**A**

**Project Report On**

**Socio-Connect**

**Designed By:**

**Mr. Ninad Vilas Shingare**

**Submitted To:**

**K.M.S.P. MANDAL’S**

**SANT RAWOOL MAHARAJ MAHAVIDYALAYA, KUDAL.**

**(NAAC Re-Accreditation “B” Grade**

**In the partial fulfillment of**

**B.Sc. Computer Science (Sem VI)**

**Under Guidance of:**

**Asst. Prof. Santosh R. Chougule(M.C.A)**

**Asst. Prof. Yashashree H. Khanolkar(M.Sc.IT)**

**Through:**

**THE HEAD,**

**DEPARTMENT OF COMPUTER SCIENCE,**

**S.R.M. COLLEGE, KUDAL.**

### PROJECT REPORT

**2022-23**

**DECLARATION**

To,

The Head,

Department of Computer Science,

S. R. M. College, Kudal.

Respected Sir,

We undersigned, hereby declare that the project on **“Socio Connect”** is developed under the guidance of our lecturer **Asst.Prof. Santosh R. Chougule. & Asst.Prof.Yashashree H. Khanolkar .**

**DATE:**

**PLACE**: KUDAL.

(Mr. Ninad Vilas Shingare)

|  |  |
| --- | --- |
|  | K.M.S.P. MANDAL’S  **SANT RAWOOL MAHARAJ MAHAVIDYALAYA,**  **KUDAL.**  **NAAC RE-ACCREDITATION “B” GRADE** |

**CERTIFICATE**

This is to certify that **Mr. Ninad Vilas Shingare** having Exam Seat number \_\_\_\_\_\_\_\_\_ have satisfactorily carried out his project work entitled.

**Socio-Connect**

As per the syllabus prescribed for **B.Sc. Computer Science Sem VI** of **Mumbai University, Mumbai**.

It is also certifying that this is his own work completed during academic year 2022-23. The work done is satisfactory and is presented as per the specifications.

**Project Guide External Examiner Head of Department**

**DATE:**

**ACKNOWLEDGEMENT**

It’s my great pleasure to get this opportunity and sincerely thank all the people, who have the way to create a successful project and help me a lot during the completion of this project.

I would like to thank my project guide, **Asst.Prof. Santosh R. Chougule & Asst.Prof. Yashashree H. Khanolkar .**

**,** who provide important information and guided me throughout the project completion.

And last but not the least! I would like to thank all my classmates.

**Thank You.**

**Mr.Ninad Vilas Shingare**

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No** | **Chapter No.** | **Title** | **Page No.** |
| **1** | **I** | **Acknowledgement** |  |
| **2** | **II** | **Synopsis of project** | 1-2 |
| **3** | **III** | **Main Report**   * Objective & Scope of Project * Theoretical Background * Survey of Literature * Feasibility Study * Software Requirement Specification * System Planning * Details of hardware and software used * Detailed lifecycle of the project * Data Flow Diagram * Use Case Diagram * Class Diagram * Entity Relationship Diagram * Activity Diagram * Object Diagram * Input and Output Screen Design * Process Involved * Methodology used for testing * Code of the Project | 3-50 |
| **4** | **IV** | **Annexure**   * List of Database models | 51-53 |
| **5** | **V** | **Conclusion** | 54 |
| **6** | **VI** | **Future Enhancement** | 55 |
| **7** | **VII** | **References & Bibliography** | 56 |

**Chapter II**

**Synopsis of the Project**

**Socio-Connect**

**Problem Statement**

Socio-Connect is a Django based social media website where users can utilize the features exactly as on other straightforward social media websites. The goal of the project is to demonstrate how the most popular web development languages might be utilized to create a straightforward social media website.

And as in other applications they provide too many time consuming useless features that make users get carried away losing the main morality of the application.

**Objective and Scope**

One of the main reasons to create a social media clone is to offer an alternative platform to users who are dissatisfied with existing social media platforms or are looking for something new. Building a social media clone requires careful planning, attention to detail, and a talented development team. This application aims to exactly provide the features only which are necessary so the users don’t get carried away which other features which tend to waste their time.

Presenting a straightforward yet distinctive website that suits the demands of a user like any other social media websites. The project created is a social media web application where visitors may register and utilize the features exactly as on other straightforward websites. The system will be built using the Django web framework and the Bootstrap framework for the frontend design.

**Chapter III**

**Main Report**

**Objective and Scope**

One of the main reasons to create a social media clone is to offer an alternative platform to users who are dissatisfied with existing social media platforms or are looking for something new. Building a social media clone requires careful planning, attention to detail, and a talented development team. This application aims to exactly provide the features only which are necessary so the users don’t get carried away which other features which tend to waste their time.

Presenting a straightforward yet distinctive website that suits the demands of a user like any other social media websites. The project created is a social media web application where visitors may register and utilize the features exactly as on other straightforward websites. The system will be built using the Django web framework and the Bootstrap framework for the frontend design.

**The objective of this system can broadly be listed as follows:**

- Website that suits the demands of a user like any other social media websites

- To offer an alternative platform to users who are dissatisfied with existing social media platforms

- No useless features that consume time of users

- More focused and concerned about user privacy and security

- Social media clones can also be monetized later through various means, such as advertising, subscription fees

**Theoretical Background**

A social media clone project is a software development project that aims to create a platform that emulates the functionality of an existing social media platform, such as Facebook, Twitter, or Instagram. The project requires a good understanding of web development, database design, and user interface design.

The theoretical background of a social media clone project involves several key concepts:

Front-end development: This involves creating the user interface of the platform using HTML, CSS, and JavaScript. The front-end should be designed to be user-friendly, aesthetically pleasing, and responsive to different screen sizes and devices.

Back-end development: This involves creating the server-side logic of the platform using programming languages such as Python, Ruby, or Java. The back-end should handle user authentication, data storage, and communication with external APIs.

Database design: This involves designing the database schema and choosing a database management system (DBMS) to store user data. The DBMS should be scalable, reliable, and secure.

User authentication and security: This involves implementing authentication mechanisms to ensure that only authorized users can access the platform. It also involves implementing security measures to protect user data from unauthorized access, such as encryption and data backups.

Testing and quality assurance: This involves testing the platform to ensure that it functions as intended, is free of bugs, and is scalable and reliable. It also involves implementing quality assurance measures, such as code reviews and automated testing, to ensure the codebase is maintainable and well-documented.

Overall, a social media clone project requires a solid understanding of web development, database design, and security principles, as well as a keen eye for user interface design and user experience.

**Survey of Literature**

This study is exploratory in nature and founded on reviewing of latest available literature on social media user typologies. The two main objectives of this study are to

* Identify the usage of social media and types of social media users and
* Determine the basis for user categorization in existing user typologies The scope of the study is pertaining to the review of existing and available literature on the internet and social media user, their media usage and most importantly the different user typology, that emerged from previous studies on user types.
* "Why We Post: The Anthropology of Social Media" by Daniel Miller et al. - This book provides a detailed analysis of social media usage in different cultures and societies around the world.
* "A review of social media research in the tourism industry" by M. Gretzel et al. - This paper provides an overview of research on social media usage in the tourism industry, including topics such as user-generated content, online reviews, and social media marketing.

