

# **SOCIAL MEDIA APPLICATION**

**A PROJECT REPORT**

*Submitted by*

**GAYATHIRI B (920422205029)**

**MONICA T (920422205062)**

**PON KARTHIGA A (920422205076)**

*in partial fulfillment for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**INFORMATION TECHNOLOGY**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**KAMARAJ COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(An Autonomous Institution - Affiliated to Anna University, Chennai)**

**K.VELLAKULAM, VIRUDHUNAGAR - 625 701**

**NOVEMBER 2024**

**KAMARAJ COLLEGE OF ENGINEERING AND TECHNOLOGY**  
**(An Autonomous Institution- Affiliated to Anna University, Chennai)**

**K.VELLAKULAM, VIRUDHUNAGAR - 625 701**

**BONAFIDE CERTIFICATE**

Certified that the project report “**SOCIAL MEDIA APPLICATION**” is the bonafide work of “**GAYATHIRI B (920422205029), MONICA T (920422205062), and PON KARTHIGA A (920422205076)**” who carried out the project work under my supervision.

**SIGNATURE**

**Dr. E. VAKAIMALA**

**Head of the Department,**

Associate Professor,  
Dept. of Information Technology,  
Kamaraj College of Engg & Tech,  
K.Vellakulam,  
Virudhunagar - 625 701.

**SIGNATURE**

**Dr. R. ARTHY**

**Supervisor,**

Assistant Professor,  
Dept. of Information Technology,  
Kamaraj College of Engg & Tech,  
K.Vellakulam,  
Virudhunagar - 625701.

## ABSTRACT

The social media application, developed using the MEAN stack (MongoDB, Express.js, Angular, and Node.js), offers users an interactive platform to share and engage with content. MongoDB serves as the database, securely storing user details, posts, comments, and interactions, ensuring scalable data management. Express.js acts as the backend framework, efficiently handling API routes and server-side logic to process requests from the front end. Angular powers the front end, providing a dynamic and responsive user interface where users can view reels, movies, and songs, post comments, and engage in real-time messaging. Node.js operates as the server environment, managing communication between the client and the database, ensuring smooth data handling and real-time updates. Users can sign up, log in, manage their profiles, and interact with content by posting comments, all while their data is securely stored in MongoDB. This platform focuses on a seamless user experience, allowing for social interaction and content sharing with real-time communication, robust user authentication, and secure data management, making it an efficient and engaging social media solution.

## **ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to Dr. E. Vakaimalar, HOD of the Department of Information Technology, and Dr. R. Arthy, my supervisor, for their invaluable guidance and support throughout the development of the SOCIAL MEDIA APP project using the MEAN stack. Their insights and expertise were instrumental in helping me understand the practical implementation of web technologies, enhancing both my technical skills and problem-solving abilities. This project has provided me with hands-on experience in full-stack development, which will be immensely helpful in future endeavors.

## TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	<b>ABSTRACT</b>	iii
	<b>LIST OF FIGURES</b>	vi
		Error!
<b>1</b>	Error! Reference source not found.	Bookmark not defined.
	1.1 HTML	1
	1.2 CSS	2
	1.3 JavaScript	3
	1.4 MEAN Stack	4
<b>2</b>	<b>METHODOLOGY</b>	<b>8</b>
	2.1 Objective	<b>8</b>
	2.2 Problem Statement	<b>8</b>
	2.3 Block Diagram	<b>9</b>
	2.4 Module Explanation	<b>10</b>
<b>3</b>	<b>RESULTS AND DISCUSSION</b>	<b>12</b>
<b>4</b>	<b>CONCLUSION</b>	<b>24</b>
<b>5</b>	<b>REFERENCE</b>	<b>25</b>

## **LIST OF FIGURES**

<b>FIGURE NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
3.1	Sign In page	13
3.2	Sign In successfully	13
3.3	Database of Sign In person	14
3.4	Login Page	14
3.5	Login successful	15
3.6	Database of Login person	15
3.7	Home Page	16
3.8	Commenting the post	16
3.9	Database of Commented person with comment	17
3.10	Reels Page	17

3.11	Message Page	18
3.12	Message Page	18
3.13	Movies Page (Horror)	19
3.14	Movies Page (Horror)	19
3.15	Movies Page (comedy and Action)	20
3.16	Movies Page (Romance)	20
3.17	Songs Page Movies Page (Animation)	21
3.18	Songs Page	21
3.19	Songs Page	22
3.20	Profile Page	22
3.21	Profile Page	23

# CHAPTER 1

## INTRODUCTION

### 1.1 HTML (Hyper Text Markup Language)

HTML (Hyper Text Markup Language) serves as the foundational building block of web applications, establishing the structural framework that dictates how content is organized and displayed on web pages. It is instrumental in creating the skeletal structure of various application pages, including the home page, movies, songs, and user profile pages. By utilizing a series of elements and tags, HTML allows developers to define the hierarchy of content, incorporating text, images, links, and multimedia components in a coherent manner.

The primary role of HTML is to ensure that content is presented clearly and accessibly, facilitating user interaction with the application. This markup language employs a variety of elements, such as headings, paragraphs, lists, tables, and forms, to organize information effectively. For instance, headings (h1, h2, etc.) create a logical flow of content, while lists and tables help present data in an easily digestible format.

In addition to content organization, HTML plays a crucial role in enhancing the overall user experience. It acts as the canvas upon which other technologies, such as CSS (Cascading Style Sheets) and JavaScript, can be layered to enrich interactivity and visual design. While CSS manages the presentation and aesthetics of web pages, JavaScript adds dynamic functionality, allowing for interactive elements like sliders, modals, and form validations.

Moreover, HTML contributes to search engine optimization (SEO) by providing a structure that search engines can easily crawl and index. Proper use of semantic HTML elements, such as `<header>`, `<footer>`, `<article>`, and `<section>`, not only improves accessibility for users with disabilities but also enhances the discoverability of the content on search engines.



## 1.2 CSS (Cascading Style Sheets)

CSS (Cascading Style Sheets) is a stylesheet language used to style and present the HTML structure of web applications, ensuring that the application is visually appealing and responsive across various devices. By separating content from presentation, CSS allows developers to control the layout, colors, fonts, and overall aesthetic of the social media app, creating a polished and user-friendly interface.

Through CSS, developers can customize the appearance of individual elements, such as buttons, links, and images, enhancing the visual hierarchy and usability of the application. Styles can be applied globally or tailored to specific components, enabling consistency throughout the app while also allowing for flexibility in design. By utilizing a variety of selectors, properties, and values, CSS provides the tools necessary to create dynamic and attractive layouts.

Responsive design techniques are a crucial aspect of CSS, allowing the app to adapt seamlessly to different screen sizes and orientations. Using media queries, developers can specify different styles for various devices, ensuring that users have an optimal experience whether they access the app on a desktop, tablet, or mobile device. This approach enhances usability by adjusting layout structures, font sizes, and other visual elements to accommodate the viewing environment.

Additionally, CSS frameworks such as Bootstrap or Tailwind CSS can be employed to expedite the development process and maintain responsive design principles. These frameworks offer pre-built classes and components, enabling developers to implement responsive grids, navigation bars, and other UI elements quickly and efficiently.

### **1.3 JavaScript**

JavaScript is a powerful, high-level programming language that is essential for adding dynamic functionality to web applications. It enables user interactions, allowing for seamless engagement with various elements within the app, such as clicking buttons, toggling between pages, and updating content in real time without the need to refresh the page. In the context of this social media app, JavaScript is the driving force behind features that enhance interactivity and user experience.

One of the key functionalities powered by JavaScript is real-time messaging, which allows users to communicate instantly within the app. By utilizing technologies such as WebSockets or AJAX, JavaScript facilitates the continuous exchange of messages, ensuring that conversations are updated in real-time without any delays. This feature is critical for maintaining user engagement and fostering community interaction within the platform.

JavaScript also enables media playback capabilities, allowing users to play videos and audio files directly within the app. Through various libraries and APIs, developers can implement controls for media playback, including play, pause, and volume adjustments, creating a rich multimedia experience for users. This is particularly important for a social media app that may feature user-generated content or shared media.

Additionally, JavaScript allows for the dynamic loading of movies, songs, and other content. Using techniques like lazy loading and infinite scrolling, the app can retrieve and display new content as users navigate through the platform, reducing load times and improving performance. This functionality ensures that users can enjoy a smooth and uninterrupted experience while exploring various features of the app.

## 1.4 MEAN Stack

The MEAN stack is the backbone of the application, offering a full-stack JavaScript solution from the front end to the back end. It is comprised of four key components:

- **MongoDB:** A NoSQL database that stores user data, media content, and interactions in a flexible format. It allows for easy scaling and handling of large volumes of data in real-time applications like social media.
- **Express.js:** A server-side web framework that simplifies routing, API integration, and backend logic. It serves as the middleware connecting the frontend Angular framework with the backend MongoDB database.
- **Angular:** A frontend framework that provides a dynamic, single-page application (SPA) experience. Angular handles data binding, page transitions, and real-time updates for a smooth user experience.
- **Node.js:** A JavaScript runtime that powers the server-side logic of the application. Node.js handles requests from the frontend, manages data communication with MongoDB, and supports real-time messaging and notifications.

Together, these technologies create a responsive, scalable, and efficient social media platform that integrates media consumption, favorites management, and real-time communication in one cohesive system.

### **MongoDB (Database Layer)**

MongoDB is a NoSQL, document-oriented database that stores data in a flexible, JSON-like format. Unlike traditional relational databases, MongoDB does not rely on tables and schemas. Instead, it uses collections and documents, allowing for easier handling of large volumes of unstructured or semi-structured data.

## Key features of MongoDB

- **Schema-less Structure:** Offers flexibility in data storage without a fixed schema, allowing developers to store varying types of data in the same collection.
- **Scalability:** Supports horizontal scaling, making it suitable for applications with growing datasets.
- **JSON Data Representation:** Stores data in BSON, seamlessly integrating with front-end JavaScript applications.

