Lab 4: Building Web-based Applications with MERN Stack

SUB: FUNDAMENTALS OF SCALABLE COMPUTING

SUB CODE: UE23CS643A

Team Member 1:DINESH M

Team Member 2:GANESH YK

Topic: Social Media

INTRODUCTION

- MongoDB: It an open-source NoSQL database, stores data in JSON-like documents, allowing for flexible data storage and querying.
- Express.Js: It a flexible Node.js web framework, simplifies building web servers and APIs by offering a straightforward interface for handling HTTP requests and responses.
- React.Js: It is a JavaScript library which facilitates building dynamic user interfaces for single-page applications, offering reusable UI components and efficient data handling through its component-based architecture and virtual DOM
- Node.Js: Node.js enables running JavaScript on servers, utilizing an event-driven, non-blocking I/O model, making it lightweight and efficient.

Together, Node.js, Express.js, MongoDB, and React.js (MERN stack) form a popular choice for modern web app development, providing a comprehensive JavaScript-based solution for building scalable and interactive web applications.

In this MERN stack implementation with CRUD (Create, Read, Update, Delete) functionality and database integration we aim to understand the very base of how a web application is created and how can it be optimised according to the requirements of the client as the designing of the application architecture is to handle potential increases in user base, optimizing database queries for performance, and implementing efficient frontend components to ensure smooth user experience even under heavy load.

Throughout this report, we will explore the various stages of development, starting from the conceptualization and planning phase to the deployment and optimization of the application.

As we embark on this journey through the development of a MERN stack Todo web application, we invite the reader to explore the intricacies of full-stack web development and discover the endless possibilities it offers in Task Organization and management.

Team Structure and Responsibilities

The two-member team for this project consists of:

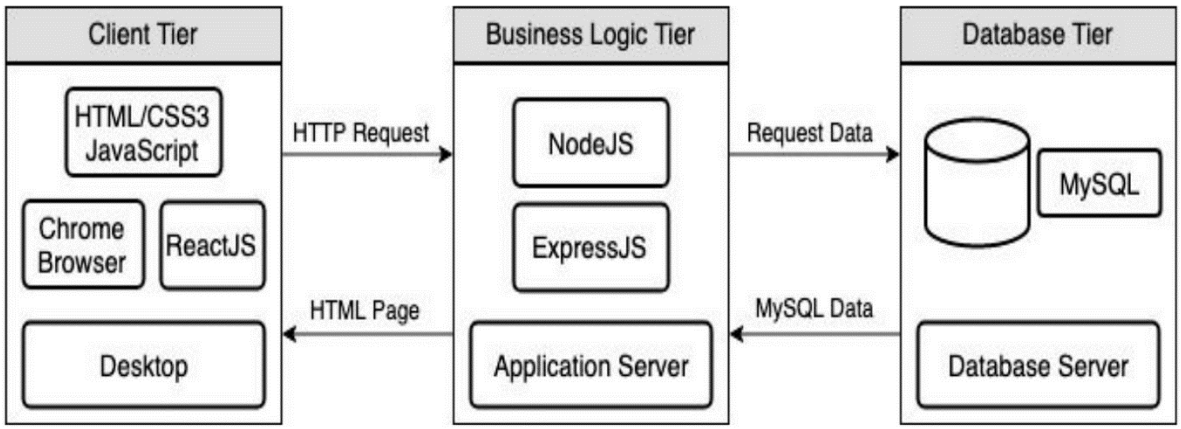
Team member 1 (SRN:PES1PG23CS012) and

Team member 2 (SRN:PES1PG23CS015) .

will handle the frontend development, user interface/user experience (UI/UX) design, and backend data management. Additionally, he will be responsible for deployment, testing of project functionality.

will focus on the backend development, building the API endpoints, implementing business logic, handling error scenarios, and securing the application by implementing Secure Sockets Layer (SSL).

