

Project On
Socio - Network for Developers

By

Sahil Baraskar (2020510005)
Shaikh Samee (2020510058)

Under the guidance of
Internal Supervisor

Prof.Nawal Dandekar



Department of Master Of Computer Application
Sardar Patel Institute of Technology
Autonomous Institute Affiliated to Mumbai University
2021-22

CERTIFICATE OF APPROVAL

This is to certify that the following students

Sahil Baraskar (2020510005)
Shaikh Samee (2020510058)

Have satisfactorily carried out work on the project
entitled

“Socio - Network for Developers”

Towards the fulfilment of project, as laid down
by
Sardar Patel Institute of Technology
during year
2022-23.

Project Guide:
Prof.Nawal Dandekar

PROJECT APPROVAL CERTIFICATE

This is to certify that the following students

Sahil Baraskar (2020510005)
Shaikh Samee (2020510058)

Have successfully completed the Project report on

“Socio - Network for Developers”,

which is found to be satisfactory and is approved

at

SARDAR PATEL INSTITUTE OF TECHNOLOGY,
ANDHERI (W), MUMBAI

INTERNAL EXAMINER

EXTERNAL EXAMINER

HEAD OF DEPARTMENT

PRINCIPAL

Contents

Abstract	i
Objectives	i
List Of Figures	ii
List Of Tables	ii
1 Introduction	1
1.1 Problem Definition	1
1.2 Objectives and Scope	1
1.2.1 Objectives	1
1.2.2 Scope	1
1.3 Existing System	2
1.4 Proposed System	2
1.5 System Requirements	3
2 Software Requirement Specification (SRS) and Design	4
2.1 Purpose	4
2.2 Definition	4
2.3 Overall Description	4
2.3.1 Product Functions	4
2.3.2 User Characteristics	5
2.4 Modules	6
2.4.1 Activity diagram	6
2.4.2 Use-Case	7
2.4.3 ER Diagram	8
2.4.4 Post Feed ER Diagram	9
2.4.5 Gantt Chart	10
2.4.6 Pert Chart	11
2.4.7 Use-Case	12
3 Project Implementation and Testing	14
3.1 Login and Register	14
3.2 Home - Landing View	15
3.3 Profile	16
3.4 Developers	17
3.5 Posts	18
3.6 Dashboard	19
3.7 DashBoard Edit	20
3.8 Add Experience	21
3.9 Add Education	22
3.10 Code 1	23
3.11 Code 2	23
3.12 Code 3	24

4	Test Cases	25
5	Limitations	26
6	Future Enhancements	26
7	User Manual	27
8	Bibliography	28
8.1	Web References	28

Socio - Network for Developers

Abstract

The Socio application is Internet-based and gives users quick electronic communication of content, such as personal information, documents and photos. Users engage via a computer, tablet, or smartphone via web-based software or applications.

The User can post questions in community forums and ask for quick responses from all the registered users.

Objectives

The web based Application "Socio" is used

- To provide users a user friendly platform for the social interaction process for.
- To provide a platform for developers and students, build their online presence.
- To provide a platform to keep personal information updated

List of Figures

2.4.1Activity Diagram	6
2.4.2Use-Case Diagram	7
2.4.3ER Diagram	8
2.4.4ER diagram for Post feed	9
2.4.5Gantt Chart	10
2.4.6Pert Chart	11
2.4.7Use-Case Diagram	12
3.1.1 Login and Register	14
3.2.1 Home View	15
3.3.1Profile	16
3.4.1Developers View	17
3.5.1Posts	18
3.6.1Dashboard	19
3.7.1Dashboard Edit	20
3.8.1Add Experience	21
3.9.1Education Details	22

List of Tables

1.5.3 Software Requirements on Client Side	3
4.2.1 Use Case Table - Register	13
4.2.2 Use Case Table - Login	13
4.2.1 Use Case Table - Add comment	13
6.1 Test Case - Login and Register	25
6.2 Test Case - Others	25

1 Introduction

1.1 Problem Definition

Social media applications are not focused on developers, We have created an application designed coherently for developers.

1.2 Objectives and Scope

1.2.1 Objectives

The web based application "Socio" is used to

- gather all the developers together and share their experience with any problems faced by new developers in the industry.
- create an enhanced new developers focused community.
- build developer profiles on social media for their online presence, focused on their industry skill sets.
- create posts, add comments, view dashboards.

1.2.2 Scope

The users can provide their details in the profile and view other added users too. Their profiles all match the developer community standards.

In the application the user must enter their personal, educational and professional details in the profile section.

Our System is being made for reducing the information loss and enhancing the communication between developers so that they can all be a part of a developer community.

Socio - Network for Developers

1.3 Existing System

All the current social media applications, are not solely focused on developer based communities. Users can make their general profile and will get a generic news feed with posts. To create a community where developers can share their posts and interact with each other is what the current system lacks.

1.4 Proposed System

Any user who wants to add their profiles can register their account. To add details start the registration process. A valid email address is required. Creating a password is necessary with valid input characters.

Making the profile requires you to add educational details, college name and degree. Adding skill sets is also required. All these functionalities curate a proper developer profile suitable for the social media application. Adding information about past industry experience is also required.

Some of the advantages of our system are as follows :

- User Friendly
It provides attractive interface to the user for navigation through a dynamic flow of social media process in the app.
The items of the application are already connected to each other using side navigation bar.
- Create posts, add comments, view added profiles of registered users.
Users can view list posts as created by other users. We can also like and comment on the post for interaction. In the news feed section we can post any kind of query related to our developer profile.

1.5 System Requirements

- Requirements on Client Side
- Software Requirements on Client Side

Table 1.5.3: Software Requirements on Client Side

Operating System	Windows, LINUX, MacOS
Server	Not Required

2 Software Requirement Specification (SRS) and Design

2.1 Purpose

The purpose of our project is to develop an application that can help users to easily access the community feed and to have an updated profile for any community.

2.2 Definition

To build a Social media Application so the users can have access to a developer community.

2.3 Overall Description

2.3.1 Product Functions

The product function includes:

1. Authentication: Users are required to Sign-up and Log-in. Users will get a verification email for successful registration.
2. Profile: This will contain information regarding the users.
3. Feed: This will contain posts created by the registered users.
4. Profiles: These profiles showcase the user's information including education, past experience and skill sets. All the registered users can browse the profiles and connect with any one they desire.
5. Add comments/like: We can add comments to any post and like it too.
6. Create Post: Create a new post for the community feed. This post is visible to all the added users in the community.