For example, an e-commerce website using MongoDB could efficiently store user data, product catalogs, and order details, evolving over time without needing extensive database modifications.

## Express.js (Backend Layer)

Express.js is a lightweight web application framework that runs on Node.js. It simplifies the process of building server-side applications by providing robust features for managing routing, middleware, and HTTP requests.

## Key features of Express.js

- **Routing:** Simplifies routing of HTTP requests to specific application endpoints. Developers can define routes for various functionalities, such as user login and data retrieval.
- **Middleware Support:** Middleware functions help handle request and response objects efficiently, such as authenticating users before granting access to certain routes.
- **Integration with MongoDB:** Works seamlessly with MongoDB for efficient data retrieval and storage.

In a typical MERN stack application, Express acts as the intermediary layer, connecting the React front end with the MongoDB database and managing API requests.

### **Angular.js (Frontend Layer)**

Angular.js is a front-end framework in the MEAN stack used for building dynamic, single-page web applications (SPAs). It provides powerful tools for creating interactive and responsive user interfaces, making it easier to manage both large and small-scale applications.

#### **Key features of Angular.js**

- **Two-Way Data Binding:** Automatically synchronizes data between the user interface and the application's logic, making it easy to manage dynamic updates in real-time.
- **Component-Based Architecture:** Organizes the app into reusable, modular components, making development and maintenance more efficient.
- **Dependency Injection:** Simplifies how different components and services interact, leading to cleaner, more maintainable code.

### **Node.js (Server Layer)**

Node.js is a powerful JavaScript runtime environment that allows developers to write server-side code using JavaScript, enabling the development of fast, efficient, and scalable server-side applications. By using the same language for both client and server-side code, Node.js simplifies the development process and promotes consistency across the application stack. It is particularly suited for building real-time applications and APIs, making it a popular choice for modern web development.

## Key features of Node.js

- **Event-Driven Architecture:** Node.js employs a non-blocking, event-driven architecture, which means that it can efficiently handle multiple requests simultaneously. This approach is ideal for applications that require high input/output (I/O) operations, such as web servers, chat applications, and APIs, as it allows the server to respond to multiple client requests without causing delays.
- **Asynchronous Programming:** Node.js operates in an asynchronous environment, meaning I/O operations such as reading files, querying databases, or handling network requests occur in the background without blocking the main execution thread. This non-blocking nature enhances performance and responsiveness, ensuring that the server remains efficient even under heavy loads.
- **NPM (Node Package Manager):** One of Node.js's most significant advantages is its robust package ecosystem, known as NPM (Node Package Manager). NPM provides developers with access to thousands of reusable libraries and modules, which can be easily integrated into applications to add functionalities, speed up development, and reduce the need for custom code. These modules range from database connectors and authentication systems to web frameworks like Express.js.

Node.js plays a crucial role in building scalable and efficient server-side applications for the social media app, managing data exchanges between the client and server, handling API requests, and integrating database operations. Its event-driven, non-blocking capabilities make it an ideal choice for real-time features like messaging, notifications, and dynamic content updates, ensuring a seamless and responsive user experience.

## **CHAPTER 2**

### **METHODOLOGY**

#### **2.1 Objective**

The primary objective of this social media application is to create a user-friendly, interactive platform that seamlessly integrates various forms of entertainment, including movies, songs, and social communication. The app is designed to enable users to engage with media content, manage their favorites, and interact with other users through personalized profiles and real-time messaging. Using the MEAN stack, the platform offers a scalable and dynamic solution for media consumption and social interaction in one place.

#### **2.2 Problem Statement**

In today's digital era, users often need to switch between multiple applications to consume media, manage their preferences, and connect with others. This fragmentation leads to inefficiency, reduced user engagement, and a disjointed experience. The challenge is to create an all-in-one solution that integrates social interactions with personalized content management across various media formats (movies, songs, posts) while maintaining a simple and smooth user experience. The goal is to develop a platform that provides a consolidated space for media consumption, favorites management, profile handling, and real-time communication, all powered by the MEAN stack.

## 2.3 Block Diagram

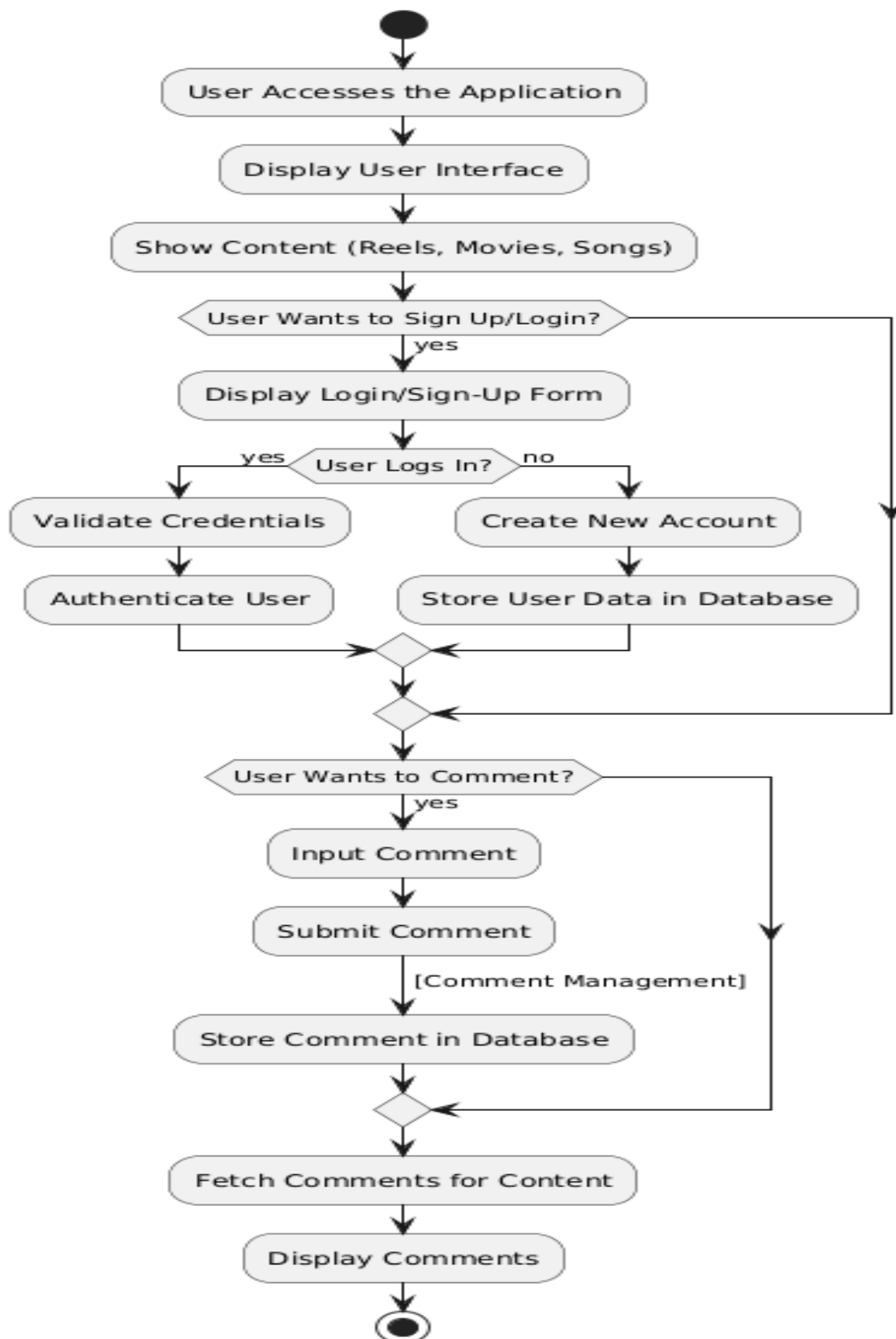


Figure 2.1 Process flow



## **2.4 MODULE EXPLANATION**

### **1. User Interface (Angular Front-End)**

- **Content Display:** This component presents various types of media, including reels, movies, and songs. Users can browse through the content, view key details such as titles, descriptions, and engagement metrics (likes/comments), and click to interact or comment on posts.
- **Comment Management:** This module allows users to add comments to posts. Users can input their thoughts in a text area, submit comments, and view existing comments in real-time. The interface is designed for ease of use, making engagement simple and intuitive.
- **User Authentication:** This module enables users to log in using their credentials or create a new account through a sign-up form. It includes validation for email and password fields to ensure secure access to the platform. Successful sign-ups and logins store user credentials securely in the database for future authentication.
- **Profile Management:** Users can view and manage their profiles, including personal information and settings, enhancing the user experience and allowing for personalized interactions.

### **2. Back-End (Content Management)**

- **Comment Handling:** This module manages the creation and retrieval of comments for posts. When users add a comment, the back-end server processes the request, stores the comment in the database, and links it to the corresponding post and user.
- **User Data Management:** This module processes user sign-up and login requests, securely storing user credentials and profile information in the MongoDB database for future authentication and profile management.

### **3. Back-End Logic (Express.js)**

- **API Management:** This module manages API routes for handling user requests, such as posting comments, user sign-up, and retrieving content. It validates incoming data from the front end and interacts with MongoDB to store or retrieve user and comment details.
- **Real-Time Communication:** Express.js facilitates real-time updates for comments, ensuring that users can see new comments and interactions without needing to refresh the page.

### **4. Server-Side Logic (Node.js)**

- **Handling Requests and Responses:** Node.js processes requests such as posting comments, user authentication, and content retrieval, managing API calls between the front end and back end effectively.
- **Interfacing with the Database:** Node.js interacts with MongoDB to securely store user information (e.g., sign-up, login details) and comment data, ensuring that all data can be accessed or updated as needed.
- **Comment Processing:** Node.js verifies user authentication before allowing comment submissions and ensures that data is correctly linked to the appropriate user and post.
- **Security and Validation:** It validates incoming data, such as ensuring required fields are filled during user authentication and comment submission, to maintain security across the platform.