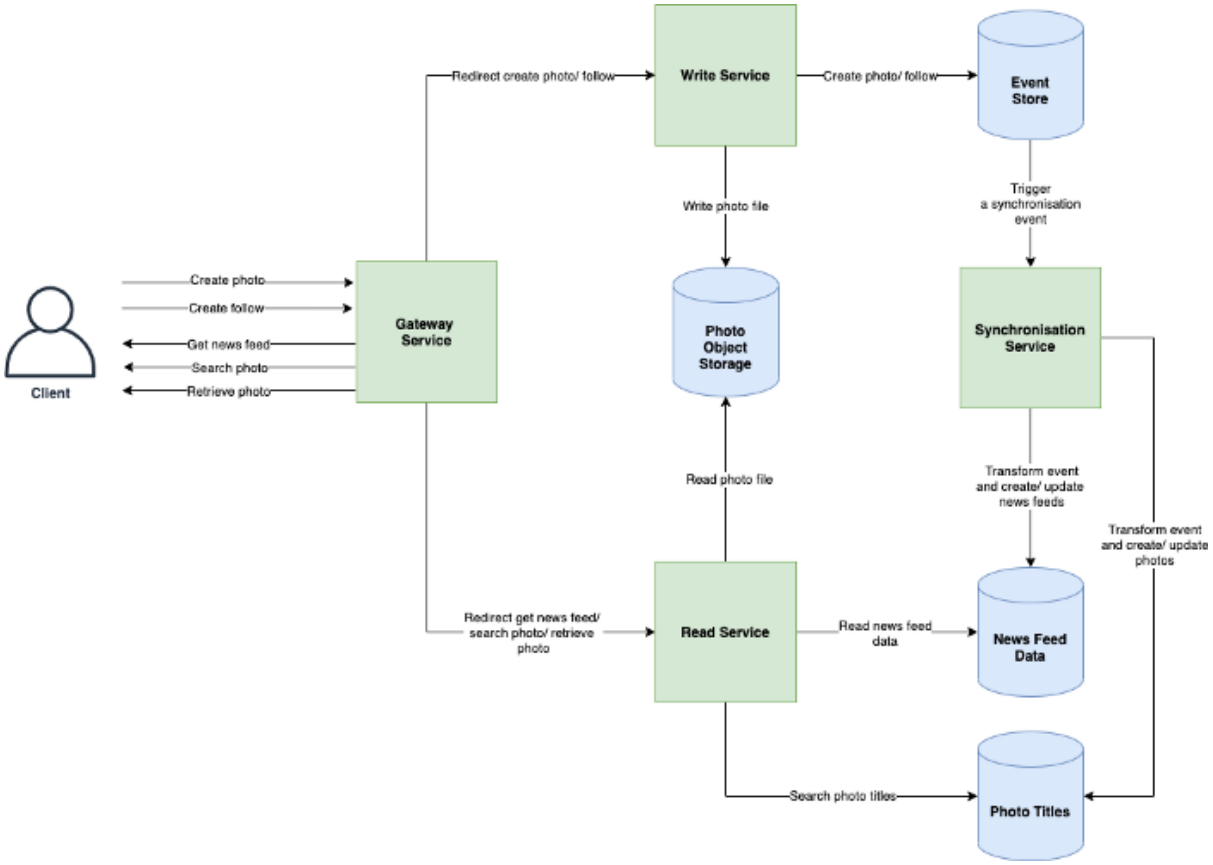
LAYERS OF MERN



Technology Stack

- **Frontend:** React.js
- **Backend:** Node.js
- **Database:** MongoDB
- **Authentication:** JWT

SYSTEM ARCHITECTURE



FRONT END

- Sign Up
- Login
- Followers
- Following
- Profile
- Comment
- Like
- Logout

BACK END

- Models
- Routers
- Controllers

DATABASE

- MongoDB

Features

1. User Profiles: Social media platforms allow users to create personal profiles where they can share information about themselves, such as their name, profile picture, bio, and interests.

2. Post Feed: The news feed is the central feature of most social media apps, displaying a continuously updated stream of content from the people and pages that a user follows. This includes posts, photos, videos, and other updates.

3. Posting: Users can create and share their own content on social media platforms. This can include text posts, photos, videos, links, and more. They can also tag other users, add hashtags, and geotag their posts.

4. Interactions: Social media platforms facilitate interactions between users through features like likes, comments, and shares. Users can engage with each other's content by reacting to it, leaving comments, or sharing it with their own followers.

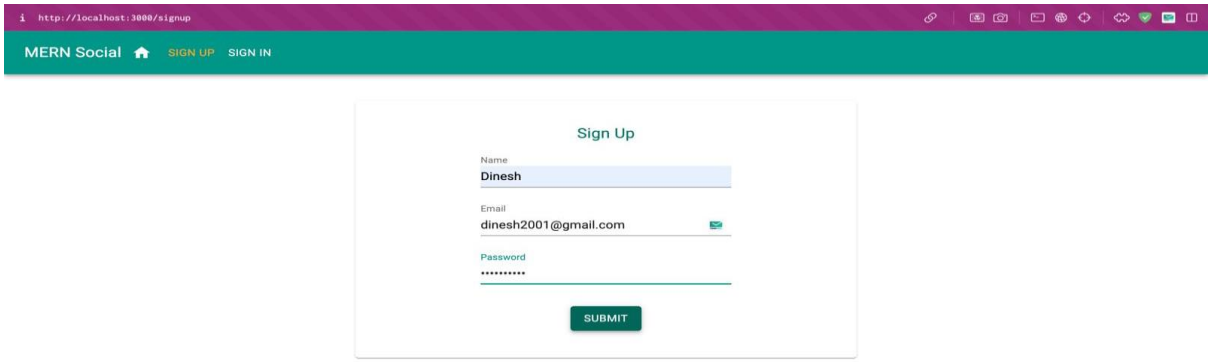
6. Notifications: Social media platforms send notifications to users to alert them of new activity, such as likes, comments, messages, or mentions. Notifications help users stay informed and engaged with the platform.

