“Socipedia” Social Media Application

Project Report

AR 98209

AF/19/15557

ICTC 3101.3

Web Application Development

Table of Contents

Introduction

Sociopedia is a social media application to connect with people. It provides a platform to interact with people. Mainly, people can manage their profiles in the application. As well as a person can share content with others by posing posts. Also, the other users are given a chance to react to the posts. So, basically, this application can be taken as a platform to share the interests of the users. After creating an account using the registration field, users are allowed to create posts.

I have developed the project using the MERN stack which stands for Mongo DB, Express JS, React and Node. I have developed my Frontend using React framework, a user-friendly framework for developing applications. As well as the React Router has been used for navigation parts. Apart from that, Formik Yup has been used for form validation. Again, I have done state management using Redux Toolkit while React Drop zone is used to upload files.

When creating the backend, I used Express JS as the backend framework. And also, I have used Mongoose to manage my database which is Mongo DB. Again, JSON web Token has been used for the authentication part in my project. Also, Multer has been used for uploading files.

Motivation

The motivation behind the development of the above-mentioned social media application is to create a media platform to have meaningful connections. Also, it has become a great problem with the privacy of the user. Mainly, user privacy and security have been considered when creating the application. Also, sharing the interests of users is considered as a main feature of the application.

Apart from that, crating the an user-friendly application with the fully responsiveness is highly considered.

Problems and Objectives

Problems

Many problems can be found when considering social media applications. Lack of personalized social experience can be taken as a main problem here. Many social media applications are failed to provide a secured system to the users while becoming the platform is user-friendly.

Also, the privacy and security concern has become a such problem here. Because of this, the usage of the social media application has been decreased and many of new users are suffering from a fear of engaging the activities through the social media.

Moreover, inefficient content moderation is also a significant challenge for social media platform. Insufficient content moderation can lead to the cyberbullying, and other forms of harmful content.

Objectives

In this application, my objective is to provide a highly personalized social media experience to the user. As well as, according to the uploading of each user, I can identify the preferences and interests of the user and improve his personalized experience by doing recommendations.

Using the like and comments section, users can share their interests and ideas on the pictures has been uploaded by several users. And lack of the functionalities such as calling, messaging with each other privately, I can decrease the fear of the users who think about more privacy.

Also, I expect improve the user interaction and improve the connections by implementing adding friends feature to the application.

System Overview

This social media platform has been built using MERN stack with incorporating the other powerful technologies. I can describe the system overview of this project using two areas which are capabilities and GUI.

Capabilities

The above-mentioned Sociopedia application offers some capabilities. They are,

User registration and authentication

User registration and authentication can be identified as the main function of this social media application. This registration functionality enables new users to create their accounts and join the platform. Especially, with this part, my objective is to create a secure system for the registration process of a user.

Mainly I have used Formik and Yup technology for form validation here. It provided a user-friendly approach to validate the input data. Not only that, the technology simplifies the process by helping to manage form state, handle form submission and enable easy integration with Yup for the form validation part. When validating, Yup helps with managing required fields, minimum and maximum length and the email format.

Here, it is very important that, I have designed a user-friendly registration form to collect the required information which are first name, last name, occupation, location, email and password. for the registration. Also, real-time validation is helpful to identify incomplete entries for users.

<u>User Login</u>

This feature allows users to login to the application. When a user login, the token given to the user is searched. If it is there, then the user can log in to the system. Also, if the user is not in the database, that user cannot access to the application. As well as if the user gives an invalid email or wrong password, again the user cannot access the application. He cannot see any posts.

Mainly, I have created and validate the user login forms using Formik Yup technologies while JSON web token has been used for the authentication.

<u>Friend Management and Connections</u>

Managing friends is another main capability in my application. It directly affects to the social enhancement of the users who are using the application. With this friend management feature, the users are allowed to interact with their friends.

Basically, under the friend component, the friend’s name, profile picture and occupation is shown. Then the user can easily identify the friend. Using the “PersonAddOutlined” icon and the “PersonRemoveOutlined” icons is helpful to add friends and remove friends from our list.

Using the PATCH method request, the list of friends is updated, after adding of removing the friends. Mainly, this process is happened based on the user Id and the friend Id.

Also, using the user navigation part, the user is given a chance to directly go to the friend’s profile and see the details. Additionally, the “navigate(0)” function is invoked to reset the navigation history, providing a good user experience to the user.

<u>Post Management Capability</u>

The post-management capability allows users to create, view and delete posts within the social media application. Here when creating the post, the details of the post, the user who creates the post, the description of the post and the date and time are taken and stores in the database user a specific collection named Post.

Also, under the “getFeedPosts” function, all posts in the application are retrieved. It feches all posts using the “Post” model. This endpoint ensures that only the authenticated users can access the posts. As well as using the “getUserPost” the posts associated with a specific user can be taken. It fetches the posts using “Post” model and then filer it using the “userId”.

Under the post-management, again I can describe the “likePost” functionality. In this functionality, a user is given the chance to like or unlike to a specific post identified by a specific Id. Also, when making likes on a post, the count is calculated by associating the post id to give the number of likes.

Then “deletePost” function allows users to delete their own posts. It deletes the posts specified by “postId” from the database. Also, in the “addComment” function, users are allowed to keep a comment under a post. That comment count is also calculated under the post id.

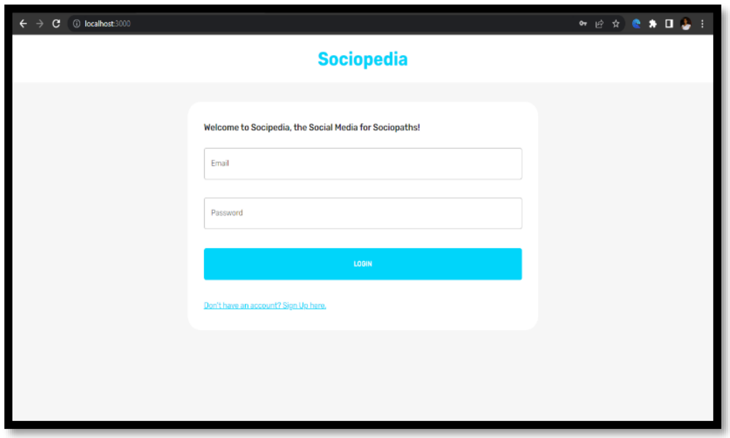
Graphical User Interface

The Sociopedia app features a visually appealing graphical user interface. The interface incorporates intuitive navigation menus, and interactive elements to enhance the user experience.

