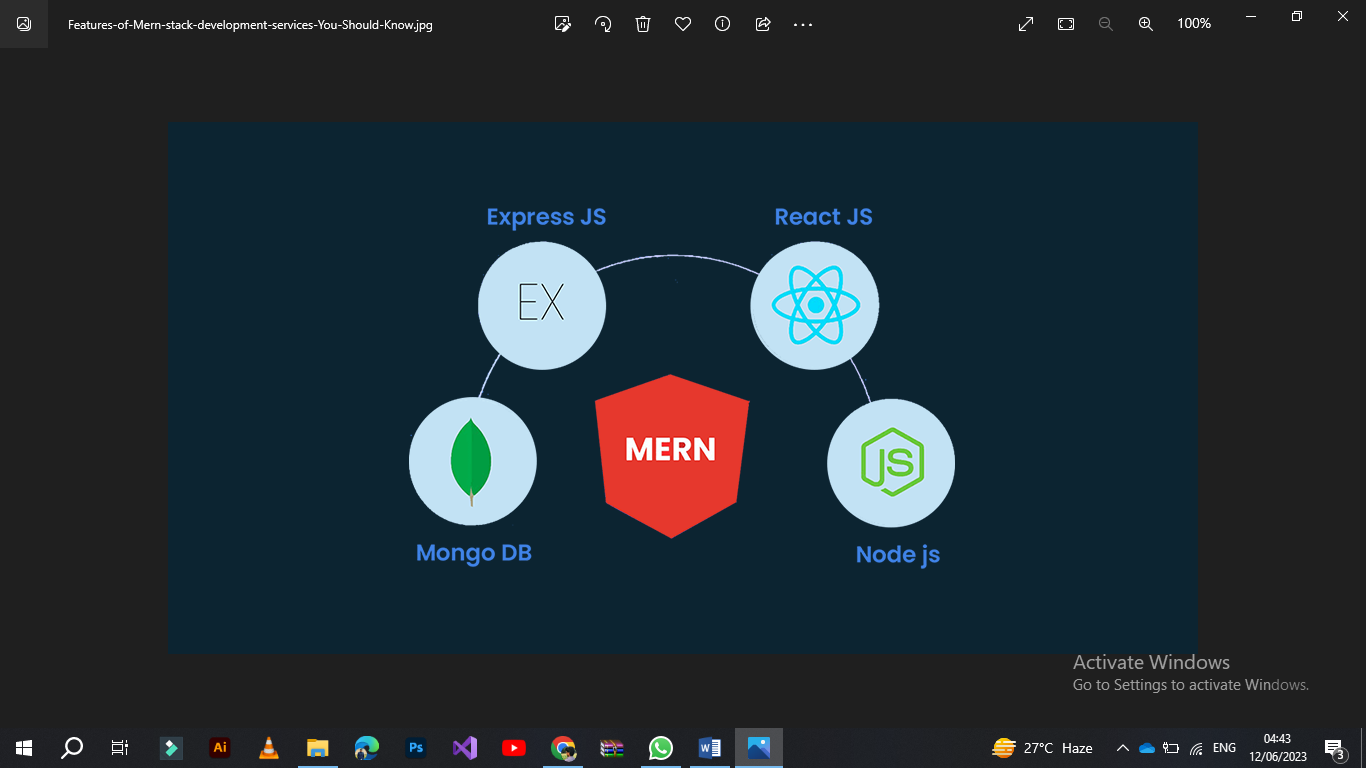
|  |
| --- |
|  |
| BULINK DEVLOPMENT  SOCIAL MEDIA USING MERN STACK |
| |  |  |  | | --- | --- | --- | | MOEEZ MUSTAFA 01-134212-088  SYED MUJADIL AHAMD 01-134212-177 | 6/11/23 | DATABASE MANAGEMENT SYSTEM | |

**1. Introduction**

The University Social Media Platform is a web-based application built using the MERN stack, which stands for MongoDB, Express.js, React.js, and Node.js. The platform aims to provide a centralized space for students, faculty, and staff to connect, share information, and collaborate within the university community. It offers various features such as user profiles, news feeds,messaging, and more.



This documentation provides an overview of the system, its architecture, the technologies used, a detailed description of its features, user roles, installation and deployment instructions, a usage guide, and potential future enhancements.

**2. System Overview**

The University Social Media Platform is designed to create an online environment that fosters communication and collaboration among university members. It enables users to create profiles, connect with others, share posts and updates. The platform provides a user-friendly interface and intuitive navigation to enhance the user experience.

**3. Architecture**

The University Social Media Platform follows a client-server architecture. The client-side is built with React.js, which handles the user interface and interactions. The server-side is implemented using Node.js and Express.js, providing the application logic and APIs for data retrieval and manipulation. MongoDB, a NoSQL database, is used for storing and retrieving data.

The architecture follows a React.redux, API approach, where the frontend communicates with the backend through HTTP requests and JSON payloads. The server handles these requests, performs necessary operations on the MongoDB database, and sends back appropriate responses to the client.

