

ConnectSphere:

A Social Media App

by

Akshay Maruti Kamble

Submitted on: 30-03-2024

Project Duration: 01-03-2024 to 30-03-2024

Contact Information: akshaykamble1431@gmail.com

GitHub: <https://github.com/akshaykamble04/Social-Media-Project.git>

Table Of Contents

Sr.No	Chapter	Page No.
1.	Introduction	1
2.	Objectives	2
3.	Technologies Used	4
4.	System Architecture	5
5.	Database Design	9
6.	Features	10
7.	Future Improvements	19
8.	Conclusion	22

Table Of Figures

Sr.No	Name	Page No.
1.	Database Design	9
2.	Login Page	10
3.	Register Page	11
4.	Dashboard	13
5.	Dark/Light Mode	14
6.	Update	17
7.	Post	18

1. Introduction

In today's digital age, social media platforms have become integral parts of our daily lives, facilitating communication, connection, and collaboration on a global scale. With the increasing demand for user-centric and feature-rich social networking experiences, the development of innovative social media applications has become paramount.

In response to this demand, the project titled "ConnectSphere" aims to deliver a comprehensive and immersive social media platform designed to connect users in meaningful ways.

2. Objectives

The primary objective of ConnectSphere is to provide users with a seamless and engaging social media experience that fosters interaction, communication, and collaboration. To achieve this objective, the project focuses on the following key goals:

Technological Innovation: Leveraging modern web development technologies and frameworks, including React.js and Node.js, to build a robust and scalable social media platform.

User Authentication and Authorization: Implementing secure user authentication and authorization mechanisms using JSON Web Tokens (JWT) to ensure the privacy and security of user data.

Real-time Data Management: Utilizing React Query to efficiently manage and fetch data from the server, enabling real-time updates and seamless user interactions.

Responsive User Interface: Designing a responsive and intuitive user interface using React.js and CSS to enhance user experience across various devices and screen sizes.

3. Technologies Used

ConnectSphere is developed using a combination of frontend and backend technologies to deliver a comprehensive and feature-rich social media experience. The key technologies and tools utilized in the development process include:

Frontend Technologies: React.js, JavaScript, JSX, React Query

Backend Technologies: Node.js, Express.js

Database: MySQL

Authentication: JSON Web Tokens (JWT)

By leveraging these technologies, ConnectSphere aims to deliver a cutting-edge social media platform that seamlessly integrates user-centric features with modern web development practices.

4. System Architecture

React.js (Frontend):

- ***Component-Based Architecture:*** React.js follows a component-based architecture, where the UI is broken down into reusable components.
- ***Virtual DOM:*** React maintains a virtual representation of the DOM in memory, allowing efficient updates and rendering of UI components.
- ***State Management:*** React uses states to manage component data and re-renders components when the state changes.
- ***Unidirectional Data Flow:*** Data flows in one direction, from parent to child components, ensuring predictable and manageable state updates.

Node.js and Express.js (Backend):

- ***Event-Driven Architecture:*** Node.js follows an event-driven architecture, where asynchronous, non-blocking I/O operations are handled through event loops.
- ***Middleware Stack:*** Express.js provides a middleware stack for handling HTTP requests and responses. Each middleware function can modify the request or response objects.
- ***Routing:*** Express.js allows you to define routes to handle different HTTP methods and URL patterns, making it easy to create RESTful APIs.
- ***Database Integration:*** Node.js can integrate with various databases, including MySQL, using libraries like Sequelize or MySQL.js to perform database operations.

MySQL (Database):

- ***Relational Database Management System (RDBMS):*** MySQL is an open-source RDBMS that organizes data into tables with rows and columns, and supports SQL queries for data manipulation.
- ***Schema Definition:*** MySQL requires defining a schema that outlines the structure of the database,

including tables, columns, data types, and relationships between tables.