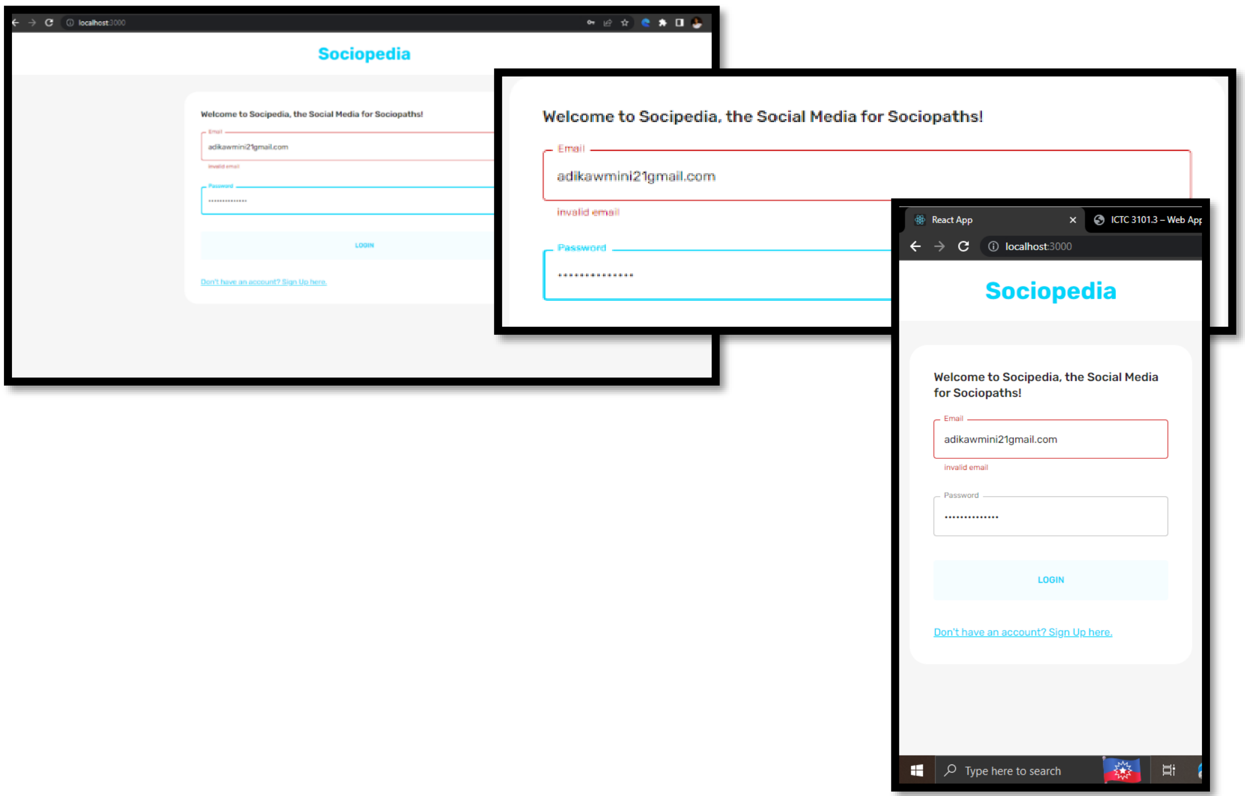
In my application, I have used a good color combination which provides a pleasant user experience. Mainly, I have used grey color and its shades throughout the application. It has become helpful to make a neutral background in my application. Also, I have used an eye-catching shade of blue as the primary color. It is helpful when identifying the main elements of the application. When coming to the dark mode of my application, I have used a somewhat deep color shade with a black color background.

Also, the graphical user interface has been created for both desktop sites and mobile sizes.

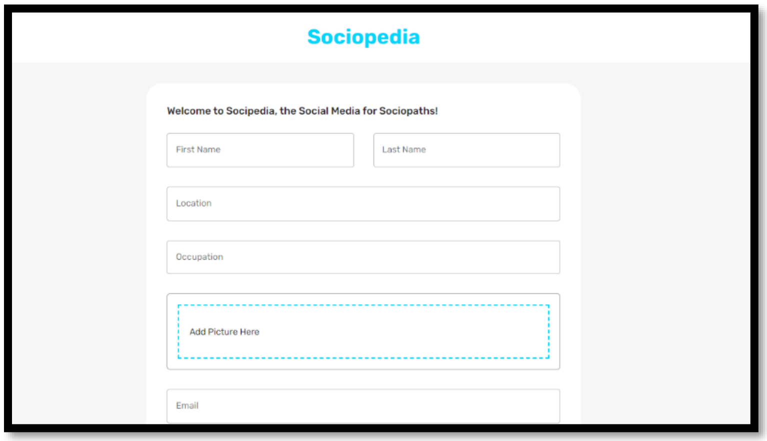
First, the user is directed to the login page. Here users can log in to the system using their email and password. Also, at the top of the page, the user is shown the name of the application and then a message saying “Welcome to Sociopedia, the social media for sociopaths.” I have used two input boxes and a sign-in button to create the GUI with the usage of Formik and Yup technologies. User is shown the required field that should be input in the input boxes.


localhost:3000
Sociopedia
Welcome to Socipedia, the Social Media for Sociopaths!
Email
Password
LOGIN
Don't have an account? Sign Up here.

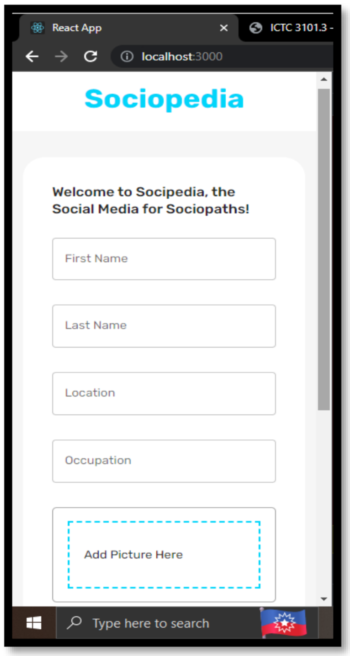
Here, if the user missed any area that should be input that is shown by becoming the text input box in to red color and error message. As well as, as I have used Formik Yup technologies for the validation, it checks whether the emails are in the correct format or not. Unless it is in the correct format, that is also shown by an error message as follows.



If that user hasn't got an account, he can move to the registration page by clicking the "don't have an account. Sign Up here" part below the sign-in button. Then the user is shown the registration page. Here, there are some input boxes to get user inputs such as first name, last name, location, occupation, email and password. Also, there is another box to upload a picture that would be appeared as the profile page of the user. That box allows to input picture formats such as png and jpg.


Sociopedia

Welcome to Socipedia, the Social Media for Sociopaths!

