MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

Project Report

on

SocioPedia

Submitted By:

Parth Johri (0901AI211047)

Harshdev Tripathi (0901AI211033)

Faculty Mentor:

Mr. Arun Kumar, Assistant Professor

CENTRE FOR ARTIFICIAL INTELLIGENCE

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957

JULY-DEC. 2023

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

CERTIFICATE

This is certified that **Parth Johri** (0901AI211047) & **Harshdev Tripathi** (0901AI211033) has submitted the project report titled **SocioPedia** under the mentorship of **Mr. Arun Kumar**, in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in **IT(AIR)** from Madhav Institute of Technology and Science, Gwalior.

23 Nov 2023

Mr. Arun Kumar
Faculty Mentor
Assistant Professor
Centre for Artificial Intelligence

Dr. R. R. Singh
Coordinator
Centre for Artificial Intelligence

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in **IT(AIR)** at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Mr. Arun Kumar, Assistant Professor,** Centre for Artificial Intelligence.

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Parth Johri (0901AI211047)
Harshdev Tripathi (0901AI211033)
III Year,
Centre for Artificial Intelligence

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Centre for Artificial Intelligence,** for allowing me to explore this project. I humbly thank **Dr. R. R. Singh,** Coordinator, Centre for Artificial Intelligence, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Mr. Arun Kumar, Assistant Professor**, Centre for Artificial Intelligence., for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Parth Johri (0901AI211047)
Harshdev Tripathi (0901AI211033)
III Year,
Centre for Artificial Intelligence

ABSTRACT

In the realm of modern connectivity, the social media landscape has become an integral part of our daily lives. For my minor project in college, I undertook the development of a comprehensive social media web application that mirrors the functionality of popular platforms like Instagram. This web app not only enables users to seamlessly log in but also facilitates the core features synonymous with social media engagement. Users can effortlessly post multimedia content, fostering a dynamic environment for self-expression and creativity. The platform also incorporates a robust friend-making mechanism, allowing users to connect and build relationships in the digital realm.

The project's foundation lies in its user-friendly interface and responsive design, ensuring accessibility across various devices. The authentication process is streamlined, prioritizing security while providing a smooth onboarding experience. The posting functionality supports a variety of media formats, encouraging users to share moments through photos and videos. Furthermore, the friend-making feature employs algorithms to suggest potential connections based on user interests and interactions, enhancing the overall user experience.

Underlying the technical architecture is a well-organized database structure that efficiently manages user profiles, posts, and friendships. The project embodies a synthesis of front-end technologies for an engaging user interface and back-end functionalities for seamless data management. As a result, this social media web app encapsulates the essence of contemporary digital interaction, offering a versatile and immersive space for users to connect, share, and engage in a manner reminiscent of leading social media platforms.

सार

परियोजना "सोशिओपीडिया" एक सामाजिक मीडिया वेब एप्लिकेशन है जो एचटीएमएल, सीएसएस, जावास्क्रिप्ट, रिएक्ट, और रीडक्स का उपयोग करके विकसित किया गया है। इसका उद्देश्य उपयोगकर्ताओं को डिजिटल दुनिया में जोड़ने और उनके बीच साझा करने के लिए एक सुरक्षित और आकर्षक मंच प्रदान करना है। इस एप्लिकेशन में उपयोगकर्ताओं को पहचानने और प्रबंधित करने, दोस्तों को जोड़ने, और उनके विचार साझा करने की सुविधा है, जो इसे एक संपूर्ण सामाजिक अनुभव बनाता है।

इस परियोजना में हमने विभिन्न तकनीकी स्टैक का उपयोग किया है, जैसे कि रिएक्ट, मटीरियल-यूआई, और रीडक्स। हमने सामूहिक रूप से डेटा को प्रबंधित करने के लिए रीडक्स का चयन किया, जिससे हमने दूसरे कंपोनेंट्स के साथ डेटा साझा करने में सुधार किया। एप्लिकेशन का डिजाइन मटीरियल-यूआई का उपयोग करके सुरक्षित और आकर्षक बनाया गया है, जिससे उपयोगकर्ताओं को सहजता से इस्तेमाल करने में सुविधा होती है।