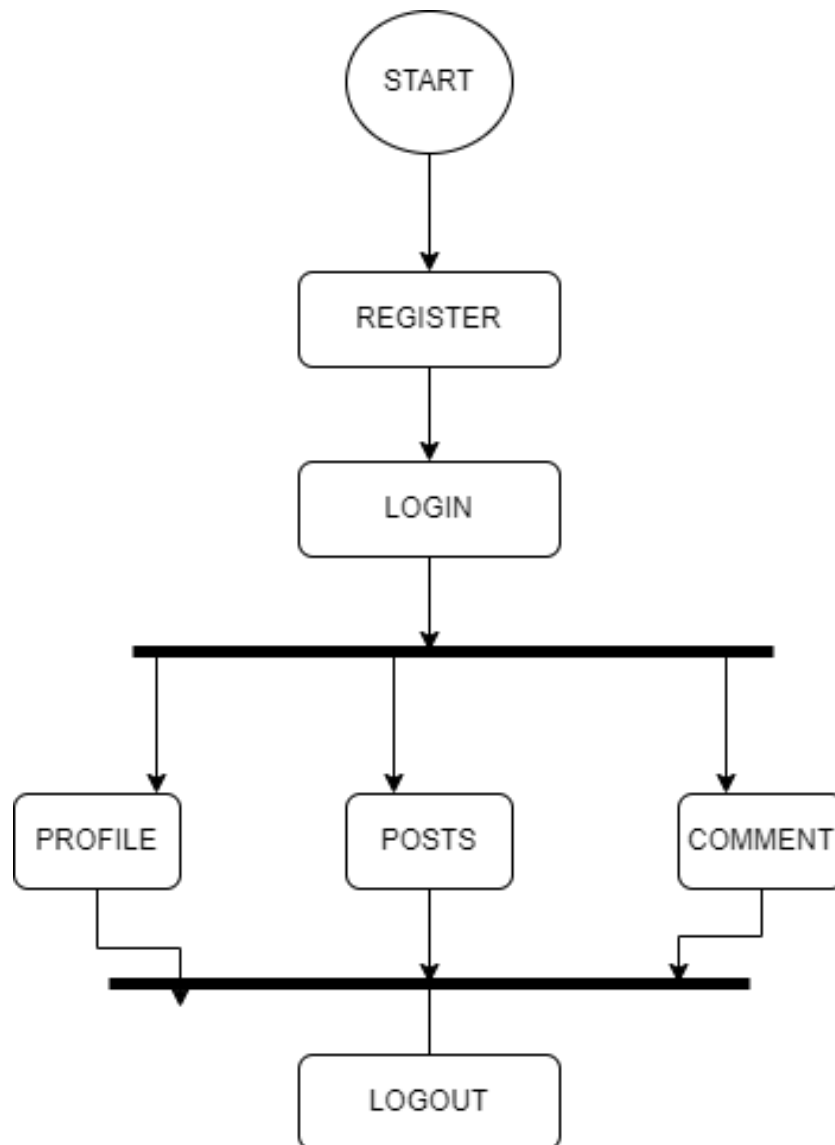
Socio - Network for Developers

2.3.2 User Characteristics

- A registered user, after creating their profile can post any content, like, comment interact with other user's posts. Only the user who has added the comment can delete it. All the users can view any other user's profile.