### **5. Database (MongoDB)**

- **MongoDB** securely stores user information, login credentials, and post comments, ensuring organized and efficient data management. It allows easy retrieval for future engagement, enhancing user interactions and experience.

## **CHAPTER 3**

### **RESULTS AND DISCUSSION**

The MEAN stack-based social media application delivered a smooth and dynamic user experience, with Angular enabling seamless navigation across pages like Home, Favourites, Movies, Songs, and Profile. Its responsive design ensured compatibility across devices, enhancing user satisfaction.

Real-time messaging was powered by Node.js and WebSockets, allowing instant communication without page reloads, which boosted user engagement. MongoDB effectively managed flexible data storage, handling media files and message histories, while its non-relational structure ensured scalability. On the backend, Express.js handled API requests, routing, and secure authentication, contributing to app maintainability and data security.

Performance challenges, particularly with real-time messaging and content management, were mitigated through caching, database indexing, and server optimizations, ensuring smooth performance as data scaled. Looking forward, future enhancements will include machine learning for personalized media recommendations, improved real-time messaging scalability, and enhanced search functionalities.

Overall, the app successfully integrated media consumption and social interaction, delivering a responsive and scalable platform, with plans for advanced features and performance improvements as the user base grows.

## SCREENSHOTS

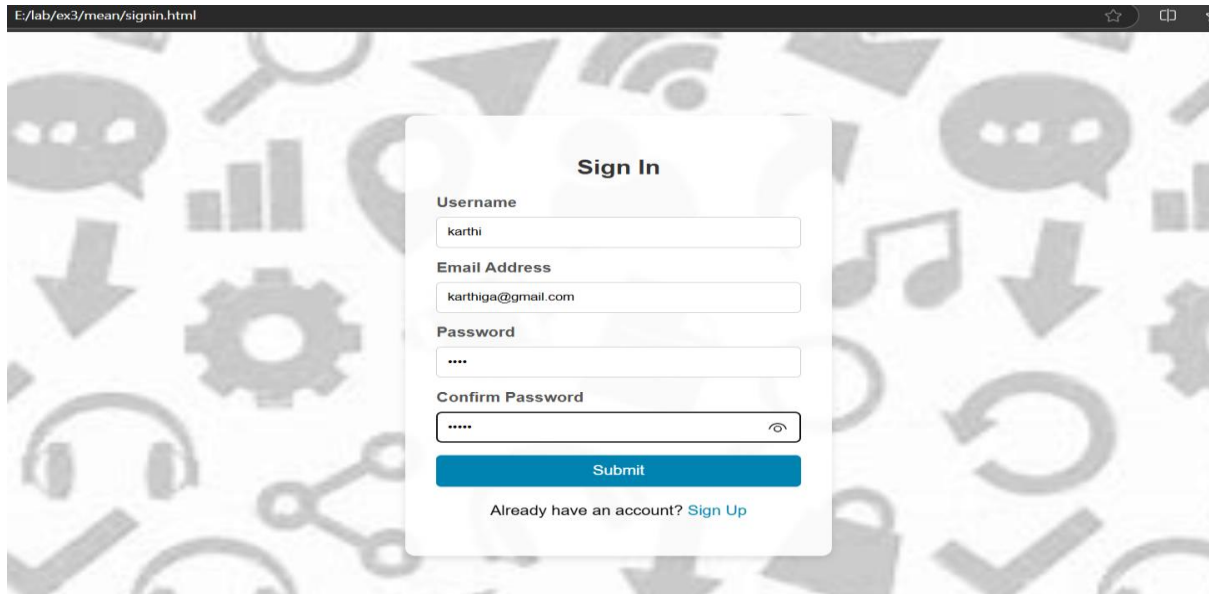


Figure 3.1 Sign In page

This figure 3.1 shows getting the details from the user.

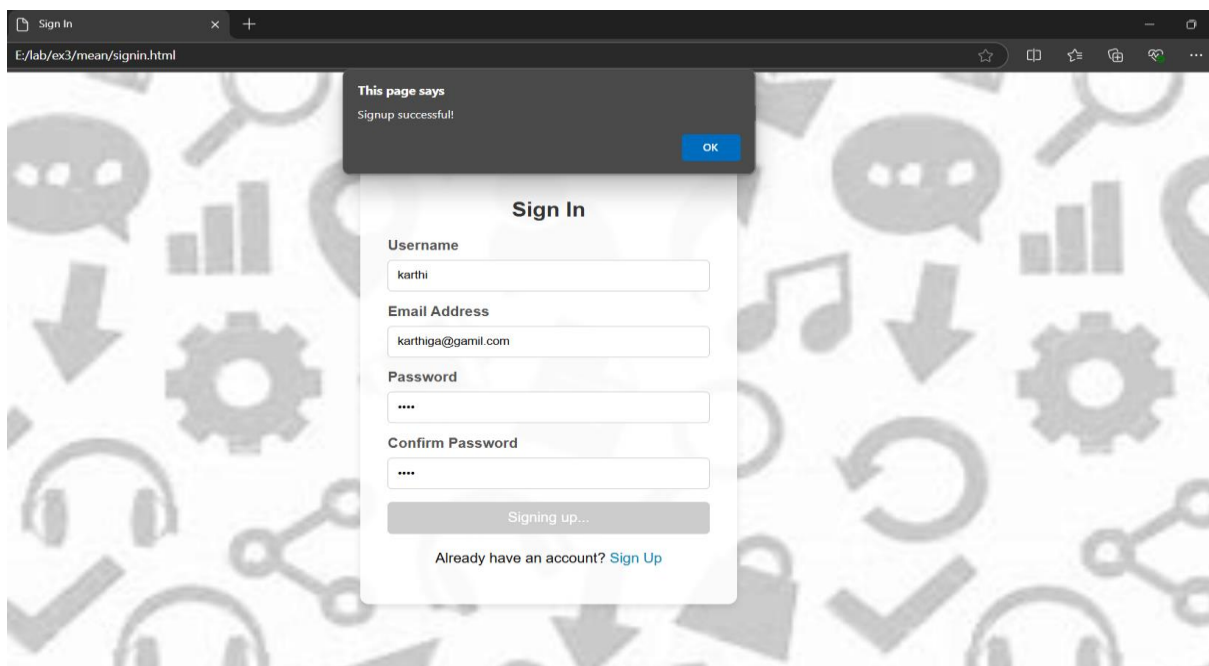


Figure 3.2 Sign In Successfully

This figure 3.2 shows allow user into the web page after the successful sign up.

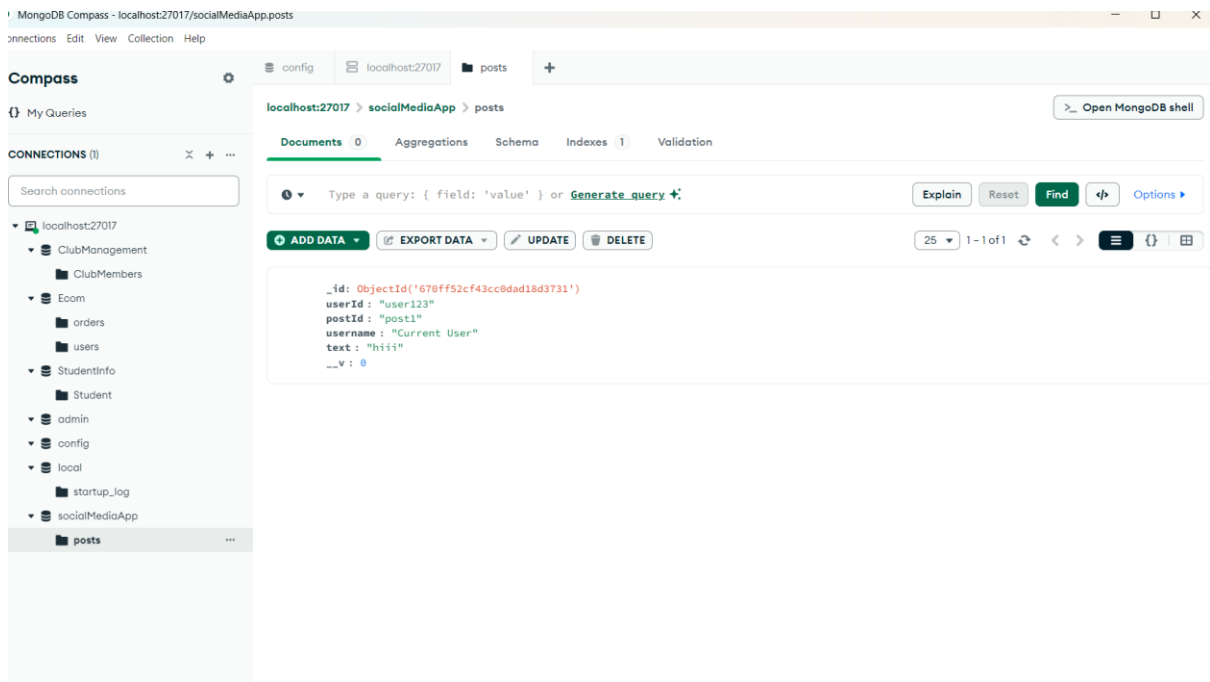


Figure 3.3 Database of Sign In person

This figure 3.3 shows the sign in person data that are stored in the database.