7. Discovery and Search: Users can discover new content and connect with new people through features like search, trending topics,

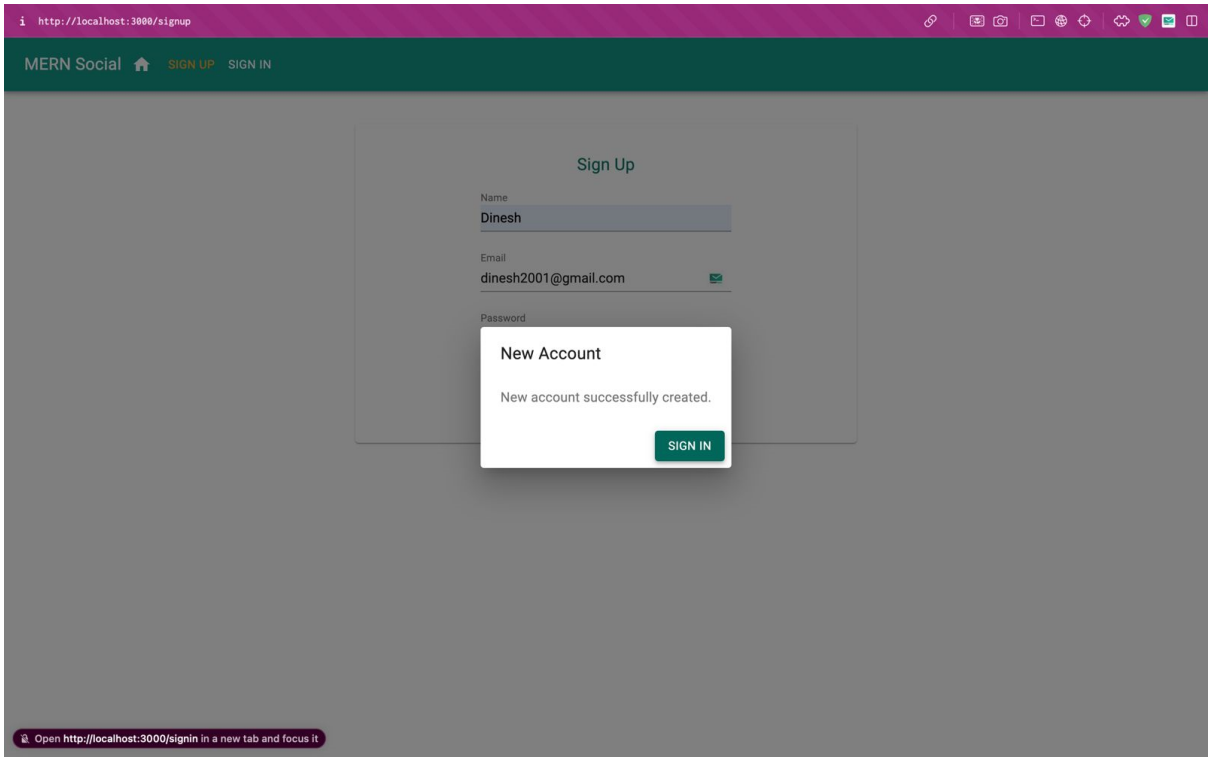
suggested friends, and explore pages. This helps users find content and users that match their interests.

Screenshot Of Sign-Up Page

The social media signup feature allows new users to create accounts on the platform. Users typically provide basic information like their name, email, and date of birth, then verify their account through email or phone. They set up their profile, choose privacy settings, agree to terms of service, and are welcomed to the platform.