2.4 Modules

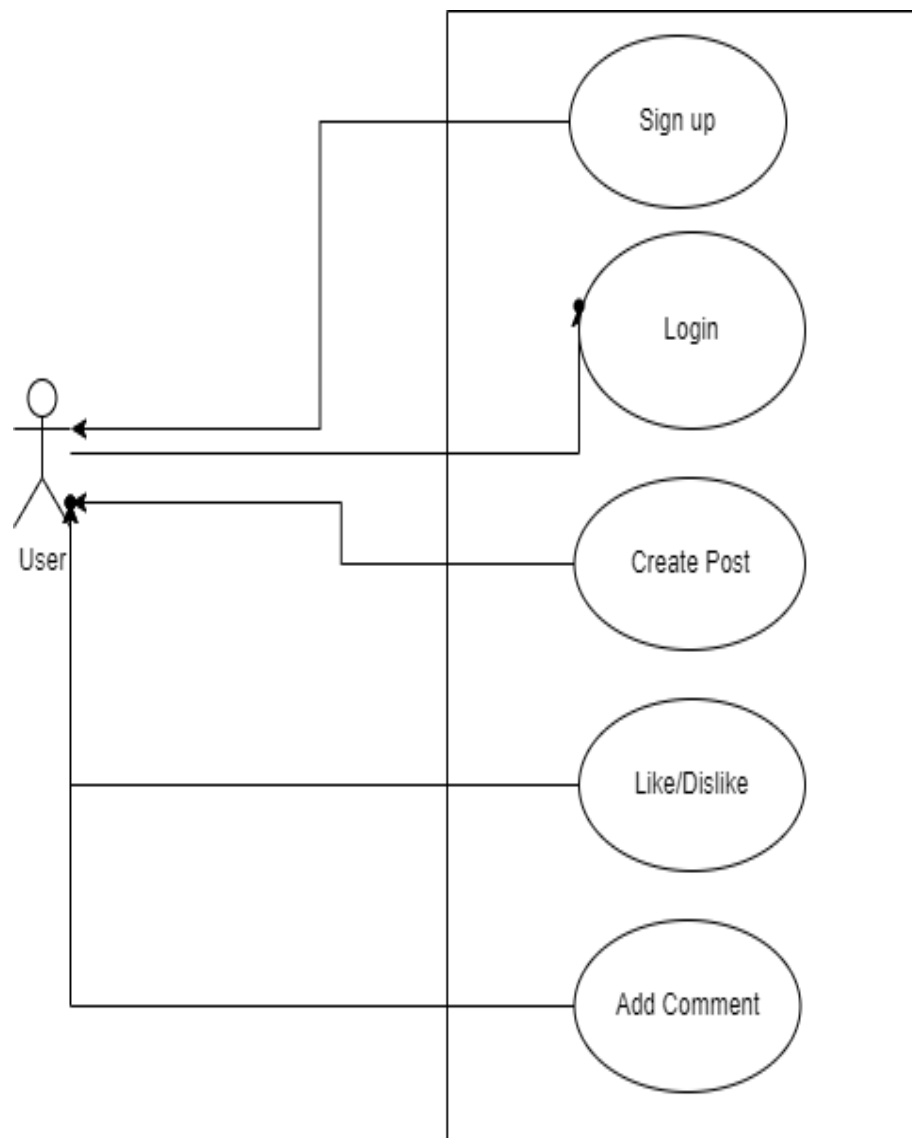
2.4.1 Activity diagram



2.4.1: Activity Diagram

Socio - Network for Developers

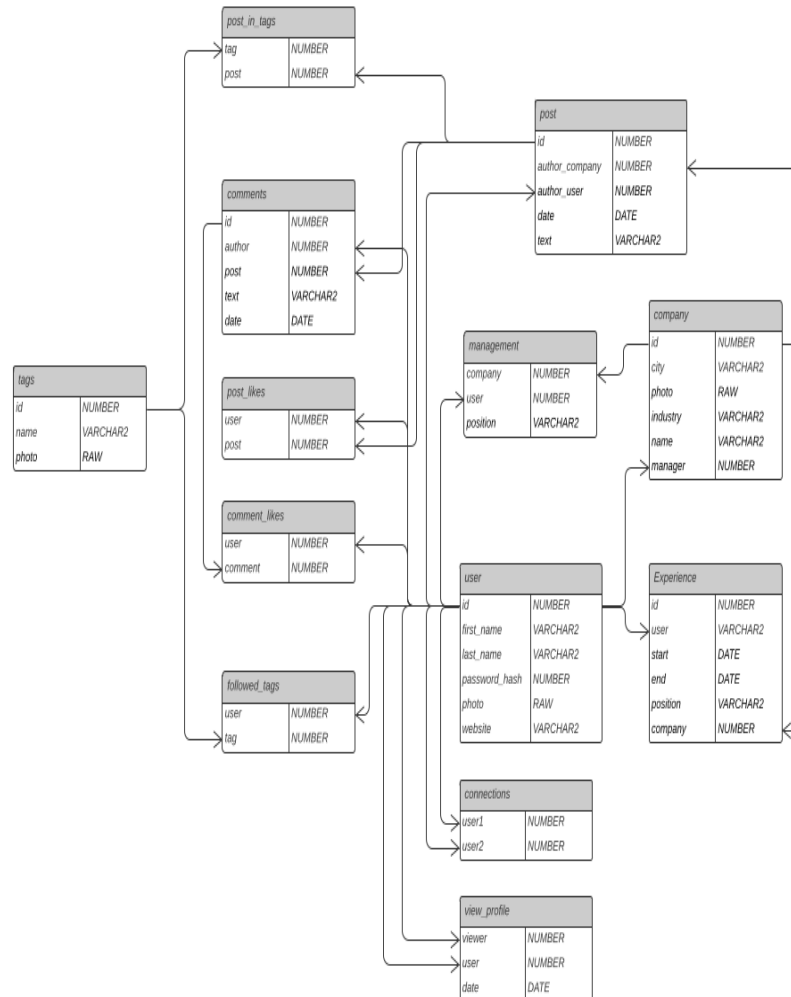
2.4.2 Use-Case



2.4.2: Use-Case Diagram

Socio - Network for Developers

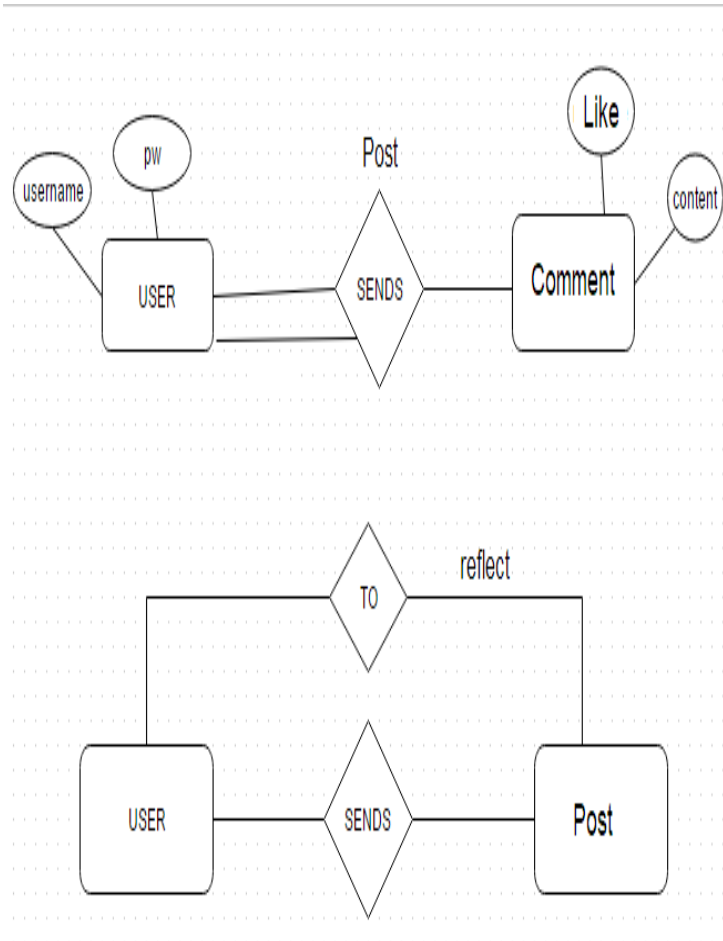
2.4.3 ER Diagram



2.4.3: ER Diagram

Socio - Network for Developers

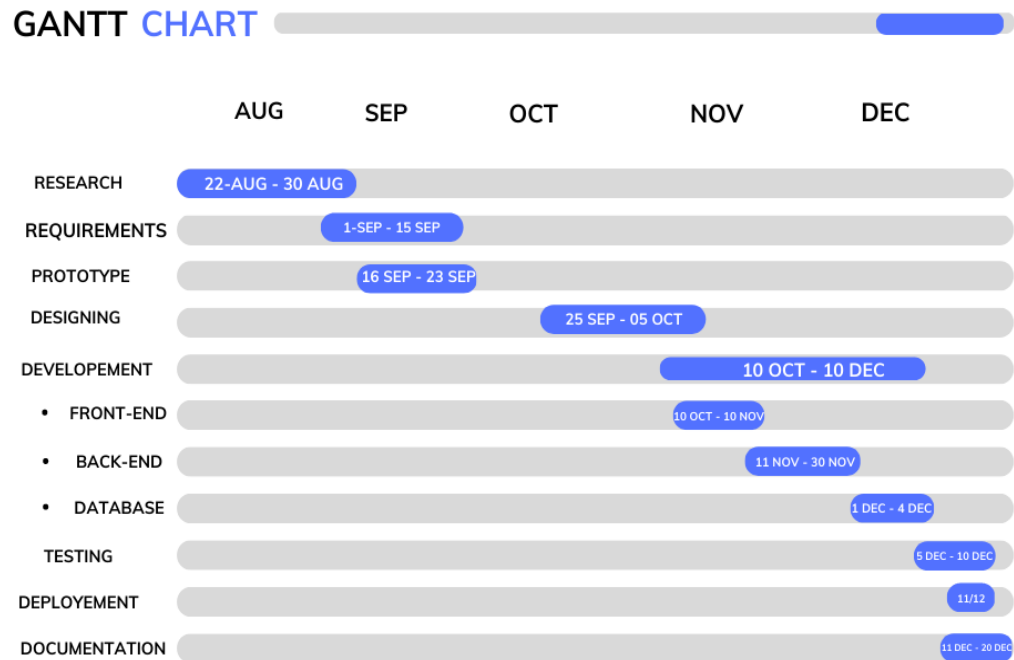
2.4.4 Post Feed ER Diagram



2.4.4: ER diagram for Post feed