**Feasibility Study**

Feasibility of a new system means ensuring that the new system, which we are going to implement, is efficient and affordable. There are various types of feasibility to be determined. They are,

1. ECONOMICALLY FEASIBILITY Development of this application is highly economically feasible. The only thing to be done is making an environment with an effective supervision. It is cost effective in the sense that has eliminated the paper work completely. The system is also time effective because the calculations are automated which are made at the end of the month or as per the user requirement.
2. TECHNICAL FEASIBILITY The technical requirement for the system is economic and it does not use any other additional Hardware and software. Technical evaluation must also assess whether the existing systems can be upgraded to use the new technology and whether the organization has the expertise to use it.
3. OPERATIONAL FEASIBILITY The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system. Technical performance include issues such as determining whether the system can provide the right information for the users, and whether the system can be organized so that it always delivers this information at the right place and on time using intranet services. Acceptance revolves around the current system and its personnel.

**Software Requirement Specification**

Functional Requirements-

* Registration and login for the both parties i.e. admin and user.
* User needs to have access to his own profile.
* User must be able to edit or update his own profile details.
* Provide user search facility.
* User must be able to discover other users profile and also follow them.
* User must be able to upload his post along with description.
* User must be able to delete his post later on if he/she wishes.
* User must be able to discover posts of the users he/she follows.
* User must be able to react to posts by clicking like button.
* Users must be able to give feedback to admin for their work.
* User must be able to discover random user whom he/she doesn’t follow
* Only Admin must be able to have full access control authorization.
* Admin must be able to manage details of registered users.
* Admin must be able to view, edit and delete user posts.
* Admin must be able to view, edit and delete user feedback.

Non-functional Requirements:

* Performance
* Security
* Data storage

1. Safety
2. Maintainability

**System Planning**

Project planning is at the heart of the project life cycle, and tells everyone involved where you’re going and how you’re going to get there. The planning phase is when the project plans are documented, the project deliverables and requirements are defined, and the project schedule is created. It involves creating a set of plans to help guide you through the implementation and closure phases of the project. The plans created during this phase will help you manage time, cost, quality, changes, risk, and related issues. They will also help you to control external suppliers to ensure that you deliver the project on time, within budget, and within schedule.

You may also need to plan your communications and procurement activities, as well as contract any third-party suppliers.

The purpose of the project planning phase is to:

* Establish business requirements
* Establish cost, schedule, list of deliverables, and delivery dates
* Establish resources plans
* Obtain management approval and proceed to the next phase

**Gantt chart**

This chart displays the expected date & originally completed date i.e. actual completion of my project.

Proposed Time

Actual Time

**Preliminary**

**Investigation**

**System**

**Analysis**

**System**

**Design**

**System**

**Coding**

**Testing &**

**Implementation**

Dec Jan Jan-Feb Feb-March March

**Details of Hardware & Software used**

* **Compatible Browsers** - Firefox, Opera, Chrome, Edge.
* **Files Included** - asgiref==3.6.0, Django==4.1.6, django-adminlte-3==0.1.6, django-adminlte-admin==0.6.1, django-adminlte3==0.1.6, Django-jazzmin==2.6.0, Pillow==9.4.0, sqlparse==0.4.3, tzdata==2022.7
* **Software version** - Python 3.11.1, VS Code 1.76.1
* **Database Server Name** - SQLITE3

* **Frameworks** -Django, Bootstrap

**Detailed Life Cycle of the Project**

**Initialization**

**Requirements**

**Planning**

**Design**

**Evaluation**

Evaluation

**Implementation**

**Deployment**

**Verification**

### Iterative and Incremental Model

### Process Model

### Iterative and Incremental Model

This Iterative and Incremental SDLC model essentially brings together an iterative design with an incremental development model which makes it amongst the best SDLC methodology for business.

The process of development in such a model begins with the simple implementation of basic product requirements. As and when the product keeps building, the product is worked on and enhanced to a more complete version of itself.

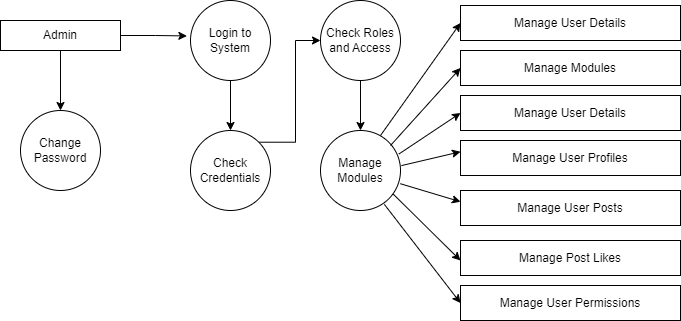
One of the most prominent features of this model is the fact that one can begin the development process without really knowing the requirements.  The iterative and incremental SDLC model, in other words, looks like a set of Mini Waterfall/V-Shaped models.

**Benefits:**

* Produces quick value for the business
* Need limited resources
* Flexible model
* Facilitates early bug detection
* Easy

**Data Flow Diagram**

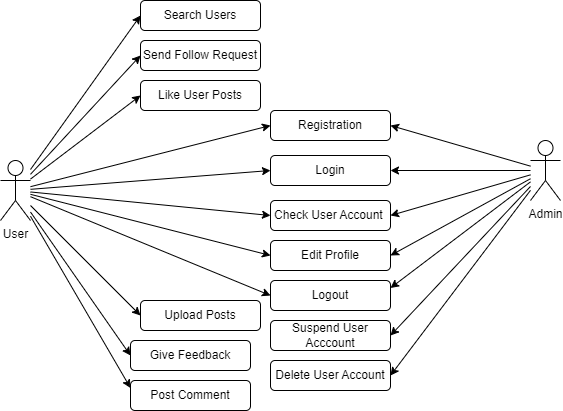
It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.



**Use Case Diagram**

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

* Scenarios in which your system or application interacts with people, organizations, or external systems.
* Goals that your system or application helps those entities (known as actors) achieve.
* The scope of your system.



**Class Diagram**

Class diagrams express, in a general way, the static structure of a system, in terms of classes and relationships between those classes

Just as a class describes a set of objects, an association describes a set of links; objects are class instances, and links are association instances.

A class diagram does not express anything specific about the links of a given object, but it describes, in an abstract way, the potential links from an object to other objects.

**Symbols used to draw Class diagram -**

**Class Name**

**Member Variables**

**Member Functions**

-Rectangular box

(To represent class)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-Connection line

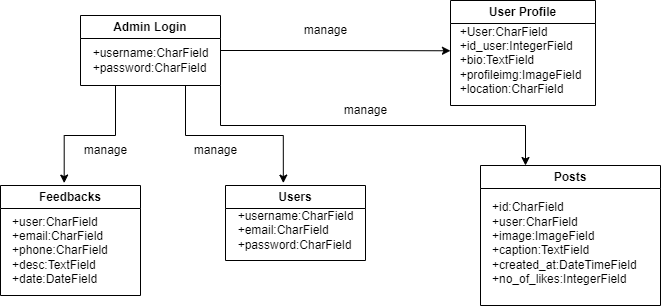
Is A AaaaAA

-Triangle (used to specialization of class)

-Triangle (used to generalization of class)

Is A

**Class Diagram**



**Entity Relationship Diagram**

**ERD –**

The traditional approach places a great deal of emphasis on data storage requirements for the new system. The model used to define the data storage requirements is called the entity-relationship diagram (ERD).