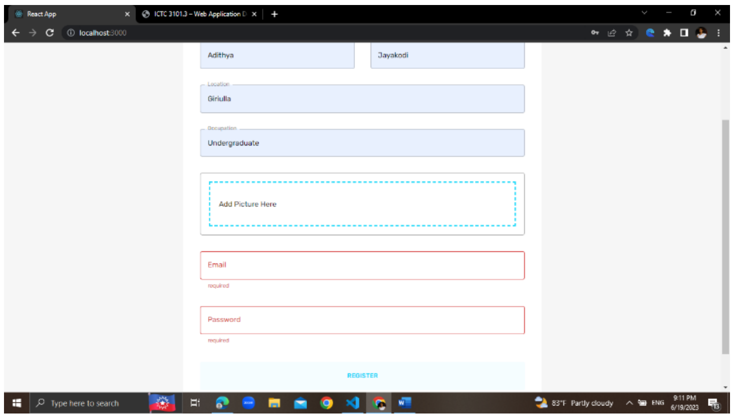


Sociopedia

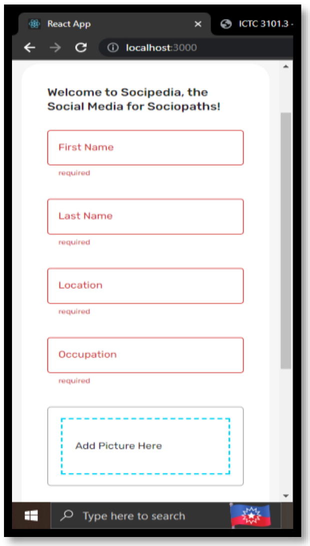
Welcome to Socipedia, the Social Media for Sociopaths!

Here, if the user remains fields without filling them, they are showing a message as they are required fields. Also, the insert box and the content in the boxes are displayed in red by sending the message that, there is an error. Also, if a user inputs an invalid email or if the format of an email is not in the input, the email input box will be red and show that

the email is invalid.

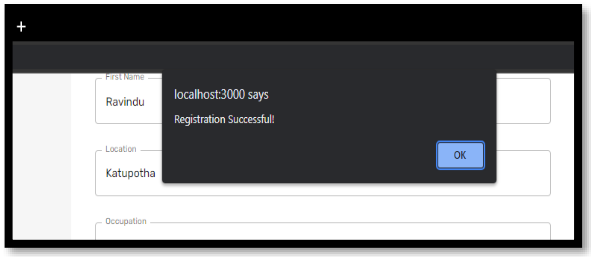


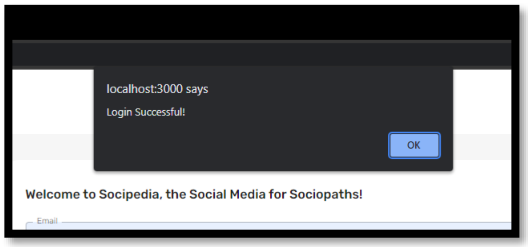
Adithya
Jayakodi
Location: Giriulla
Occupation: Undergraduate



Welcome to Socipedia, the Social Media for Sociopaths!

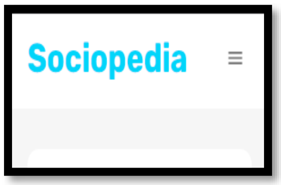
After completing the registration, the user is moved to the login page. Before moving to the login page, the user is shown an alert saying the registration is successful. As well as, when the user is login to the application, this type of alert is shown which is saying the login is successful.

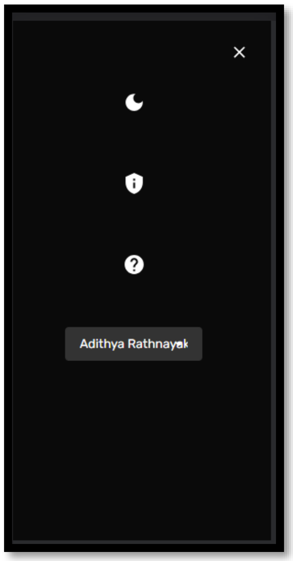


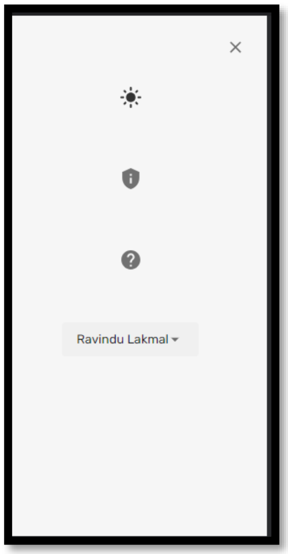


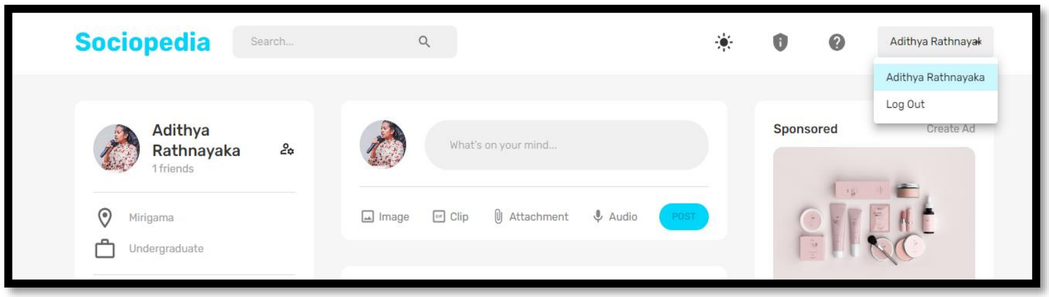
After login to the system using the email and password used when creating the account, the user is carried to the home page. The home page is created with four sections.

At the top of the home page, the navigation bar is shown. In the mobile size, this section appears as a hide section. That means that is shown as three short lines (menu icon). After clicking on it, the navigation bar appears.







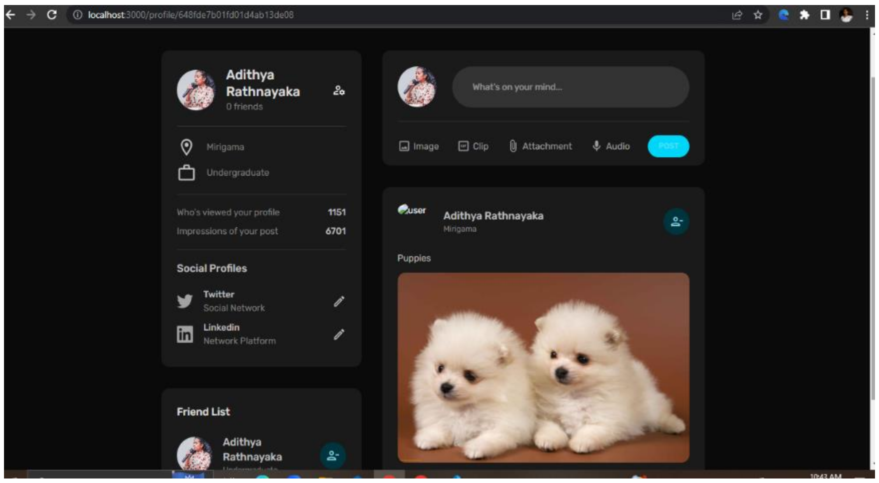


In the left-side corner of the navigation bar, the name of the social media application is shown which is “Sociopedia”. And I have used light blue color as the text color there. Also, I have made that word clickable. So, we are able to refresh the page by clicking the word.