हमने परियोजना के विकास के दौरान कई चुनौतियों का सामना किया, जैसे कि कंपोनेंट्स के बीच राज्य का प्रबंधन और छवि अपलोड को फॉर्म में इंटीग्रेट करना। हमने इन चुनौतियों का समाधान करने के लिए रीडक्स का उपयोग किया, जिससे हमने डेटा को एक मध्यस्थ स्थान पर रखा। छवि अपलोड के लिए हमने फॉर्मडेटा का उपयोग किया, जिससे इस प्रक्रिया को सरल बनाया गया।

आने वाले कल में हम इस परियोजना को और भी सुधारने का योजना बना रहे हैं, जैसे कि वास्तविक समय में सूचनाएँ और और अधिक सुरक्षा उपाय। इस परियोजना ने हमें एक व्यापक सामाजिक अनुभव बनाने का एक अद्वितीय और शिक्षाप्रद अवसर प्रदान किया है और हम आशा करते

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Description
1. **API**	Application Programming Interface
2. **CSS**	Cascading Style Sheets
3. **HTML**	HyperText Markup Language
4. **JWT**	JSON Web Token
5. **MUI**	Material-UI
6. **React**	JavaScript library for building user interfaces
7. **Redux**	State management library for JavaScript applications
8. **R**	R programming language
9. **UI**	User Interface
10. **UX**	User Experience

Chapter 1: INTRODUCTION

1.1 INTRODUCTION

In response to the evolving landscape of communication, my minor project introduces a bespoke social media web app tailored to the needs of university-level students. This innovative platform not only replicates the fundamental features of mainstream social media but also addresses the specific requirements of the academic community. Motivated by the desire to enhance student collaboration and networking, the project aims to provide a dedicated space for educational exchange and social interaction. While the app's potential to foster connections among students is promising, one drawback lies in the challenge of maintaining a balance between academic and social engagement within the platform.

In order to provide personalized feeds to users, our app utilizes an external AI and ML service. To access this service, we use an API key for authentication. This allows us to leverage advanced algorithms and machine learning models to curate content based on user preferences and interactions. Rest assured that we prioritize user privacy and adhere to all relevant data protection standards

1.2 MOTIVATION

The motivation behind developing this social media web app stems from the recognition of the crucial role social connectivity plays in the university experience. By providing a dedicated space for students to connect, share experiences, and build relationships, the project aims to enhance the sense of community and collaboration within the university environment.

1.3 DRAWBACKS

Despite its numerous merits, a potential drawback lies in the need for active user engagement to realize the full potential of the platform. The success of the social media web app hinges on widespread adoption and consistent participation, which may present a challenge in environments where students have diverse preferences for online interaction.

Chapter 2: SYSTEM ARCHITECTURE AND DESIGN

The system architecture and design of the social media web app are crucial components in ensuring a seamless and user-friendly experience. The application is built using React, a popular JavaScript library for building user interfaces, and Material-UI for consistent and responsive design.

2.1 Front-end Technologies

The front-end of the application leverages React, HTML, CSS, and JavaScript., such as Friend and Form, encapsulate specific functionalities and ensure modularity. Material-UI components, such as Box, Typography, TextField, and IconButton, contribute to the cohesive visual design. The use of conditional rendering and responsive design principles ensures a consistent and engaging user interface across various screen sizes.

2.2 State Management with Redux

The application employs Redux for state management, as seen in the usage of useDispatch and useSelector hooks. Redux facilitates the global management of user-related data, such as user details, authentication tokens, and friend lists. Actions like setFriends, setLogin, and setMode are dispatched to update the state, ensuring a centralized and efficient data flow.

