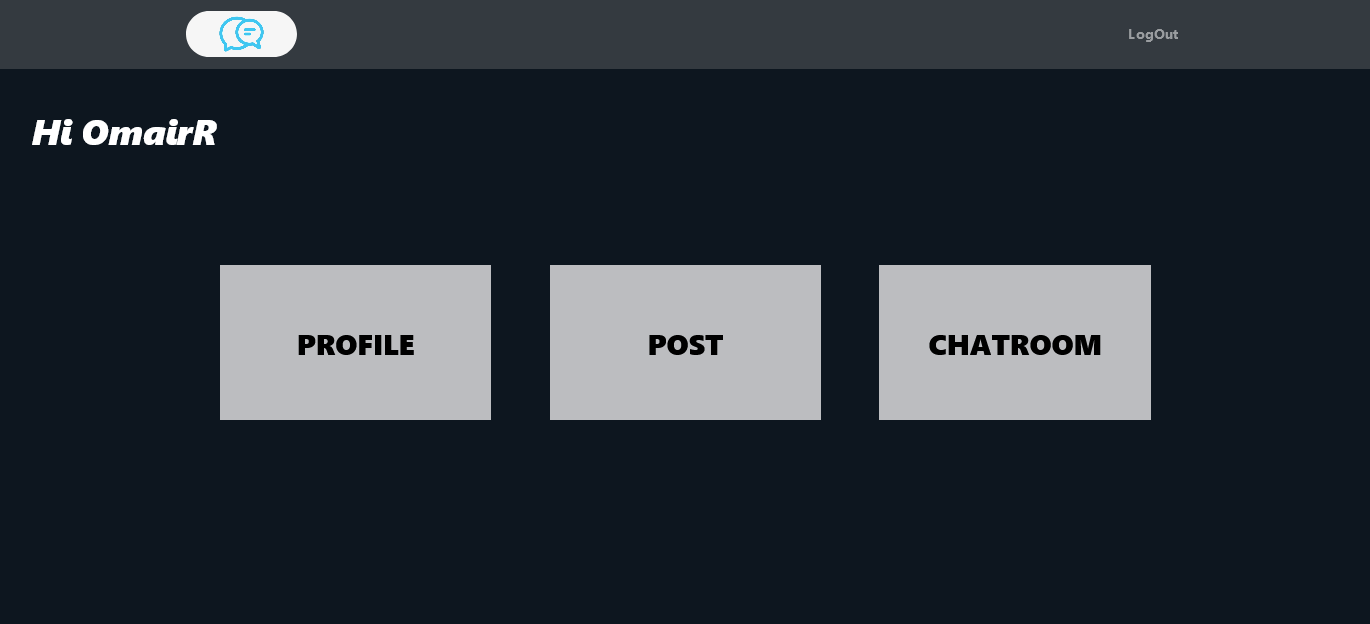


Figure: Dashboard Page

1. Users can see all registered accounts. They can select any one to communicate from the list.

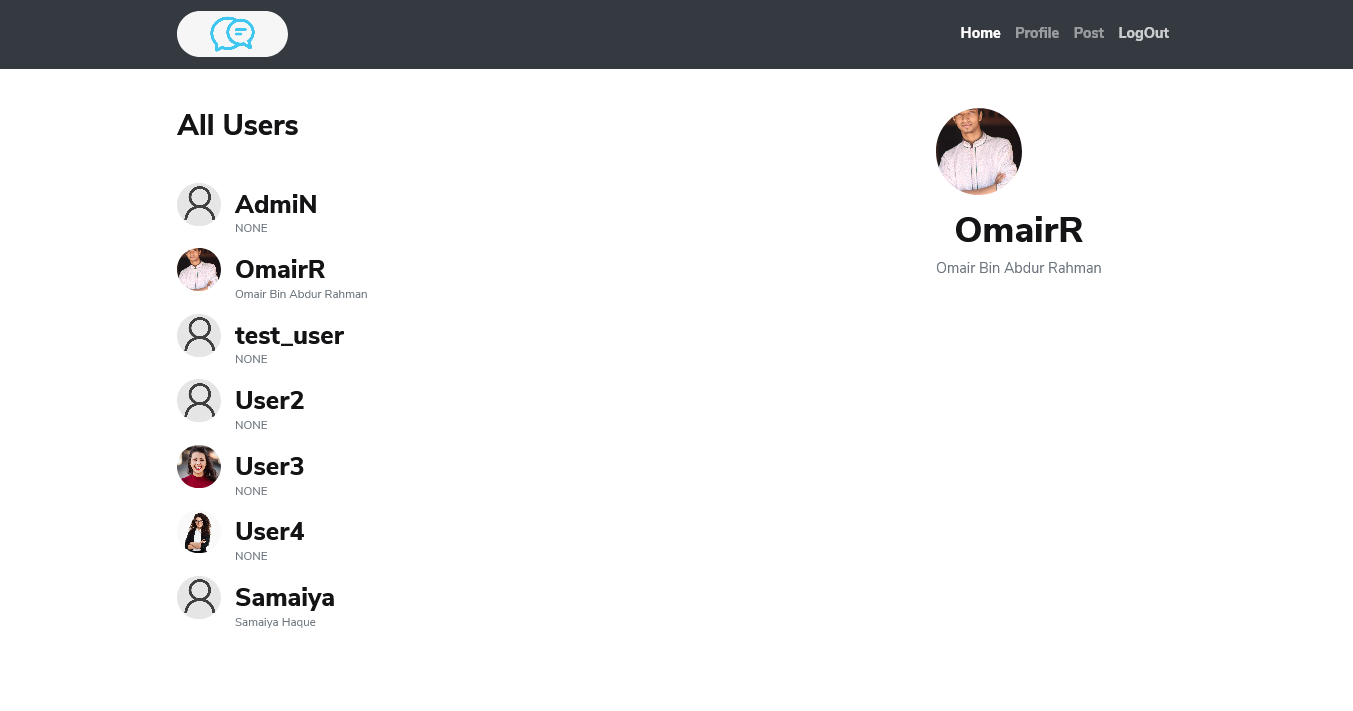


Figure: All User List

1. All users have a default profile. Users can view their profile information any time.

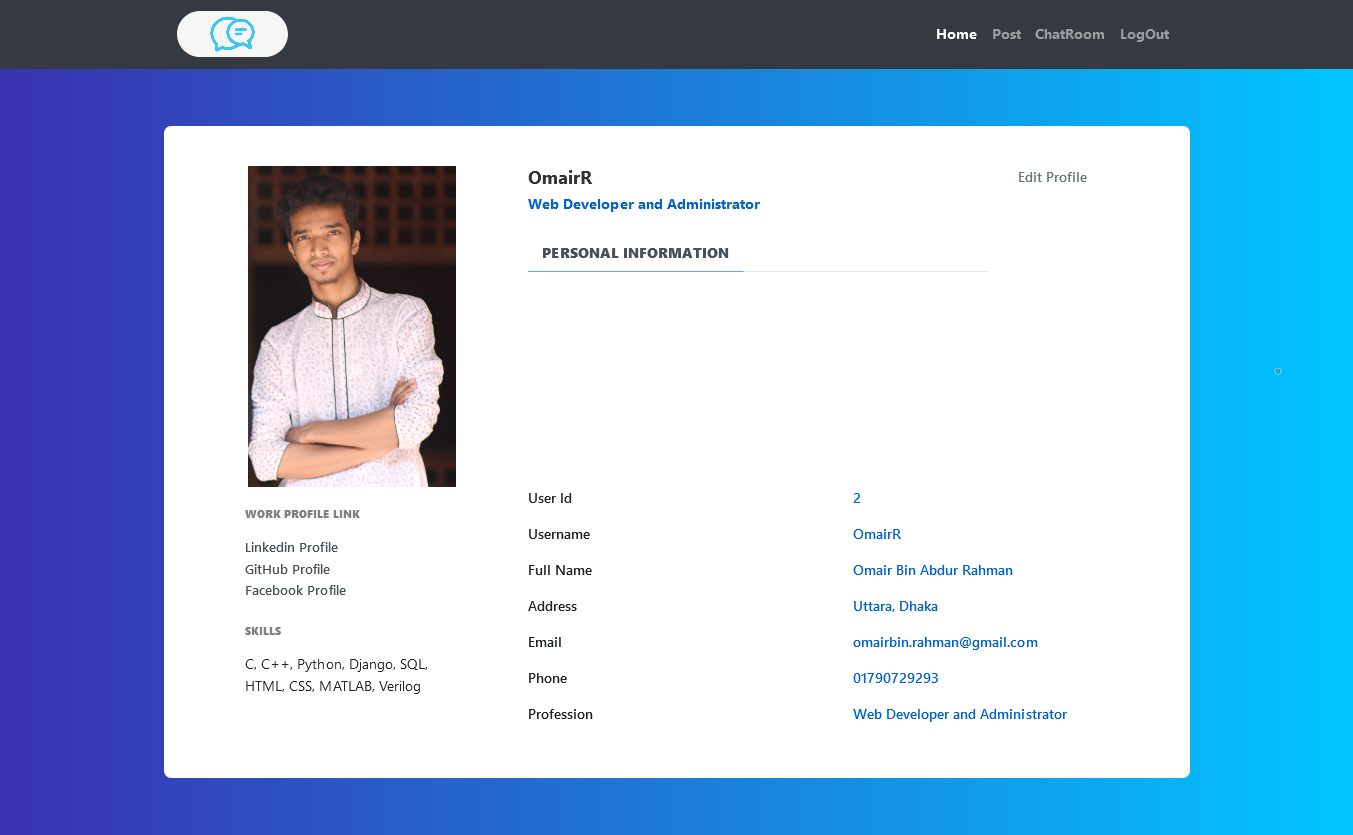