After that, I have used a search box with a search icon. In this part, users are allowed to search for the people who have created accounts. So, that feature is useful when finding friends and adding them to our friend list.

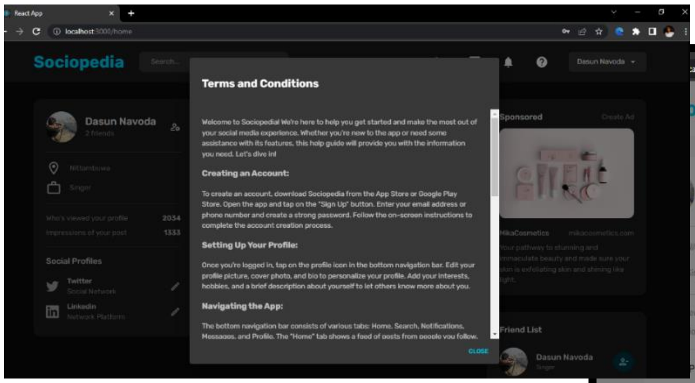
Also, then there is an icon to change the mode from light mode to dark mode. By clicking that users are allowed to set the application into dark mode. After clicking the icon, the icon also changes. Clicking that icon user can move to the light mode again. In the dark mode, I have used somewhat deep colors,

.



Then the next icon shows the privacy policies of the application. After clicking the icon, the user is shown a dialog box and in it, the content of the privacy policy is shown. Also, at the end of the dialog box, there is a close button. I have created functions to close the dialog box after clicking the close button.

Then, there is an icon showing the “helping” section. By clicking that icon, the user is shown a dialog box the same as the previous icon. And in that dialog box, the user is shown all the instructions for creating an account properly, managing it and issues may occur.



Terms and Conditions

Welcome to Sociopedia! We're here to help you get started and make the most out of your social media experience. Whether you're new to the app or need some assistance with its features, this help guide will provide you with the information you need. Let's dive in!

Creating an Account:

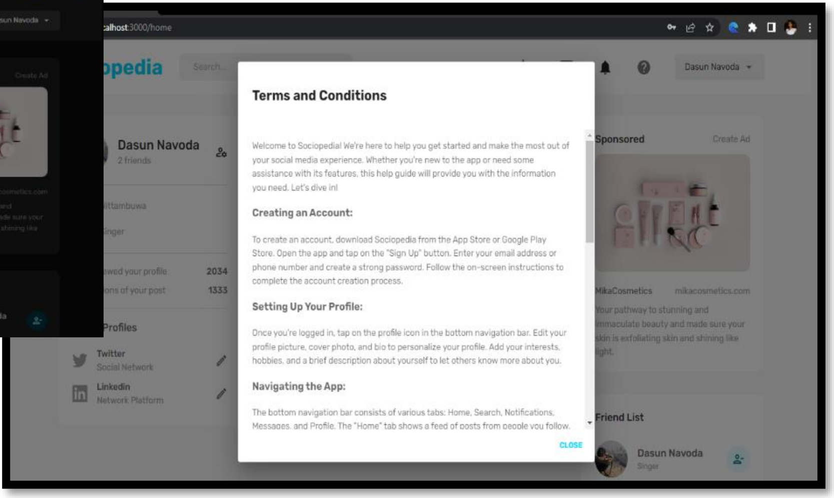
To create an account, download Sociopedia from the App Store or Google Play Store. Open the app and tap on the "Sign Up" button. Enter your email address or phone number and create a strong password. Follow the on-screen instructions to complete the account creation process.

Setting Up Your Profile:

Once you're logged in, tap on the profile icon in the bottom navigation bar. Edit your profile picture, cover photo, and bio to personalize your profile. Add your interests, hobbies, and a brief description about yourself to let others know more about you.

Navigating the App:

The bottom navigation bar consists of various tabs: Home, Search, Notifications, Messages, and Profile. The "Home" tab shows a feed of posts from people you follow.



Terms and Conditions

Welcome to Sociopedia! We're here to help you get started and make the most out of your social media experience. Whether you're new to the app or need some assistance with its features, this help guide will provide you with the information you need. Let's dive in!

Creating an Account:

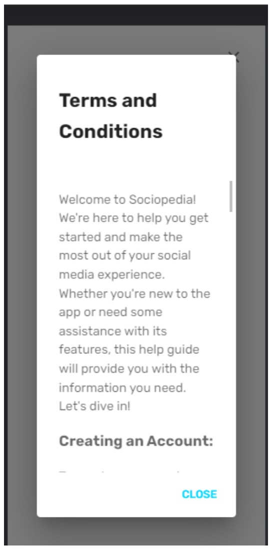
To create an account, download Sociopedia from the App Store or Google Play Store. Open the app and tap on the "Sign Up" button. Enter your email address or phone number and create a strong password. Follow the on-screen instructions to complete the account creation process.

Setting Up Your Profile:

Once you're logged in, tap on the profile icon in the bottom navigation bar. Edit your profile picture, cover photo, and bio to personalize your profile. Add your interests, hobbies, and a brief description about yourself to let others know more about you.

Navigating the App:

The bottom navigation bar consists of various tabs: Home, Search, Notifications, Messages, and Profile. The "Home" tab shows a feed of posts from people you follow.

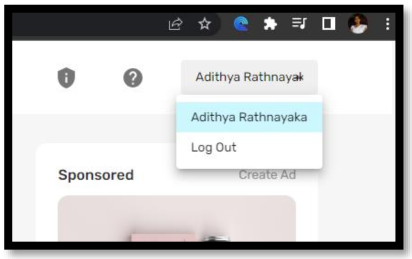


Terms and Conditions

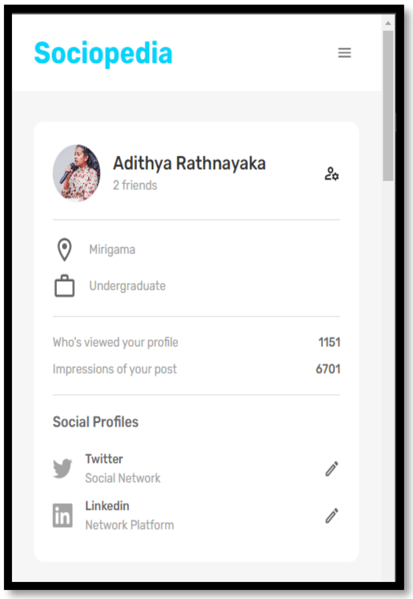
Welcome to Sociopedia! We're here to help you get started and make the most out of your social media experience. Whether you're new to the app or need some assistance with its features, this help guide will provide you with the information you need. Let's dive in!