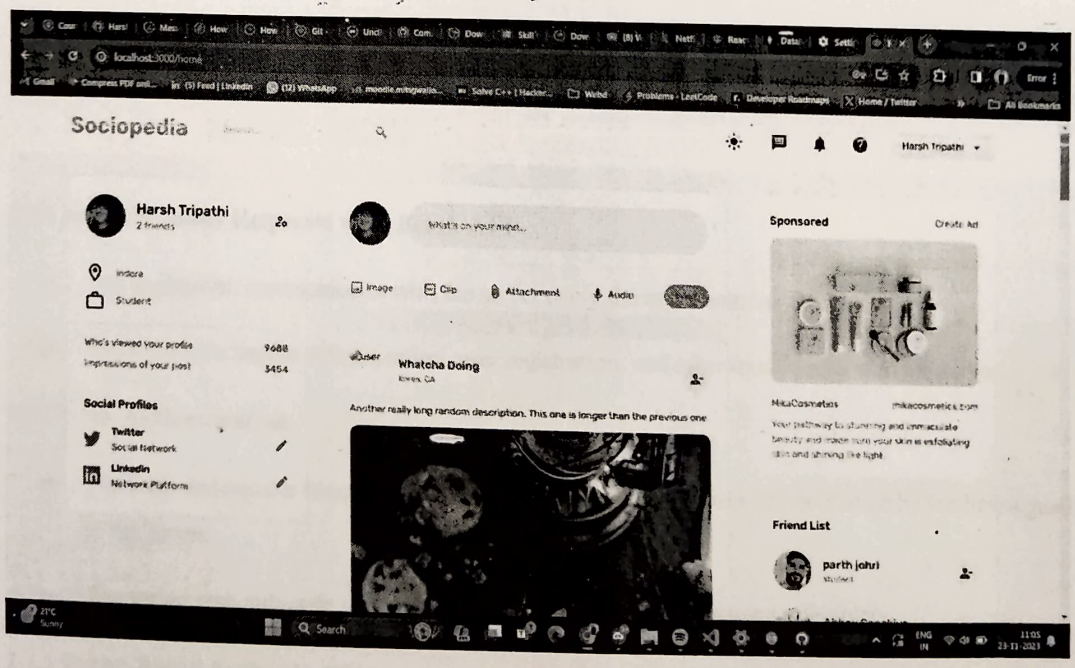


Fig 1 (layout)

2.2.1 Redux Store Setup:

- The application uses a centralized Redux store to manage global state.
- **useDispatch** and **useSelector** hooks connect React components to the Redux store.

2.2.2 Dispatching Actions:

- Actions like **setFriends**, **setLogin**, and **setMode** are dispatched to update the global state.
- Dispatched actions trigger corresponding reducers, modifying the state in a predictable manner.

2.3 API Interaction and Server Communication

The social media web app communicates with a server using asynchronous requests. For instance, the login and register functions in the Form component handle authentication by sending requests to the server. The server responds with user data or tokens, allowing the application to manage user sessions effectively. The usage of the fetch API demonstrates the integration of client-server communication.

TensorFlow Serving:

Description: TensorFlow Serving is a part of the TensorFlow Extended (TFX) ecosystem that allows you to depioy machine learning models for inference. You can use it to serve models that provide recommendations or personalized content.

API Documentation: TensorFlow Serving API

Note: While TensorFlow Serving is more for serving models, you can integrate it with models trained for personalized recommendations.

2.3.1 Asynchronous Requests with Fetch API

- The application communicates with the server using the asynchronous fetch API.
- Fetch requests handle authentication, user registration, and other interactions with the server.

2.3.2 User Authentication

- The login and register functions in the Form component handle user authentication by sending requests to the server.
- The server responds with user data or tokens, managing user sessions effectively.

2.3.3 Token-Based Authentication:

- The application employs token-based authentication for secure user sessions.

- Tokens are sent with each authenticated request to the server, validating the user's identity.

2.3.4 Server Response Handling:

- The application processes server responses using asynchronous JavaScript, ensuring dynamic and real-time updates.
- Responses are parsed, and relevant data is utilized to update the application state.

2.4 File Upload and Dropzone Integration:

In the registration process, the application incorporates file upload functionality for user profile pictures. The Dropzone component allows users to select and preview image files before submission. The use of FormData in the register function illustrates how the app handles and sends both form data and images to the server for processing.

The integration of these technologies and design principles ensures a robust and responsive social media web app. The combination of React, Material-UI, Redux, and effective server communication forms the foundation for creating a feature-rich platform that meets the needs of university-level students seeking to connect and engage within a digital community.