-Rectangles represent the data entities.

-Lines connecting the rectangles show relationships among the data entities.

**Symbols used to draw ERD-**

-Entity

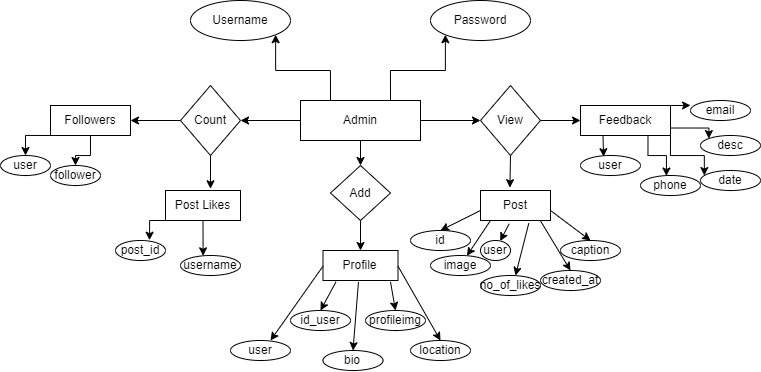
-Relationships

-Connection line

-Weak Entity

-Generalization

**Entity Relationship Diagram**



**Activity Diagram**

An activity diagram is a variant of state chart diagram organized according to actions, and mainly targeted towards representing the internal behavior of a method or a use case. An activity is represented by a rounded rectangle.

**How to develop activity diagram:**

1] Identify swim length.

2] Identify input message.

3] Describe message from external actor to system using message notation.

4] Identify and add any special condition on the input messages including iteration and true or false condition.

5] Identify and add the output return messages.

**Symbols used to draw Activity diagram -**

-Start

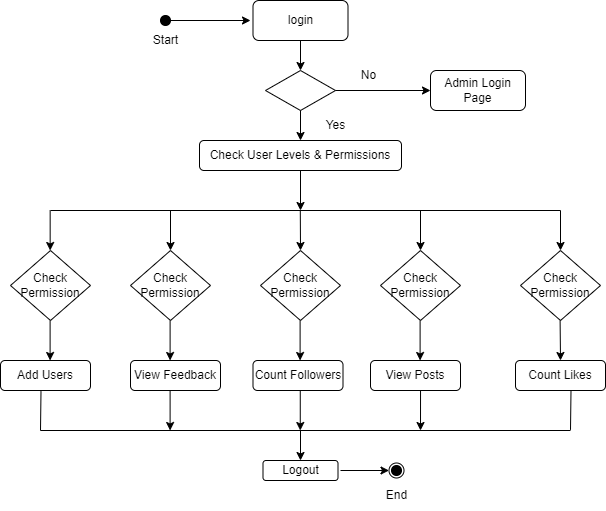
-Activity

-Connection line

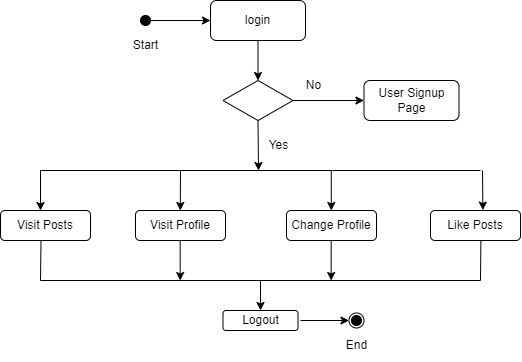
- End

**Activity Diagram**

For admin-



For User-



**Object Diagram**

Object diagrams, or instance diagram, illustrate objects and links. As in the case of class diagrams, object diagrams represent the static structure .The notation used for object diagrams is derived from that of class diagrams; elements that are instances are underlined.

Object diagrams are primarily used to show a context –before or after an interaction, for example. However, they are also used to aid the understanding of complex data structures, such as recursive structure.

**Symbols used to draw Object diagram -**

-Rectangular box

**: Object**

(To represent object)

-Connection line

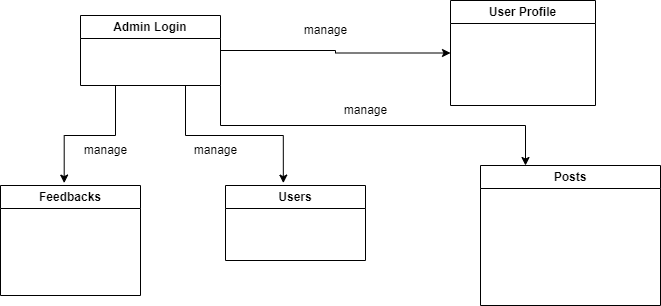
Is a

-Triangle (for specialization)

Is a

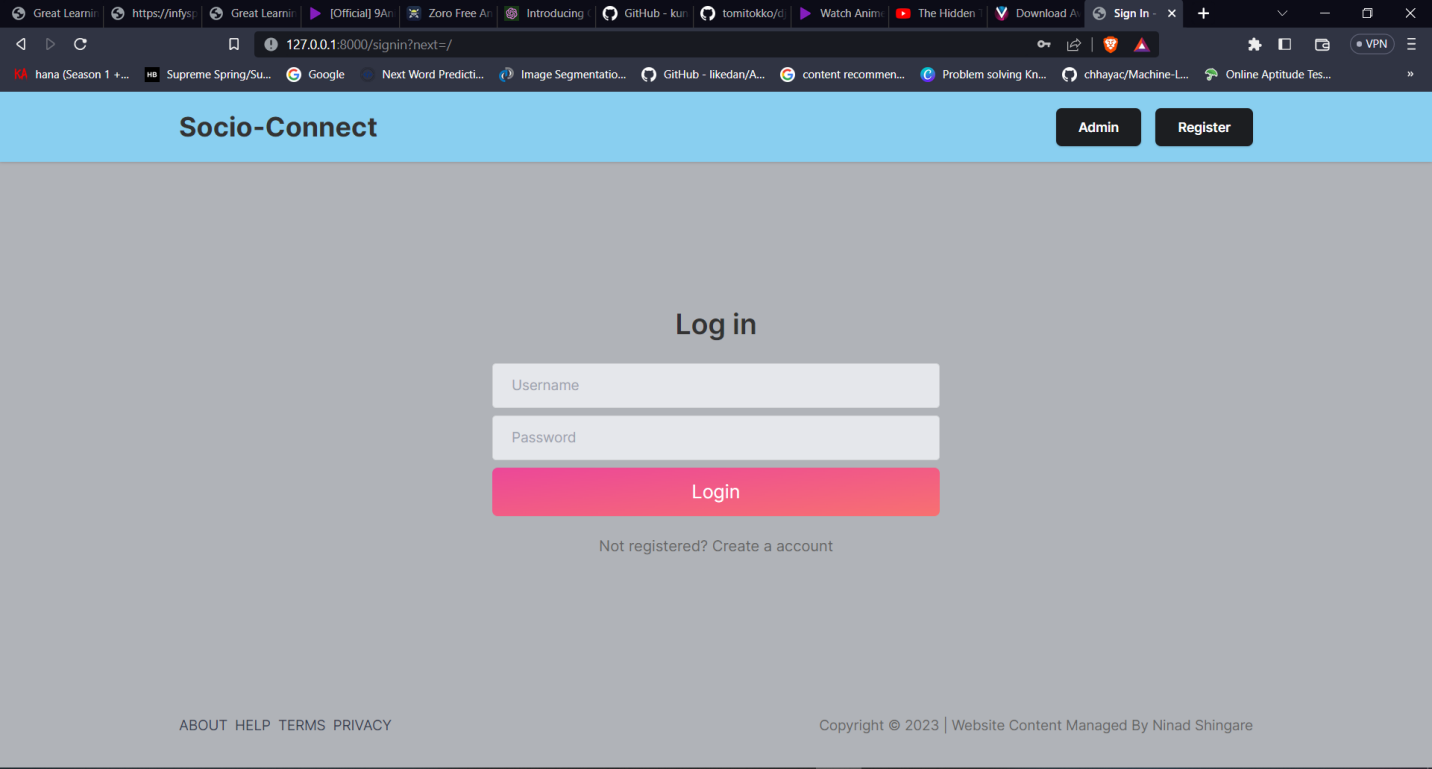
-Triangle (for generalization)