Figure: Profile Information

1. Default user profile. If a user creates an account to the system, this default profile will be created automatically.

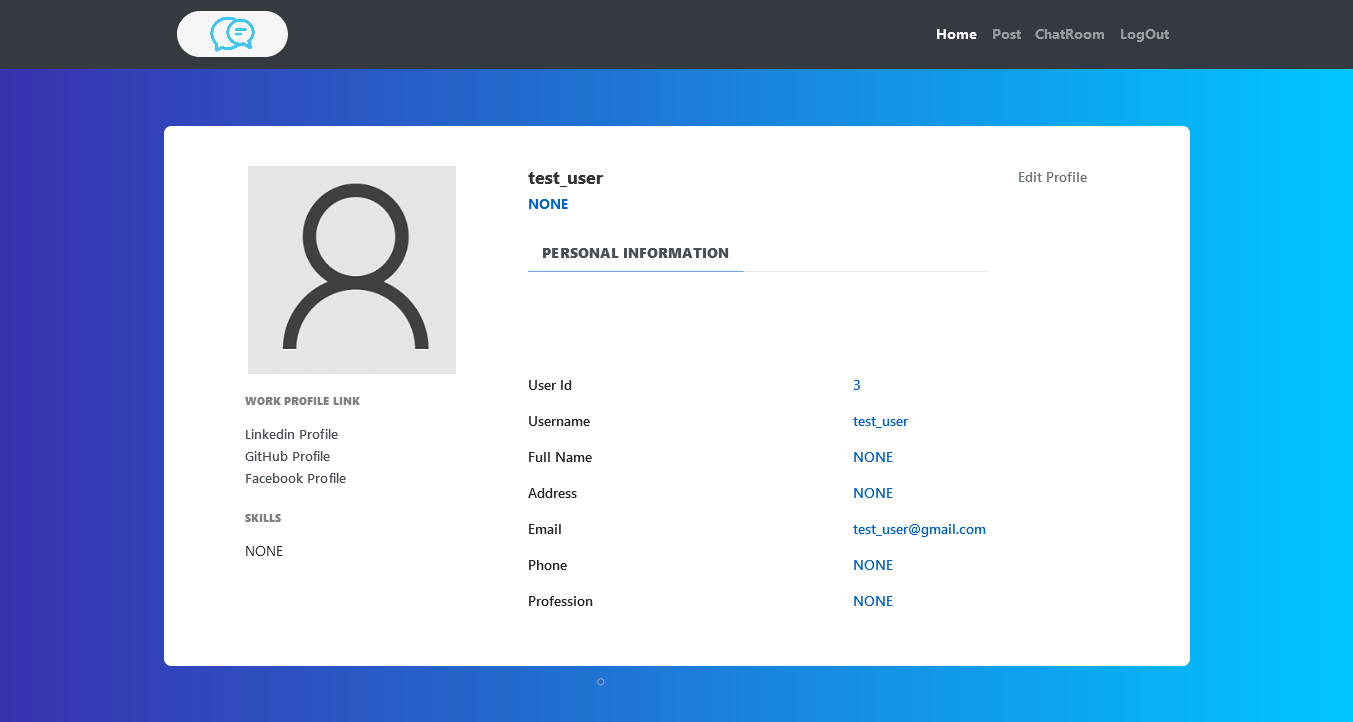


Figure: Default Profile

1. Users can view all the posts which are being posted.

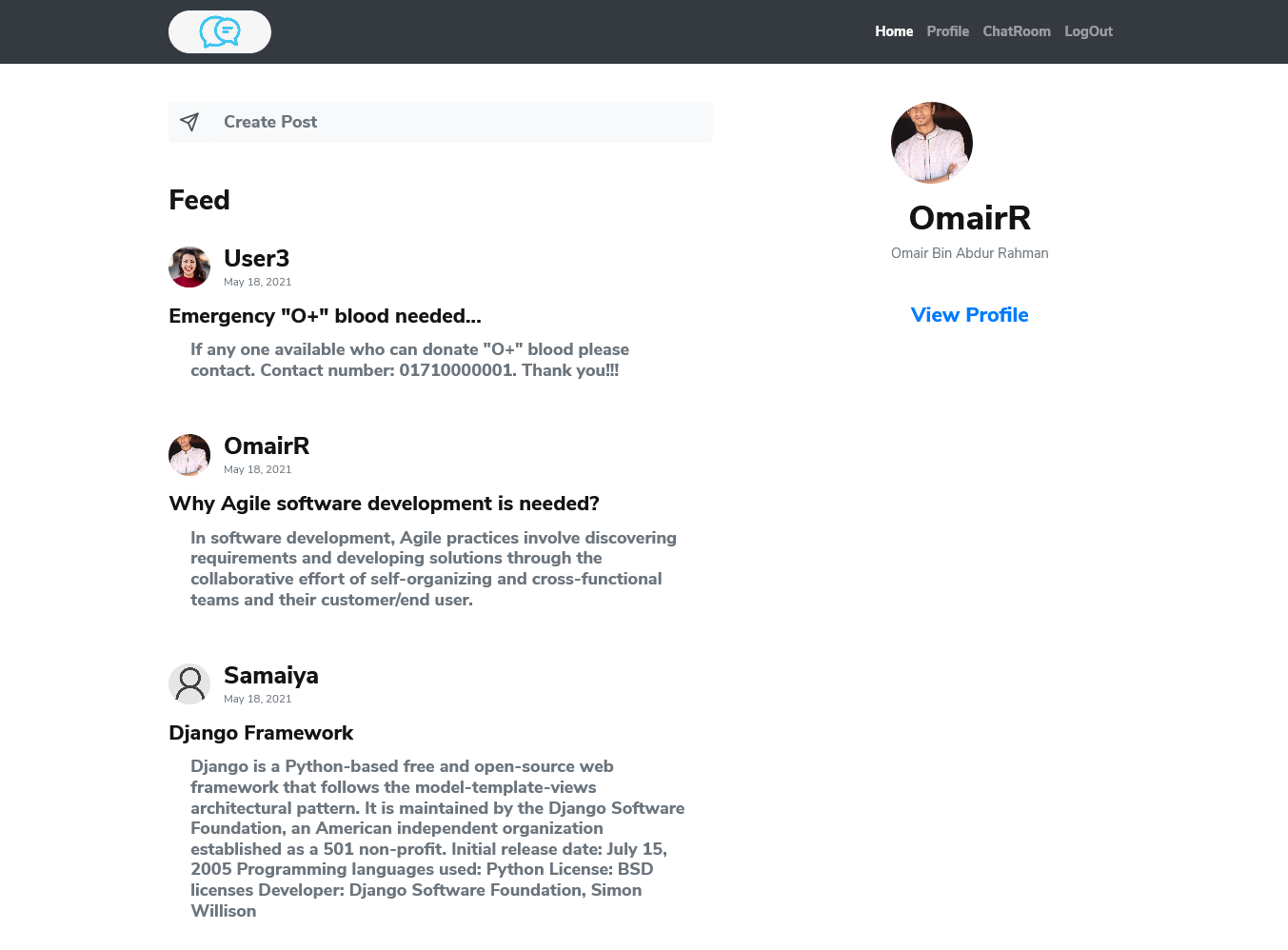


Figure: All Posts