- **Query Processing:** MySQL processes SQL queries efficiently, utilizing indexes and query optimization techniques to retrieve data quickly.
- **Concurrency Control:** MySQL implements concurrency control mechanisms to ensure data consistency and isolation in multi-user environments.

JSON Web Token (JWT) (Authentication):

- **Token-Based Authentication:** JWT is a compact, URL-safe token format used for authentication and authorization.
- **Stateless Authentication:** JWT enables stateless authentication, where the server does not need to store session data for each user.
- **Payload:** JWT consists of three parts: header, payload, and signature. The payload contains claims or attributes about the user.
- **Cryptographic Signing:** JWT tokens are digitally signed by the server to ensure their integrity and prevent tampering.

React Query (Data Fetching):

- ***Data Fetching and Caching:*** React Query provides hooks for fetching and caching data from APIs in React applications.
- ***Query Invalidation:*** React Query automatically invalidates and refetches stale data based on configurable criteria, such as cache timeouts or manual triggers.
- ***Optimistic Updates:*** React Query supports optimistic updates, allowing UI to be updated optimistically before the server responds, providing a smoother user experience.

5. Database Design

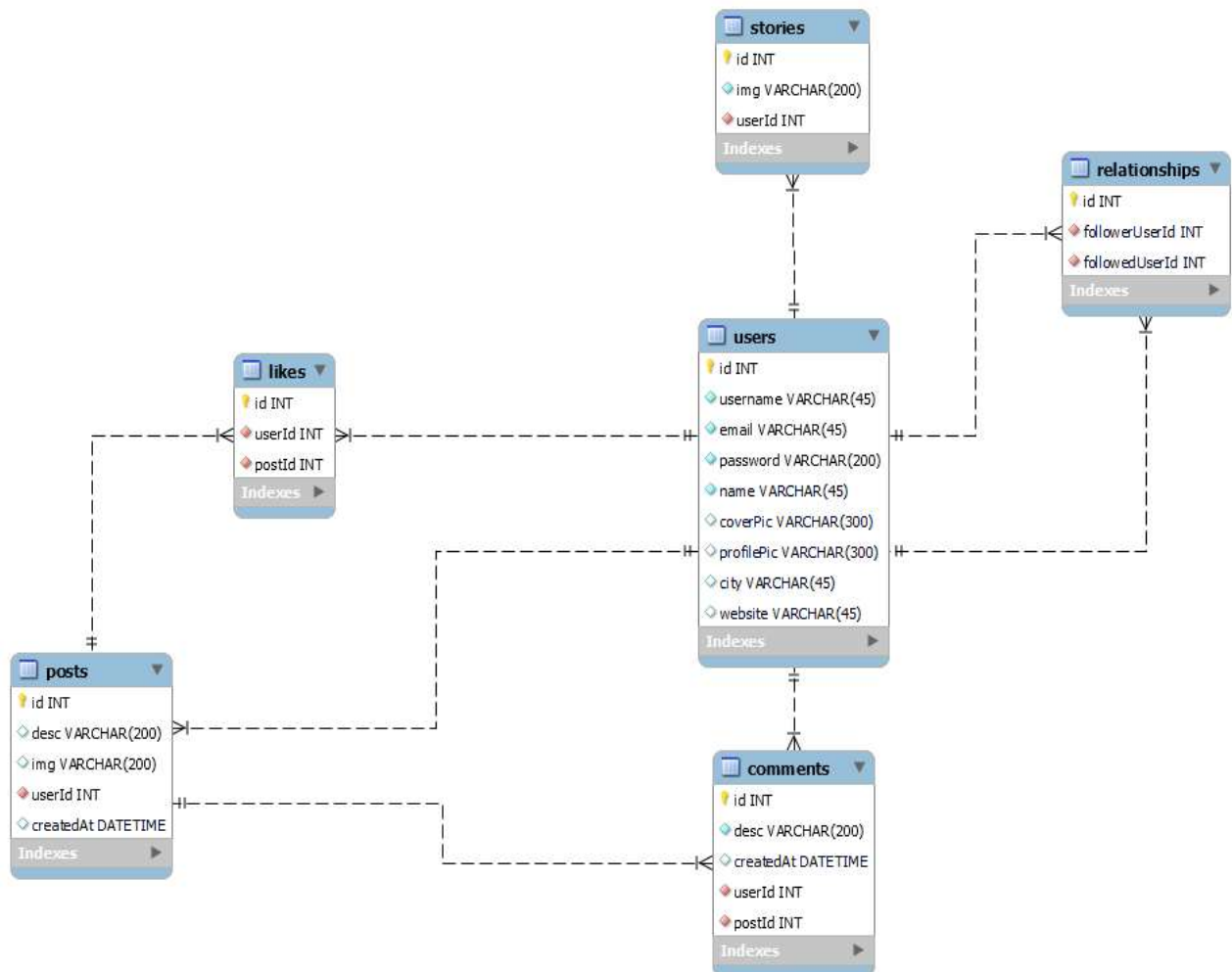


Figure 5.1 Database Design

6. Features

1. User Authentication and Authorization

User registration and login functionality with secure password hashing.

Authorization mechanisms to control access to certain features or content based on user roles.