2.4.5 Gantt Chart

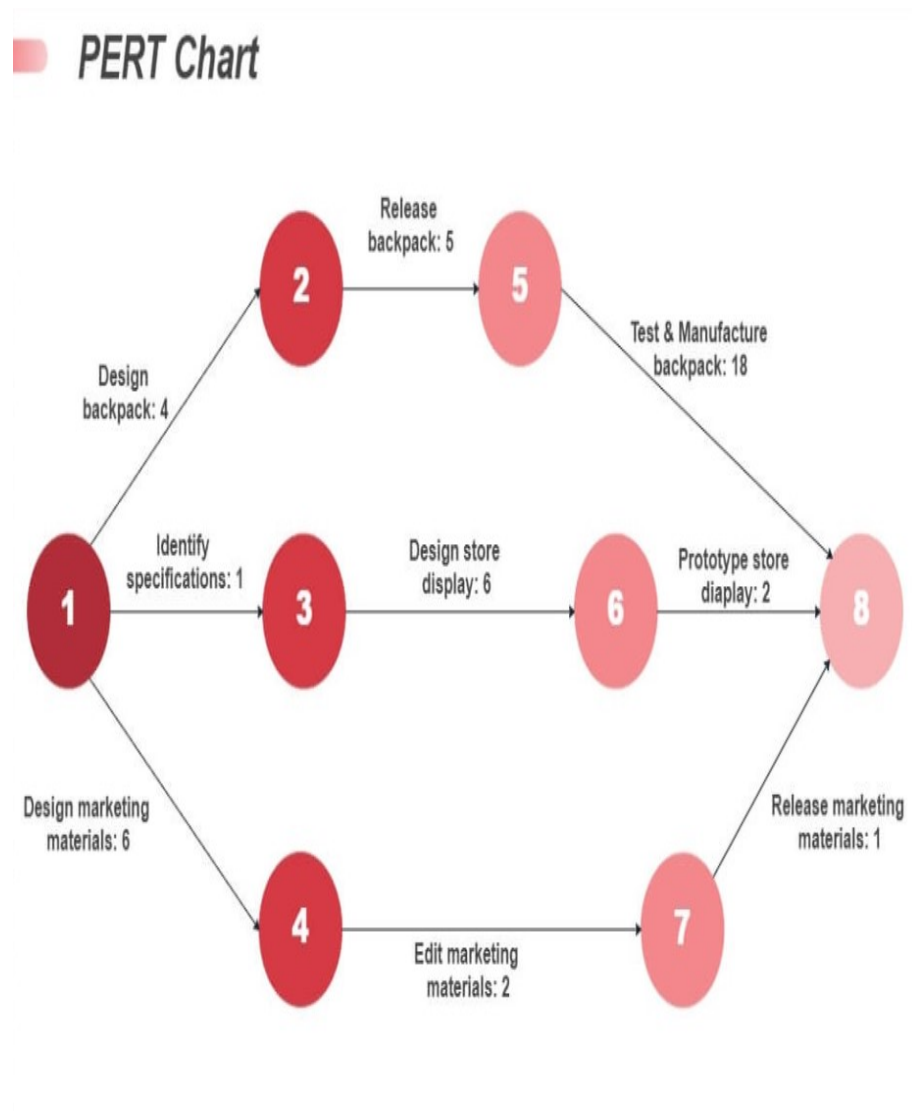
Socio - Netwoking Site For Developer



2.4.5: Gantt Chart

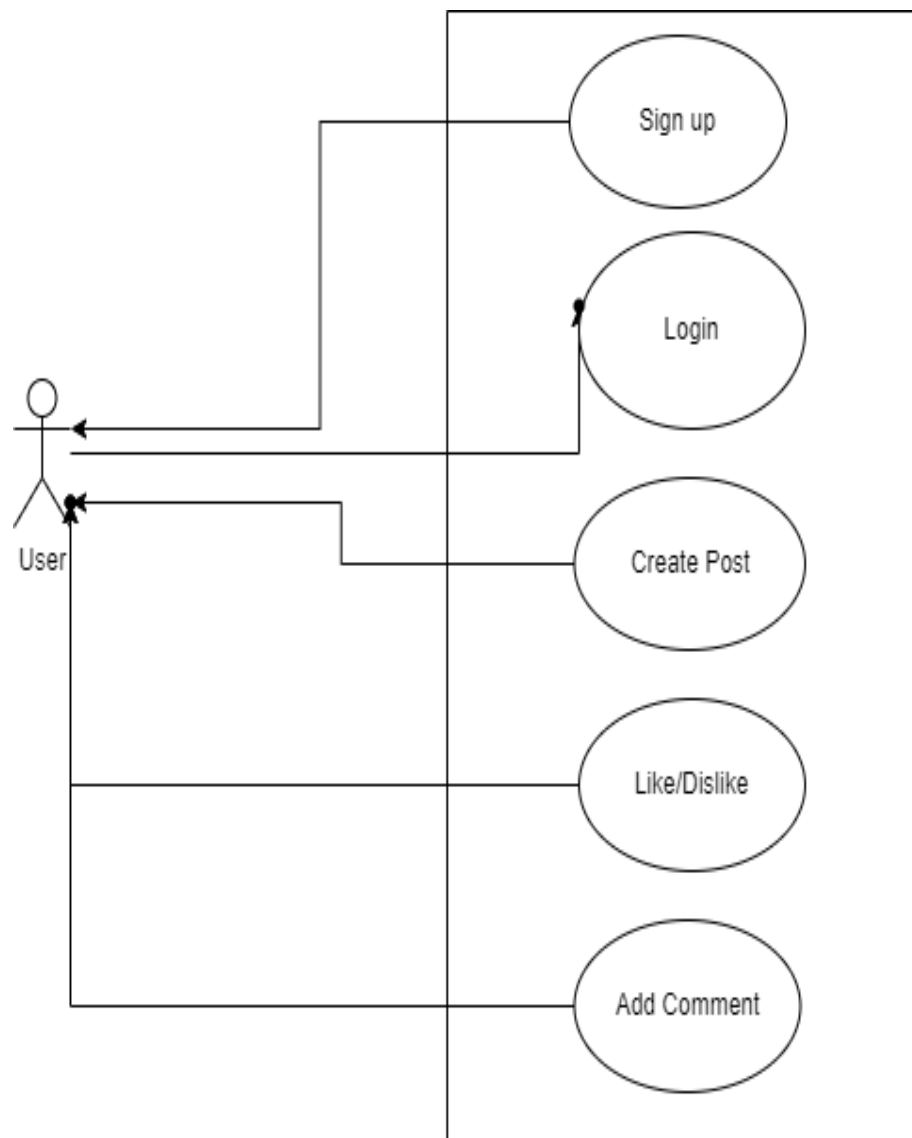
Socio - Network for Developers

2.4.6 Pert Chart



2.4.6: Pert Chart

2.4.7 Use-Case



2.4.7: Use-Case Diagram

Socio - Network for Developers

Use Cases:

1. Register
2. Login
3. Add Comment
4. Add Post
5. View Post

Table 4.2.1: Use Case Table - Register

Use Case ID	1