1. Users can edit their profile information and update them any time. They can change their profile picture as well.

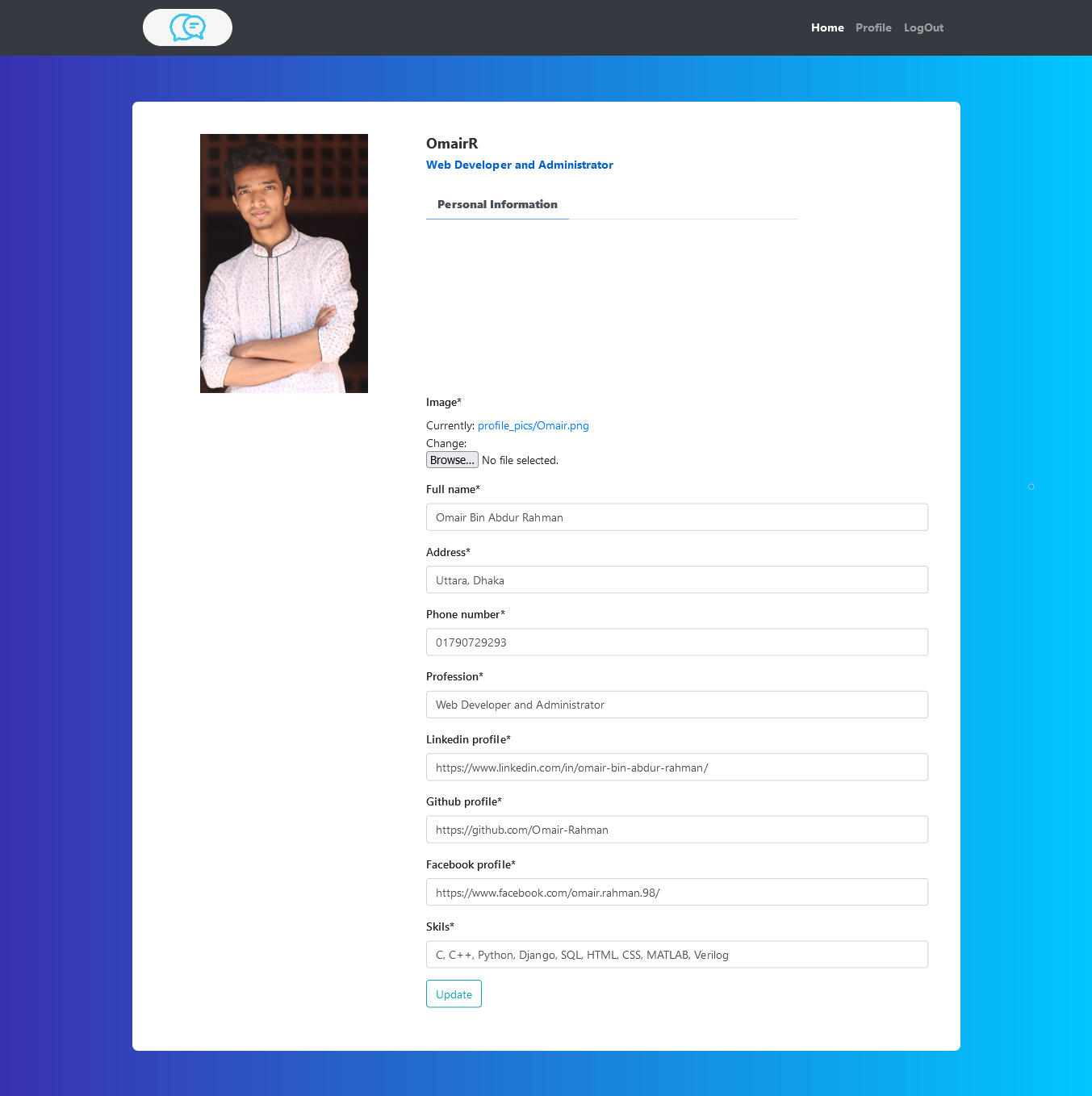


Figure: Profile Update Form

1. Users can view all the posts which are being posted.

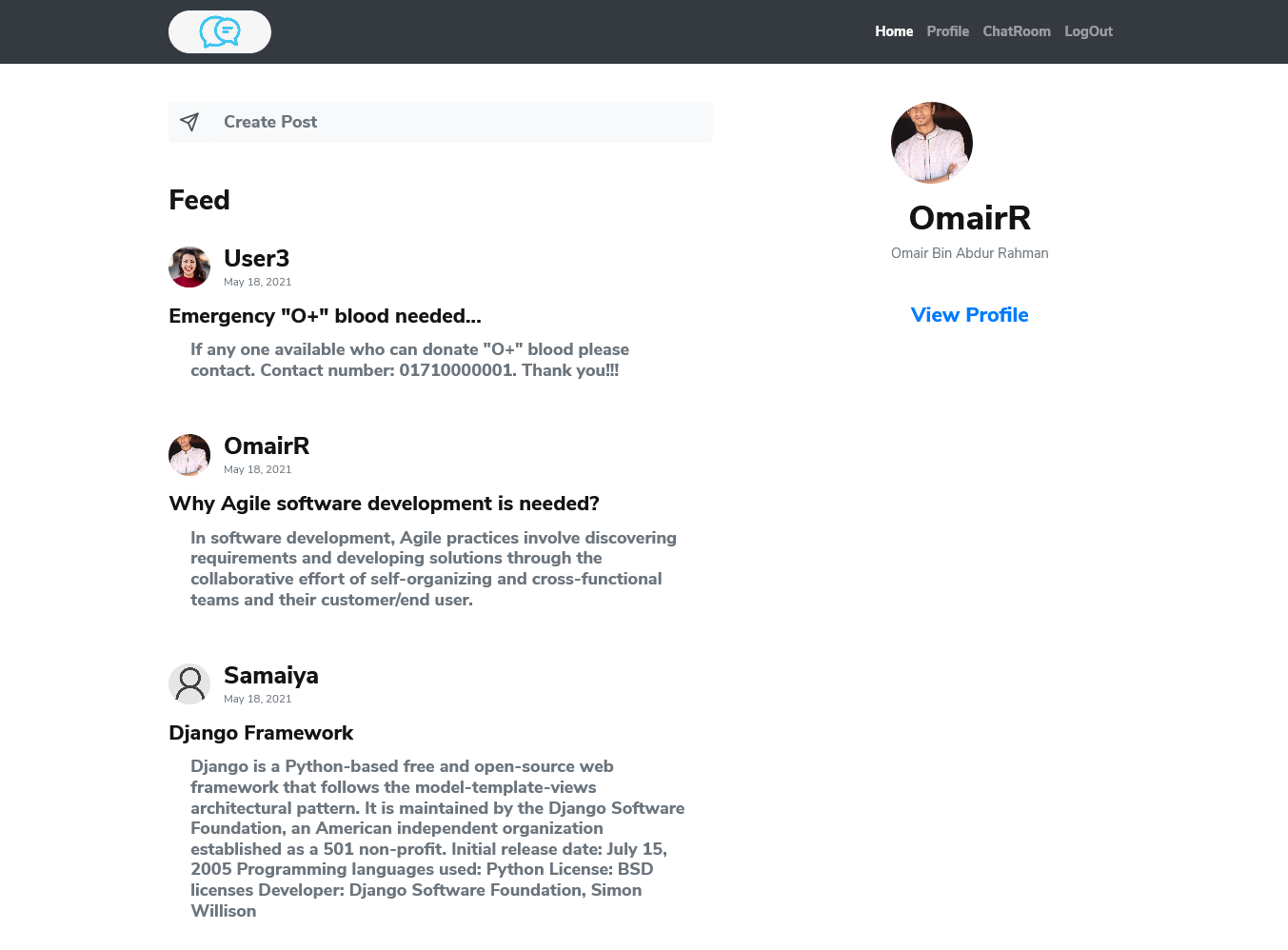


Figure: All Posts

1. Users can view individual post details.

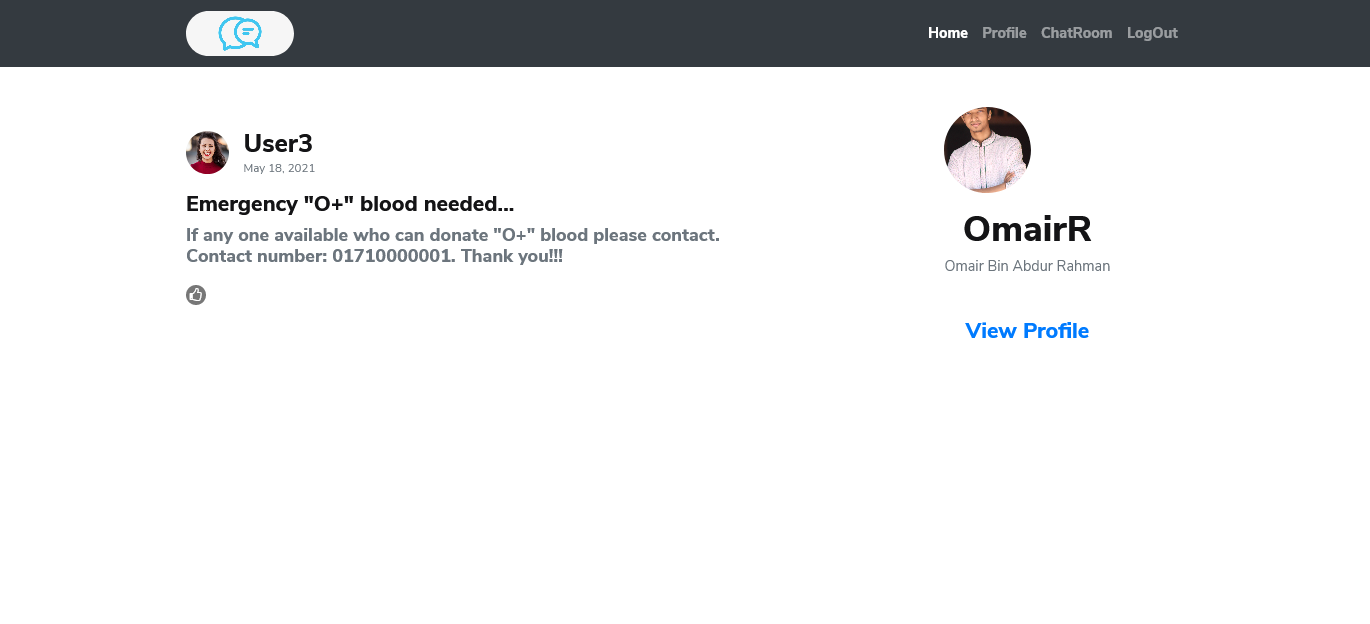