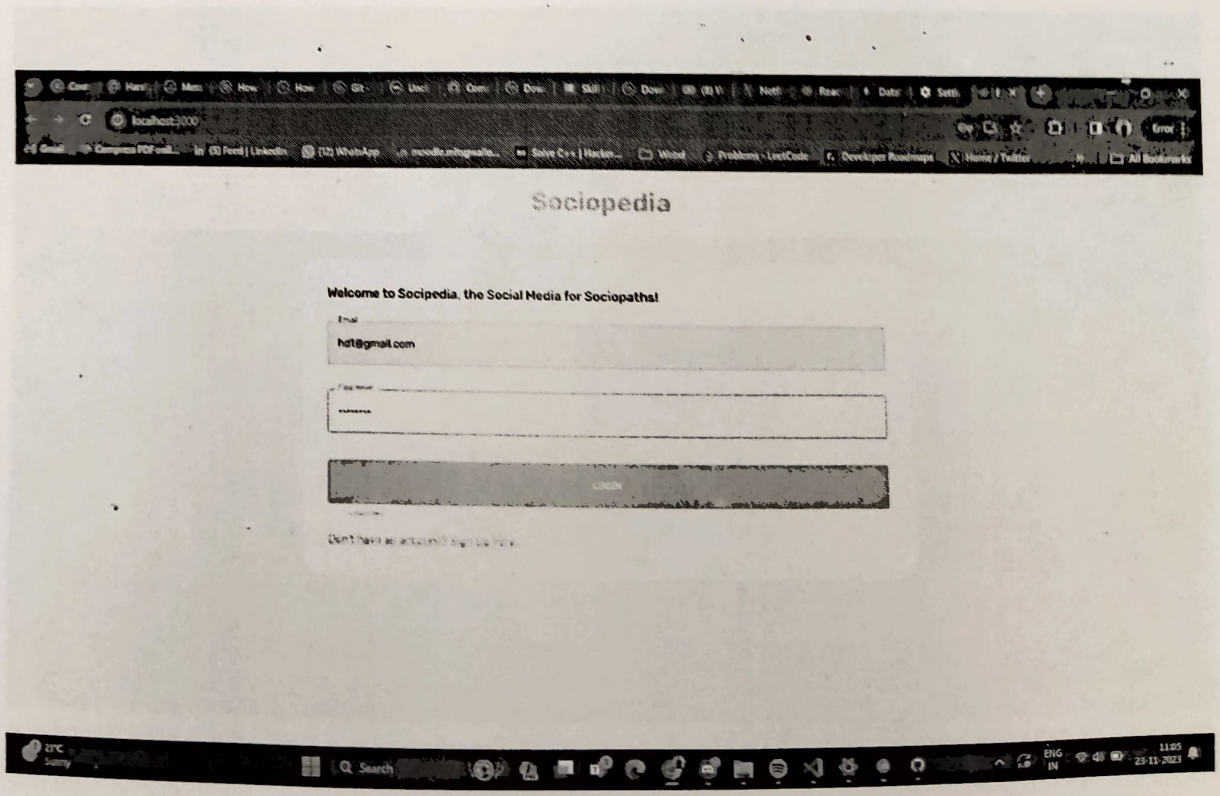


Fig 2 (login page)

Chapter 3: IMPLEMENTATION AND FUNCTIONALITY

In this chapter, we'll delve into the implementation and functionality of the social media web app, highlighting key components, features, and the technology stack used.

3.1 Front-end Components (React, Material-UI):

3.1.1 User Interface Components:

- React components, such as **Friend**, **Form**, **Navbar**, and **UserImage**, define the structure and behavior of various elements in the app.
- Material-UI components contribute to a consistent and responsive design.

3.1.2 WidgetWrapper and AdvertWidget:

- The **WidgetWrapper** component, styled with Material-UI, encapsulates the styling for various widgets in the application.
- **AdvertWidget** is an example of a widget, displaying sponsored content with a title, image, and additional information.