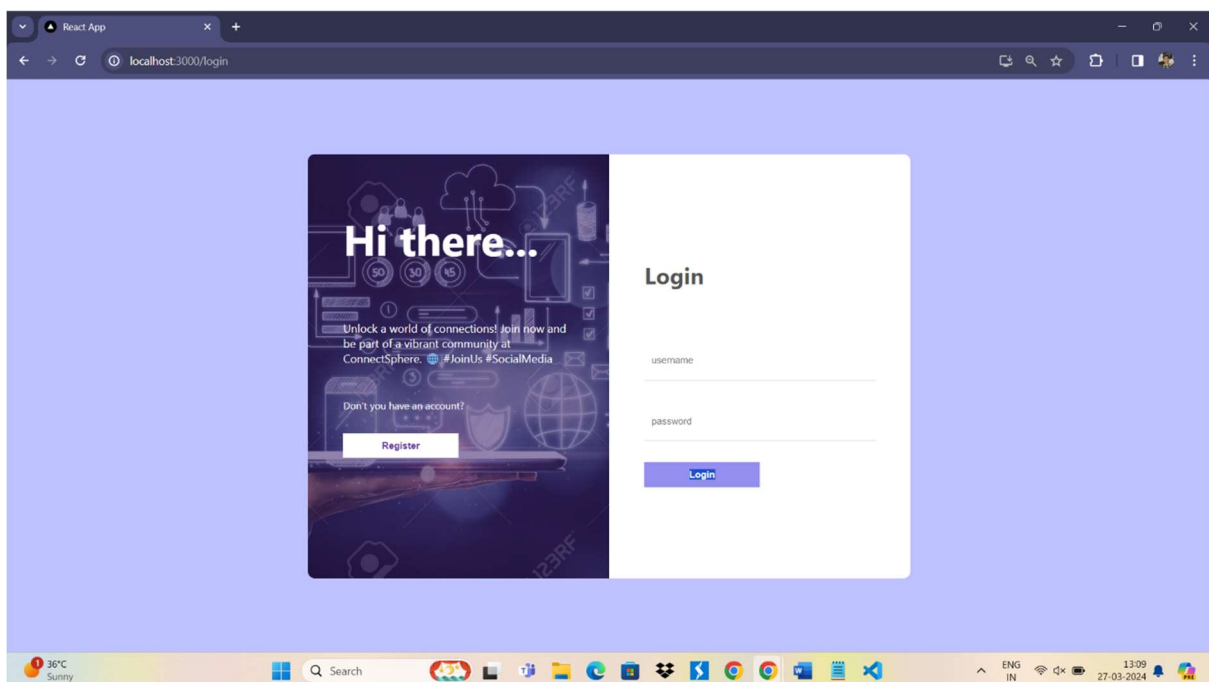


figure 6.1 Login Page

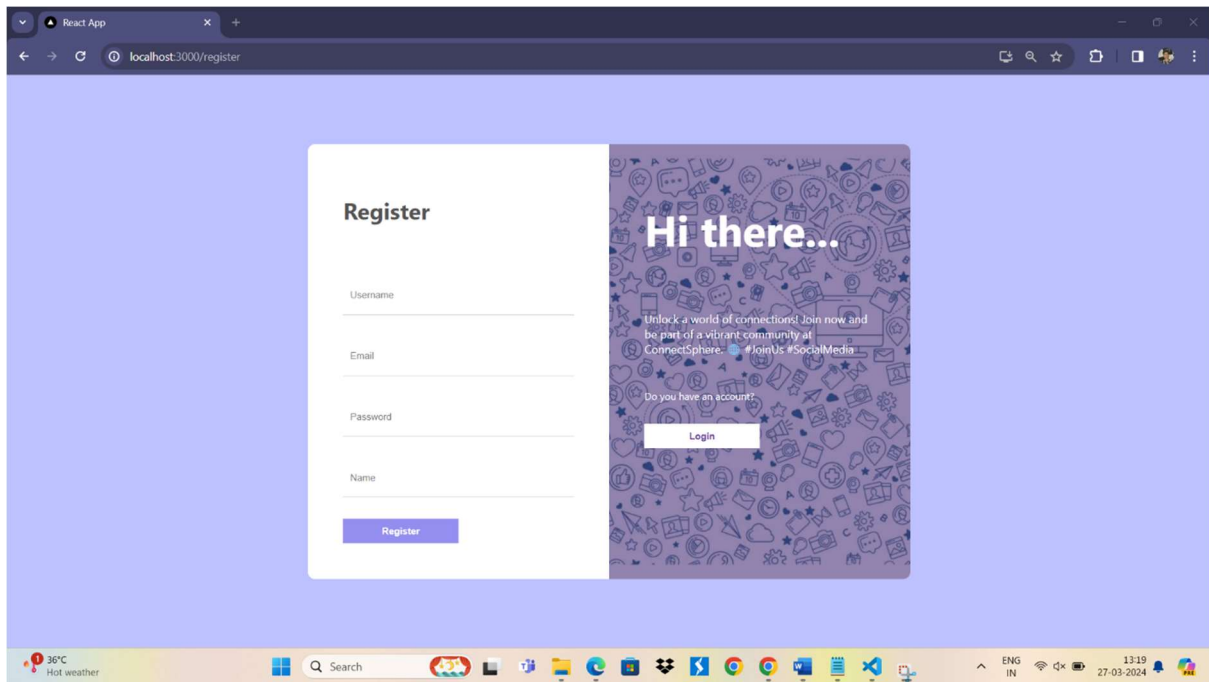


figure 6.2 Register Page

The Login component represents the login page of social media app. It allows users to authenticate and access their accounts.

State Management:

Utilizes the useState hook to manage state variables:

inputs: Tracks the values of username and password input fields.

err: Stores any error messages encountered during the login process.

Navigation:

Uses the useNavigate hook from React Router to handle navigation within the app.

Event Handling:

Defines a handleChange function to update the inputs state when user inputs change.

Implements a handleLogin function to handle the login process when the user submits the login form.

Calls the login function from the AuthContext to authenticate the user.

If successful, navigates the user to the home page ("/").

If an error occurs, displays the error message in the UI.