**4. Technologies Used**

The University Social Media Platform is built using the following technologies:

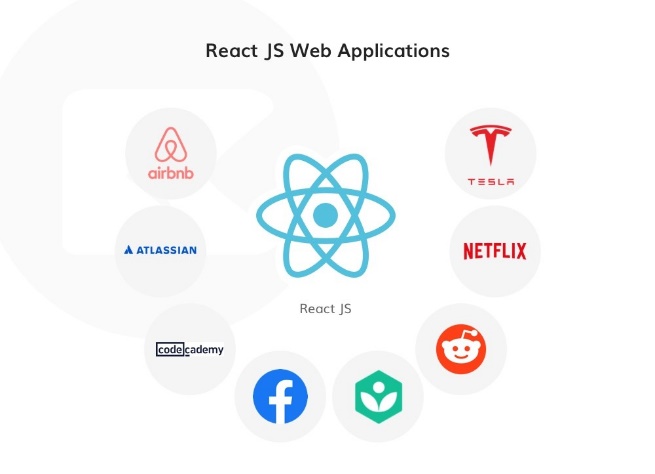
* MongoDB: A NoSQL database used for data storage.

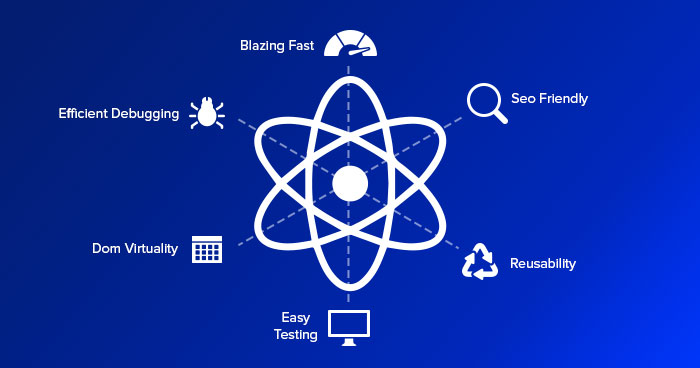


* Express.js: A web application framework for Node.js, used to build the backend.

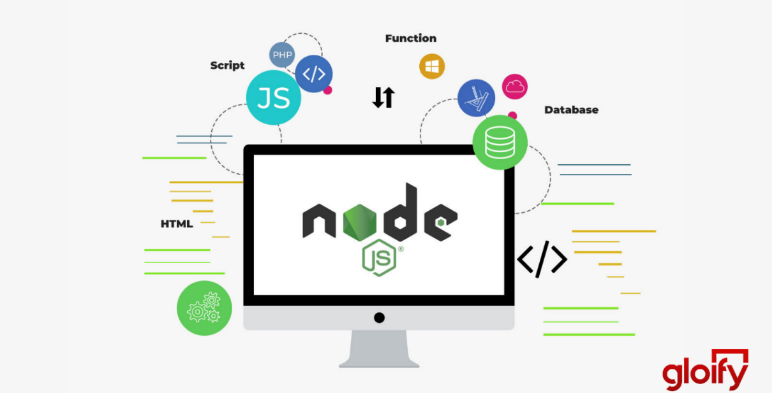


* React.js: A JavaScript library used for building the user interface.





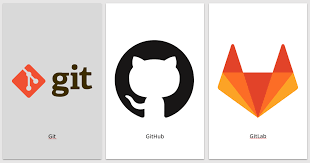
* Node.js: A JavaScript runtime environment used for server-side development.



* JavaScript: The primary programming language used for both frontend and backend development.
* React Redux: The architecture follows the principles of react redux for handing that state and communication between frontend and backend.



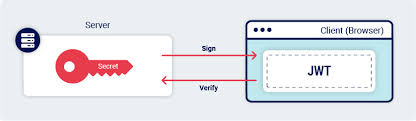
* Git: Version control system for tracking changes and collaborating with a development team.



* NPM: Package manager for installing and managing project dependencies.



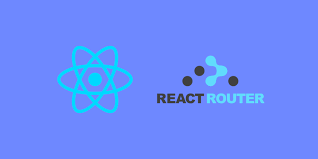
* JWT: JSON Web Token allows the application to authenticate users and authorize access to resources without maintaining a session state on the server.



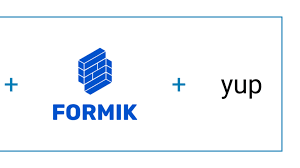
* Mullter: Mullter is a middleware for handling multipart/form-data, which is commonly used for uploading files in web applications.



* React Router: popular library for handling navigation and routing in React applications. It allows you to create declarative and dynamic routing in your application, enabling navigation between different views or components based on the current URL.



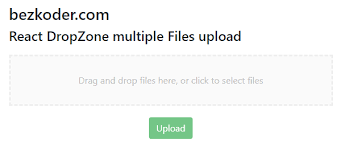
* Formik &Yup: are two popular libraries in the React ecosystem used for form management and form validation, respectively.