Creating an Account:

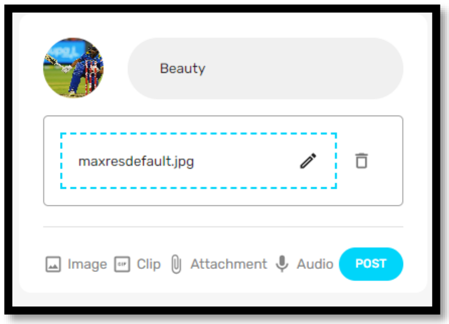
Also, at the right corner of the navigation bar, the first name and the last name of the user who is logged in at the time is presented. And there is a dropdown arrow in there. After clicking that arrow, user name and the logout message can be seen. By clicking that logout button, the user can log out from the application and the user is directed to the login page again

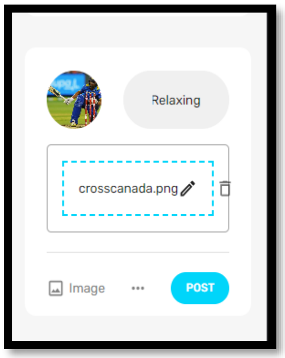


And at the right side of the page, a user profile has appeared. In that area, the profile picture, number of friends, location, and occupation is shown. And the number of views and the impressions are shown there. Also, there is a space to add social media networks as well.

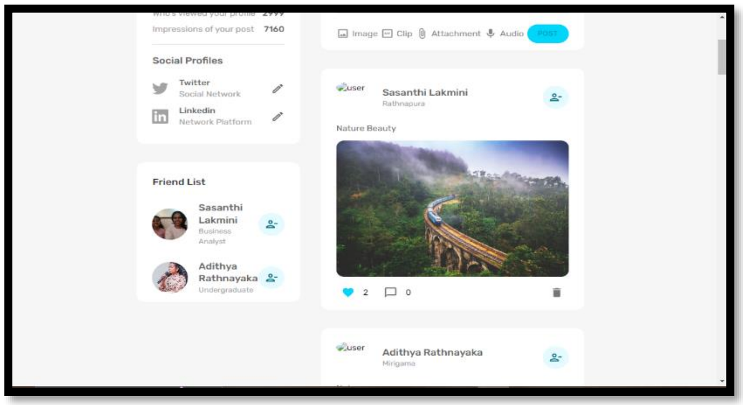


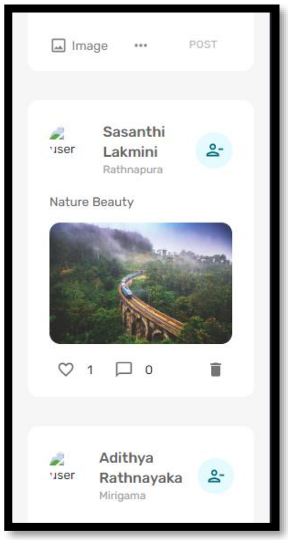
Also, top of the middle column, the post-creating area is shown. In this area, the user can upload images by sending a comment to that through the space "What's on your mind". The image can be posted using the "post" button.





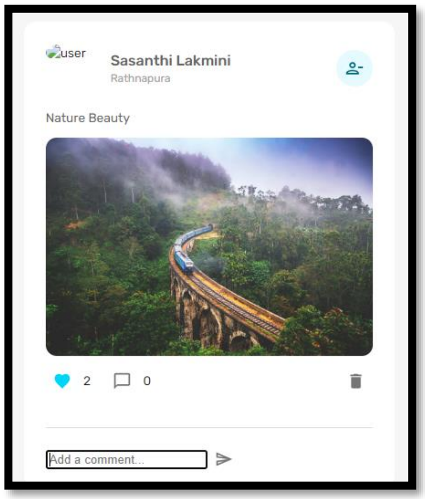
When a new post is added, it is added to the end of the posts list. That post area is shown in the middle of the application. And by going down in the page, the new posts which have been uploaded by several users can be seen.

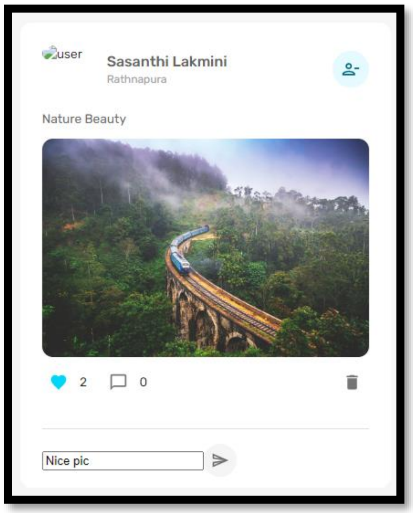




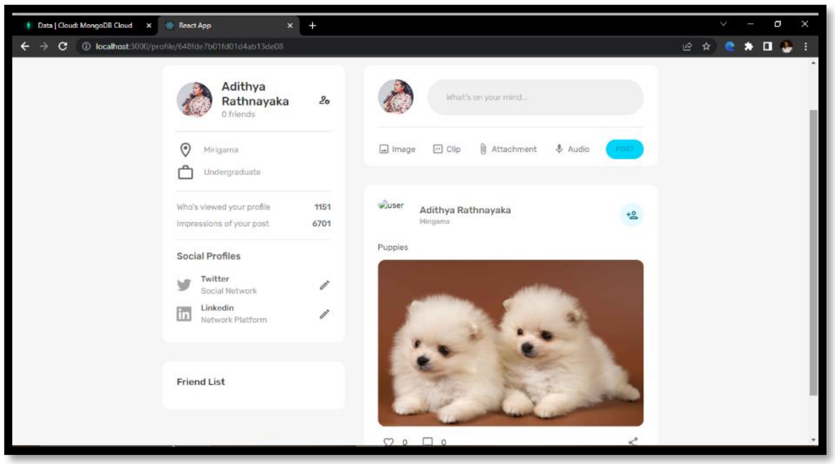
Also, if the user has gone to his own profile and sees, the user can his own posts uploaded. Newly uploaded pictures can be seen at the end of the posts list.