II. Dashboard

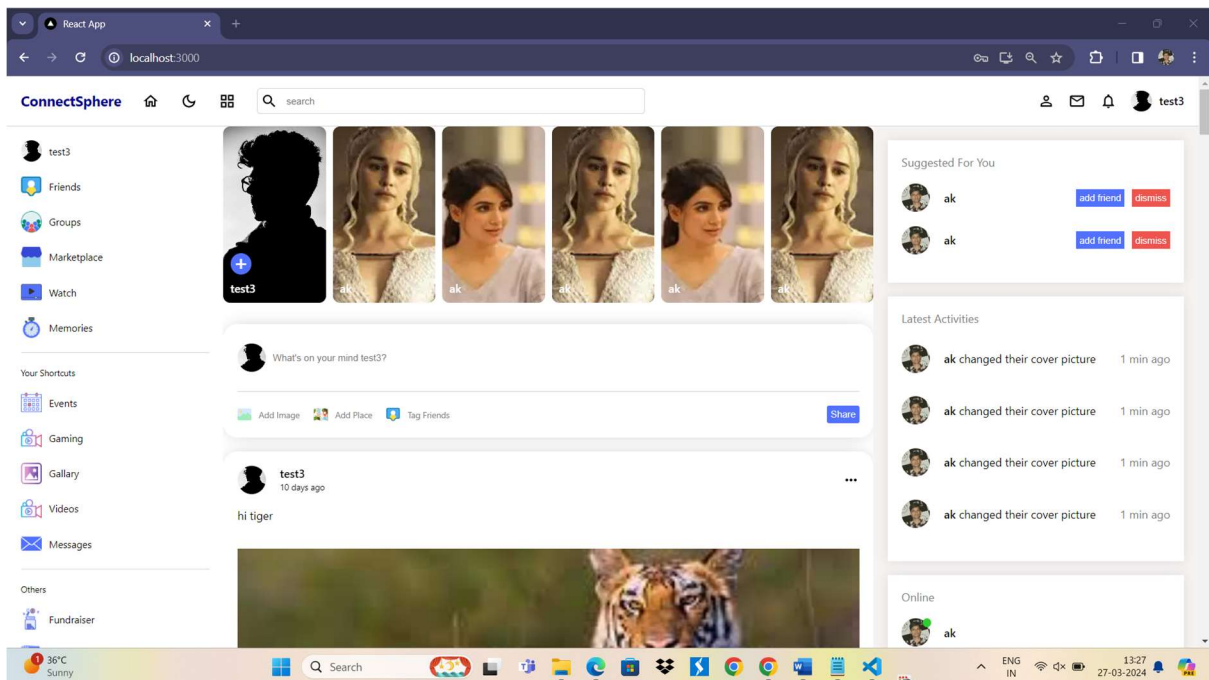


figure 6.3 Dashboard

The dashboard serves as the central hub for users to access and interact with various features and functionalities of the ConnectSphere social media platform. Designed with user convenience and engagement in mind, the dashboard provides a comprehensive overview of personalized content, social interactions, and account management options.