**Object Diagram**



**Input And Output Screen Design**

**Login Page**

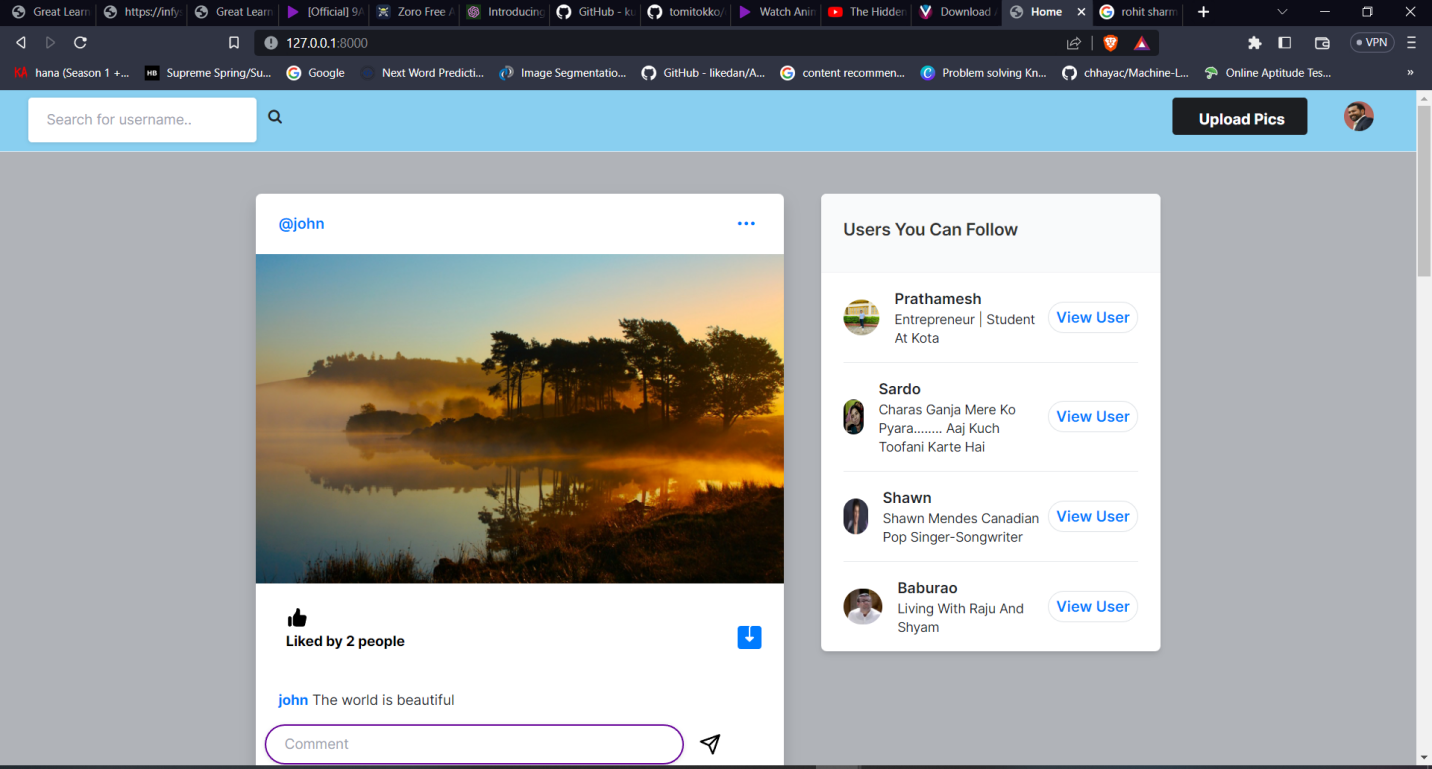


**Signup Page**

Graphical user interface

Description automatically generated

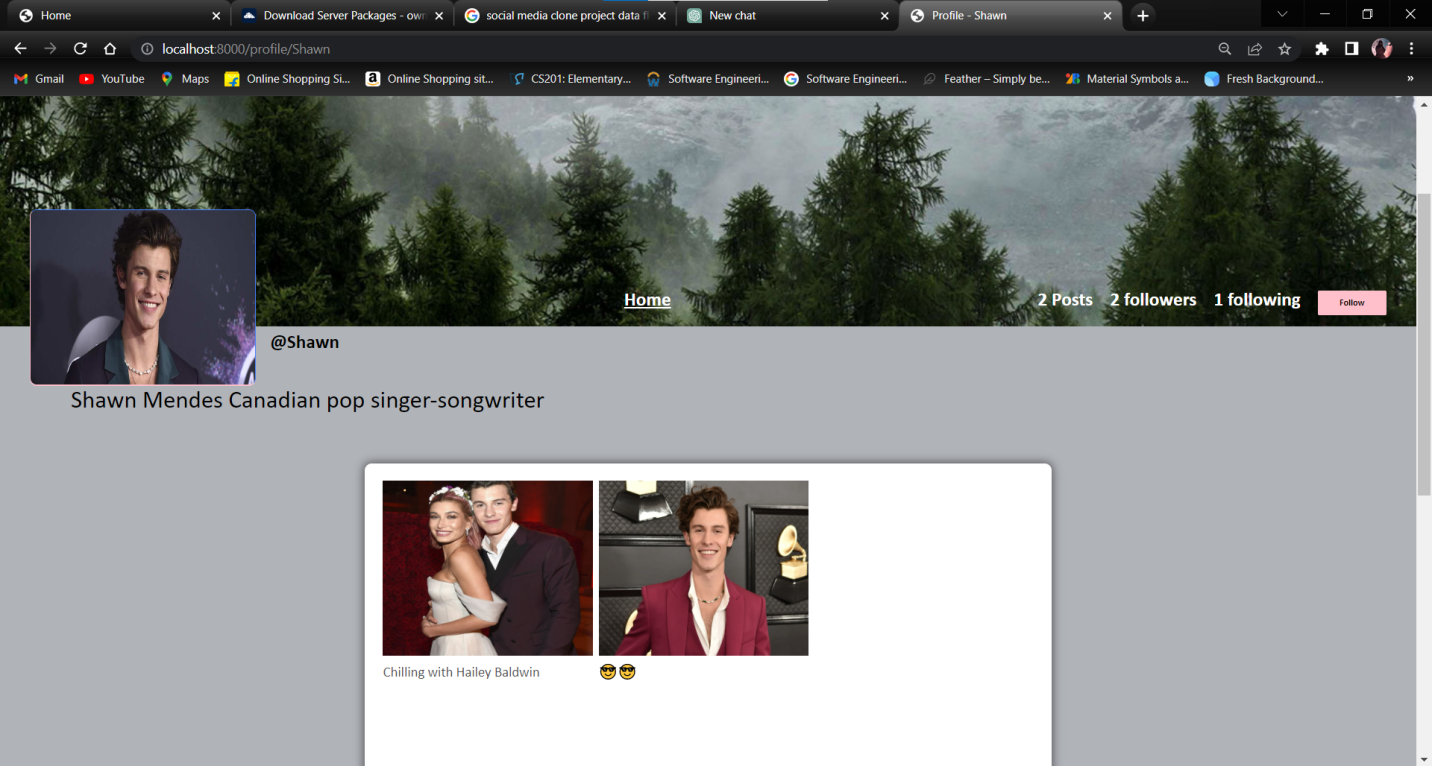
**Home Page**



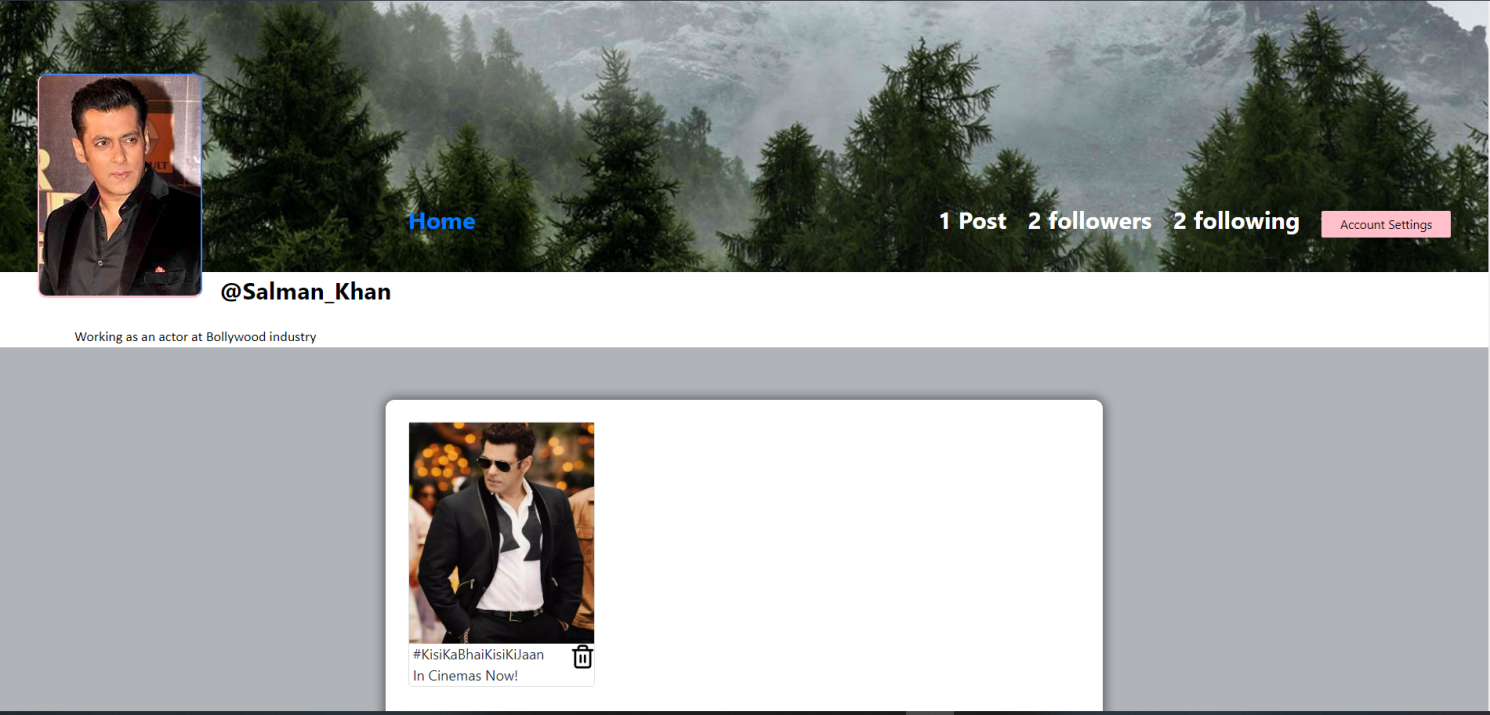
A screenshot of a computer

Description automatically generated with medium confidence

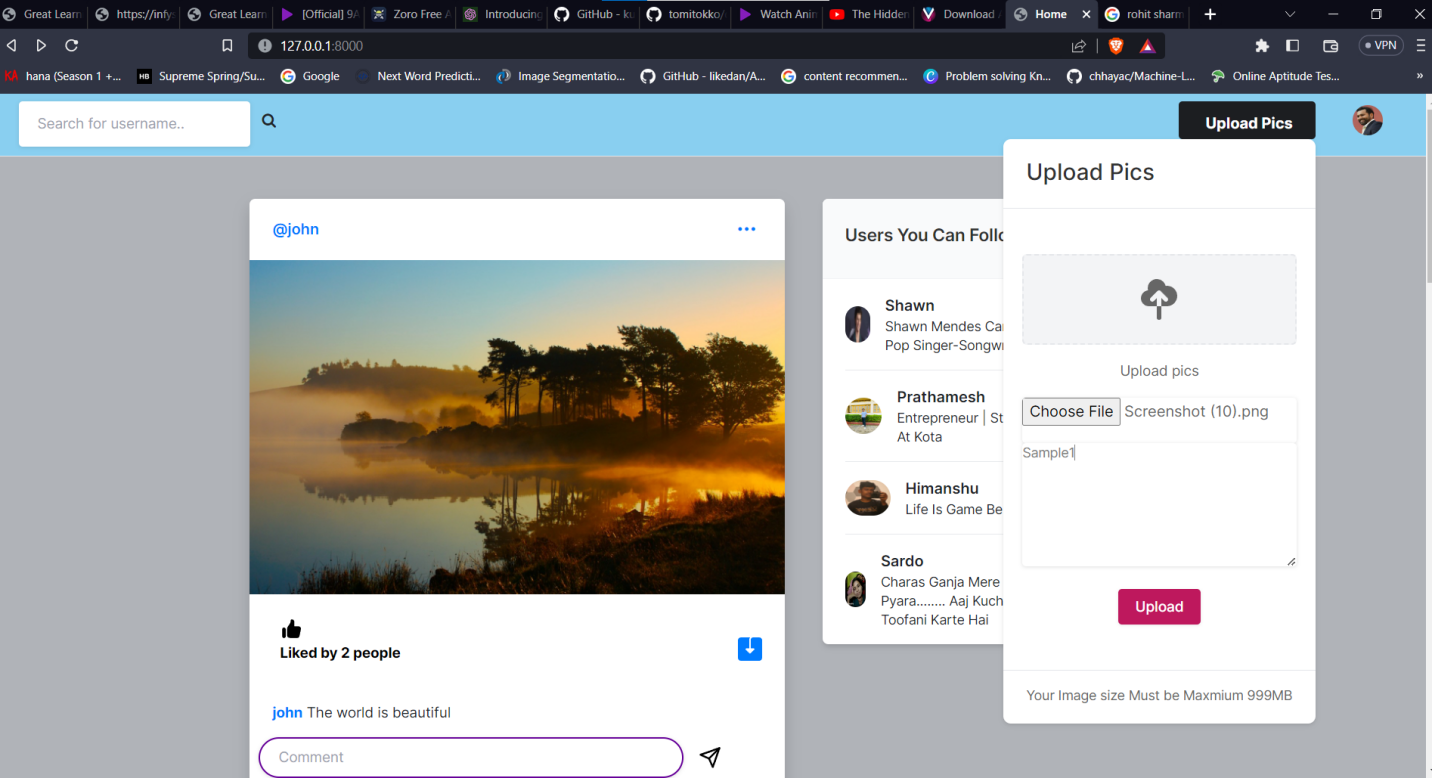
**User Profile Page**



**User’s own Profile Page**



**Home Page**



**Feedback Page**

A screenshot of a computer

Description automatically generated

**Admins Dashboard**

A screenshot of a computer

Description automatically generated with medium confidence

**Admins Users Page**

Graphical user interface, text, application, email

Description automatically generated

**Admins User List**

A screenshot of a computer

Description automatically generated

**Admins Feedbacks Page**

Graphical user interface, application

Description automatically generated

**Process Involved**

**Functionality of project**

First type ***localhost:8000/*** on your browser. Then you get **login page**. The users which are not registered can register by clicking on **Do not have an account**. And enter the details to register. There are two types of user in this project.

Admin first have to login then redirected to **Admin Dashboard** there are many options that admin can handle easily such as **Users**, **Groups, Posts, Followers, Feedbacks*.***

In **User** section admin can **Add, Edit, Delete** user by entering the details easily. In **Groups**section admin can Add or Remove Users and assign them specific permissions***.*** In **Followers** section admin can Add, View, Delete Followers of users feedbacks.

Users first have to create account by entering details then after login they get redirected to **user**Homepage there they are suggested users whom they can follow***.*** When user follows another user he/she gets to see their posts on their homepage on which they can react by liking or commenting them .There are many Options that student can access easily such as **User Profile, Add Post, Feedback*.***

In **Add Post** section user can add post to be seen by other users and for them to like or comment them. User needs to upload image and give it an caption to make an successful post.