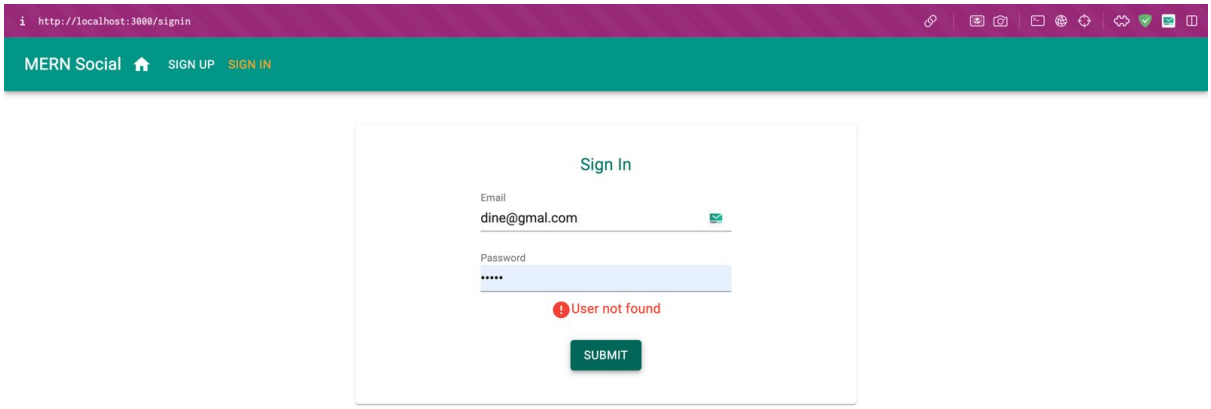


Screenshot Of Successful signup



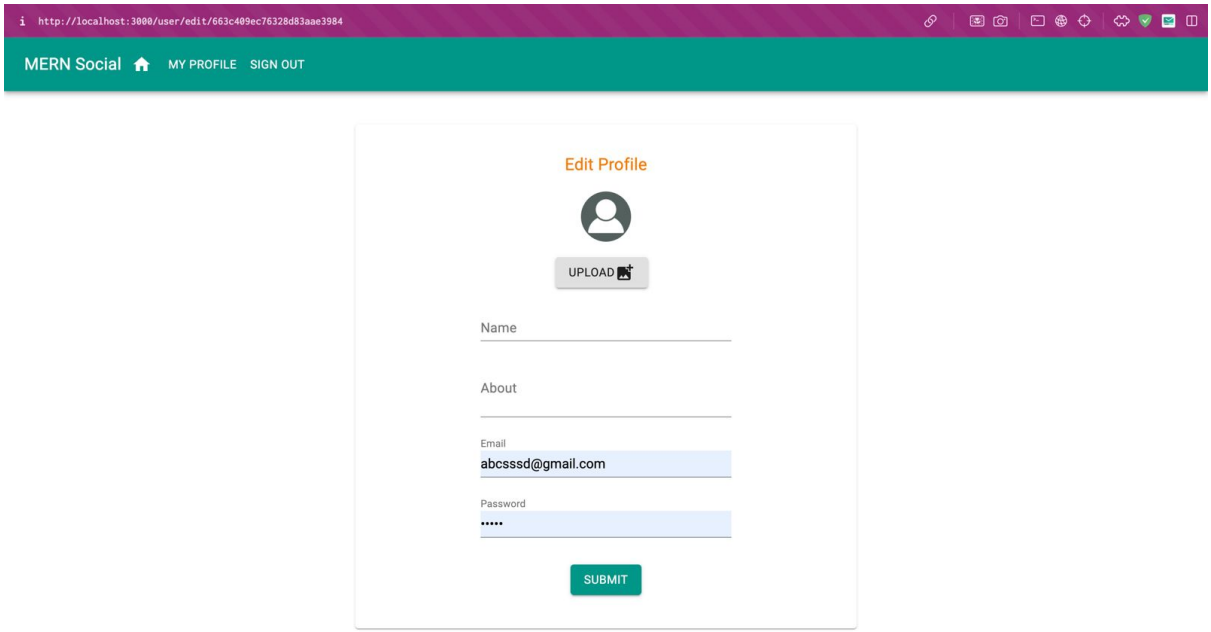
Screenshot Of Sign In

The social media login feature allows users who have already signed up for an account to access the platform quickly and easily. Users can log in using their username/email and password or through third-party authentication providers like Google or Facebook. After entering their credentials, users gain access to their personalized feed, messages, and other features without needing to go through the signup process again.