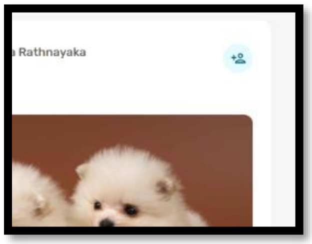
Then as shown in the picture, people who have created accounts can like to the posts or comment to the pictures. The number of likes and the number of comments is shown under each picture while it is saved in the database.



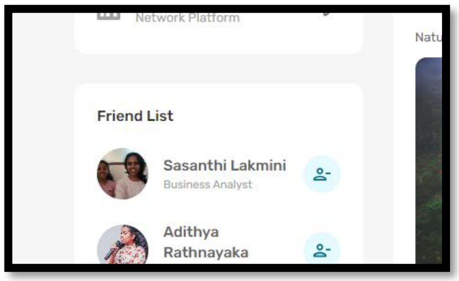


The friend list can be seen below the profile details of the logged-in user. A user can add a person as a friend by clicking the human icon placed in the right corner of the published post. The added friend list is shown as below.



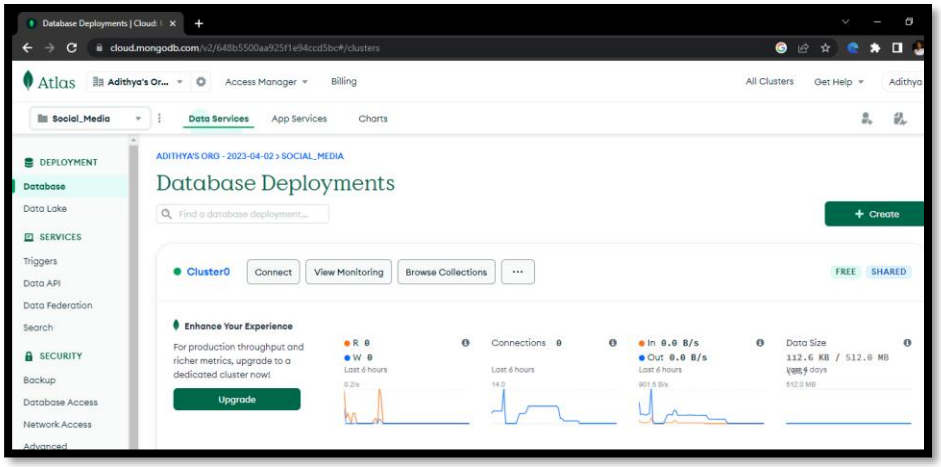


Here, after adding a person to the friend list, the plus mark (+) with the human icon has become the minus mark (-). Then, by clicking that human icon with the minus mark, the user can remove the friend from the icon. And then, the number of friends under the user's name is decreased. Also, that person is removed from the friend list.



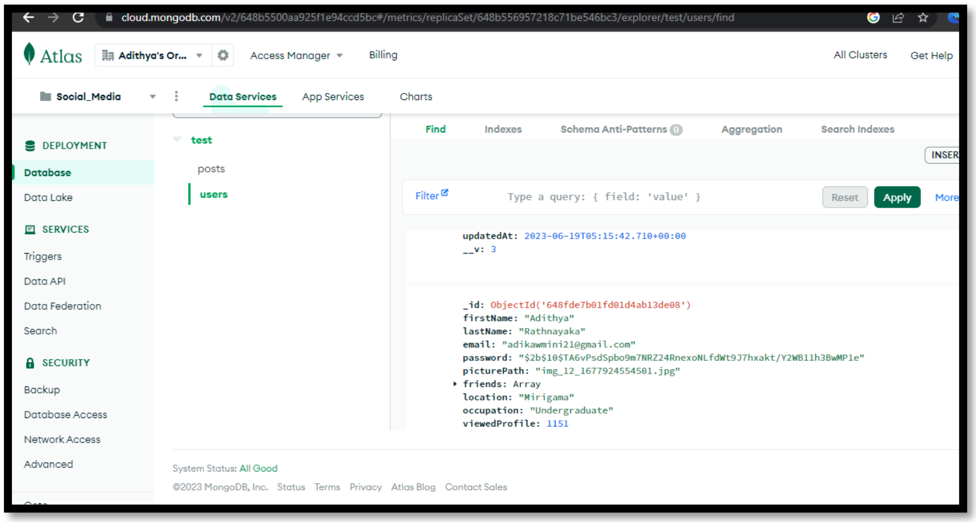
Database Structure

My social media application utilizes MongoDB as the database which is a non-SQL document-based database. Here, I have used MongoDB Atlas as the database service. That provides a cloud-based platform for hosting and managing the database. And I have named my database as "Social_Media."



For each component which are needed to be stored in a database, I have created collections. Mainly, I have created a user's collection which is stored the details of the users who come and create accounts in the social media application. Here, in the user collection, there are some key fields named, _id, first name, last name, email, password, picture path, friends, location, occupation, viewed profile, created date and time and the updated time and date.

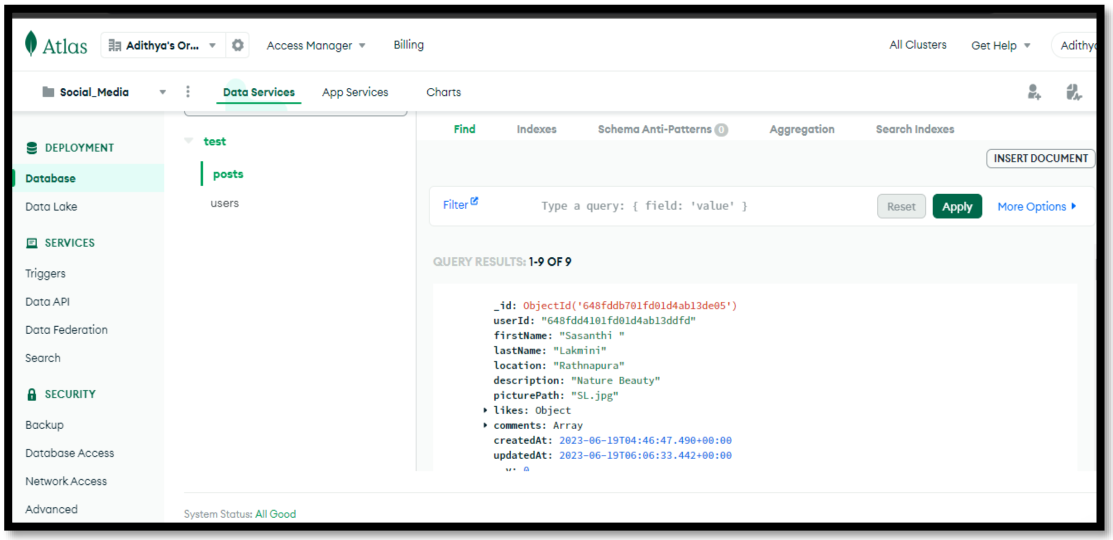
In the _id field, each user document is given a unique id number. So, it helps to manage users and avoid conflicts when there are users with the same user name. in the profile-picture field, the URL pointing to the profile picture image file is shown. Also, in the "friends" field, details are stored as an array.



```
updatedAt: 2023-06-19T05:15:42.710+00:00
__v: 3

_id: ObjectId('648fde7b01fd01d4ab13de08')
firstName: "Adithya"
lastName: "Rathnayaka"
email: "adikawmini21@gmail.com"
password: "$2b$10$TA6vPsdSpbo9m7NRZ24RnexoNLfdWt9J7hxakt/Y2WB11h3BwMP1e"
picturePath: "img_12_1677924554501.jpg"
▸ friends: Array
location: "Mirigama"
occupation: "Undergraduate"
viewedProfile: 1151
```