III. DarkMode / LightMode

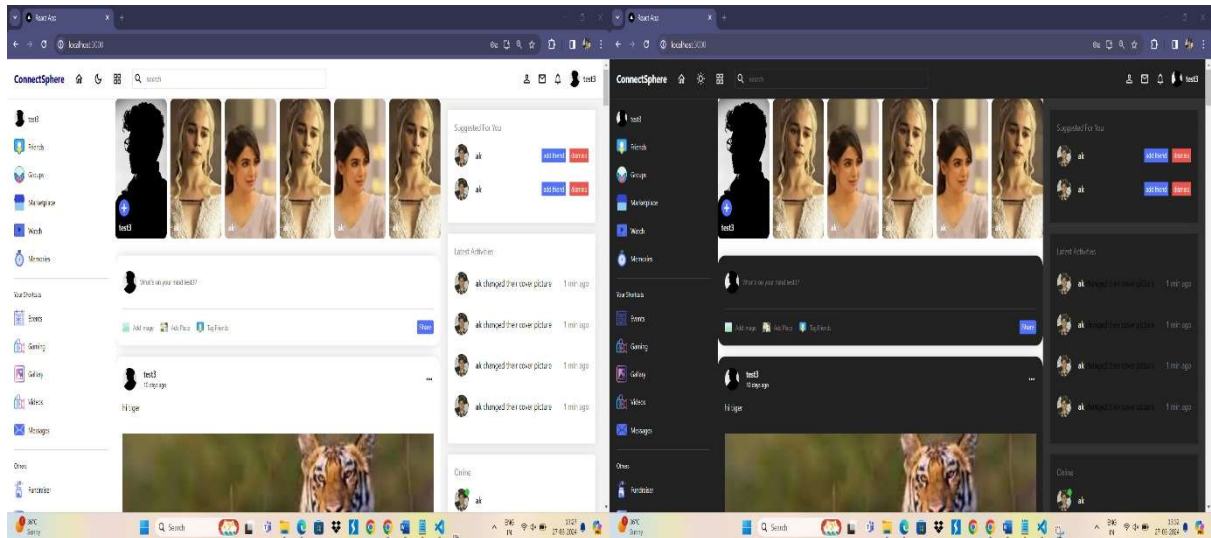


Figure 6.4 Dark/Light Mode

DarkModeContextProvider Component:

The `DarkModeContextProvider` component is responsible for managing the dark/light mode state and providing it to child components within the application. It receives children as a prop, representing the child components wrapped by the provider.

State Management:

Utilizes the `useState` hook to define a state variable `darkMode`, which stores the current mode (true for dark mode, false for light mode).

The initial state is determined by retrieving the `darkMode` value from local storage using

localStorage.getItem. If no value is found, the default state is set to false (light mode).

Toggle Functionality:

Defines a toggle function that toggles the darkMode state between true and false when called.

This function is passed down to child components via the context value, allowing them to trigger the mode toggle.

Effect Hook:

Uses the useEffect hook to persist the darkMode state to local storage whenever it changes.

This ensures that the user's preferred mode is maintained across page refreshes and sessions.

DarkModeContext Context:

Creates a context object DarkModeContext using the createContext function.

This context provides a way to pass the darkMode state and toggle function down the component tree.

Context Provider:

Wraps the child components with the DarkModeContext.Provider component, passing down

the `darkMode` state and `toggle` function as context values.

Child components can consume these values using the `useContext` hook or by accessing the context directly.

Usage:

To enable dark/light mode in your application, wrap the root component or relevant sections with the `DarkModeContextProvider`.

Child components can then access the `darkMode` state and `toggle` function via the `DarkModeContext` context, allowing them to adjust their appearance and behaviour accordingly.