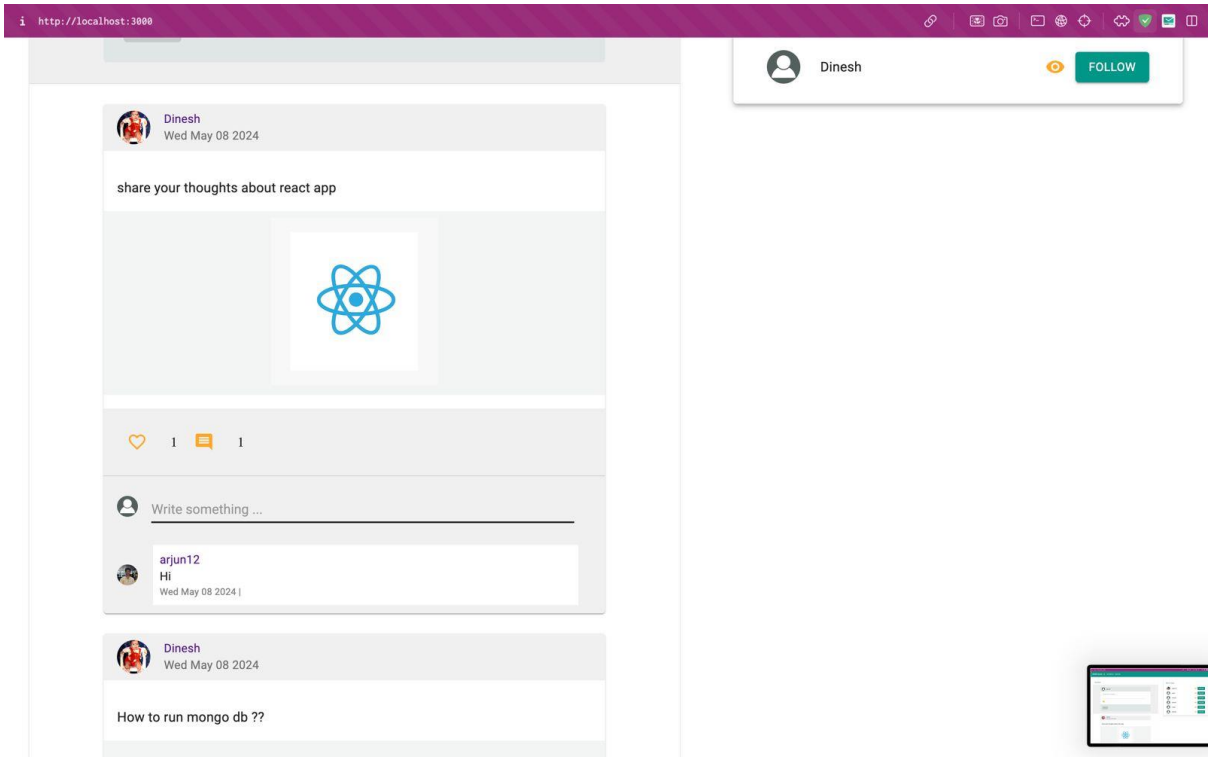
Screenshot Of Profile

The social media profile feature is where users can showcase information about themselves on the platform. Users typically include details such as their name, profile picture, bio, and interests. They can also customize privacy settings to control who can view their profile and its contents. The profile serves as a hub for users to share updates, connect with others, and express their personality within the social media community.



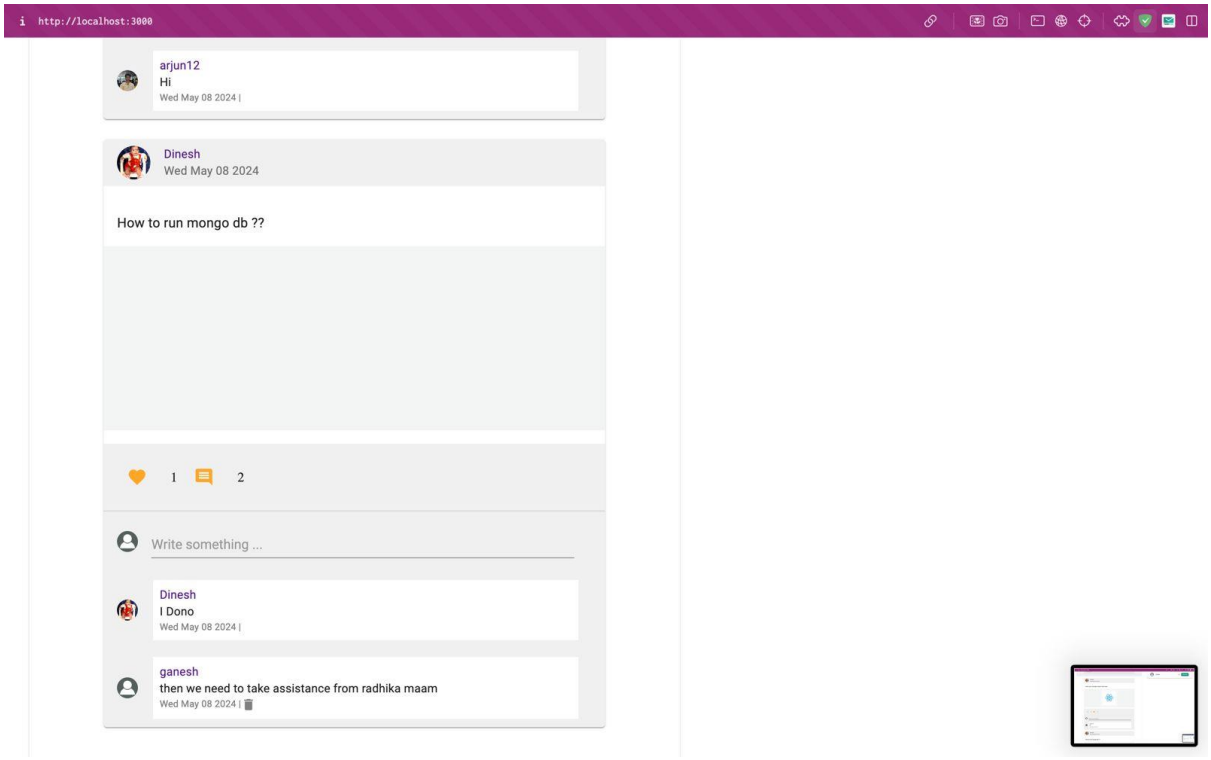
Screenshot Of Comment

The social media comment feature allows users to engage with posts by leaving feedback, reactions, or opinions. Users can respond directly to posts by typing their comments in a designated section below the post. Comments can range from simple reactions like "like" or "wow" to more elaborate responses or discussions. This feature fosters interaction and communication among users, creating a sense of community and facilitating conversations around shared interests or topics.