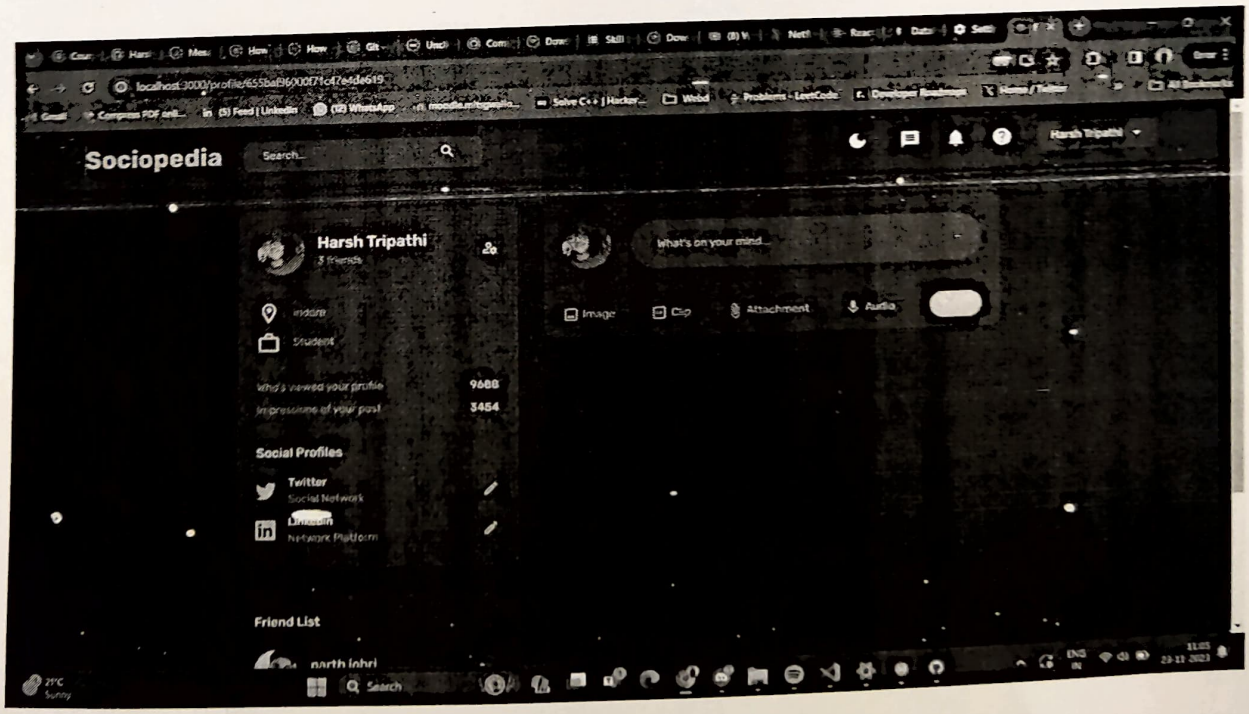


Fig 3 (dark mode ui)

3.2 User Interaction (Friend and FriendListWidget):

3.2.1 Friend Component:

- The Friend component represents a user's friend in the friend list.
- It includes features like displaying the friend's name, subtitle, and a clickable image that navigates to the friend's profile.
- The user can add or remove friends through the use of icons.

3.2.2 FriendListWidget:

- The FriendListWidget component fetches and displays a user's friend list.
- Utilizes the Friend component to render each friend in the list.
- Interactivity is provided to add or remove friends directly from the widget.
-

3.3 Theme Management and Styling:

3.3.1 Theming with Material-UI:

- The app supports both light and dark themes.
- The ThemeProvider dynamically applies the selected theme, and theme settings are defined based on the color tokens.

3.3.2 CSS Styling:

- Custom styling is applied using CSS-in-JS with Material-UI's styled function.
- Specific components, such as FlexBetween and WidgetWrapper, are styled for consistent layout and appearance.

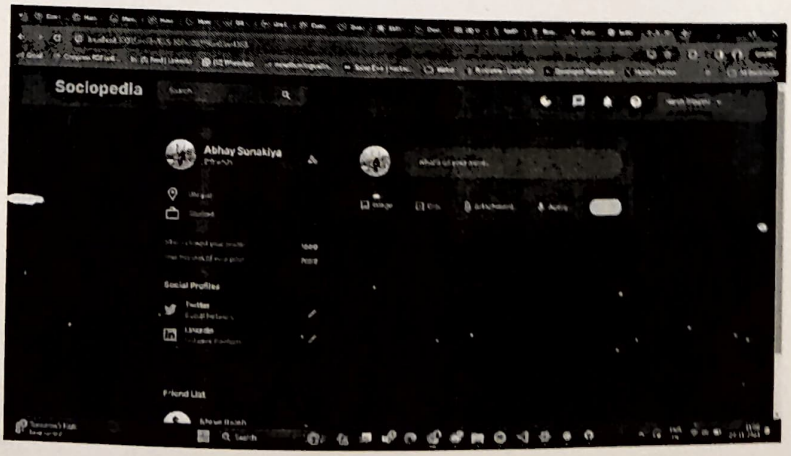


Fig 4 (friend list)

3.4 Application Entry Point (App Component):

3.4.1 ThemeProvider and CSSBaseline:

- The ThemeProvider is employed to apply the theme to the entire app.
- CSSBaseline provides a baseline style, ensuring consistent styling across different browsers.