IV. Update Profile

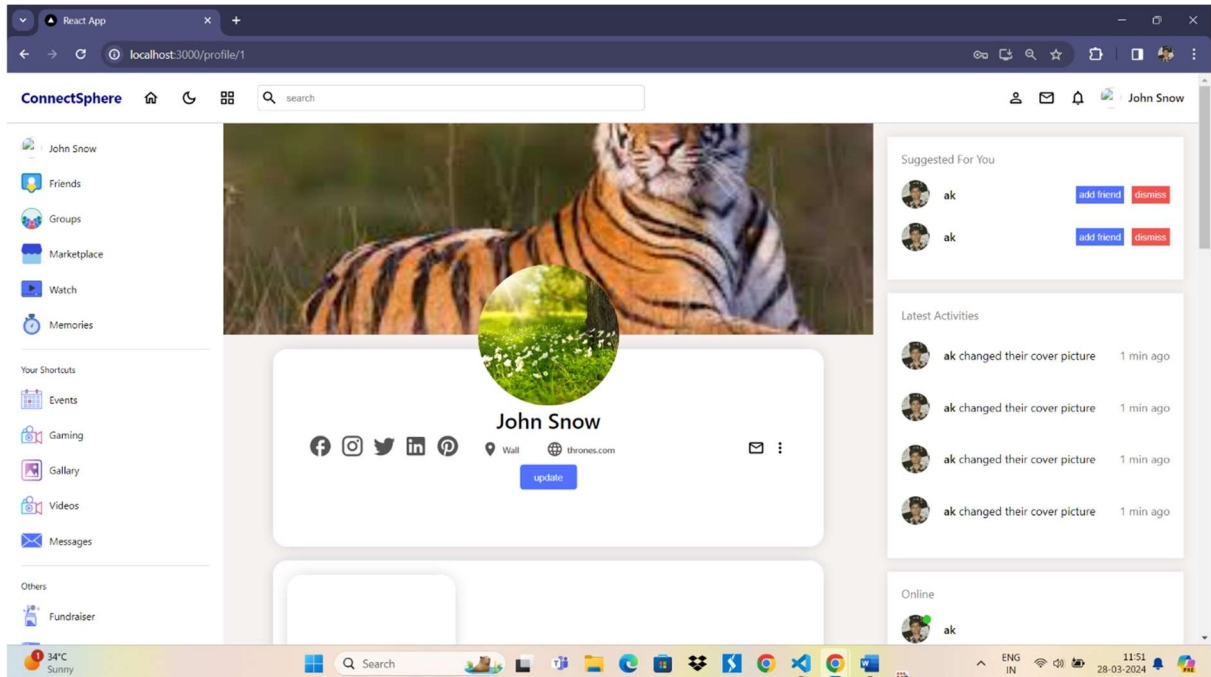


Figure 6.5 Update

The Update component is a form-based interface designed to allow users to update their profile information, including their name, city, website, profile picture, and cover picture.

V. Share Posts

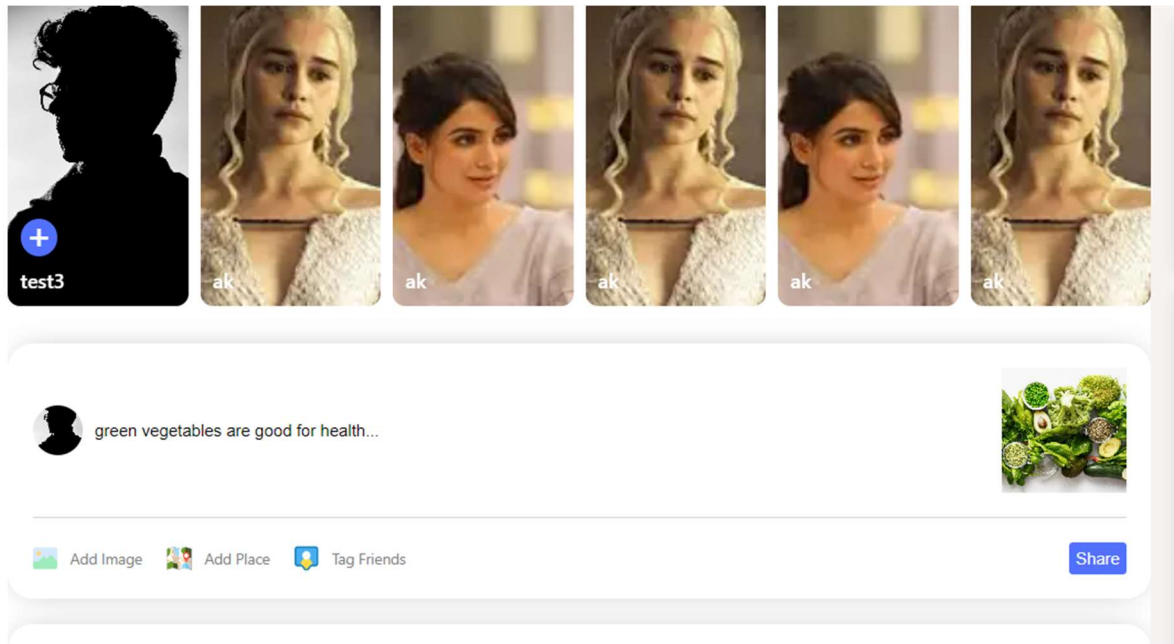


figure 6.6 Post

The Share component provides users with the functionality to create and share posts within the social media application.