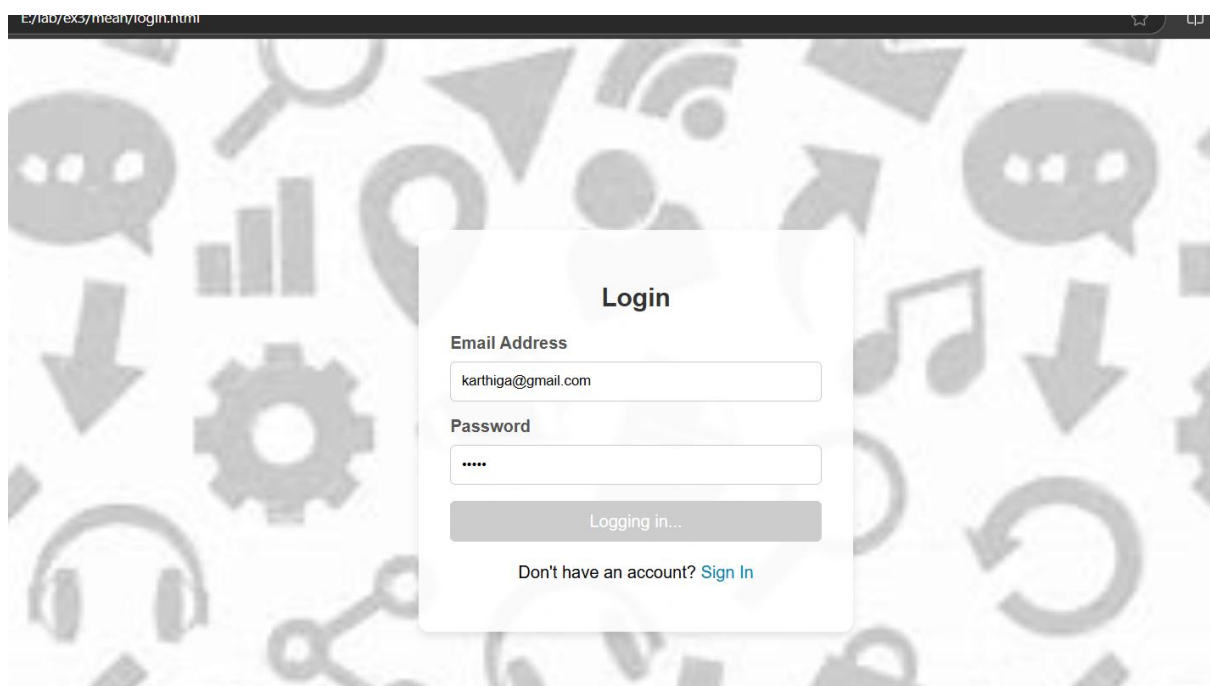


Figure 3.4 Login Page

This figure 3.4 shows getting the details of the user to log into the website.

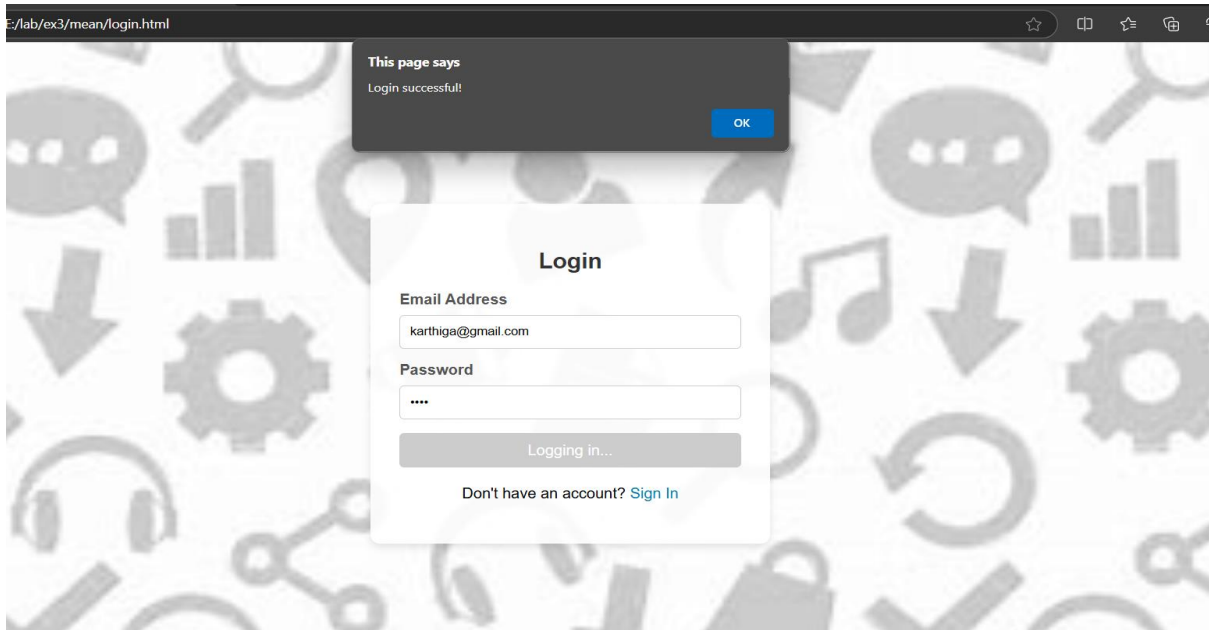


Figure 3.5 Login Successful

This figure 3.5 shows verify the details and send the successful notification.

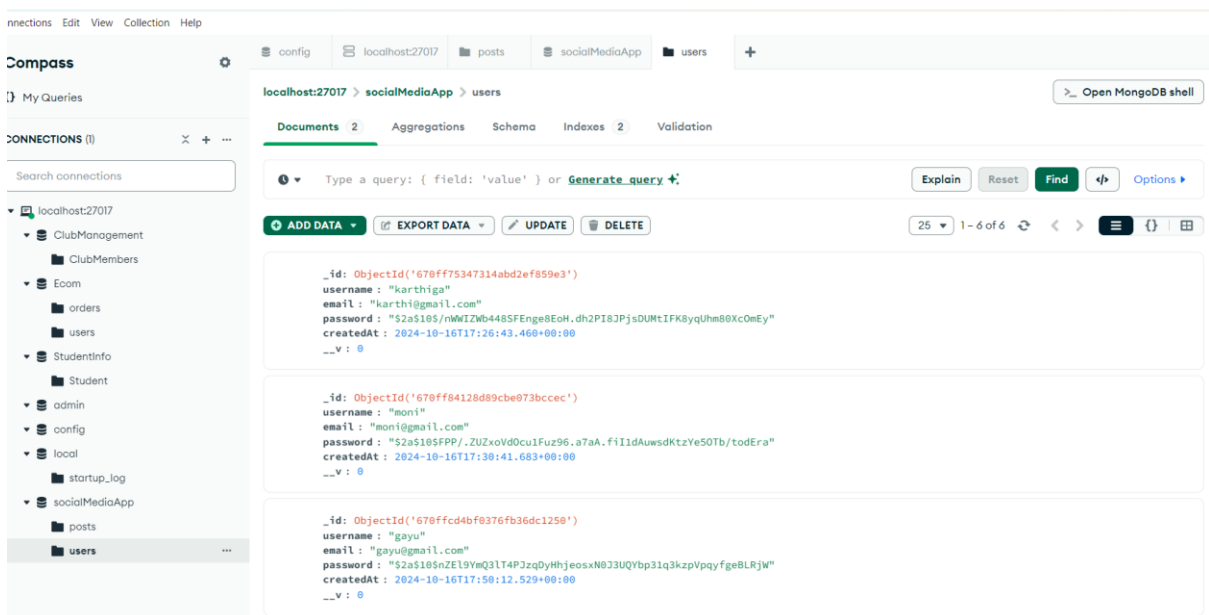


Figure 3.6 Database of Login person

This figure 3.3 shows the login person data are stored in the database.

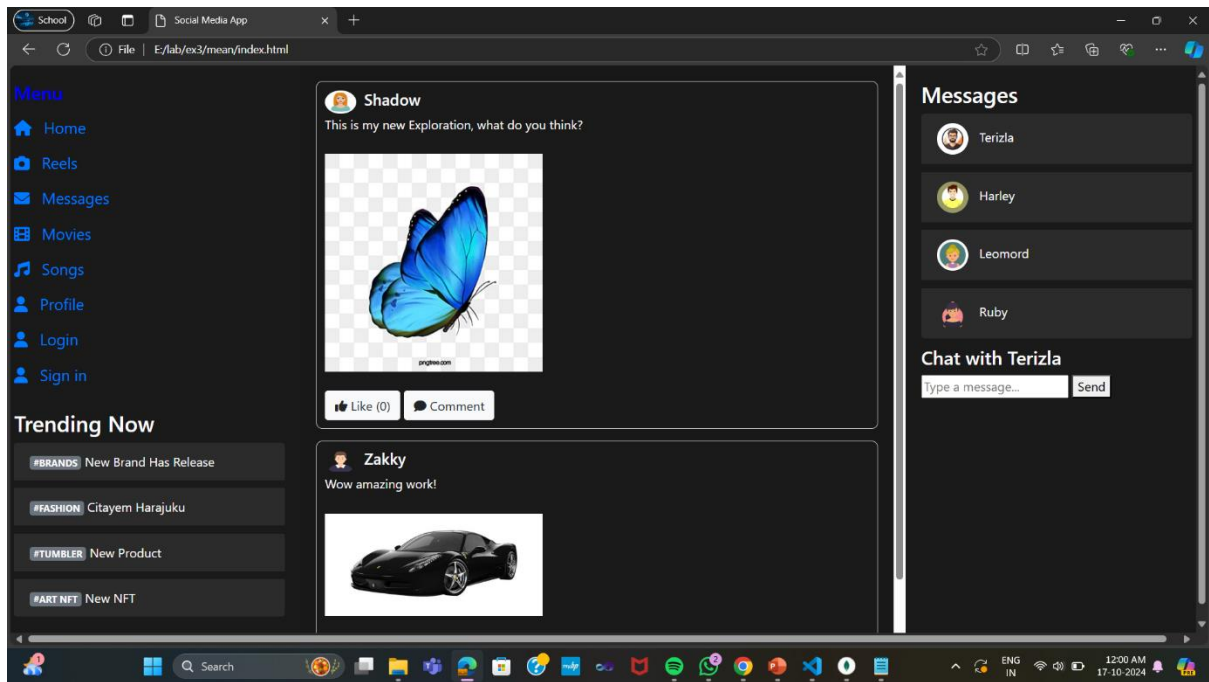


Figure 3.7 Home Page

This figure 3.7 shows the posts that are posted by the other users.