Use Case Name	Register
Actor	Any user
Pre-Condition	They must register themselves first
Post-Condition	User can login
Flow of events	Login,Register or Edit details

Table 4.2.2: Use Case Table - Login

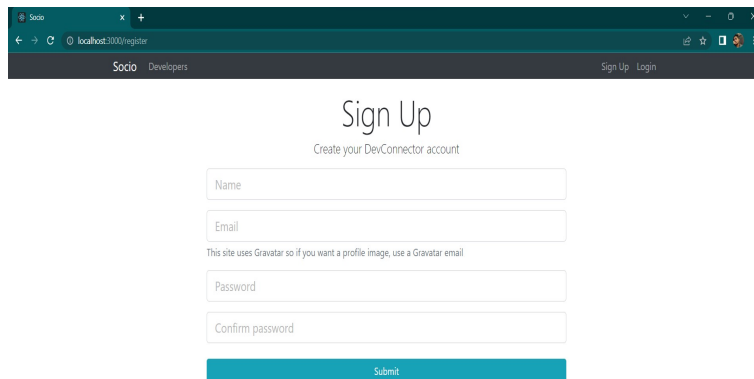
Use Case ID	2
Use Case Name	Login
Actor	Any user
Pre-Condition	They must register themselves first
Post-Condition	User can view the profiles and posts
Flow of events	Login,Register or Edit details abd add comments