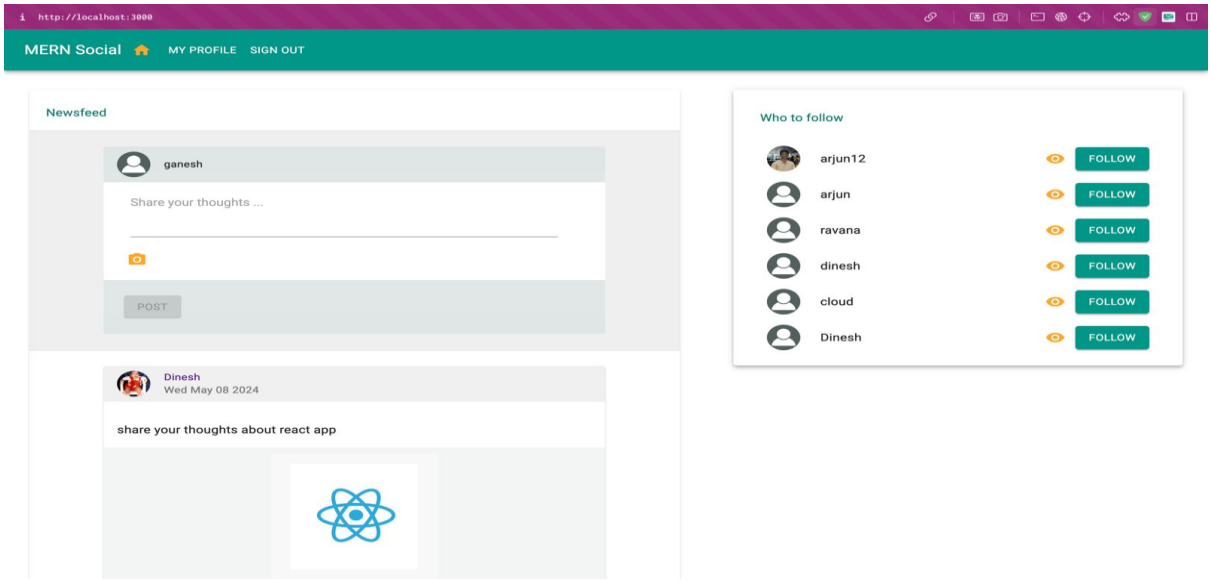
Screenshot Of Like

The social media likes feature allows users to express appreciation or approval for content shared by others. With a simple click or tap, users can "like" a post, photo, or video, indicating that they enjoyed it or found it interesting. Likes serve as a form of social validation and can help content creators gauge the popularity and impact of their posts. They also contribute to the algorithmic ranking of content, influencing its visibility in users' feeds.



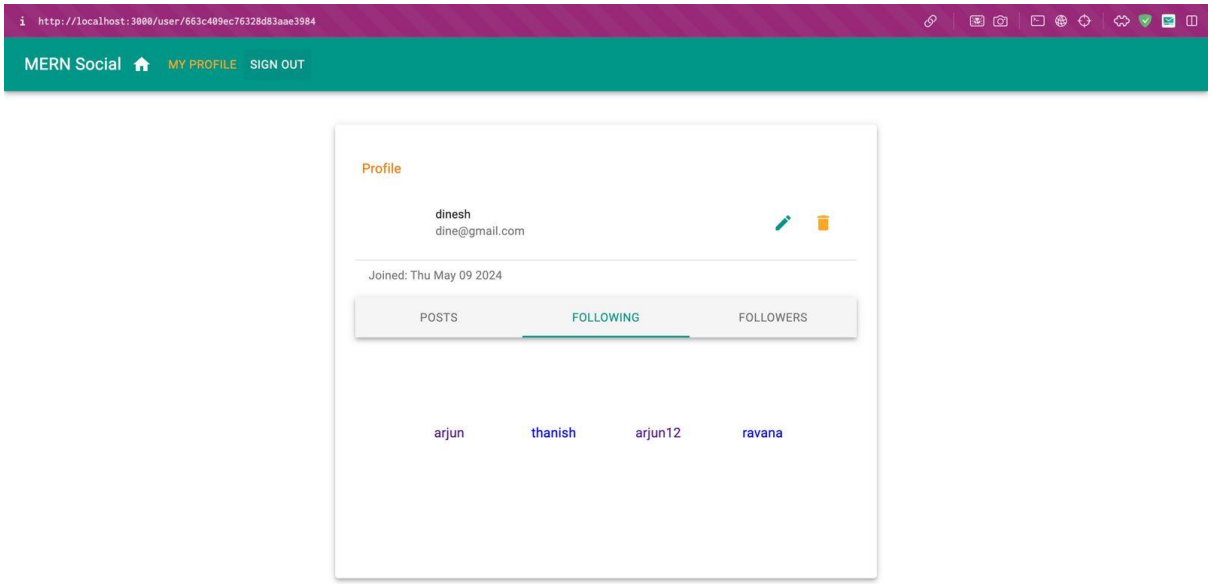
Screenshot Of Followers

The social media followers feature allows users to connect with others and stay updated on their activity. Users can follow other accounts to see their posts, updates, and content in their feed. The number of followers a user has often serves as a measure of their influence or popularity on the platform. Followers can engage with the user's content by liking, commenting, or sharing it, fostering interaction and community within the social media ecosystem.