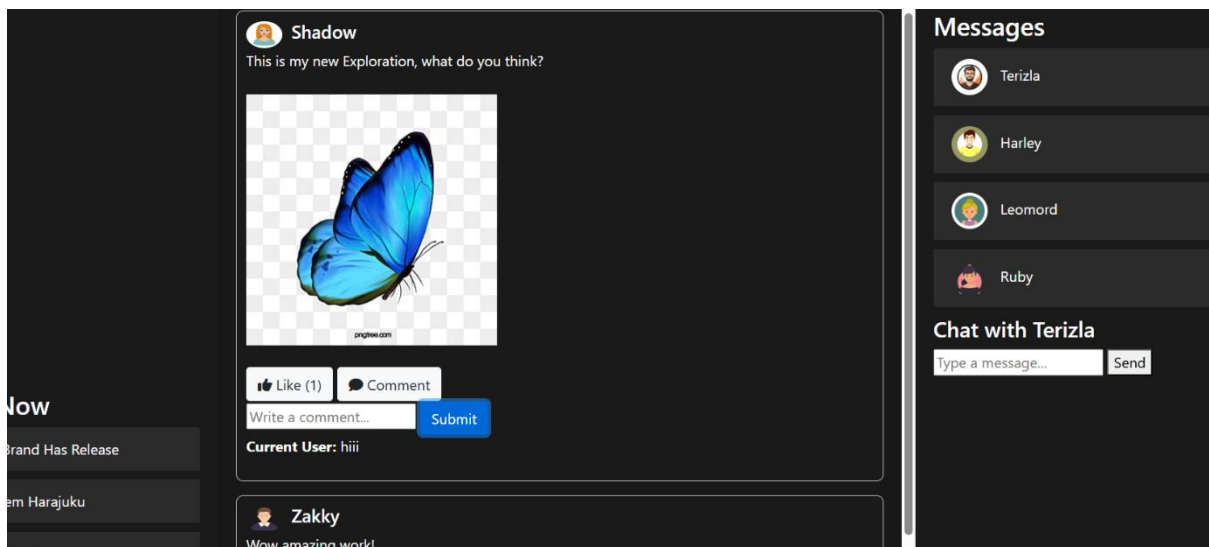


Figure 3.8 Commenting the post

This figure 3.8 shows the like and comment options that are present in the home page.

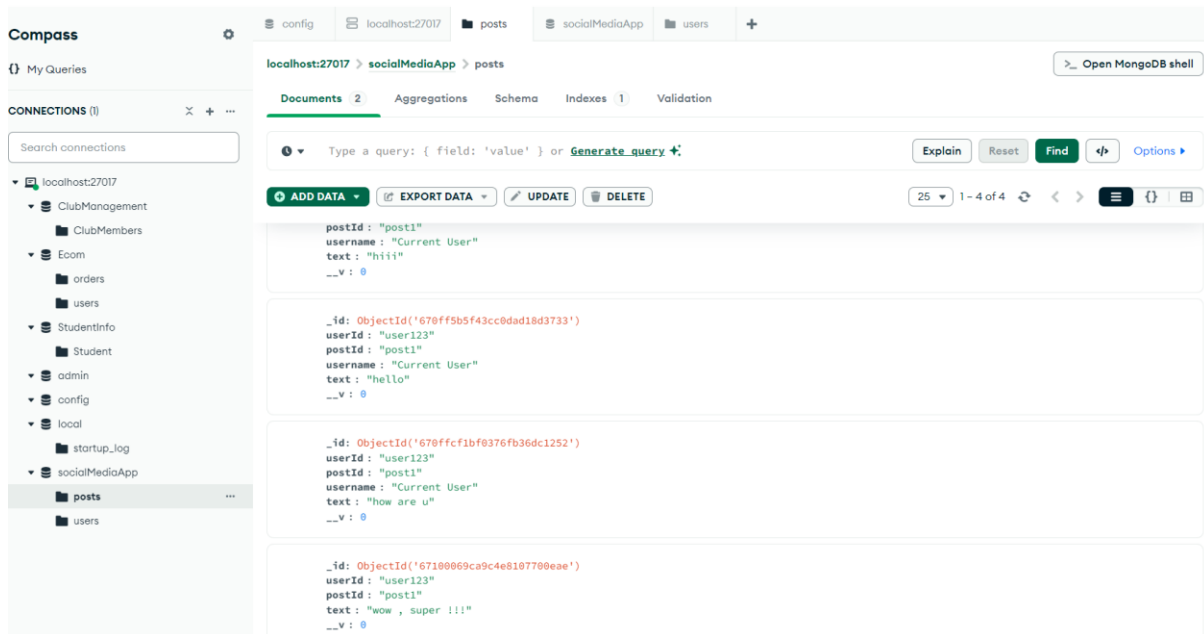


Figure 3.9 Database of Commented person with comment.

This figure 3.9 shows the data of the liked and commented person can be stored in the database.

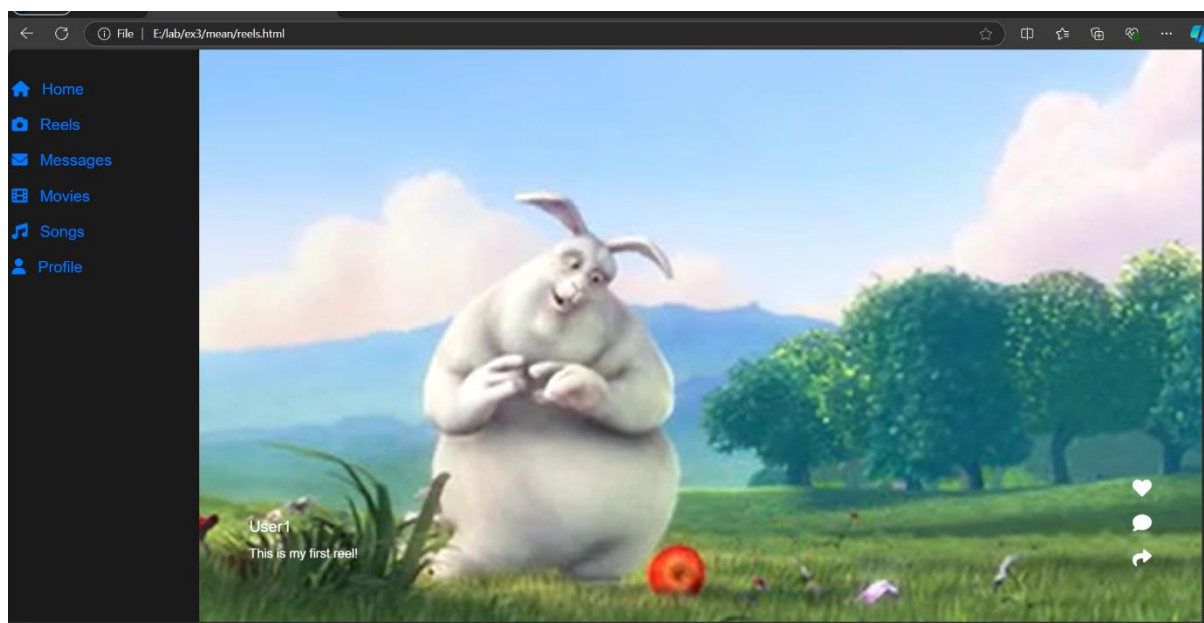


Figure 3.10 Reels Page

This figure 3.10 shows the videos that are posted by the other users.



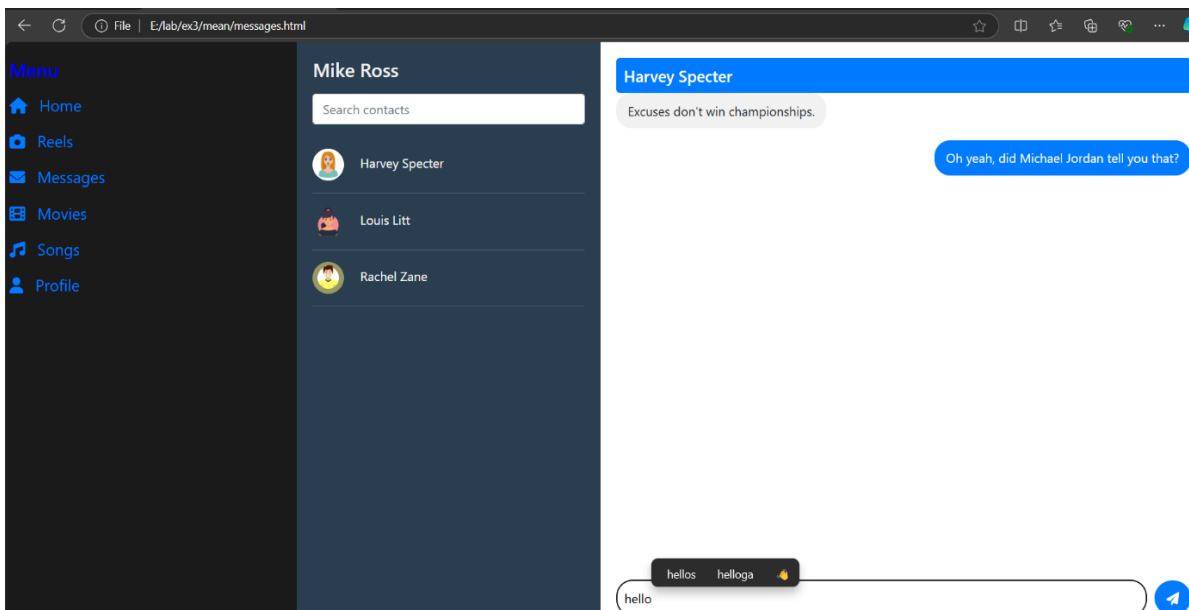


Figure 3.11 Message Page

This figure 3.11 shows page in that the user can interact with the other users.