Table 4.2.1: Use Case Table - Add comment

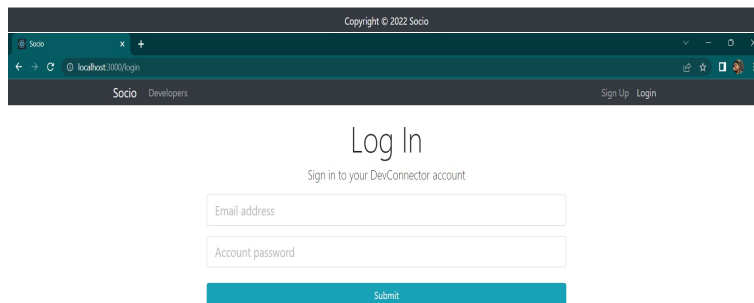
Use Case ID	1
Use Case Name	Comment
Actor	Any user
Pre-Condition	They must register themselves first
Post-Condition	User can login
Flow of events	add a new comment

3 Project Implementation and Testing

3.1 Login and Register



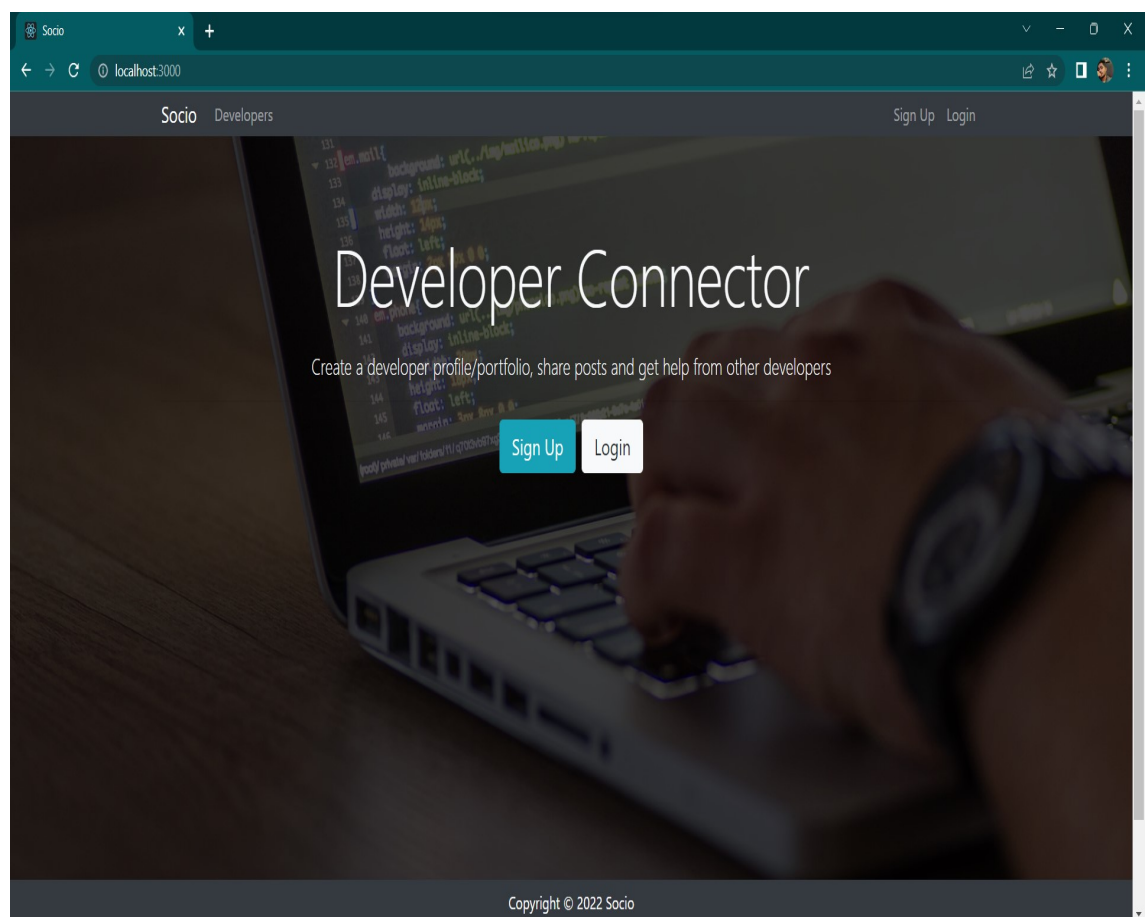
The screenshot shows a web browser window with the address bar displaying 'localhost:3000/register'. The page title is 'Socio Developers'. The main heading is 'Sign Up' with the subtext 'Create your DevConnector account'. Below this, there are four input fields: 'Name', 'Email', 'Password', and 'Confirm password'. A small note states 'This site uses Gravatar so if you want a profile image, use a Gravatar email'. At the bottom is a blue 'Submit' button.



The screenshot shows a web browser window with the address bar displaying 'localhost:3000/login'. The page title is 'Socio Developers'. The main heading is 'Log In' with the subtext 'Sign in to your DevConnector account'. Below this, there are two input fields: 'Email address' and 'Account password'. At the bottom is a blue 'Submit' button.

Socio - Network for Developers

3.2 Home - Landing View



3.2.1: Home View

Socio - Network for Developers

3.3 Profile

The screenshot shows a web browser window with the address bar at `localhost:3000/create-profile`. The page title is "Socio" and the navigation bar includes "Socio", "Developers", "Posts Feed", "Dashboard", and a "Logout" button. The main content area is titled "Create your profile" with the subtitle "Let's get some information to make your profile look beautiful". The form contains the following fields and labels:

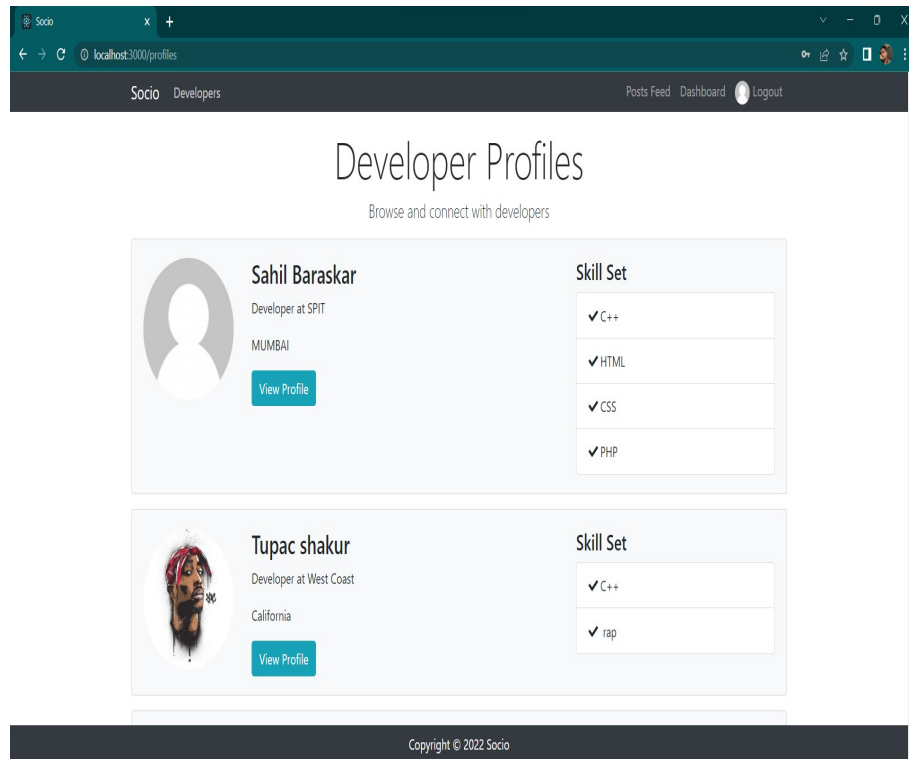
- * Desired profile handle (text input)
- A unique handle for your profile URL.
- * Select Professional Status (dropdown menu)
- Give us an idea where you are at in your carrier (text input)
- Company (text input)
- Could be your own company or one you work for
- Website (text input)
- Could be your own website or a company one
- Location (text input)
- City or city & state
- * Skills (text input)
- Please use comma seperated values (eg. HTML,CSS,JavaScript,PHP)
- Github Username (text input)
- If you want your lates repositories and a github link, include your username
- Short Bio (text input)
- Tell us about yourself