**Methodology Used For Testing**

**Manual Testing**

**Manual Testing** is a type of software testing in which test cases are executed manually by a tester without using any automated tools. The purpose of Manual Testing is to identify the bugs, issues, and defects in the software application. Manual software testing is the most primitive technique of all testing types and it helps to find critical bugs in the software application.

Any new application must be manually tested before its testing can be automated. Manual Software Testing requires more effort but is necessary to check automation feasibility. Manual testing concepts does not require knowledge of any testing tool. One of the Software Testing Fundamental is "**100% Automation is not possible**". This makes Manual Testing imperative.

* Each module has been tested separately (Unit testing).
* Identifying and categorizing data entity types.
* Testing exceptions for registration of new user using different data types.
* Checking of input data validation testing.
* Comparison of actual and expected results based on the executed test cases Keeping track of newly encountered bugs and their fixes
* Ensures that the software system meets the functionality.
* Testing to identifying any potential bug or performance issues.
* Testing for spelling mistakes and broken links.
* Tested on machines with the lowest specification to test loading times and any latency problems.
* Testing the Speed (i.e. Response Time, data rendering and accessing), Capacity, Stability and Scalability.
* Testing compatibility across multiple browsers.

**Code Of The Project**

**index.html**

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="{% static 'favicon.png' %}" rel="icon" type="image/png">

<title>Home</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="{% static 'assets/css/icons.css' %}">

<link rel="stylesheet" href="{% static 'assets/css/uikit.css' %}">

<link rel="stylesheet" href="{% static 'assets/css/style.css' %}">

<link rel="stylesheet" href="{% static 'assets/css/tailwind.css' %}">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css" integrity="sha384-xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N" crossorigin="anonymous">

<script src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-Fy6S3B9q64WdZWQUiU+q4/2Lc9npb8tCaSX9FK7E8HnRr0Jz8D6OP9dO5Vg3Q9ct" crossorigin="anonymous"></script>

</head>

<body style="background-color:#B9D9EB">

<header style="background-color:#89CFF0">

<div class="header\_inner">

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

{% csrf\_token %}

<div class="left-side">

<!-- Logo -->

<div id="logo" class=" uk-hidden@s">