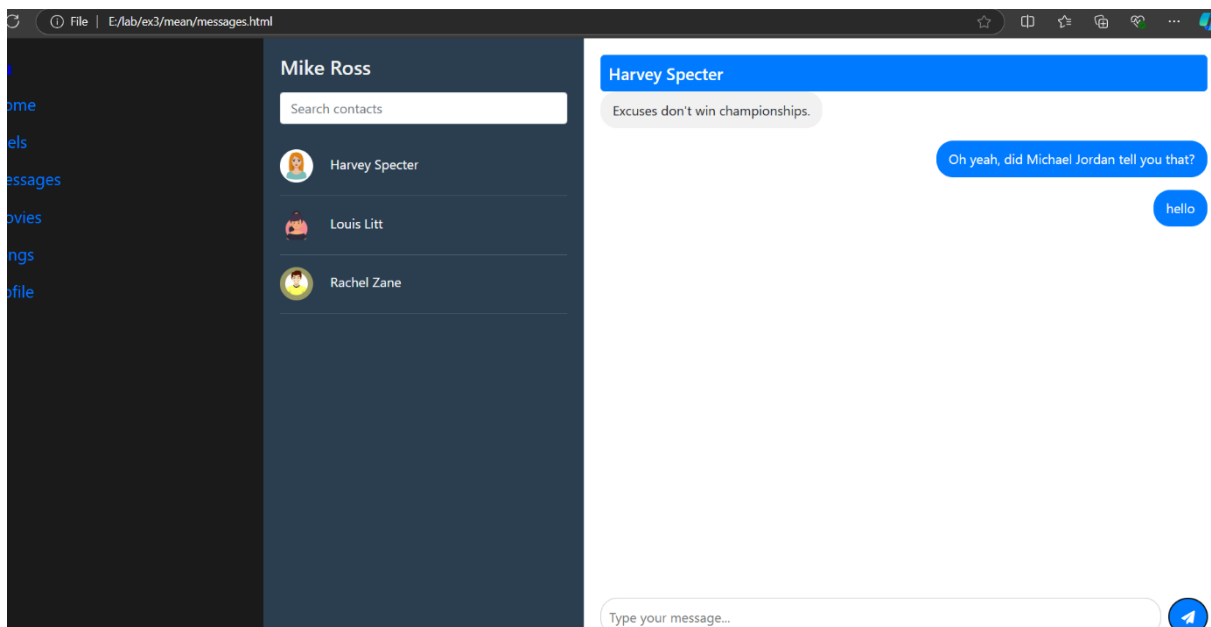


Figure 3.12 Message Page

This figure 3.12 show page in that the user can interact with the other users.

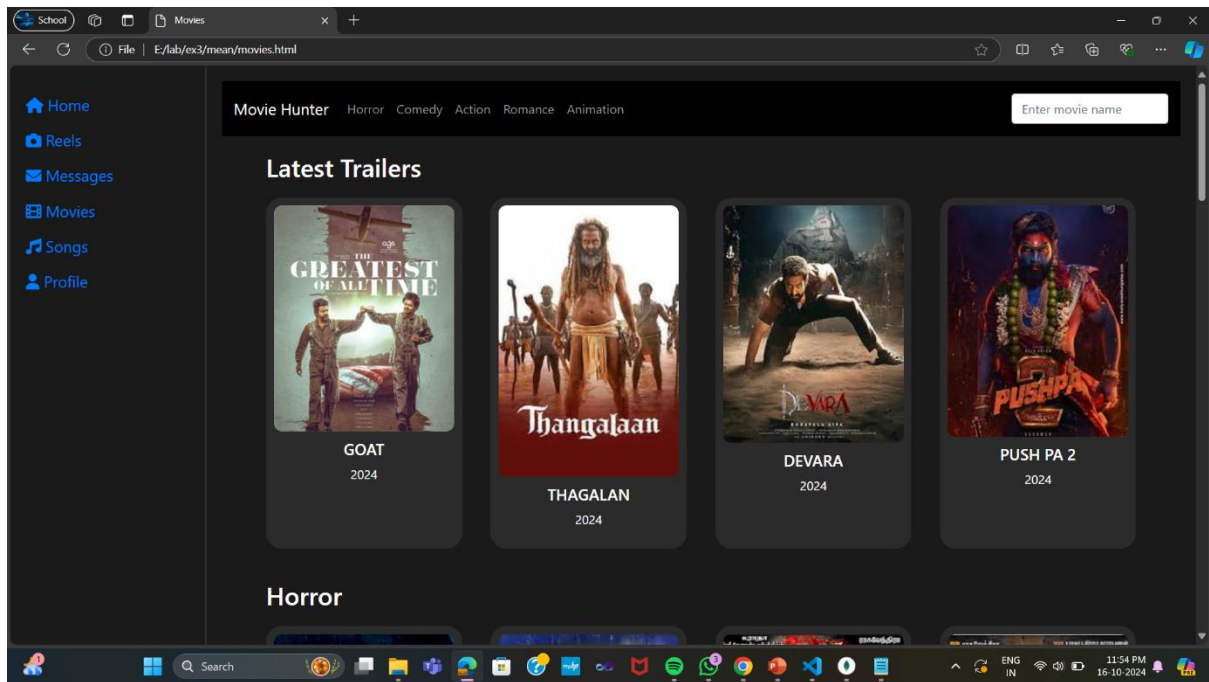


Figure 3.13 Movies Page

This figure 3.13 shows the trailers that are recently released.

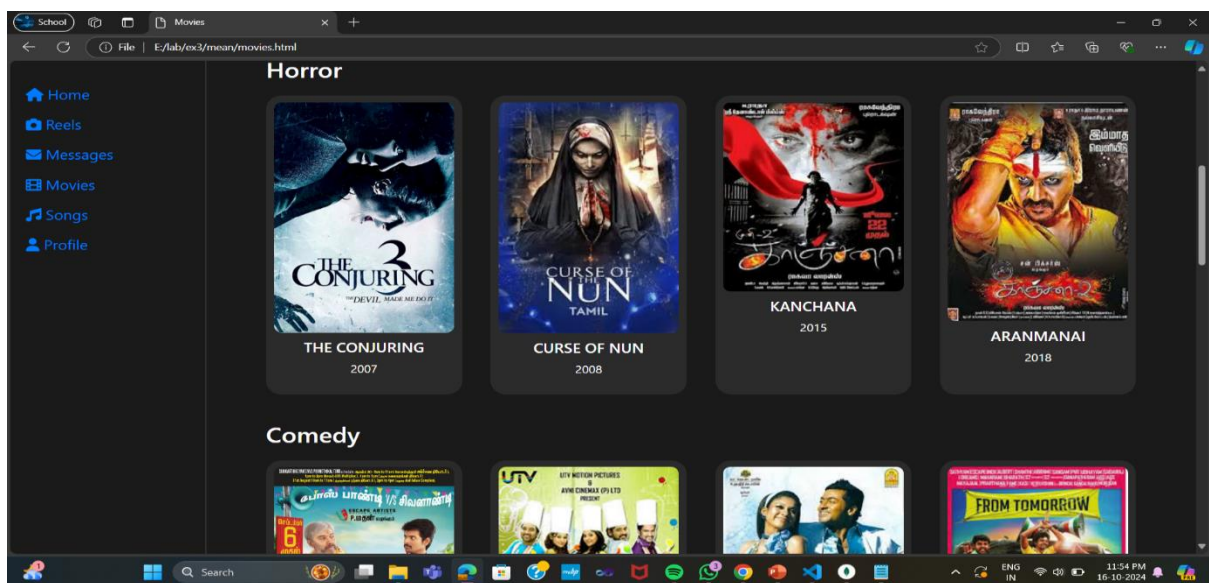


Figure 3.14 Movies Page (Horror)

This figure 3.14 shows the Horror movies.

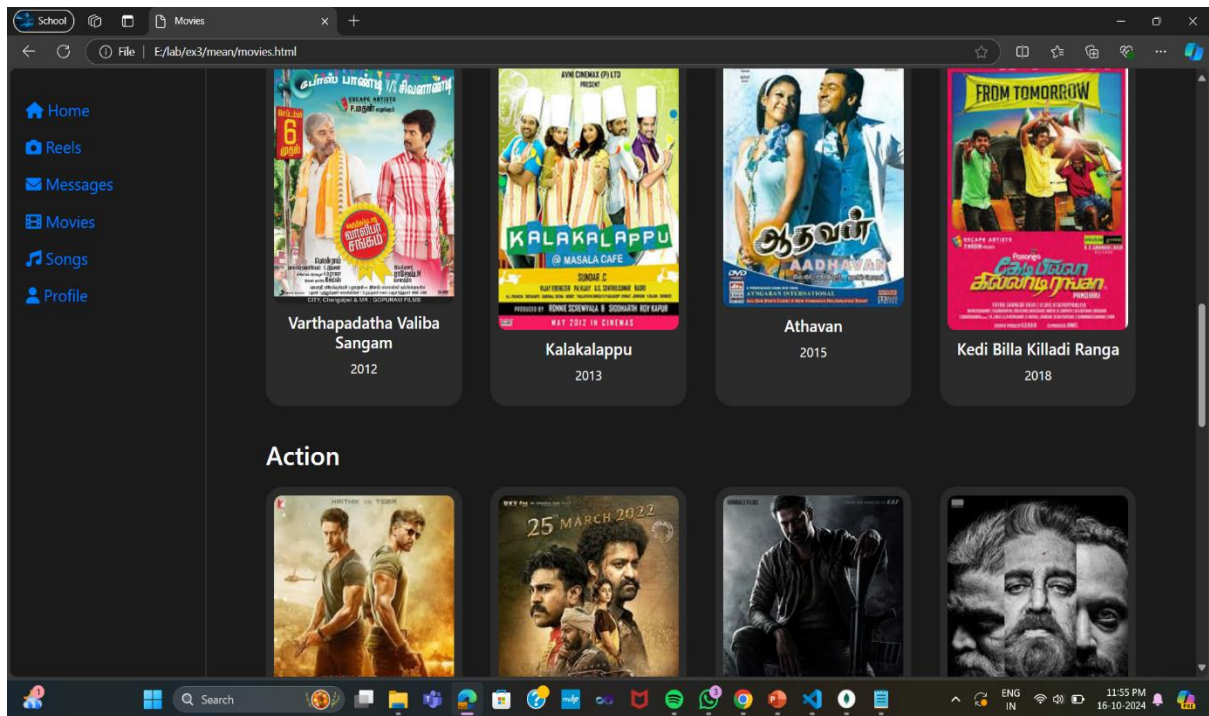


Figure 3.15 Movies Page (comedy and Action)

This figure 3.15 shows the comedy and action movies.