Figure: Other’s Post Details

1. If users view their own post they will find update and delete options. Otherwise they can only view the post details.

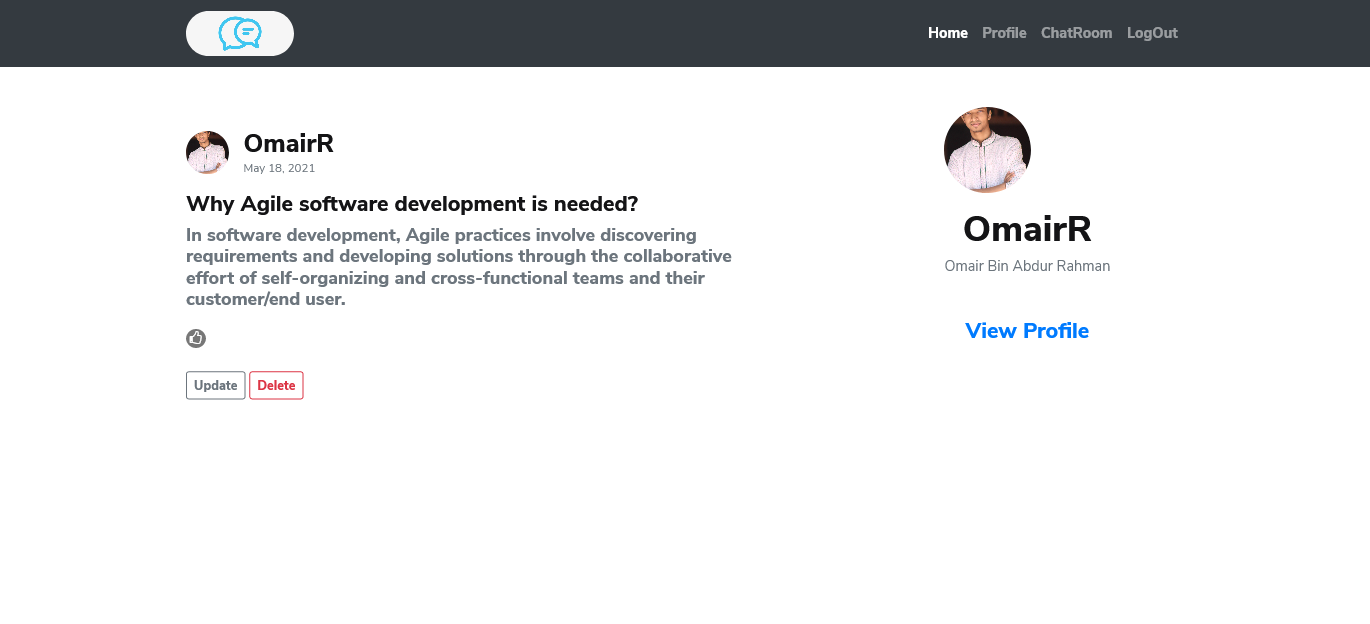


Figure: Own Post Details

1. Users can react such as ‘like’, ‘love’, ‘haha’ on a post.

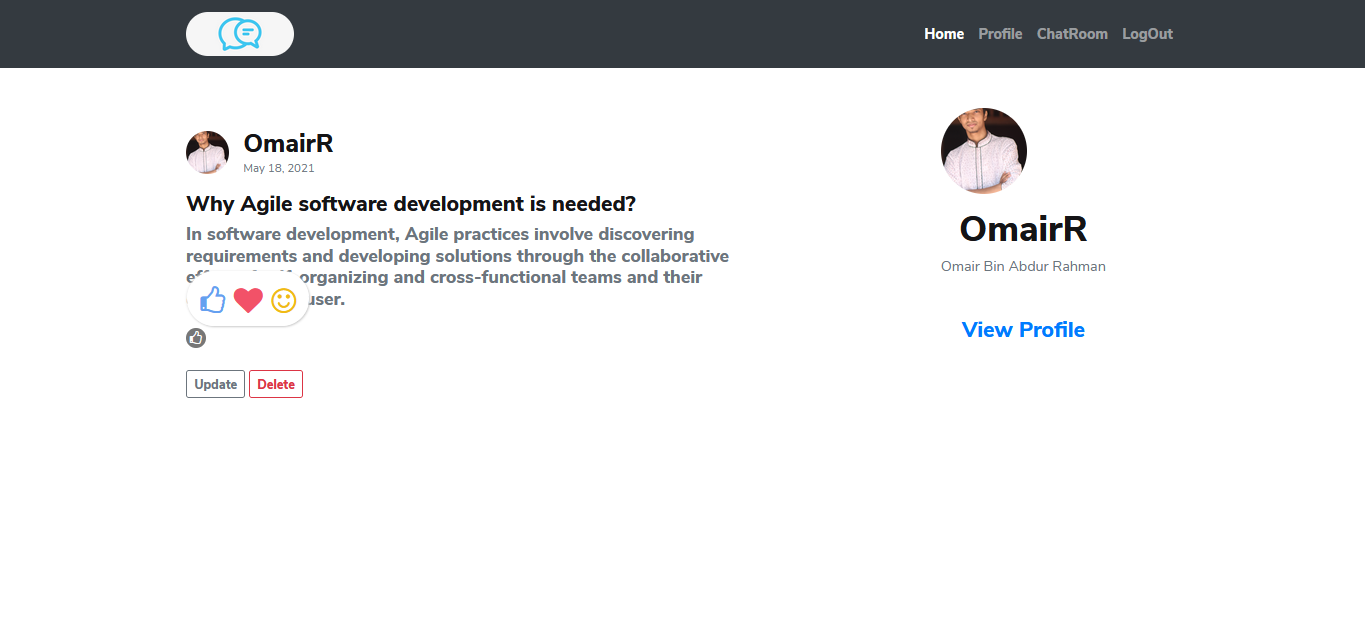


Figure: Post React