<a href="home.html">

<b><h1 style="text-transform: uppercase;">Social Book</h1></b>

</a>

</div>

<!--<div class="triger" uk-toggle="target: #wrapper ; cls: sidebar-active">

<i class="uil-bars"></i>

</div>-->

<!-- <div class="header\_search"> -->

<input class="mt-2"type="text" name="username" placeholder="Search for username..">&nbsp; &nbsp;

<button type="submit"><i class="fa fa-search fa-1x"></i></button>

<!-- </div> -->

</div>

</form>

<div class="right-side lg:pr-4">

<!-- upload -->

<a href="#"

class="bg-pink-500 flex font-bold hidden hover:bg-pink-600 hover:text-white inline-block items-center lg:block max-h-10 mr-4 px-4 py-2 rounded shado text-white">

<ion-icon name="add-circle" class="-mb-1

mr-1 opacity-90 text-xl uilus-circle"></ion-icon> Upload Pics

</a>

<!-- upload dropdown box -->

<div uk-dropdown="pos: top-right;mode:click ; animation: uk-animation-slide-bottom-small" class="header\_dropdown">

<!-- notivication header -->

<div class="px-4 py-3 -mx-5 -mt-4 mb-5 border-b">

<h4>Upload Pics</h4>

</div>

<!-- notification contents -->

<div class="flex justify-center flex-center text-center">

<div class="flex flex-col choose-upload text-center">

<div class="bg-gray-100 border-2 border-dashed flex flex-col h-24 items-center justify-center relative w-full rounded-lg ">

<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 20 20" fill="currentColor" class="w-12">

<path d="M5.5 13a3.5 3.5 0 01-.369-6.98 4 4 0 117.753-1.977A4.5 4.5 0 1113.5 13H11V9.413l1.293 1.293a1 1 0 001.414-1.414l-3-3a1 1 0 00-1.414 0l-3 3a1 1 0 001.414 1.414L9 9.414V13H5.5z" />

<path d="M9 13h2v5a1 1 0 11-2 0v-5z" />

</svg>

</div>

<form action="upload" method="POST" enctype="multipart/form-data" >

{% csrf\_token %}

<p class="my-3 leading-6"> Upload pics

</p>

<div>

<input type="file" name="image\_upload">

<!-- <a href="#" class="button soft-warning small"> Choose file</a> -->

<textarea class="button soft-warning small" name="caption" placeholder="caption..." ></textarea>

<br>

<button class="button bg-blue-700" type="submit" >Upload</button>

</div>

</form>

<!-- <a href="#" class="uk-text-muted mt-3 uk-link"

uk-toggle="target: .choose-upload ; animation: uk-animation-slide-right-small, uk-animation-slide-left-medium; queued: true">

Or Import Video </a> -->

</div>

<div class="uk-flex uk-flex-column choose-upload" hidden>

<div class="mx-auto flex flex-col h-24 items-center justify-center relative w-full rounded-lg">