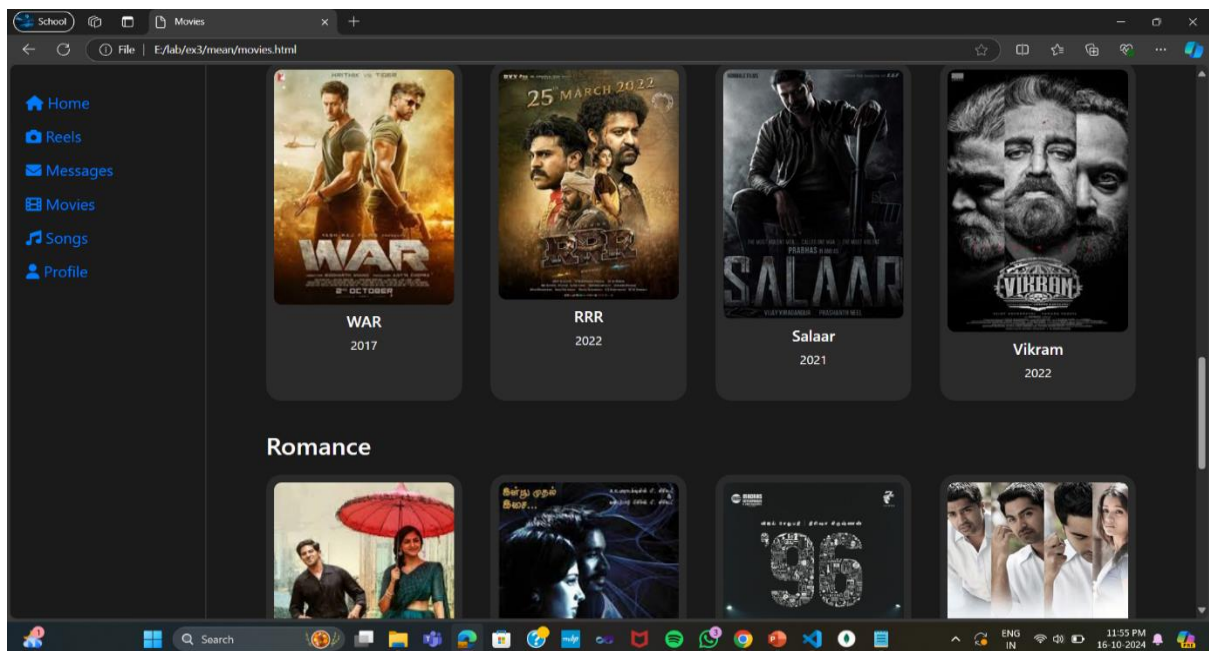


Figure 3.16 Movies Page (Romance)

This figure 3.16 shows the romantic movies.

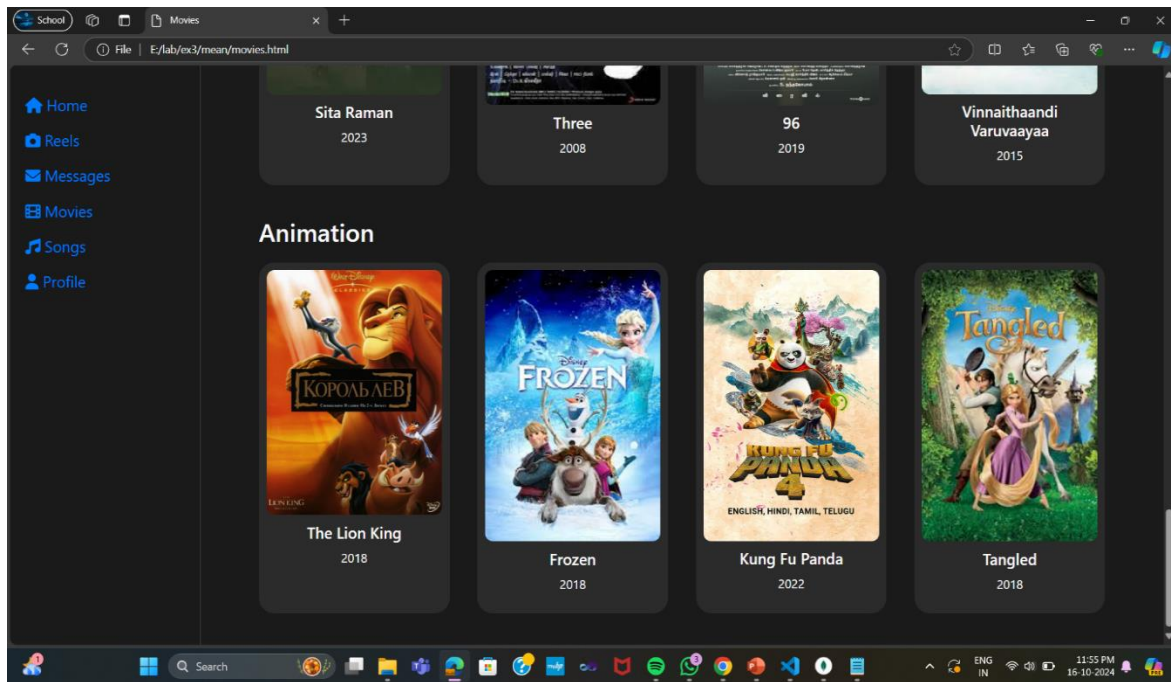


Figure 3.17 Movies Page (Animation)

This figure 3.17 shows the animated movies.

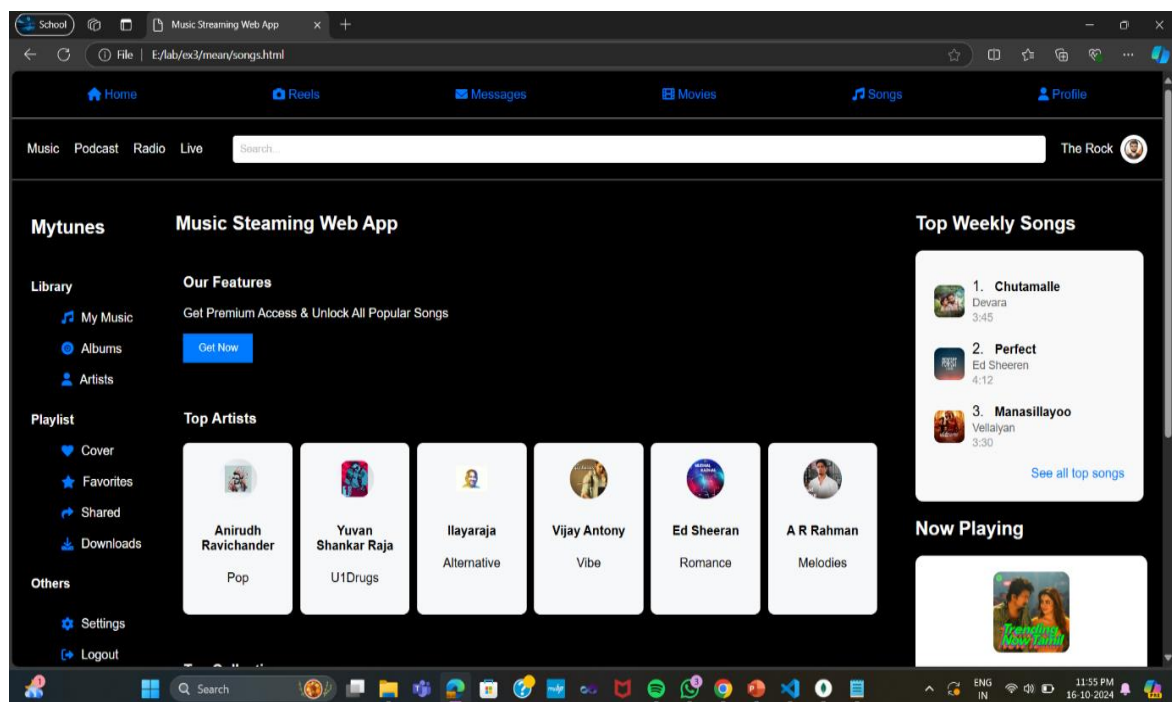


Figure 3.18 Songs Page

This figure 3.18 shows the trending and the user's favourites songs.