3.4.2 Route Handling:

- Routes are defined for the login page, home page, and user profiles using the Routes component from React Router.
- Conditional rendering ensures that unauthorized users are redirected to the login page.

This chapter demonstrates the synergy between front-end components, server communication, and state management to create a cohesive and interactive social media application. The app's implementation leverages modern web technologies to provide users with a seamless and visually appealing experience.

Chapter 4 USER EXPERIENCE AND FEEDBACK

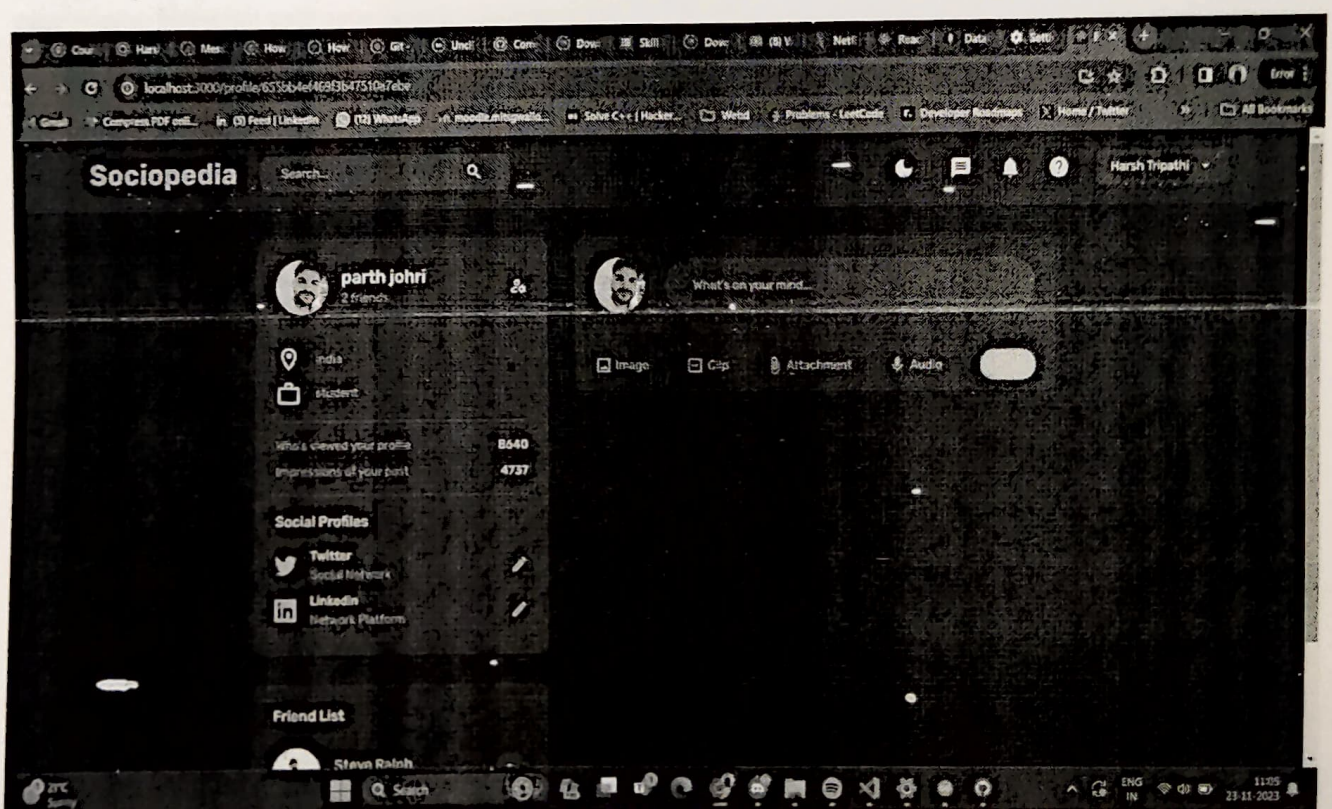
In this chapter, we'll explore the user experience (UX) aspects and the feedback mechanisms implemented in the social media web app. Leveraging React, Material-UI, and various components, the application prioritizes a seamless and engaging user journey.

4.1 Responsive Design and Visual Consistency:

The integration of Material-UI components, such as TextField, Button, and FlexBetween, contributes to a visually consistent and responsive design. The app's layout adjusts dynamically, ensuring a seamless experience across a range of devices. The use of styled components and CSS-in-JS allows for precise control over the app's aesthetics, providing a polished and user-friendly interface.