For posts also, I have a separate collection called posts. It is stored the details related to the posts. Here also, _id, user Id, first name, last name, location, description, picture path, likes, comments and the created date and time are stored. Here, the comments are stored as an array while the likes are stored as objects. In the user id, it stores the id of the user who creates the post. Also, the post is also given a unique id to each post.



QUERY RESULTS: 1-9 OF 9

```
  _id: ObjectId('648fddb701fd01d4ab13de05')
  userId: "648fdd4101fd01d4ab13ddfd"
  firstName: "Sasanthi "
  lastName: "Lakmini"
  location: "Rathnapura"
  description: "Nature Beauty"
  picturePath: "SL.jpg"
▸ likes: Object
▸ comments: Array
  createdAt: 2023-06-19T04:46:47.490+00:00
  updatedAt: 2023-06-19T06:06:33.442+00:00
```

API Implementation and Endpoints

In this social media application, I have implemented several APIs for user authentication, user-related some operations and post management and friend management. In this application, I have used Express.js framework in Node.js while following RESTful principles for the process of implementation of APIs. Here I have provided relevant pictures from Postman and Network category in the console to show the status of the APIs.

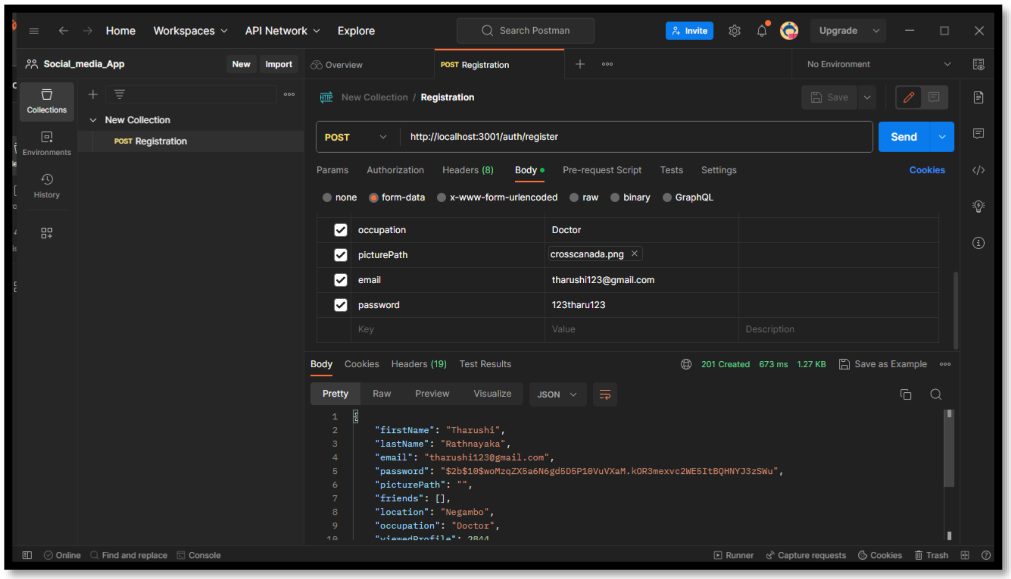
01. User Authentication API

In this part, there is a registration endpoint. Registration endpoint allows user to create a new account by providing their personal information such as first name, last name, location, occupation, email and password. It validates the data and creates a new user entry in the database. Here, API handles the register function in the auth.js file in under the controllers folder in the server folder. Also, if the registration process is success, the user is added to the database.

- API : http://localhost:3001/auth/register

- Endpoint: '/auth/register'
- Method: POST

This can be shown in postman as follows.



POST http://localhost:3001/auth/register

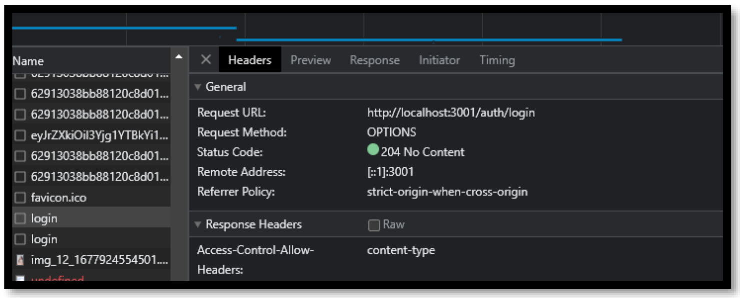
	Key	Value	Description
✓	occupation	Doctor	
✓	picturePath	crosscanada.png	
✓	email	tharushi123@gmail.com	
✓	password	123tharu123	

201 Created 673 ms 1.27 KB

```
{
    "firstName": "Tharushi",
    "lastName": "Rathnayaka",
    "email": "tharushi123@gmail.com",
    "password": "$2b$10$woMzqZX5a6N6gd5D5P10VuVXaM.kOR3mexvc2WE5ItBQHNYJ3zSWu",
    "picturePath": "",
    "friends": [],
    "location": "Negambo",
    "occupation": "Doctor",
    "viewedProfile": 2844
```

Under the routes, there is another API that handles the other authentication related to routes such as login and logout.