1. Users can create posts any time.

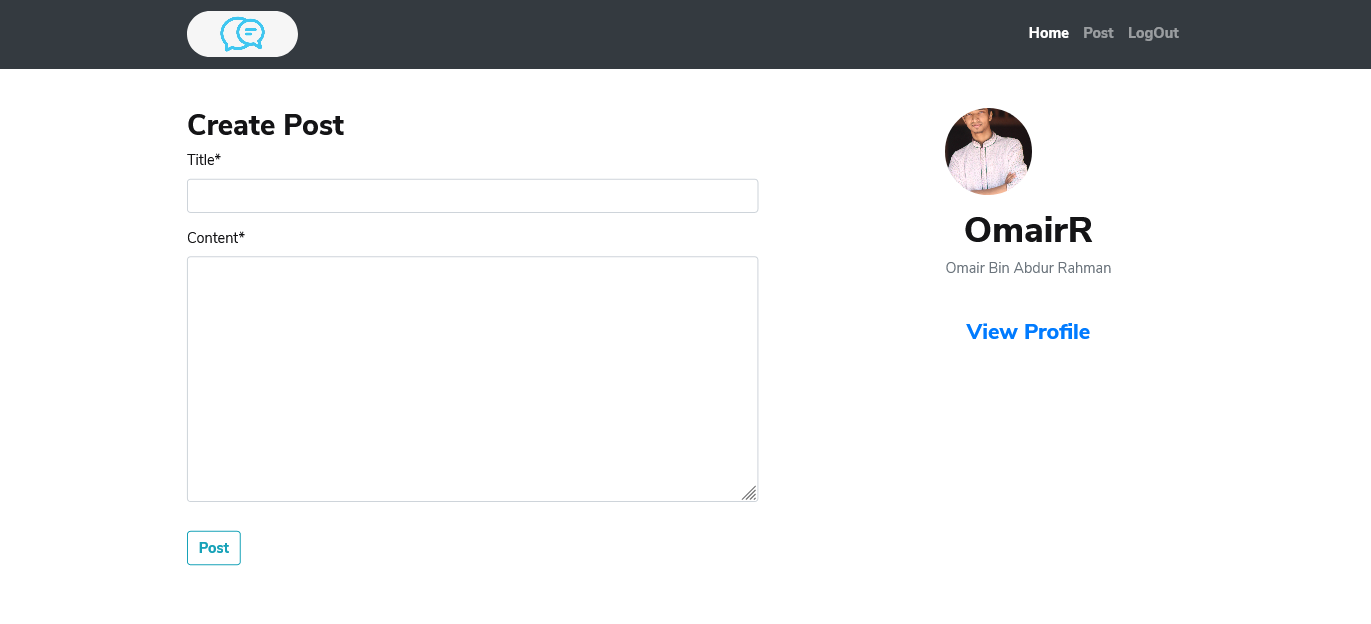


Figure: Post Create Form

1. Users can update their own post any time.

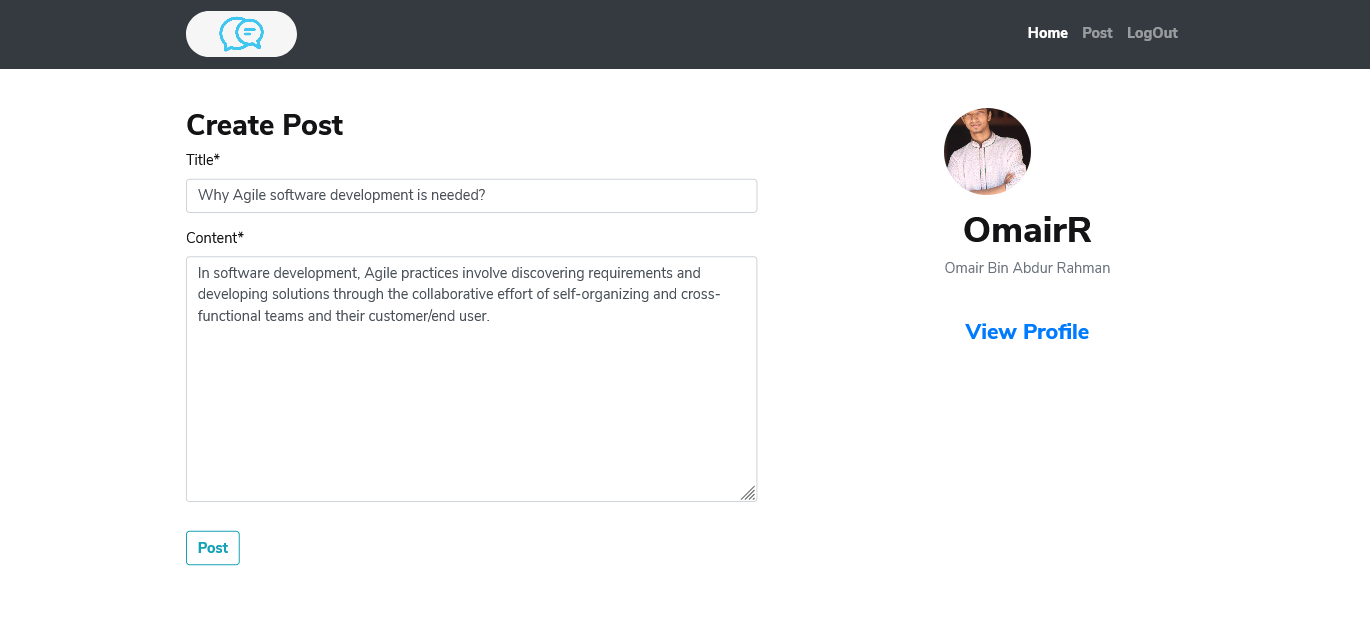


Figure: Post Update Form

1. Users can delete their own post any time.

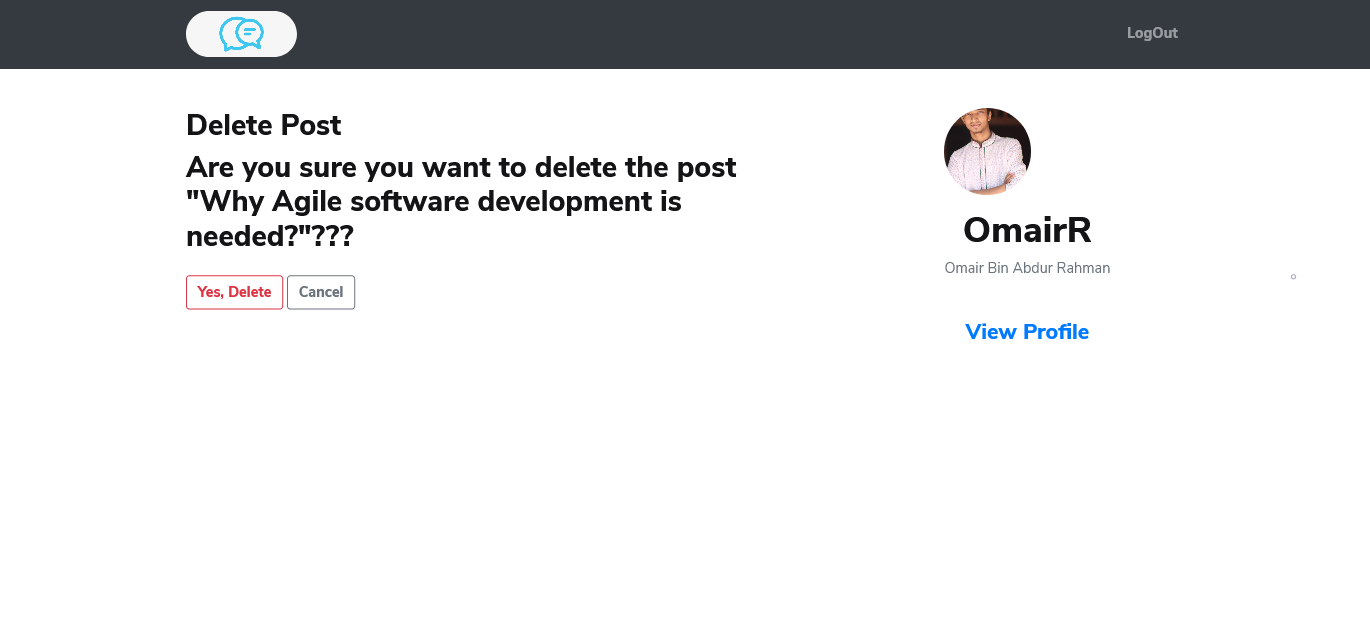


Figure: Post Delete