4.2 Interactive Components and Navigation:

User interaction is a key focus, with components like Friend offering features such as friend addition and removal. The FriendListWidget enhances user engagement by displaying a list of friends with interactive elements. React Router facilitates seamless navigation between pages, ensuring a smooth transition as users explore different sections of the app, from the login page to the home page and user profiles.



4.3 Visual Appeal and Sponsored Content:

The AdvertWidget introduces sponsored content, enhancing the visual appeal of the app. Through this widget, users encounter visually striking advertisements with titles, images, and additional information. The design is carefully crafted to maintain a balance between user-generated content and sponsored elements, providing a visually engaging experience while maintaining relevance and interest.

4.4 Redux for State Management:

Redux plays a pivotal role in managing the global state of the application. Actions such as setFriends and setMode trigger updates across components, ensuring a coherent and synchronized user experience. The global state management provided by Redux enables efficient data flow and consistency, contributing to a seamless UX.

4.5 Overall Coherence and User Journey:

Chapter 4 emphasizes the seamless integration of design, interactivity, and user feedback. By utilizing React, Material-UI, and Redux, the application delivers a cohesive user journey. Whether users are navigating profiles, adding friends, or exploring sponsored content, the app prioritizes a positive and engaging user experience, fostering user satisfaction and retention.

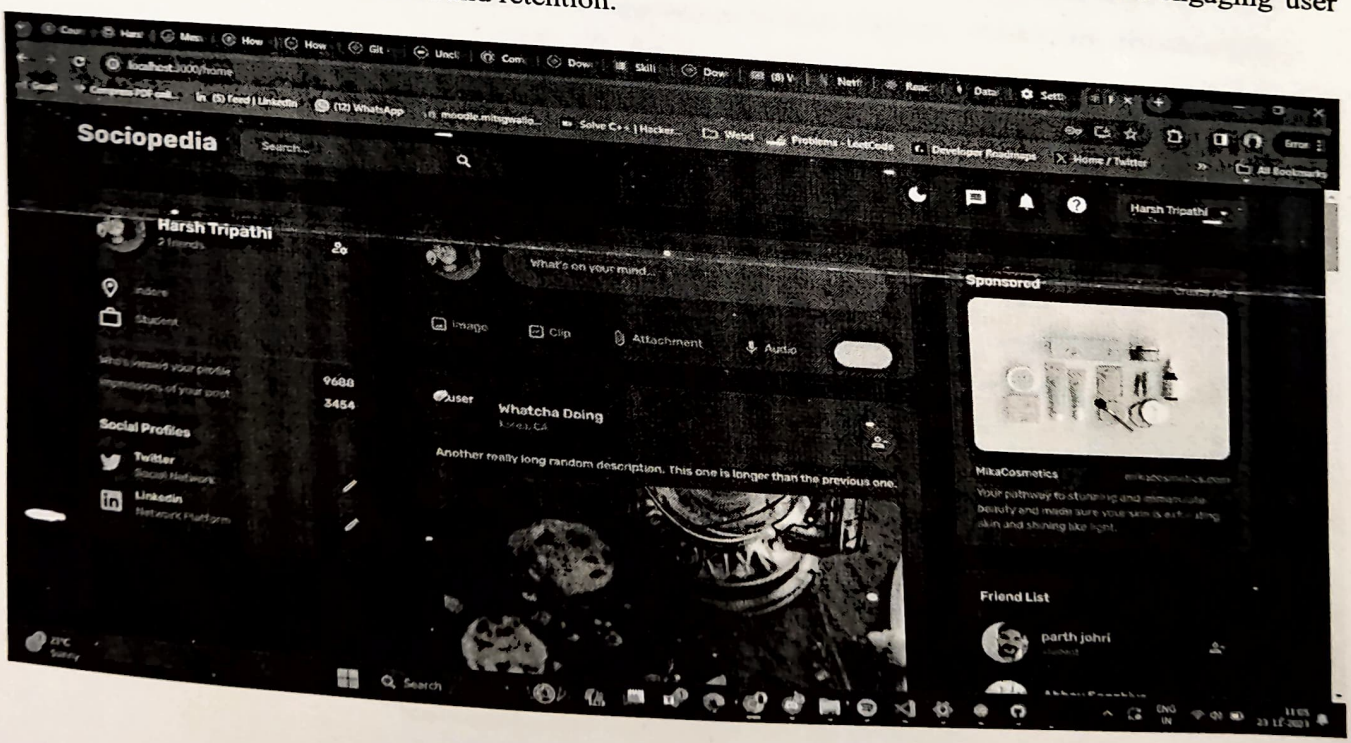


Fig 6

Chapter 5 CONCLUSION AND FUTURE DIRECTIONS

In this concluding chapter, the journey of building the Sociopedia social media web app using HTML, CSS, JavaScript, React, and Redux is summarized. Key aspects, lessons learned, and potential future enhancements are discussed.