- API: 'http://localhost:3001/auth/login'
- Endpoint: 'auth/login'
- Method: OPTIONS



```
62913038bb88120c8d01...
62913038bb88120c8d01...
eyJrZXkiOil3Yjg1YTBkYi1...
62913038bb88120c8d01...
62913038bb88120c8d01...
favicon.ico
login
login
img_12_1677924554501....
undefined
```

General

Request URL:	http://localhost:3001/auth/login
Request Method:	OPTIONS
Status Code:	● 204 No Content
Remote Address:	[::1]:3001
Referrer Policy:	strict-origin-when-cross-origin

Response Headers ☐ Raw

Access-Control-Allow-Headers:	content-type

02. User Operation API

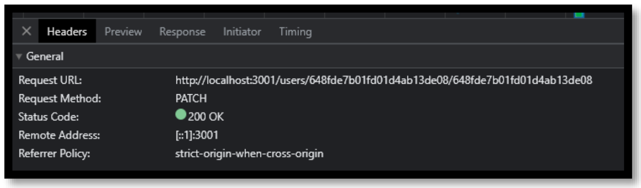
Here, the user endpoint retrieves user information. This API fetch all the details about all registered users in this application.

- Endpoint: '/users'

- Method: POST

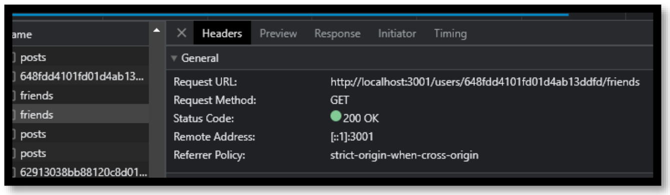
Here also, under routes, I have created API related to the user operations. The API is relates to the user operations such as retrieving information, and adding or removing friends.

- API: ‘http://localhost:3001/users/${_Id}/${friendId}’
- Endpoint: ‘/users/${userId}’
- Method: GET



```
Request URL:      http://localhost:3001/users/648fde7b01fd01d4ab13de08/648fde7b01fd01d4ab13de08
Request Method:   PATCH
Status Code:      ● 200 OK
Remote Address:   [::1]:3001
Referrer Policy:  strict-origin-when-cross-origin
```

- API: ‘http://localhost:3001/users/${userId}/friends’
- Endpoint: ‘/friends’
- Method: GET



```
Request URL:      http://localhost:3001/users/648fdd4101fd01d4ab13ddfd/friends
Request Method:   GET
Status Code:      ● 200 OK
Remote Address:   [::1]:3001
Referrer Policy:  strict-origin-when-cross-origin
```

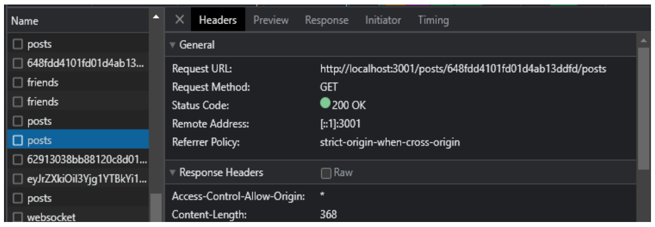
03. Post-management API

In this API, the registered are allowed to create a new post. Here, there is an authentication part. Mainly, when creating this post, the token should be there. As well as post cannot be published without a certain image file and the post content. Those three should be included in the request body. API calls the createPost function which has been created in posts.js file under the controllers folder in server folder.

- API: ‘http://localhost:3001/posts’
- Endpoint: ‘/posts’
- Method: POST

Also, another API is used to get the post of the selected user. Here, here posts are taken under a given user Id.

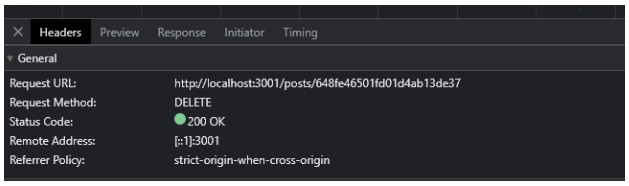
- API: ‘http://localhost:3001/posts/${userId}/posts’
- Endpoint: ‘/posts’
- Method: GET



```
Name                              Headers  Preview  Response  Initiator  Timing
posts                             ▾ General
648fdd4101fd01d4ab13...           Request URL:       http://localhost:3001/posts/648fdd4101fd01d4ab13ddfd/posts
friends                           Request Method:    GET
friends                           Status Code:       ● 200 OK
posts                             Remote Address:    [::1]:3001
posts                             Referrer Policy:   strict-origin-when-cross-origin
62913038bb88120c8d01...           ▾ Response Headers  Raw
eyJrZXkiOil3Yjg1YTBkYi1...        Access-Control-Allow-Origin:  *
posts                             Content-Length:    368
websocket
```

If a user wants to delete a post that was posted before, it can be done using another API. Here, I get the post by user Id and delete the post.

- API: ‘http://localhost:3001/posts/${userId}’
- Endpoint: ‘/posts/${userId}’
- Method: DELETE

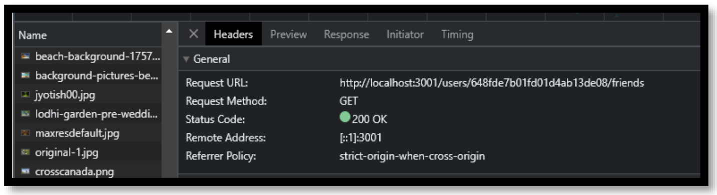


```
Headers  Preview  Response  Initiator  Timing
▾ General
Request URL:       http://localhost:3001/posts/648fe46501fd01d4ab13de37
Request Method:    DELETE
Status Code:       ● 200 OK
Remote Address:    [::1]:3001
Referrer Policy:   strict-origin-when-cross-origin
```

04. User Friends API

This API has been created to have the friend list under each user. Same as the post-creating API, this API requires an authentication with a token and the id if the user. API fetches the friends of a specific user from the database.

- Endpoint: ‘/users/:id/friends’
- Method: GET



```
Name                              Headers  Preview  Response  Initiator  Timing
beach-background-1757...          ▾ General
background-pictures-be...         Request URL:       http://localhost:3001/users/648fde7b01fd01d4ab13de08/friends
jyotish00.jpg                     Request Method:    GET
lodhi-garden-pre-weddi...         Status Code:       ● 200 OK
maxresdefault.jpg                 Remote Address:    [::1]:3001
original-1.jpg                    Referrer Policy:   strict-origin-when-cross-origin
crosscanada.png
```

Summary

In conclusion, Sociopedia is a social media application that prioritizes user privacy, security and meaningful connections. It offers a range of features including user authentication, profile management, friend connection and post creation.

The implemented APIs and endpoints ensure the communication between client application and the server while providing smooth operations. The system report provides a detailed explanation of the app’s functionalities, database structure and API implementations to give a comprehensive understanding of the Sociopedia system.

Moreover, the application has been implemented using MERN stack which means using the Mongo dB as the database, express.js, node.js and React framework. Apart from the above technologies, react router and the redux have been used for navigation and state management. Also, JSON web token has been used for the authentication process in the system.

However, the system provides a user-friendly graphical user interface and interactive and flexible platform while using the “Sociopedia” social media application.