1. Users can send messages and view the messages at the same time. It is a real time chat system.

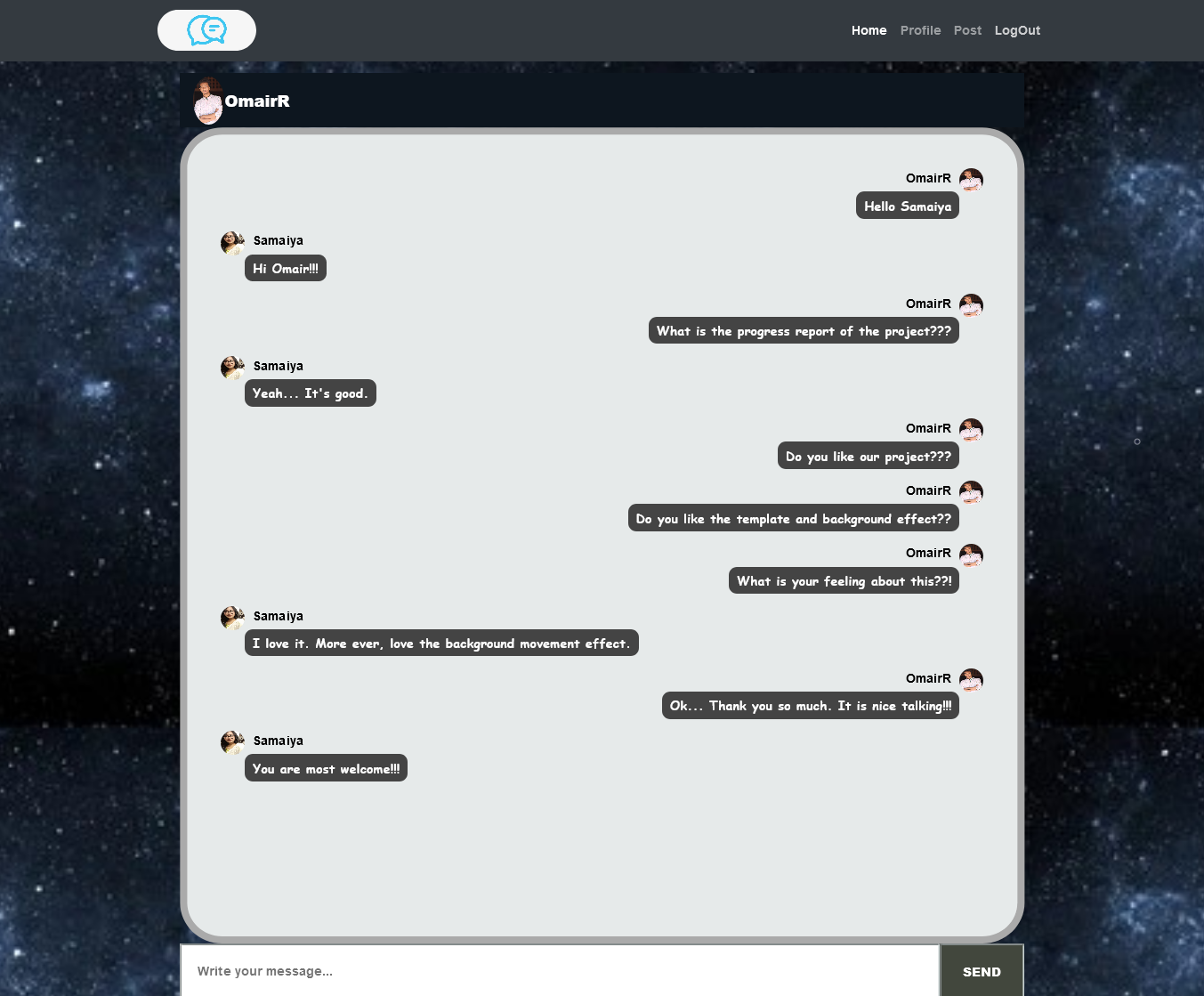


Figure: Message Form

1. If the password does not match during registration, it will show an error message.

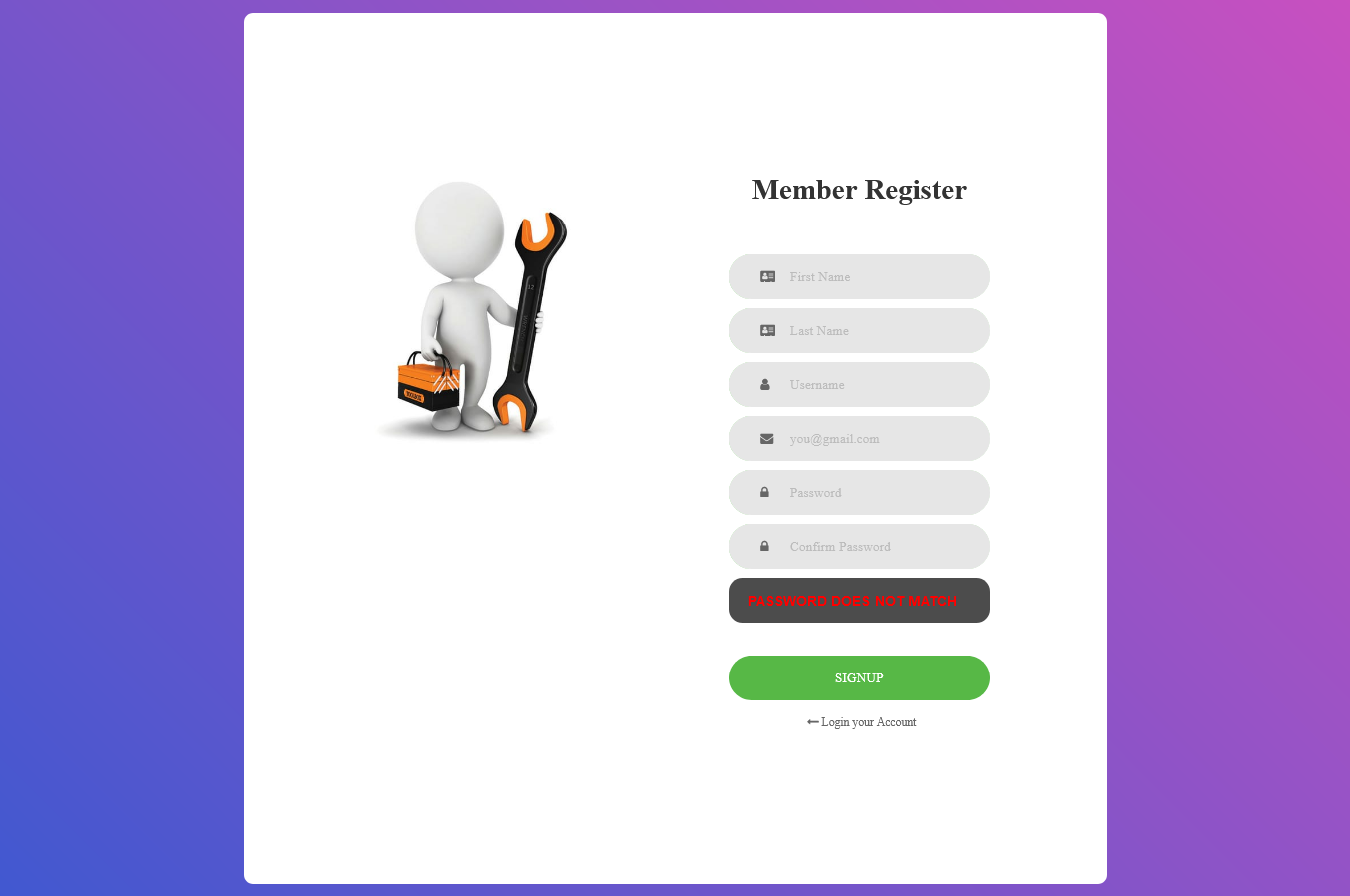


Figure: Password Not Match

1. If a given username already exists while registration, it will show an error message.