* Redux toolkit: Redux team that provides a set of utilities and abstractions to write and maintain Redux code.
* Redux Persistent: save the Redux store's state to a storage mechanism (such as localStorage or AsyncStorage) and retrieve it when the application reloads or restarts.



* React dropzone: React library that provides a drag-and-drop file uploader component.



**5. Features**

* The University Social Media Platform offers the following features:
* User Registration and Login: Users can create accounts with their email addresses and passwords and securely log in to the platform.
* User Profiles: Users can create and manage their profiles, including personal information, profile pictures, and interests.
* News Feed: Users can view and interact with a personalized news feed that displays posts, updates, and events from users and groups they follow.
* Posts and Comments: Users can create posts, comment on existing posts, and like or dislike posts and comments.
* Notifications: Users receive real-time notifications for activities such as new followers, comments on posts, event invitations, and group updates.
* Search Functionality: Users can search for other users, groups, events, or specific posts using relevant keywords.
* Privacy and Security: The platform ensures the privacy and security of user data through secure authentication mechanisms and access controls.

**6. User Roles**

* The University Social Media Platform supports the following user roles:
* Student: Students are the primary users of the platform. They can access all features, including creating profiles, connecting with others and providing news for events.
* Faculty: Faculty members have similar privileges as students but may have additional administrative capabilities, such as creating and managing news of events.
* Staff: Staff members can access the platform and participate in various activities, but they may have limited privileges compared to students and faculty.
* Administrator: Administrators have elevated privileges and can manage the overall system, including user management, system configuration, and content moderation.

**7. Installation and Deployment**

To install and deploy the University Social Media Platform, follow these steps:

1. Install Node.js and MongoDB on the server.
2. Clone the project repository from the provided source.
3. Navigate to the project directory and run the command **npm install** to install project dependencies.
4. Create a **.env** file in the root directory and configure environment variables such as database connection details, API keys, and secret keys.
5. Run the command **npm run build** to build the frontend React.js application.
6. Run the command **npm start** to start the server.
7. The application should be accessible on the specified port. Configure any necessary reverse proxy or load balancer for production deployment.

**8. Usage Guide**

Once the University Social Media Platform is installed and running, users can access it using a web browser. The following steps provide a high-level usage guide:

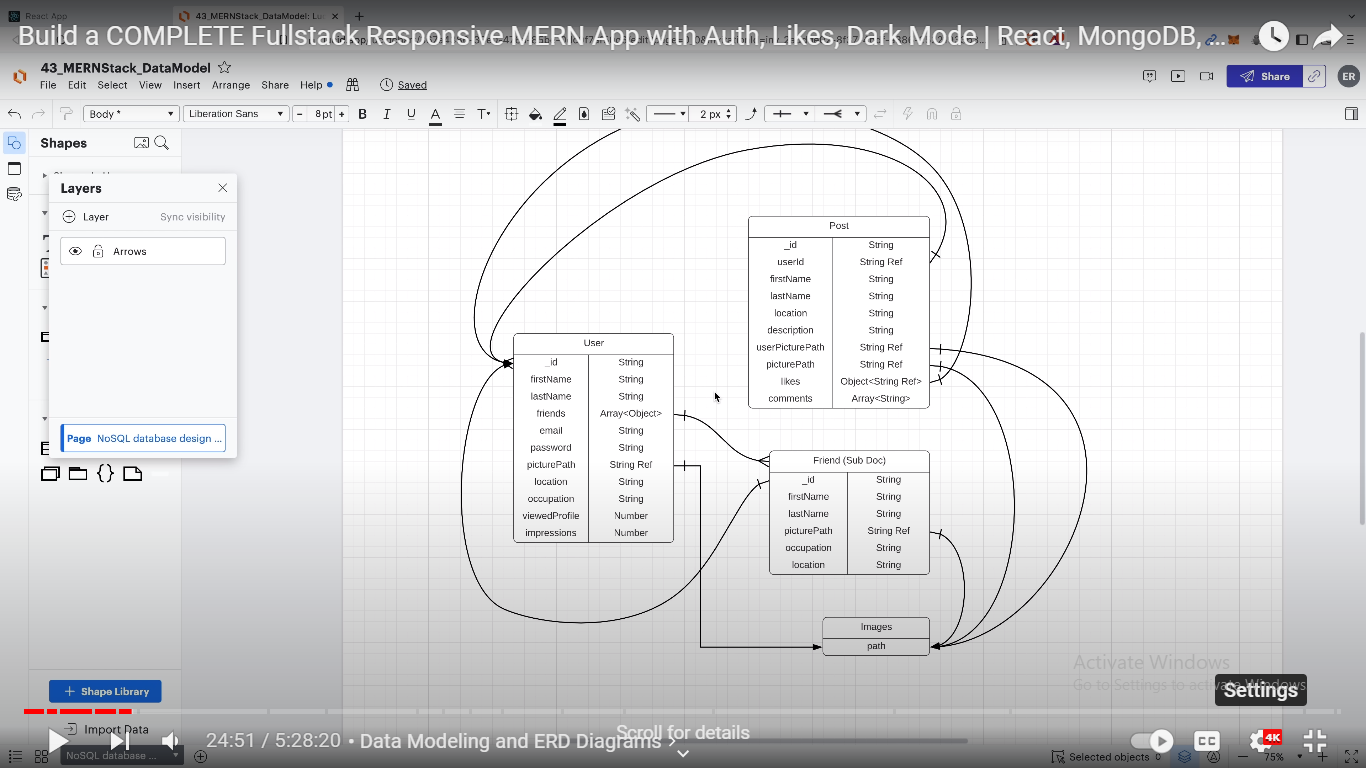
* Register an account using a valid email address and password.
* Log in to the platform using the created credentials.
* Create and complete your user profile, including personal information, profile picture, and interests.
* Explore the news feed to view posts, updates, and events from other users and groups.
* Follow users or join groups of interest to see their content in your news feed.
* Create posts to share updates, ask questions, or share resources with the community.
* Comment on posts to engage in discussions and provide feedback.
* Join existing groups or create your own groups to connect with like-minded individuals.
* Create, manage, or participate in events to stay informed about university activities.
* Use the messaging feature to collaborate and communicate with other users privately.
* Configure notification preferences to receive updates and alerts based on your preferences.
* Utilize the search functionality to find specific users, groups, events, or posts.
* Log out from the platform when you're finished.