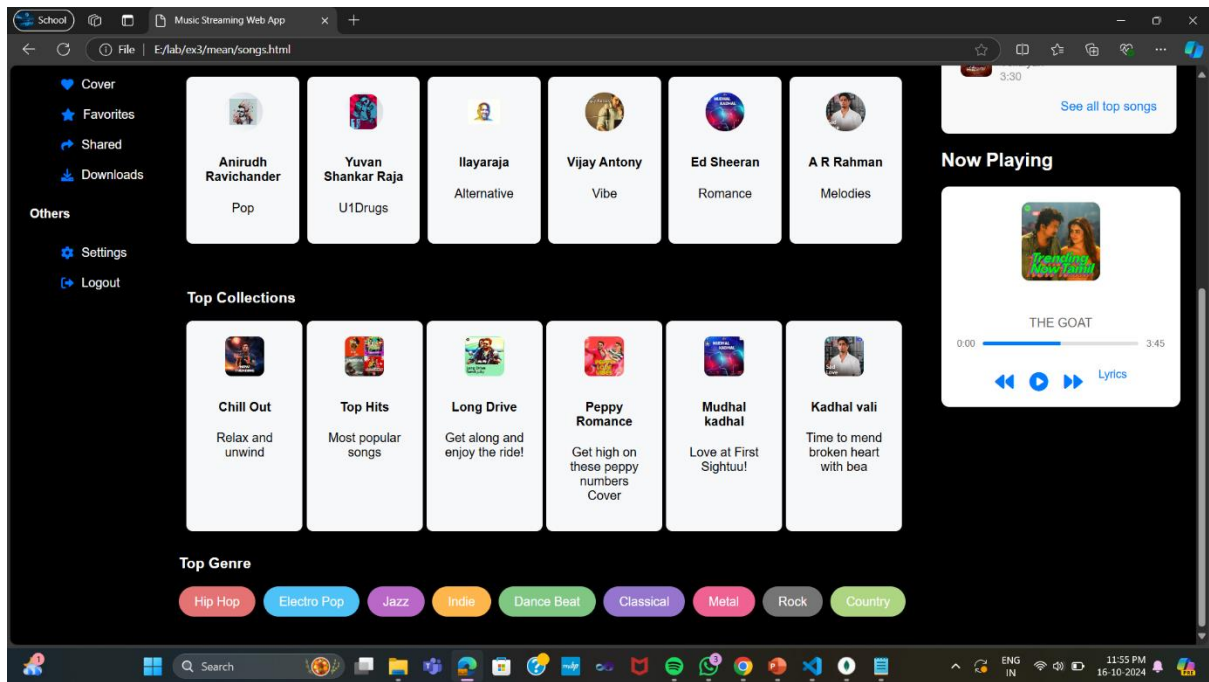


Figure 3.19 Songs Page

This figure 3.19 shows the trending and the top collection songs.

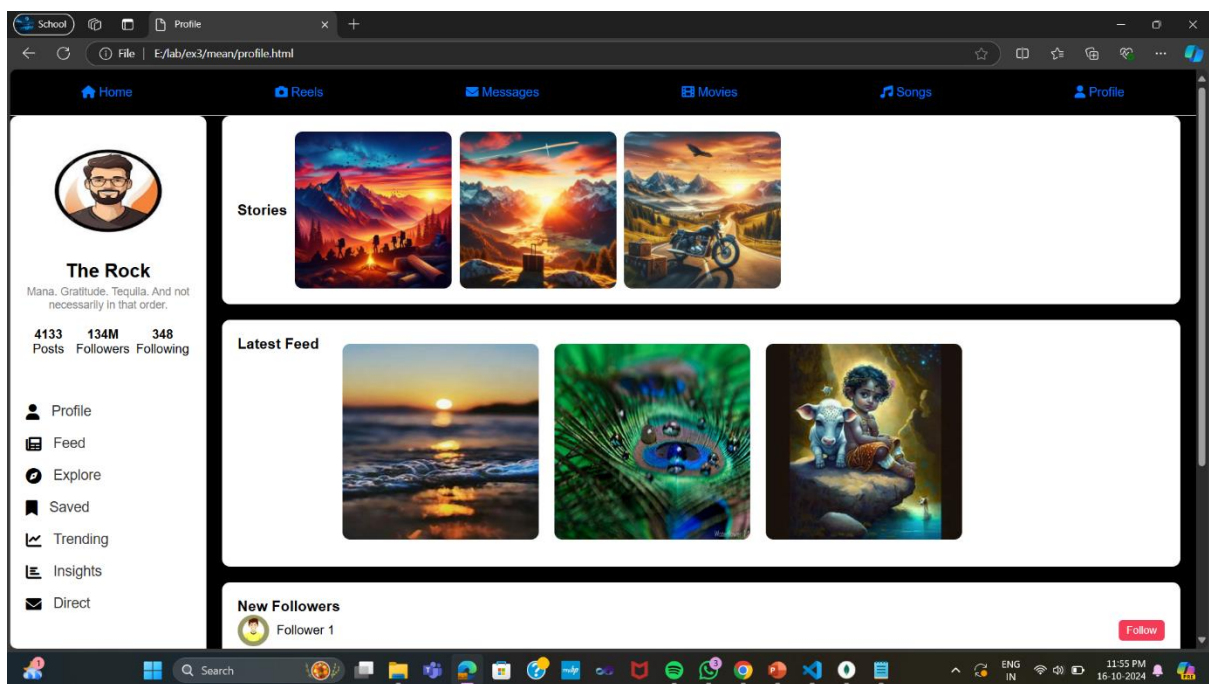


Figure 3.20 Profile Page

This figure 3.20 shows the stories and posts that are posted by the user.

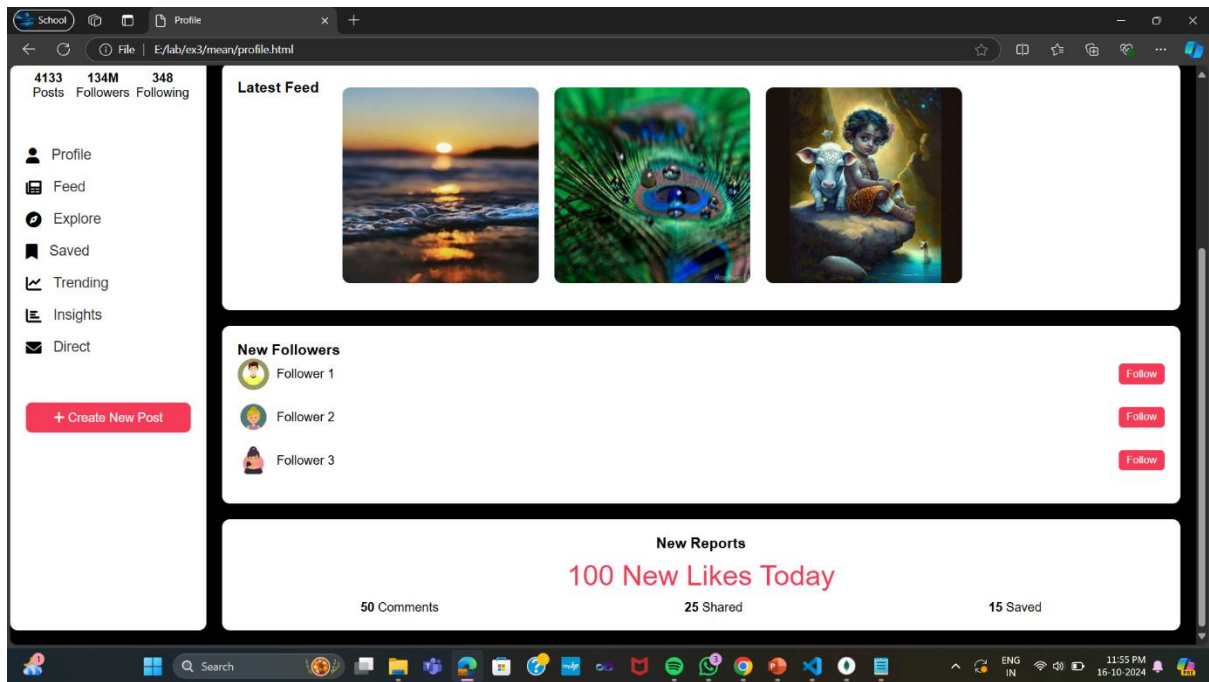


Figure 3.21 Profile Page

This figure 3.21 shows the followers list and current status of the posts.

## **CHAPTER 4**

### **CONCLUSION**

In conclusion, the development of this social media application using the MEAN stack successfully demonstrates the integration of modern web technologies to create a cohesive platform for media consumption and social interaction. The application effectively combines HTML, CSS, and JavaScript to deliver a dynamic user experience, while the MEAN stack components—MongoDB, Express.js, Angular, and Node.js—ensure robust data management, efficient backend operations, and real-time communication capabilities.

The project meets its objectives by providing users with an engaging environment to browse movies and songs, manage favorites, and connect with others through instant messaging. While the current version showcases a solid foundation, ongoing enhancements will be essential to adapt to user needs, scale effectively, and incorporate advanced features. Overall, this social media app represents a significant step toward creating an integrated digital space that enhances user interaction and media accessibility, positioning it well for future growth and innovation.

## REFERENCE

- 1) An in-depth analysis of user behavior and engagement patterns on social media platforms, focusing on content sharing and interaction through reels and comments.

[ResearchGate - Social Media Engagement Analysis](#)

- 2) A comprehensive overview of the latest trends in real-time messaging and social interaction on media-centric platforms, highlighting user preferences and engagement drivers.

[Academia.edu - Social Media Messaging Trends](#)

- 3) An article discussing the integration of multimedia (reels, movies, songs) within social platforms and its impact on user interaction and time spent on the platform.

[Market Research Future - Multimedia in Social Platforms](#)

- 4) A detailed market analysis report on the growth of social media applications, emphasizing content-sharing features like reels, comments, and messaging.

[ResearchGate - Social Media App Growth Trends](#)

- 5) A dataset exploring user interaction and comment behavior on social media posts, focusing on engagement metrics, user demographics, and content preferences.

[Kaggle - Social Media Comment and Interaction Dataset](#)