**signin.html**

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="{% static 'assets/images/favicon.png' %}" rel="icon" type="image/png">

<title>Sign In - Socio Connect</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="{% static 'assets/css/icons.css' %}">

<link rel="stylesheet" href="{% static 'assets/css/uikit.css' %}">

<link rel="stylesheet" href="{% static 'assets/css/style.css' %}">

<link rel="stylesheet" href="{% static 'assets/css/tailwind.css' %}">

</head>

<body class="bg-gray-100">

<div id="wrapper" class="flex flex-col justify-between h-screen">

<!-- header-->

<div class="bg-white py-4 shadow dark:bg-gray-800" style="background-color:#89CFF0">

<div class="max-w-6xl mx-auto">

<div class="flex items-center lg:justify-between justify-around">

<a href="trending.html">

<b><h1 style="font-size: 1.8rem;">Socio-Connect</h1></b>

</a>

<div class="capitalize flex font-semibold hidden lg:block my-2 space-x-3 text-center text-sm">

<a href="/admin" class="bg-pink-500 pink-500 px-6 py-3 rounded-md shadow text-white" >Admin</a>

<a href="/signup" class="bg-pink-500 pink-500 px-6 py-3 rounded-md shadow text-white" >Register</a>

</div>

</div>

</div>

</div>

<!-- Content-->

<div>

<div class="lg:p-12 max-w-md max-w-xl lg:my-0 my-12 mx-auto p-6 space-y-">

<h1 class="lg:text-3xl text-xl font-semibold mb-6" align="center"> Log in</h1>

<div>

<style>

h5{

color: red;

}

</style>

{% for message in messages %}

<h5>{{message}}</h5>

{% endfor %}

</div>

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

{% csrf\_token %}

<input type="text" name="username" placeholder="Username" class="bg-gray-200 mb-2 shadow-none dark:bg-gray-800" style="border: 1px solid #d3d5d8 !important;">

<input type="password" name="password" placeholder="Password" class="bg-gray-200 mb-2 shadow-none dark:bg-gray-800" style="border: 1px solid #d3d5d8 !important;">

<!-- <div class="flex justify-between my-4">

<div class="checkbox">

<input type="checkbox" id="chekcbox1" checked>

<label for="chekcbox1"><span class="checkbox-icon"></span>Remember Me</label>

</div>

<a href="#"> Forgot Your Password? </a>

</div> -->

<button type="submit" class="bg-gradient-to-br from-pink-500 py-3 rounded-md text-white text-xl to-red-400 w-full">Login</button>

<div class="text-center mt-5 space-x-2">

<p class="text-base"> Not registered? <a href="/signup" class=""> Create a account </a></p>

</div>

</form>

</div>

</div>

<!-- Footer -->

<div class="lg:mb-5 py-3 uk-link-reset">

<div class="flex flex-col items-center justify-between lg:flex-row max-w-6xl mx-auto lg:space-y-0 space-y-3">

<div class="flex space-x-2 text-gray-700 uppercase">

<a href="#"> About</a>

<a href="#"> Help</a>

<a href="#"> Terms</a>

<a href="#"> Privacy</a>

</div>

<p class="capitalize">Copyright © 2023 | Website Content Managed By Ninad Shingare </p>

**signup.html**

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="{% static 'assets/images/favicon.png' %}" rel="icon" type="image/png">

<title>Sign Up - Socio Connect</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="{% static 'assets/css/icons.css' %}">

<link rel="stylesheet" href="{% static 'assets/css/uikit.css' %}">

<link rel="stylesheet" href="{% static 'assets/css/style.css' %}">

<link rel="stylesheet" href="{% static 'assets/css/tailwind.css' %}">

</head>

<body class="bg-gray-100">

<div id="wrapper" class="flex flex-col justify-between h-screen">

<!-- header-->

<div class="bg-white py-4 shadow dark:bg-gray-800" style="background-color:#89CFF0">

<div class="max-w-6xl mx-auto">

<div class="flex items-center lg:justify-between justify-around">

<a href="trending.html">

<b><h1 style="font-size: 1.8rem;">Socio-Connect</h1></b>

</a>

<div class="capitalize flex font-semibold hidden lg:block my-2 space-x-3 text-center text-sm">

<a href="/signin" class="bg-pink-500 pink-500 px-6 py-3 rounded-md shadow text-white" >Login</a>

</div>

</div>

</div>

</div>

<!-- Content-->

<div>

<div class="lg:p-12 max-w-md max-w-xl lg:my-0 my-12 mx-auto p-6 space-y-">

<h1 class="lg:text-3xl text-xl font-semibold mb-6" align="center"> Register</h1>

<!--<p class="mb-2 text-black text-lg"> Register to manage your account </p>-->

<div>

<style>

h5{

color: red;

}

</style>

{% for message in messages %}

<h5>{{message}}</h5>

{% endfor %}

</div>

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

{% csrf\_token %}

<!--<div class="flex lg:flex-row flex-col lg:space-x-2">

<input type="text" placeholder="First Name" class="bg-gray-200 mb-2 shadow-none dark:bg-gray-800" style="border: 1px solid #d3d5d8 !important;">

<input type="text" placeholder="Last Name" class="bg-gray-200 mb-2 shadow-none dark:bg-gray-800" style="border: 1px solid #d3d5d8 !important;">

</div>-->

<input type="text" name="username" placeholder="Username" class="bg-gray-200 mb-2 shadow-none dark:bg-gray-800" style="border: 1px solid #d3d5d8 !important;">

<input type="email" name="email" placeholder="Email" class="bg-gray-200 mb-2 shadow-none dark:bg-gray-800" style="border: 1px solid #d3d5d8 !important;">

<input type="password" name="password" placeholder="Password" class="bg-gray-200 mb-2 shadow-none dark:bg-gray-800" style="border: 1px solid #d3d5d8 !important;">

<input type="password" name="password2" placeholder="Confirm Password" class="bg-gray-200 mb-2 shadow-none dark:bg-gray-800" style="border: 1px solid #d3d5d8 !important;">

<!-- <div class="flex justify-start my-4 space-x-1">

<div class="checkbox">

<input type="checkbox" id="chekcbox1" checked>

<label for="chekcbox1"><span class="checkbox-icon"></span> I Agree</label>

</div>

<a href="#"> Terms and Conditions</a>

</div> -->

<button type="submit" class="bg-gradient-to-br from-pink-500 py-3 rounded-md text-white text-xl to-red-400 w-full">Sign Up</button>

<div class="text-center mt-5 space-x-2">

<p class="text-base"> Alreadyvs have an account? <a href="/signin"> Login </a></p>

</div>

</form>

</div>

</div>

<!-- Footer -->

<div class="lg:mb-5 py-3 uk-link-reset">

<div class="flex flex-col items-center justify-between lg:flex-row max-w-6xl mx-auto lg:space-y-0 space-y-3">

<div class="flex space-x-2 text-gray-700 uppercase">

<a href="#"> About</a>

<a href="#"> Help</a>

<a href="#"> Terms</a>

<a href="#"> Privacy</a>

</div>

<p class="capitalize">Copyright © 2023 | Website Content Managed By Ninad Shingare </p>

</div>

</div>

</div>

**profile.html**

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta name="description" content="" />