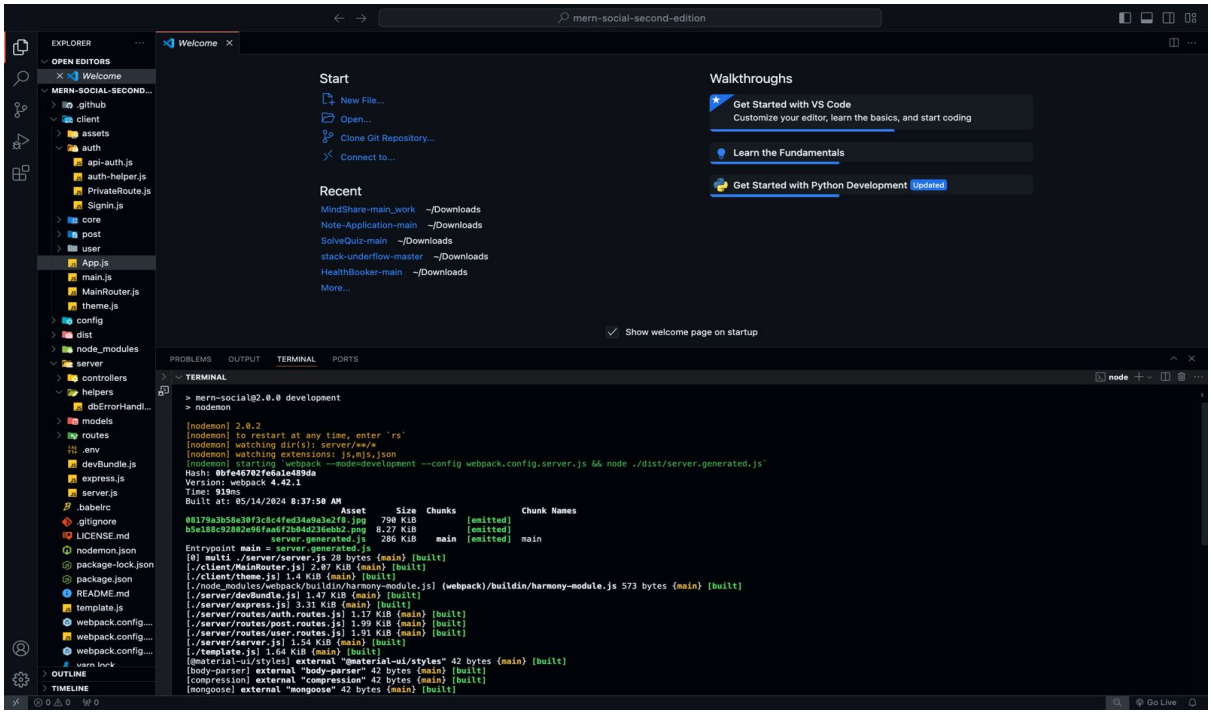


Screenshot Of Following

The social media following feature allows users to subscribe to the updates and posts of other users or accounts on the platform. When a user follows another user or account, the updates and content shared by that user will appear in their feed. This feature enables users to stay connected with friends, family, celebrities, brands, and other entities whose content they find interesting or relevant. Following someone on social media facilitates engagement and interaction by keeping users updated on the latest posts and activities of those they follow.



Screenshot Of Code



The application Focus on:

Simplicity: The login page has minimal features. It focuses on capturing username/email and password for a quick and easy login experience**.**

Secure Communication: Communication should be encrypted (HTTPS) to prevent unauthorized interception of login credentials.

Overall, a well-designed MERN Assignment Social Media page which is both user-friendly and secure.

Implementation Details

Frontend Development (React.js):

React.js is a popular JavaScript library used for building user interfaces, and it's often employed in social media applications for its efficiency and flexibility. Here's how React.js is typically utilized in a social media app:

1. Component-Based Architecture: React.js follows a component-based architecture, where the user interface is broken down into reusable components. In a social media app, components might include the news feed, profile, post, comment, like button, and so on. Each component manages its own state and can be easily reused throughout the application.

2. Dynamic User Interface: Social media apps require dynamic user interfaces that update in real-time as new content is posted or interactions occur. React.js excels at efficiently updating the UI when the underlying data changes, making it well-suited for displaying real-time updates such as new posts, comments, likes, and messages.

3. Efficient Rendering: React.js uses a virtual DOM (Document Object Model) to optimize rendering performance. When the state of a component changes, React.js calculates the minimal set of DOM mutations needed to update the UI, resulting in faster rendering compared to traditional methods.

4. State Management: Social media apps often have complex user interfaces with multiple components that need to share state.

React.js provides tools for managing component state, such as useState and useContext hooks, as well as state management libraries like Redux or React Context API. This allows developers to easily synchronize the state between different components and maintain a consistent user experience.

5. Integration with Backend Services: Social media apps typically rely on backend services to store and retrieve data, such as user profiles, posts, comments, and likes. React.js can communicate with these backend services using APIs (Application Programming Interfaces) to fetch and update data asynchronously. This enables features like posting updates, fetching news feed content, and sending/receiving messages.

6. Responsive Design: React.js supports responsive design principles, allowing developers to create user interfaces that adapt to different screen sizes and devices. This is essential for social media apps, which are accessed on a variety of devices ranging from smartphones to desktop computers.

7. Third-Party Libraries and Plugins: React.js has a vast ecosystem of third-party libraries and plugins that extend its functionality. Social media app developers can leverage these libraries for features like rich text editing, image and video processing, authentication, and more, saving time and effort in development.

Overall, React.js is a powerful tool for building social media apps due to its component-based architecture, efficient rendering, state management capabilities, and support for real-time updates. By

leveraging React.js, developers can create engaging and responsive user interfaces that meet the needs of modern social media users.