**9. Future Enhancements**

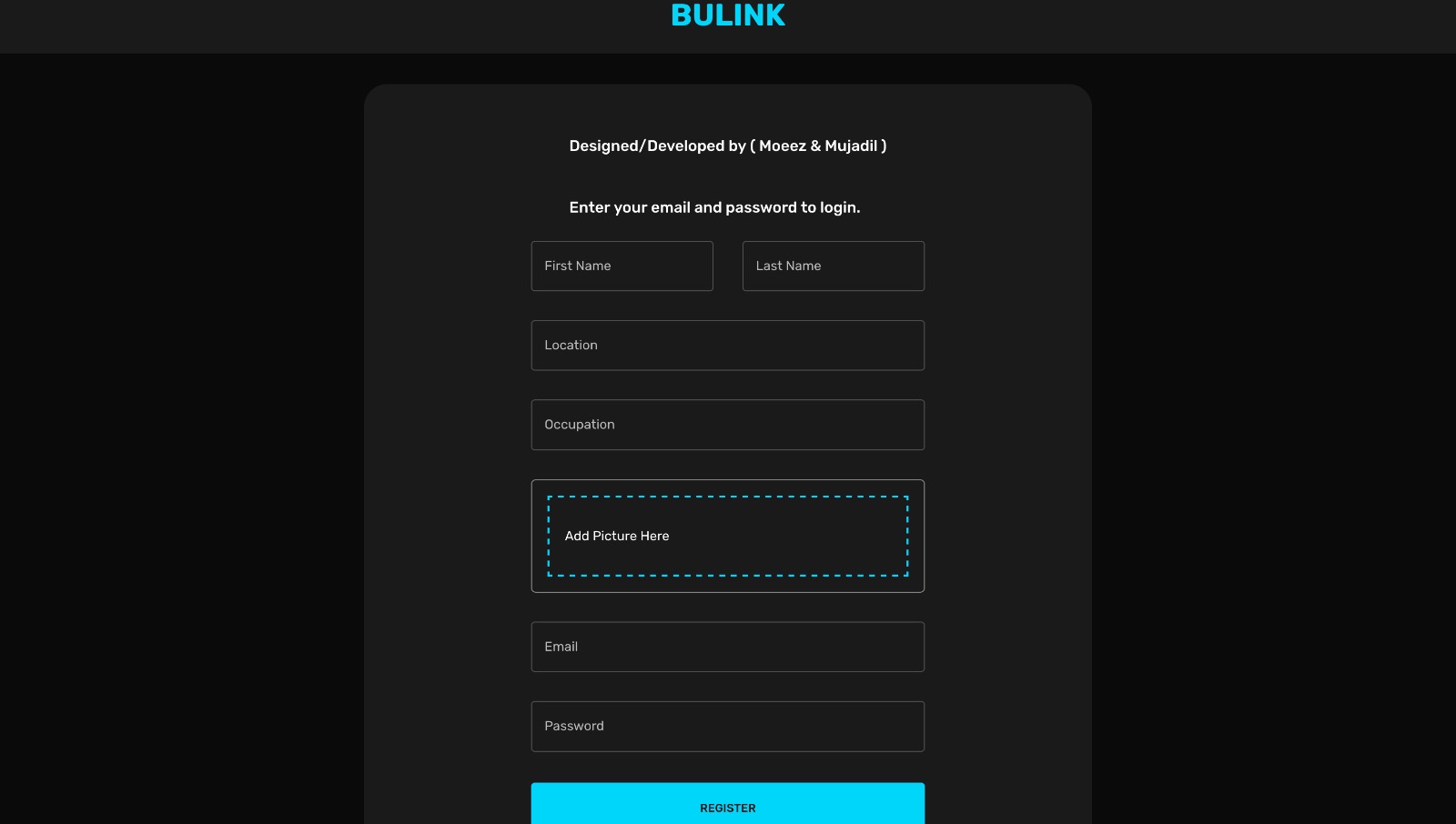
The University Social Media Platform has great potential for future enhancements. Some possible features and improvements include:

* Integration with other university systems, such as course management or grading systems.
* Enhanced analytics and reporting capabilities to provide insights into user engagement and platform usage.
* Integration with external APIs for additional functionalities, such as integrating with popular social media platforms or online resources.
* Integration of real-time collaboration tools, such as document editing or video conferencing.
* Mobile application development for better accessibility and user experience on mobile devices.
* Users will join existing groups or create their own groups based on interests, courses, or departments to facilitate collaboration and discussions.
* Users will create, manage, and participate in events happening within the university community.
* Continuous performance optimization to handle a growing user base and increasing data volume.Additional privacy and security features, such as two-factor authentication and data encryption.

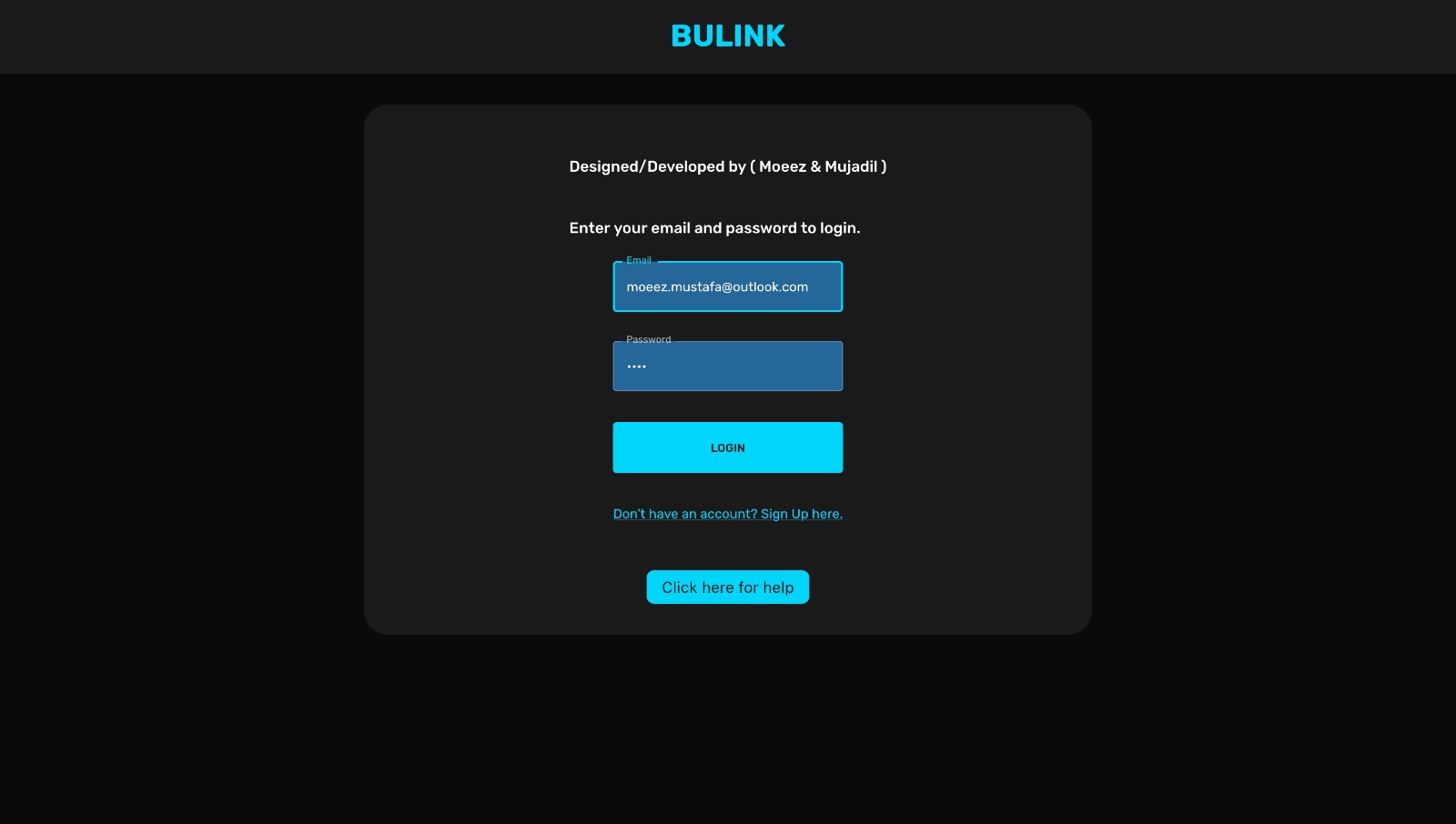