Renders a share form with input fields for the post description and file upload.

Displays the current user's profile picture alongside the description input field.

Provides options for users to upload images, add location information, and tag friends to the post.

Preview the selected image file before sharing.

7. Future Improvements

Real-time Notifications: Implement a real-time notification system to alert users about new likes, comments, friend requests, and other relevant activities instantly.

Direct Messaging: Introduce a direct messaging feature to allow users to communicate privately with their connections, fostering closer relationships and facilitating seamless communication.

Advanced Search Functionality: Enhance the search functionality to enable users to discover content, users, groups, and events more efficiently. Implement filters, sorting options, and advanced search criteria to improve search accuracy and relevance.

Explore and Discover: Create an "Explore" or "Discover" section where users can explore trending topics, popular posts, recommended users, and relevant content based on their interests and activity.

Analytics and Insights: Provide users with insights and analytics tools to track their post-performance, audience demographics, engagement metrics, and overall social media presence. Enable users to gain valuable insights into their content strategy and audience engagement.

Group and Event Management: Introduce features for creating, managing, and participating in groups and events within the platform. Allow users to create public or private groups, organize events, and invite members to join.

Customizable Profiles: Enhance user profiles with customizable themes, cover photos, and bio sections. Allow users to personalize their profiles to reflect their personality and interests.

Integration with External Services: Integrate with external services and platforms such as social media sharing, authentication providers, content management systems, or productivity tools to expand functionality and improve user experience.

Accessibility Improvements: Ensure that the platform is accessible to users with disabilities by implementing accessibility features such as screen reader support, keyboard navigation, and text alternatives for multimedia content.

Security Enhancements: Strengthen security measures to protect user data, privacy, and account integrity. Implement features such as two-factor authentication, encryption, and regular security audits to mitigate risks and ensure user trust.

Gamification Elements: Introduce gamification elements such as badges, rewards, and challenges to incentivize user engagement and participation. Encourage users to interact with the platform regularly and achieve milestones.

Internationalization and Localization: Support multiple languages and regions to cater to a diverse user base. Allow users to choose their preferred language and customize content based on their location and cultural preferences.

8. Conclusion

In conclusion, ConnectSphere stands as a testament to the power of individual initiative and dedication in software development. This social media application represents a journey of innovation, learning, and persistence.

Throughout the development process, a variety of modern technologies and frameworks were employed. Leveraging the versatility of React.js for the frontend and Node.js with Express.js for the backend, ConnectSphere offers a seamless user experience with its responsive and dynamic interface. MySQL serves as the backbone of the application's data management, ensuring robustness and efficiency.

The core features of ConnectSphere include user authentication powered by JSON Web Tokens (JWT), profile management, post creation, and interactive social interactions. Through diligent implementation and continuous refinement, these features have been seamlessly integrated to deliver a cohesive and engaging user experience.

While the journey was not without its challenges, each obstacle served as an opportunity for growth and learning. Overcoming technical complexities, ensuring smooth integration of various components, and refining

the user interface were among the key challenges encountered and successfully addressed.

Looking ahead, ConnectSphere holds immense potential for further enhancement and expansion. Future iterations may include the introduction of real-time notifications, direct messaging capabilities, and advanced search functionalities to enrich the user experience and foster deeper engagement within the platform.

As the sole creator of ConnectSphere, I am immensely proud of the accomplishments achieved thus far. This project not only showcases my technical skills and expertise but also demonstrates my passion for innovation and commitment to delivering high-quality software solutions.

In closing, ConnectSphere stands as a testament to the power of individual creativity and determination in the realm of software development. With its foundation laid and its potential limitless, ConnectSphere represents not just a project, but a journey of growth, exploration, and possibility.