User Interaction and Data Flow:

- **Fetch API:** Libraries used to make HTTPS requests to the backend API endpoints for CRUD operations.
- **State Management:** A state management solution like Redux or Context API will be implemented to manage assignment data across components.
- **Adding Assignments:** Upon submitting the AddAssignmentForm, data will be sent in a POST request to the backend's "create assignmetn" API endpoint. The response from the backend (confirmation or error) will update the application state and potentially trigger a re-render of the assignment list.
- **Deleting Assignments:** A delete button on each assignment can be implemented. Clicking it will send a DELETE request to the backend's "delete assignment" endpoint with the assignment ID. Upon successful deletion, the application state and assignment list will be updated accordingly.

Backend Development (Express.js):

Server Setup:

- Express.js will be used to create a Node.js server that listens for incoming requests from the frontend.
- Middleware like body-parser will be used to parse incoming request bodies containing assignment data.

API Endpoints:

API (Application Programming Interface) endpoints in a social media app are specific URLs or routes that allow external systems or applications to interact with the app's backend server. These endpoints serve as gateways for accessing and manipulating data stored in the app's database. Here's how API endpoints are typically used in a social media app:

1. User Authentication : Social media apps often have API endpoints for user authentication, such as `/login` and `/register`. These endpoints handle user login and registration processes, verifying user credentials and generating authentication tokens upon successful authentication.

2. User Profile Management : API endpoints like `/user/profile` or `/user/settings` allow users to view and update their profiles. Users can retrieve their profile information, such as name, email, profile picture, bio, and preferences, and modify it as needed.

3. Content Creation and Interaction : Social media apps provide API endpoints for creating and interacting with content, such as posting updates, commenting on posts, liking posts, and sharing content. These endpoints typically include routes like `/posts/create`, `/posts/{postId}/comments`, `/posts/{postId}/like`, and `/posts/{postId}/share`.

4. News Feed : API endpoints are used to fetch news feed content for users. These endpoints retrieve posts and updates from users and pages that the user follows, filtering and sorting them based on relevance and recency. Endpoints like `/feed` or `/newsfeed` are commonly used for this purpose.

5. Messaging and Notifications : Social media apps often have API endpoints for messaging and notifications. These endpoints handle tasks such as sending messages between users, fetching message history, and delivering notifications about new messages, comments, likes, or mentions.

6. Search and Discovery : API endpoints enable users to search for other users, pages, or content within the app. These endpoints provide search functionality and return relevant results based on the user's query. Routes like `/search/users`, `/search/posts`, and `/search/tags` are typical examples.

7. Analytics and Insights : Some social media apps offer API endpoints for retrieving analytics and insights data. These endpoints allow users or administrators to access metrics such as post engagement, follower growth, demographic information, and more.

8. Security and Permissions : API endpoints may also include routes for managing security and permissions, such as verifying user permissions before performing certain actions or restricting access to sensitive data.

Overall, API endpoints play a crucial role in enabling communication between the frontend client (e.g., web or mobile app) and the backend server of a social media app, allowing users to interact with the app's features and access its data programmatically

Key Features:

- Enhanced User Experience: Intuitive interface simplifies assignment creation, and deletion.
- Scalability: It offers a robust foundation for future application growth and feature additions.

Challenges Faced:

- Efficient Data Retrieval: As the number of assignments grows, optimizing database queries to ensure fast retrieval of assignment information becomes important.

Conclusion

In conclusion, the development of a MERN (MongoDB, Express.js, React.js, Node.js) social media app presents a powerful and comprehensive solution for building a dynamic and engaging platform. By leveraging the strengths of each component within the MERN stack, developers can create a feature-rich application that offers seamless user experiences across various devices.

The flexibility of MongoDB as a NoSQL database allows for scalable and efficient data storage, accommodating the diverse and evolving needs of social media platforms. Express.js serves as a robust backend framework, facilitating the creation of API endpoints for user authentication, content management, messaging, and more.

React.js, with its component-based architecture and efficient rendering capabilities, empowers developers to craft highly responsive and interactive user interfaces. Its ability to handle real-time updates makes it particularly well-suited for displaying dynamic content in a social media context.

Node.js rounds out the MERN stack by providing a fast and scalable runtime environment for server-side JavaScript execution. Its event-driven architecture enables non-blocking I/O operations, ensuring optimal performance for handling concurrent user requests.

Overall, the MERN stack offers a modern and efficient solution for building social media applications that meet the demands of today's users. By combining MongoDB, Express.js, React.js, and Node.js,

developers can create a versatile and powerful platform that delivers a seamless and engaging social media experience..

Challenges Overcome:

- **Mongoose Indexing:** Mongoose allows creating indexes on frequently used fields. This significantly improves the speed of database queries when filtering or searching for assignment based on these criteria.
- **User-friendly Interface:** The frontend can provide a clear and intuitive interface for users to filter assignments. By offering a well-designed filtering system, the complexity of managing a large number of assignments is reduced for users.

Project Repository

https://github.com/ganesh-yk/MERN_SOCIAL-MEDIA/

https://github.com/Dinesh12122001