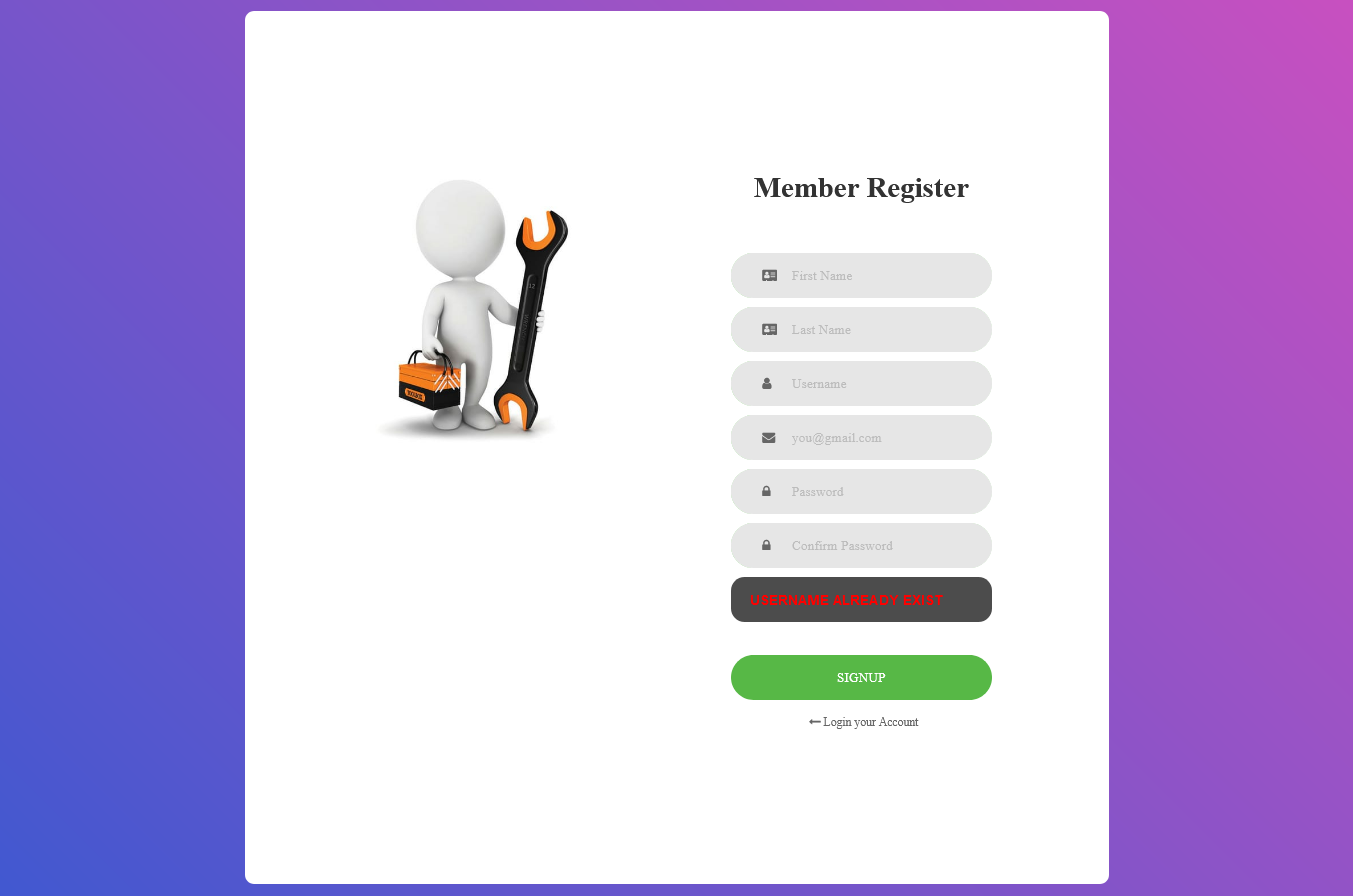


Figure: Username Already Exist

1. If a given email already exists while registration, it will show an error message.

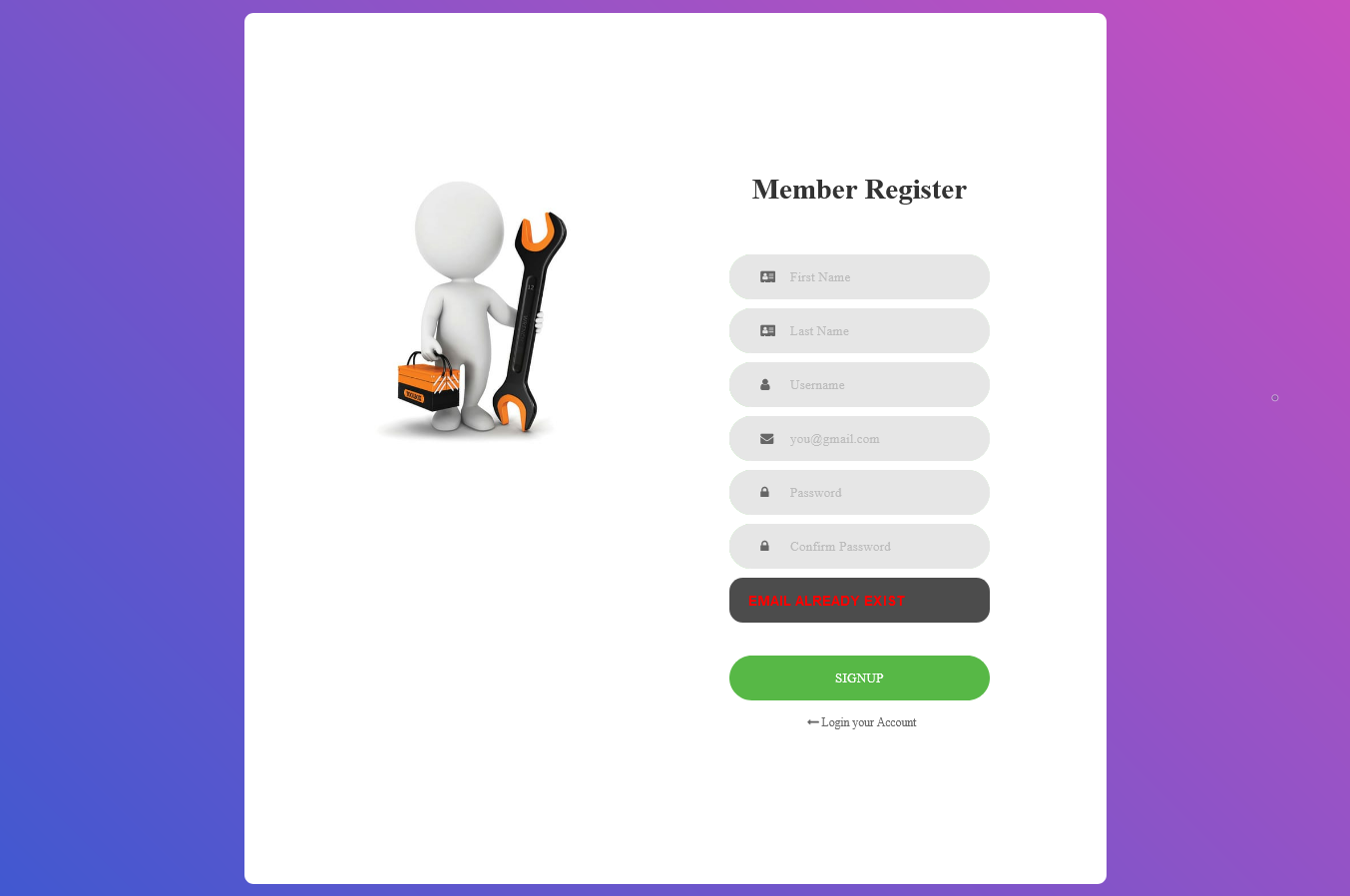


Figure: Email Already Exist

**Hardware Interfaces**

The minimum hardware requirements of that supported to install a web browser that can easily access the internet. Django needs to be installed on the computer with a MongoDB database to locally run this system.

**Software Interfaces**

To run the application, users need a browser. Django, any command line interface and MongoDB database need to locally run this system.

**Communications Interfaces**

A good internet connection is needed for this application system.