The footer of the page reads "Copyright © 2022 Socio".

3.3.1: Profile

Socio - Network for Developers

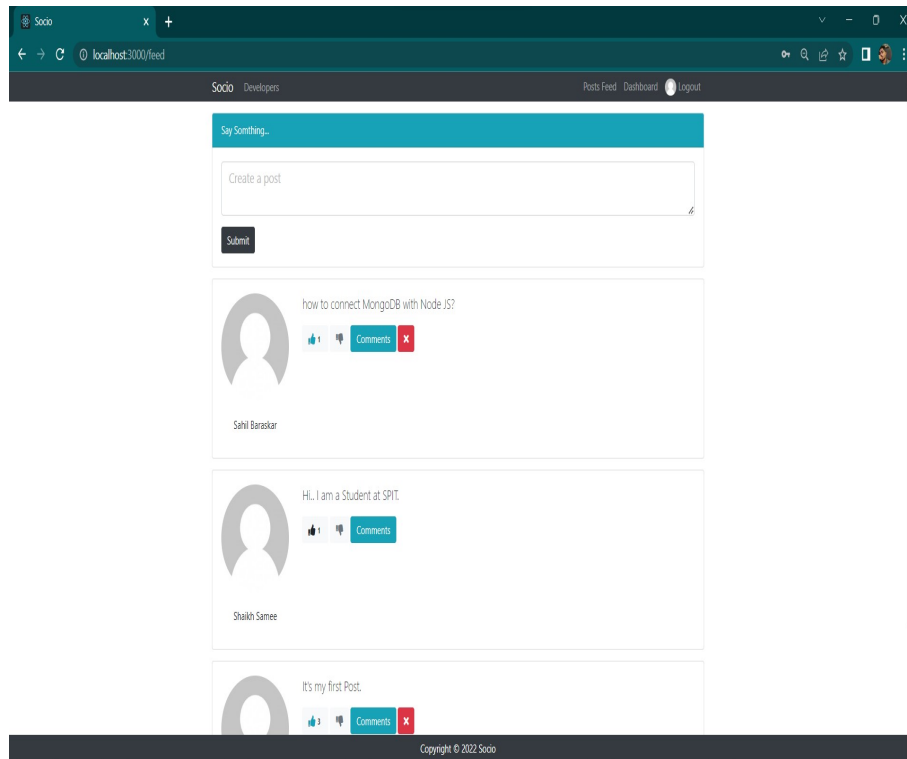
3.4 Developers



3.4.1: Developers View

Socio - Network for Developers

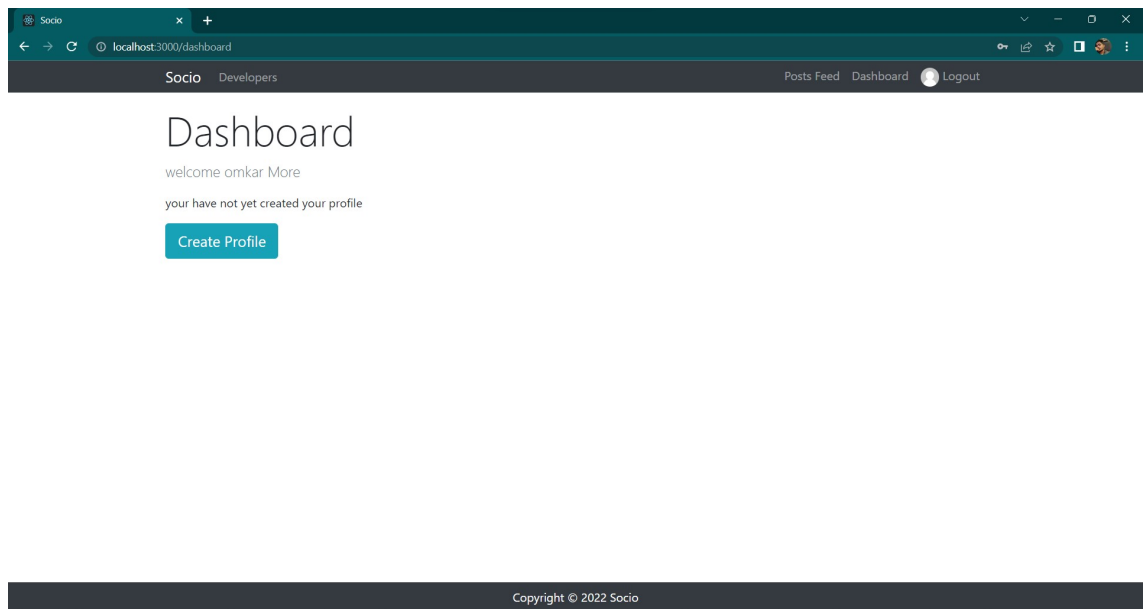
3.5 Posts



3.5.1: Posts

Socio - Network for Developers

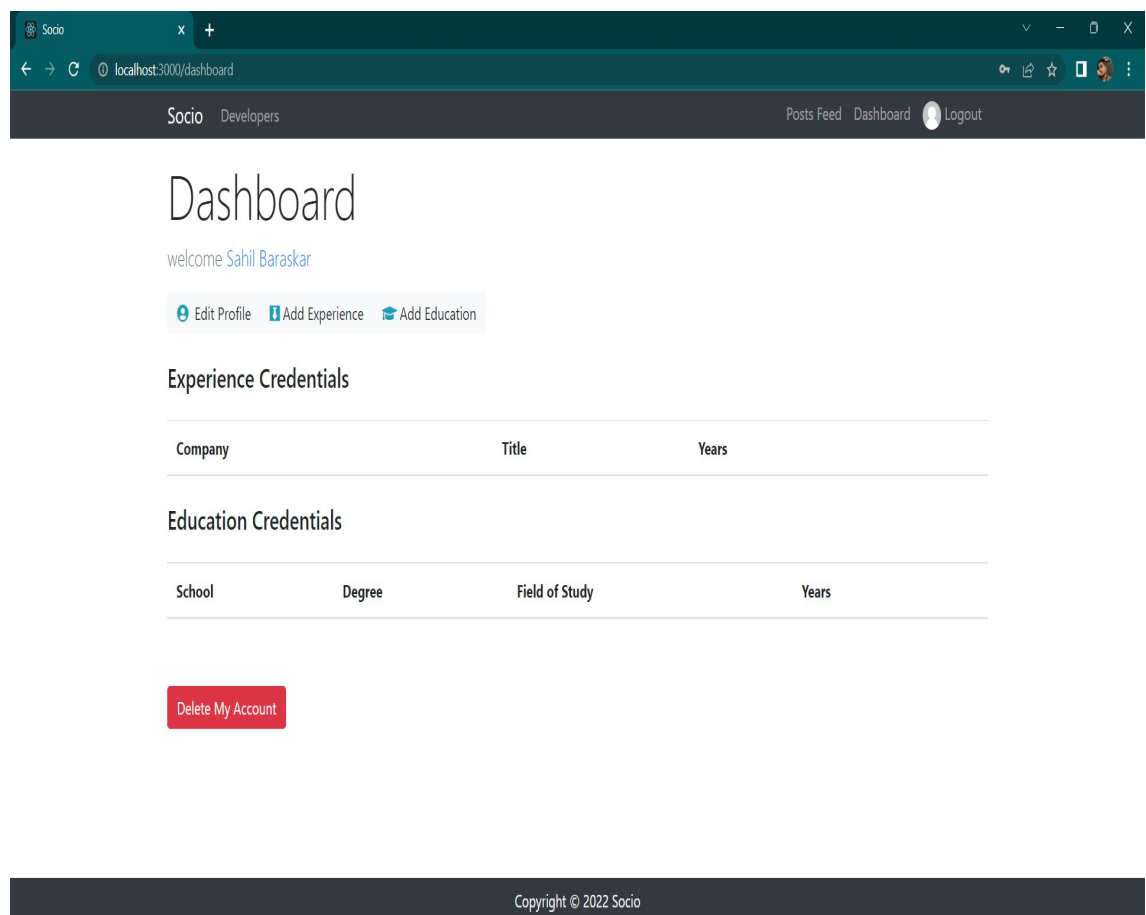
3.6 Dashboard



3.6.1: Dashboard

Socio - Network for Developers

3.7 DashBoard Edit



3.7.1: Dashboard Edit

3.8 Add Experience

The screenshot shows a web browser window with the address bar at `localhost:3000/add-experience`. The page title is "Socio" and the navigation bar includes "Socio", "Developers", "Posts Feed", "Dashboard", and a "Logout" link. The main content area is titled "Add Experience" with the subtitle "Add any job or post you have had in the past or current". A "Go Back" link is at the top left. A legend indicates that fields with an asterisk (*) are required. The form contains the following fields: "Company" (required), "Job Title" (required), "Location", "From Date" (date picker), "To Date" (date picker), a checkbox for "Current Job", and a "Job description" text area with the placeholder "Tell us about your position". A teal "submit" button is at the bottom. The footer shows "Copyright © 2022 Socio".

Go Back

Add Experience

Add any job or post you have had in the past or current

* = required fields

* Company

* Job Title

Location

From Date

dd-mm-yyyy

To Date

dd-mm-yyyy

☐ Current Job

Job description

Tell us about your position

submit

Copyright © 2022 Socio

3.8.1: Add Experience

3.9 Add Education

The screenshot shows a web browser window with the URL `localhost:3000/add-education`. The page title is "Add Education" with the subtitle "Add any school or college you have attended." A "Go Back" link is at the top left. The form includes several input fields: "School" (marked with an asterisk), "Degree" (marked with an asterisk), "Field of study", "From Date" (with a date picker icon), "To Date" (with a date picker icon), and a checkbox for "Current Job". Below these is a text area for "Program description" with a placeholder "Tell us about your education". A teal "submit" button is at the bottom. The footer shows "Copyright © 2022 Socio".

Go Back

Add Education

Add any school or college you have attended.