<meta name="keywords" content="" />

<title>Profile - {{user\_profile.user.username}}</title>

<!-- <link rel="icon" href="images/fav.png" type="image/png" sizes="16x16"> -->

<link rel="stylesheet" href="{% static 'css/main.min.css' %}">

<link rel="stylesheet" href="{% static 'css/style.css' %}">

<link rel="stylesheet" href="{% static 'css/color.css' %}">

<link rel="stylesheet" href="{% static 'css/responsive.css' %}">

</head>

<body>

<!--<div class="se-pre-con"></div>-->

<div class="theme-layout"></div>

</div><section>

<div class="feature-photo"><figure><img src="{% static 'images/resources/timeline-1.jpg' %}" alt=""></figure>

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

{% csrf\_token %}

<div class="add-btn">

<span style="color: white; font-size: 27px; margin-right: 520px;margin-left: 50px;"><b><u><a href="/">Home</a></u></b></span>

{% if user\_post\_length == 0 %}

<span style="color: white; font-size: 27px;"><b>No Post</b></span>

{% elif user\_post\_length == 1 %}

<span style="color: white; font-size: 27px;"><b>{{user\_post\_length}} Post</b></span>

{% else %}

<span style="color: white; font-size: 27px;"><b>{{user\_post\_length}} Posts</b></span>

{% endif %}

{% if user\_followers == 0 or user\_followers == 1 %}

<span style="color: white; font-size: 27px;"><b>{{user\_followers}} follower</b></span>

{% else %}

<span style="color: white; font-size: 27px;"><b>{{user\_followers}} followers</b></span>

{% endif %}

<span style="color: white; font-size: 27px;"><b>{{user\_following}} following</b></span>

<input type="hidden" value="{{user.username}}" name="follower" />

<input type="hidden" value="{{user\_object.username}}" name="user" />

{% if user\_object.username == user.username %}

<a href="/settings" data-ripple="">Account Settings</a>

{% else %}

<a data-ripple=""><button type="submit" style="background-color: #ffc0cb; border: #ffc0cb;">{{button\_text}}</button></a>

{% endif %}

</div>

</form>

**settings.html**

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="{% static 'assets/images/favicon.png' %}" rel="icon" type="image/png">

<title>Settings</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="{% static 'assets/css/icons.css' %}">

<link rel="stylesheet" href="{% static 'assets/css/uikit.css' %}">

<link rel="stylesheet" href="{% static 'assets/css/style.css' %}">

<link rel="stylesheet" href="{% static 'assets/css/tailwind.css' %}">

</head>

<body>

<div class="container m-auto">

<h1 class="text-2xl leading-none text-gray-900 tracking-tight mt-3"><a href="/">Home</a> / Account Setting for <b>{{user.username}}</b> </h1>

<br>

<hr>

<div class="grid lg:grid-cols-3 mt-12 gap-8">

<div>

<h3 class="text-xl mb-2"> General</h3>

<p></p>

</div>

<div class="bg-white rounded-md lg:shadow-lg shadow col-span-2">

<form action="" method="POST" enctype="multipart/form-data">

{% csrf\_token %}

<div class="grid grid-cols-2 gap-3 lg:p-6 p-4">

<div class="col-span-2">

<label for=""> Profile Image</label>

<img width="100" height="100" src="{{user\_profile.profileimg.url}}" />

<input type="file" name="image" value="" class="shadow-none bg-gray-100">

</div>

<div class="col-span-2">

<label for="about">Bio</label>

<textarea id="about" name="bio" rows="3" class="shadow-none bg-gray-100">{{user\_profile.bio}}</textarea>

</div>

<div class="col-span-2">

<label for=""> Location</label>

<input type="text" name="location" value="{{user\_profile.location}}" class="shadow-none bg-gray-100">

</div>

</div>

<div class="bg-gray-10 p-6 pt-0 flex justify-end space-x-3">

<button class="p-2 px-4 rounded bg-gray-50 text-red-500"> <a href="/">Cancel</a> </button>

<button type="submit" class="button bg-blue-700"> Save </button>

</div>

</form>

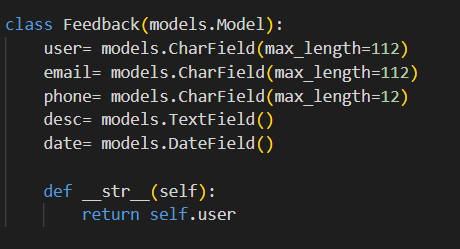
</div>

**Chapter IV**

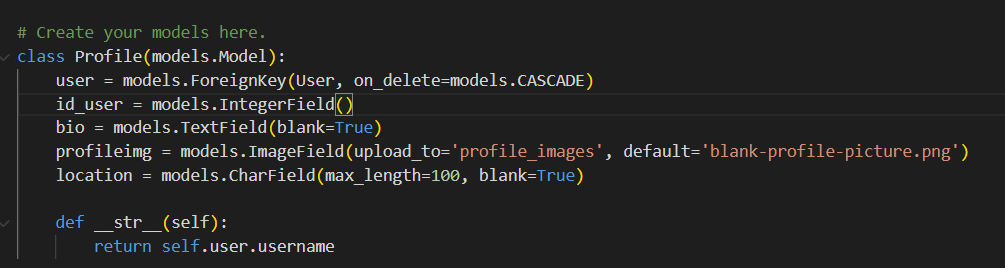
**Annexure**

**List of Database Models**

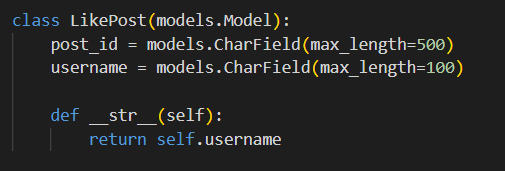
**Feedbacks**



**Profiles**



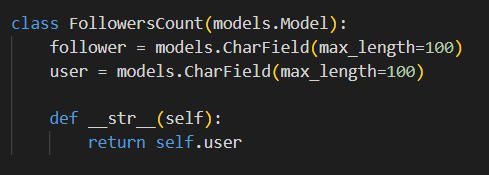
**Like Posts**



**Posts**



**FollowersCount**



**Chapter V**

**Conclusion**

As the goal was set most of the implementations and requirements have be passes and test cases been solved. Some problems for developer like error correction did become a hassle as we are new to this subject, but at the end a fruitful website was created together.

One of the main reasons to create a social media clone is to offer an alternative platform to users who are dissatisfied with existing social media platforms or are looking for something new. Building a social media clone requires careful planning, attention to detail, and a talented development team

Overall, creating a social media clone is a challenging task, but it can be rewarding if done well. It's important to consider the market competition, the features you want to offer, and how you will attract and retain users. Additionally, it's crucial to prioritize user privacy and security as these aspects have become increasingly important in today's digital age.

Social media clones can also be monetized through various means, such as advertising, subscription fees, or through the sale of user data.

**Chapter VI**

**Future Enhancement**

* Commenting feature to add comments to be seen by others.
* Share post feature to be able to directly share a particular post.
* Show notifications for user likes and follows.
* User be able to keep his/her account private or public as per wish.
* Integrating our application with a real time chat application.

**Chapter VII**

**References and Bibliography**

Google - <https://www.google.com/>

YouTube - <https://www.youtube.com/>

Stack Overflow- <https://stackoverflow.com/>

Django documentation- <https://www.djangoproject.com/>

Python documentation- <https://docs.python.org/3/>