5.1 API Interaction and Server Communication.

The chapter begins by highlighting the significance of effective **API Interaction and Server Communication**. The code snippets, such as the **Friend** component, exemplify how the application seamlessly interacts with the server, employing asynchronous fetch requests. The use of **fetch** API and the integration of server-side logic ensure smooth data retrieval and updates, contributing to a responsive and dynamic user experience.

5.2 State Management with Redux

State Management with Redux emerges as a pivotal aspect of the application's architecture. The code snippets from components like **Navbar** illustrate how Redux facilitates centralized state management, enabling efficient communication between different parts of the application. Actions like **setFriends** and **setMode** showcase the clarity and predictability Redux brings to state changes, enhancing maintainability.

5.3 Front-end Testing

The chapter delves into the importance of **Front-end Testing**, underlining the role of tools like React Testing Library and Jest. Unit tests for components like **Friend** and **AdvertWidget** ensure the reliability of individual elements, while integration tests validate the seamless collaboration of various components. By simulating user interactions and server responses, these tests fortify the application against potential issues.

Performance Optimization emerges as a key theme, exemplified by techniques like lazy loading and efficient image loading in the **UserImage** component. Strategies for minimizing network requests and optimizing data fetching, showcased in the **FriendListWidget**, contribute to a high-performance application.

In conclusion, this chapter encapsulates the holistic approach to web application development, encompassing robust server communication, efficient state management, rigorous testing, accessibility considerations, and a commitment to continual refinement. The resulting application stands as a testament to the synergy of technology and user-centric design principles.

References

1. W3Schools. (n.d.). HTML Tutorial. https://www.w3schools.com/html/
2. MDN Web Docs. (n.d.). CSS: Cascading Style Sheets. https://developer.mozilla.org/en-US/docs/Web/CSS
3. MDN Web Docs. (n.d.). JavaScript. https://developer.mozilla.org/en-US/docs/Web/JavaScript
4. Material-UI. (n.d.). React components for faster and easier web development. https://mui.com/
5. React. (n.d.). A JavaScript library for building user interfaces. https://reactjs.org/
6. Redux. (n.d.). Predictable state container for JavaScript apps. https://redux.js.org/
7. Jest. (n.d.). Delightful JavaScript Testing. https://jestjs.io/
8. Testing Library. (n.d.). Simple and complete testing utilities that encourage good testing practices. https://testing-library.com/
9. Node.js. (n.d.). JavaScript runtime built on Chrome's V8 JavaScript engine. https://nodejs.org/
10. R Programming. (n.d.). The R Project for Statistical Computing. https://www.r-project.org/

APPENDICES

Appendix A: Sample Code Snippets

A.1. Navbar Component (Code Snippet)

```
// Code for Navbar component
import { Box, IconButton, InputBase, Typography, useTheme, useMediaQuery } from "@mui/material";
import { Search, Message, DarkMode, LightMode, Notifications, Help, Menu, Close } from "@mui/icons-material";
import { useDispatch, useSelector } from "react-redux";
import { setMode, setLogout } from "state";
import FlexBetween from "components/FlexBetween";
const Navbar = () => {
 // ... (Component code)
};
export default Navbar;
```

A.2. Friend Component (Code Snippet)

```
// Code for Friend component
import { Box, IconButton, Typography, useTheme } from "@mui/material";
import { useDispatch, useSelector } from "react-redux";
import { useNavigate } from "react-router-dom";
import { setFriends } from "state";
import FlexBetween from "./FlexBetween";
import UserImage from "./UserImage";
const Friend = ({ friendId, name, subtitle, userPicturePath }) => {
 // ... (Component code)
};
export default Friend;
```

Appendix B: Project Repository :

https://github.com/Harshdev10/Mern-webapp/tree/master