* = required fields

* School

* Degree

Field of study

From Date

dd-mm-yyyy

To Date

dd-mm-yyyy

☐ Current Job

Program description

Tell us about your education

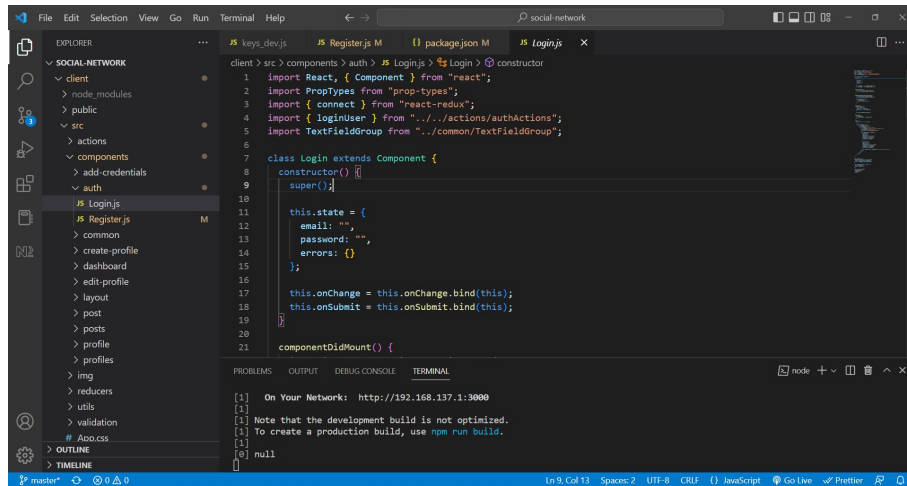
submit

Copyright © 2022 Socio

3.9.1: Education Details

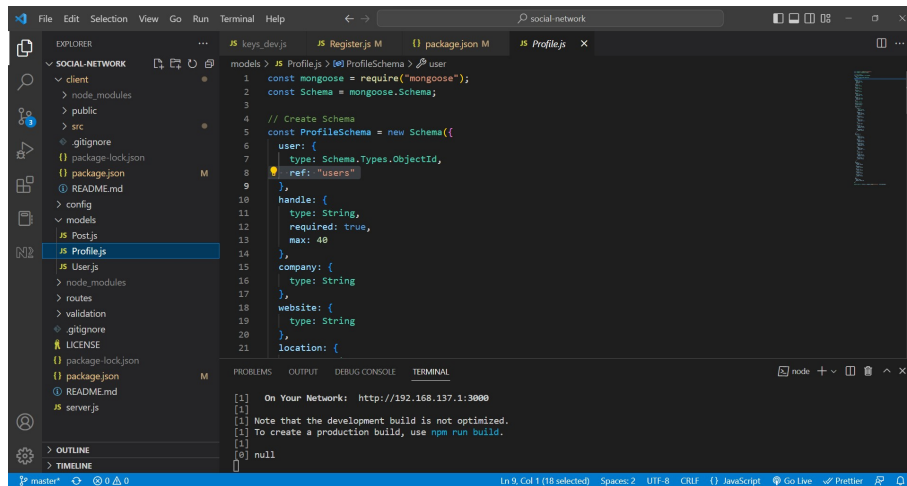
Socio - Network for Developers

3.10 Code 1



```
client > src > components > auth > Login.js > constructor
1 import React, { Component } from "react";
2 import PropTypes from "prop-types";
3 import { connect } from "react-redux";
4 import { loginUser } from "../../actions/authActions";
5 import TextFieldGroup from "../../common/TextFieldGroup";
6
7 class Login extends Component {
8   constructor(props) {
9     super(props);
10
11     this.state = {
12       email: "",
13       password: "",
14       errors: {}
15     };
16
17     this.onChange = this.onChange.bind(this);
18     this.onSubmit = this.onSubmit.bind(this);
19   }
20
21   componentDidMount() {
22
23   }
24
25   constructor() {
26     super();
27   }
28
29   state = {
30     email: "",
31     password: "",
32     errors: {}
33   };
34
35   onChange = (e) => {
36     this.setState({
37       [e.target.name]: e.target.value
38     });
39   };
40
41   onSubmit = (e) => {
42     e.preventDefault();
43     loginUser(this.state);
44   };
45
46   render() {
47     const { email, password, errors } = this.state;
48
49     return (
50       <div>
51         <TextFieldGroup
52           type="text"
53           value={email}
54           onChange={this.onChange}
55           error={errors.email}
56         />
57         <TextFieldGroup
58           type="password"
59           value={password}
60           onChange={this.onChange}
61           error={errors.password}
62         />
63         <button type="submit">Login</button>
64       </div>
65     );
66   }
67 }
68
69 Login.propTypes = {
70   // ...
71 };
72
73 export default connect(null, loginUser)(Login);
```

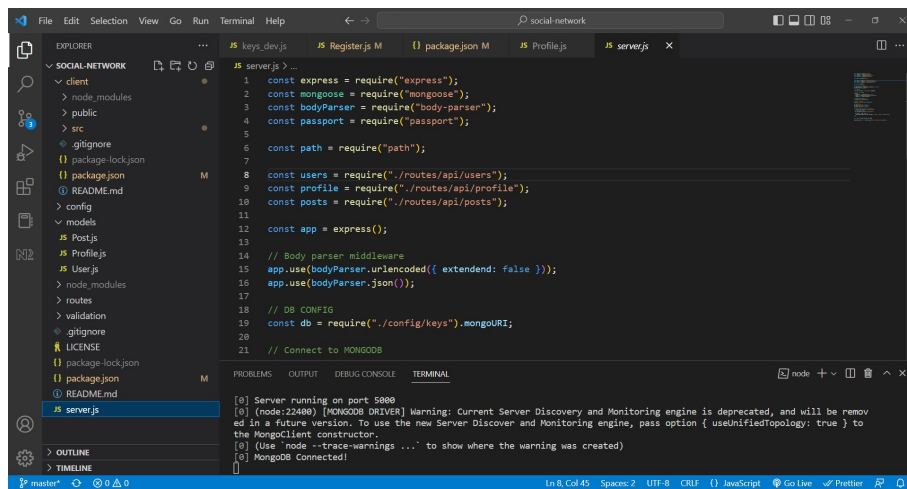
3.11 Code 2



```
models > Profile.js > ProfileSchema > user
1 const mongoose = require("mongoose");
2 const Schema = mongoose.Schema;
3
4 // Create Schema
5 const ProfileSchema = new Schema({
6   user: {
7     type: Schema.Types.ObjectId,
8     ref: "users"
9   },
10   handle: {
11     type: String,
12     required: true,
13     max: 40
14   },
15   company: {
16     type: String
17   },
18   website: {
19     type: String
20   },
21   location: {
22
23   }
24 });
25
26 export default mongoose.model("profiles", ProfileSchema);
```

Socio - Network for Developers

3.12 Code 3



The screenshot shows the Visual Studio Code editor with a project named 'social-network'. The Explorer sidebar on the left shows the file structure, including 'client', 'node_modules', 'public', 'src', and various configuration files. The main editor window displays the code for 'server.js', which is a Node.js application using Express, Mongoose, and Passport. The code includes routes for users, profile, and posts. The terminal at the bottom shows the output of running the application, indicating it is running on port 5000 and connected to MongoDB.

```
1 const express = require("express");
2 const mongoose = require("mongoose");
3 const bodyParser = require("body-parser");
4 const passport = require("passport");
5
6 const path = require("path");
7
8 const users = require("../routes/api/users");
9 const profile = require("../routes/api/profile");
10 const posts = require("../routes/api/posts");
11
12 const app = express();
13
14 // Body parser middleware
15 app.use(bodyParser.urlencoded({ extended: false }));
16 app.use(bodyParser.json());
17
18 // DB CONFIG
19 const db = require("../config/keys").mongoURI;
20
21 // Connect to MONGODB
```

Terminal Output:

```
[0] Server running on port 5000
[0] (node:22480) [MONGODB DRIVER] Warning: Current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
[0] (Use 'node --trace-warnings ...' to show where the warning was created)
[0] MongoDB Connected!
```

4 Test Cases

Table 6.1: Test Case - Login and Register

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	User enter user id and password	Enters the correct user id and password	Log in Successful	Home Page	Pass
2	User enter user id and password	Enters the user id and password	Prompt error	Prompt error	Pass
3	User enter user id and password	Valid user id and password which doesn't exist in Database	Registered Successfully	Login Page	Pass
4	User enter user id and password	Invalid user id and password which contains in Database	Prompt error	Prompt error	Pass

Table 6.2: Test Case - Others

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	User enters invalid information in developer section	Invalid developer details	Prompt Message showing error	Successful Updated	Fail
2	User enters valid information in Profile	Valid Profile Details	Profile Updated successful	Profile Updated successful	Pass

5 Limitations

- It needs internet to be accessed.
- It does not have the functionality of privatizing the profile, all the profiles are public.
- Email verification is not provided, so authentication of a correct email not available.
- It does not have a feature to add various projects other than github.

6 Future Enhancements

- User will be able post videos.
- Users can report to another user or post.
- Users can add their various pictures.

7 User Manual

Part 1 – Register

Upon opening the application, user will be greeted with the registration screen.

If the user has no account, user can click on register and register self.

After verification of user id, User's account will be saved in our database. The user can now proceed to login.

Part 2 – Login

User needs to enter email first and then password. If there is an active internet connection, user can proceed to login.

Part 3 – Profile

User can access/add/update personal details. User can update this details any-time.

Part 4 – Developer Section

All the Users details and profile are visible on the developers page, anyone can view a profile and can check their Github repos and other social media accounts.

Part 5 – Create Post

User can create a new post, add comment and can like/ dislike other posts.

8 Bibliography

8.1 Web References

- [1.] <https://link.springer.com/article/10.1007/s10796-017-9810-y>
- [2.] <https://www.liebertpub.com/doi/10.1089/cyber.2020.0134>
- [3.] <https://mospace.umsystem.edu/xmlui/bitstream/handle/10355/59903/research.pdf>
- [4.] <https://stackoverflow.com/>
- [5.] <https://www.draw.io/>
- [6.] <https://www.geeksforgeeks.org/>