**10. ER Modeling**



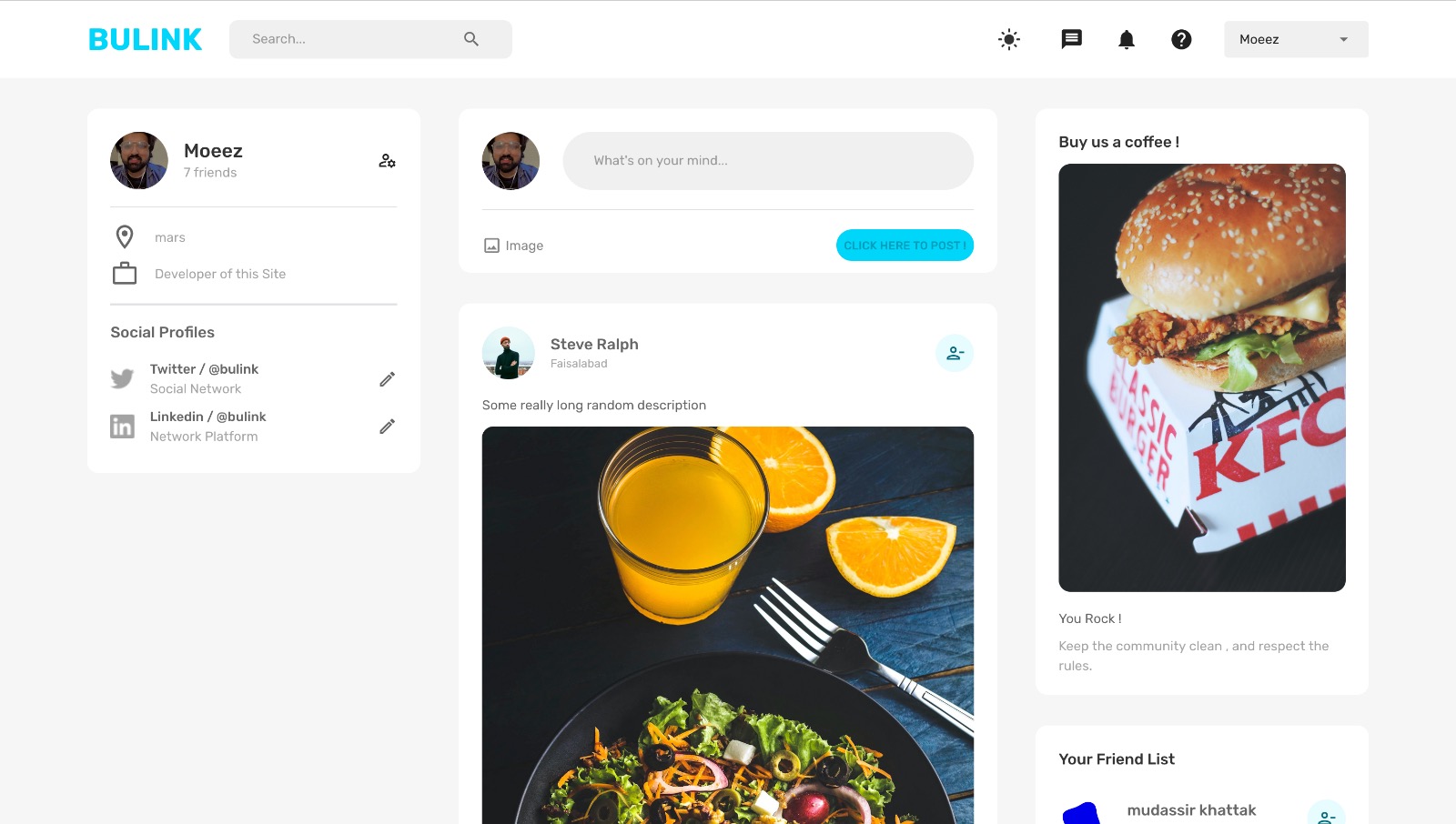
**11.ScreenShot**



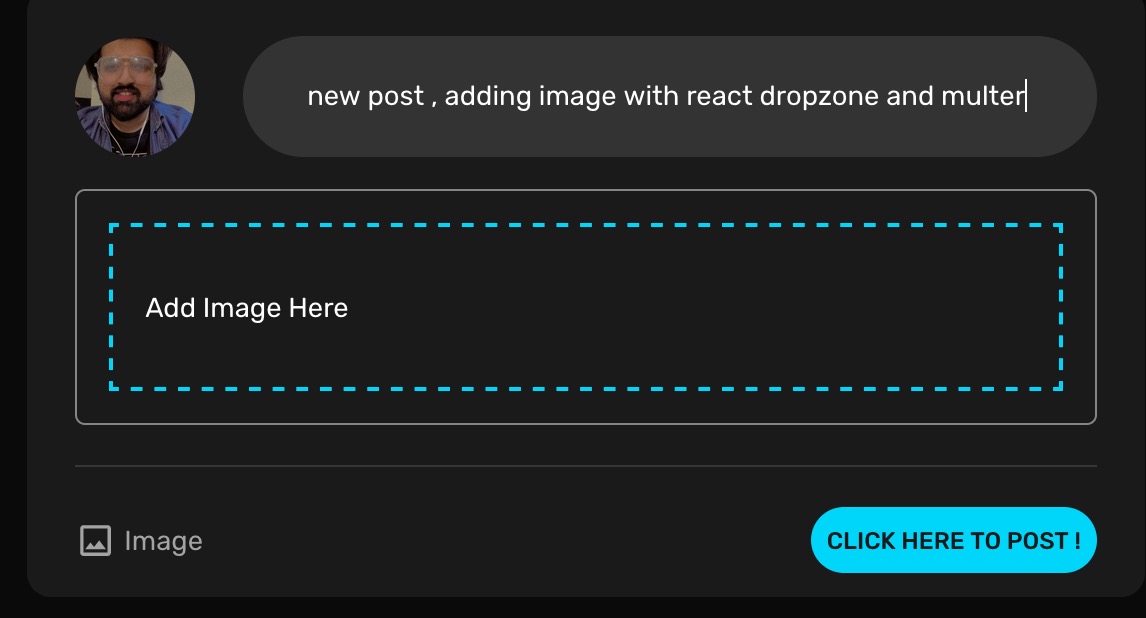
**Figure: Signup page**

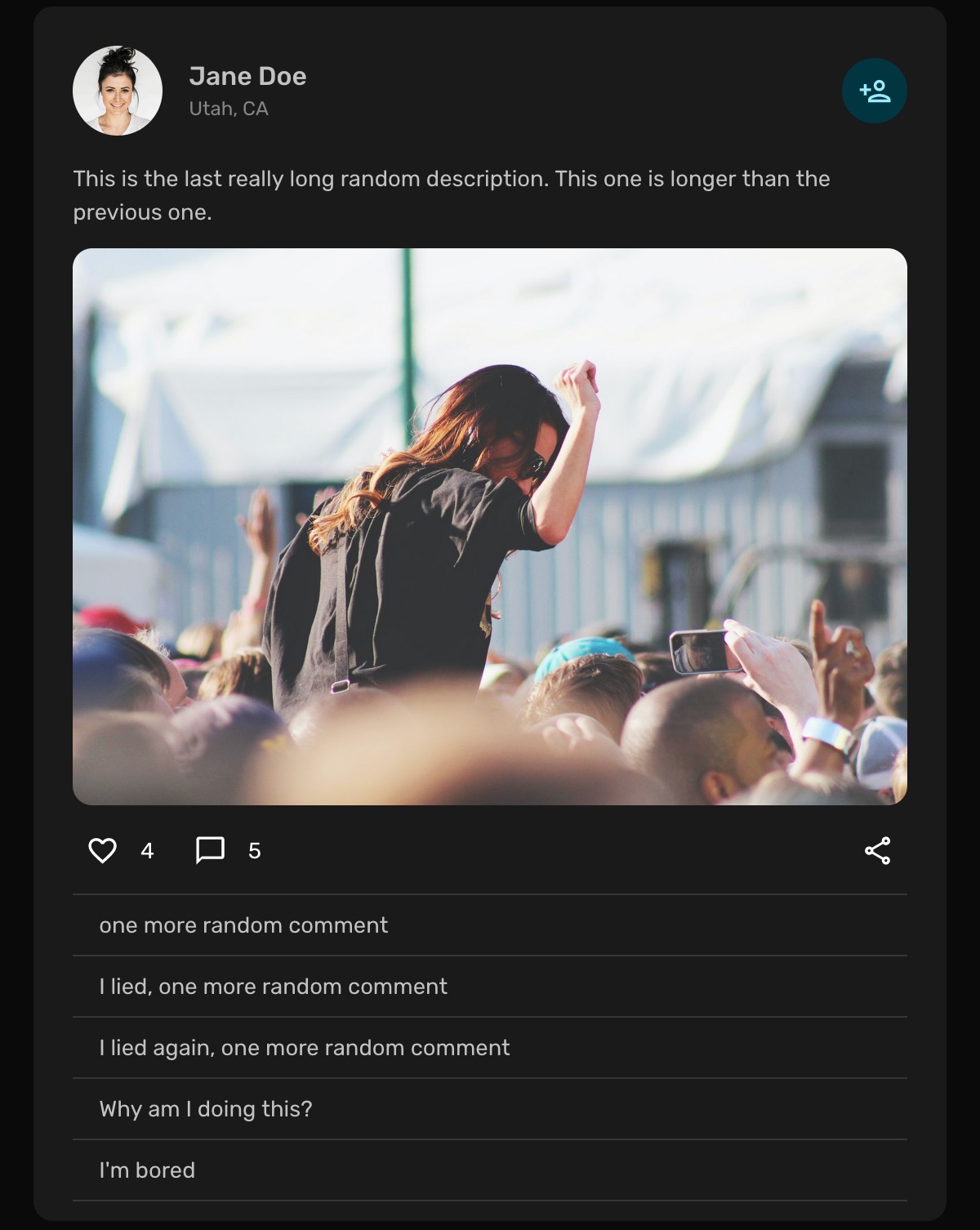


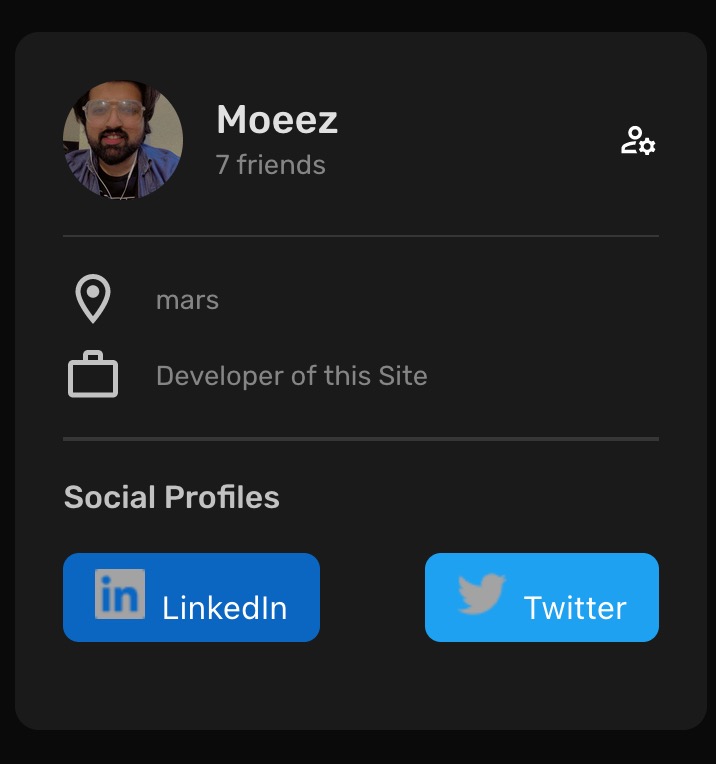
**Figure: Login page**



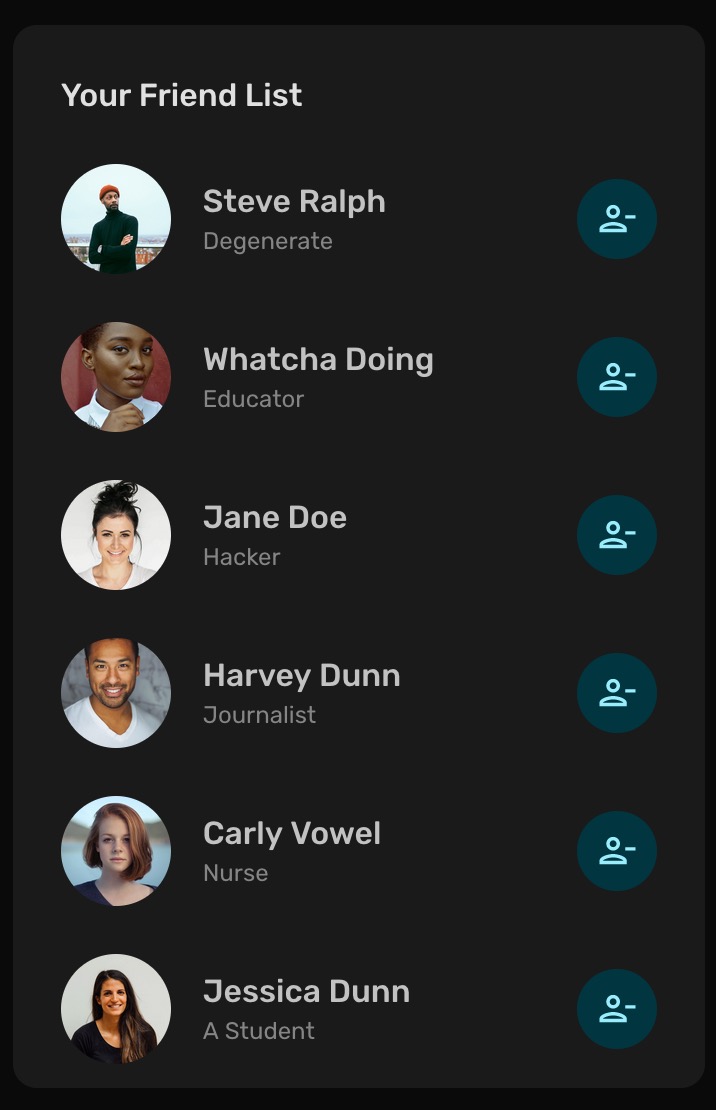
**Figure: Main page**



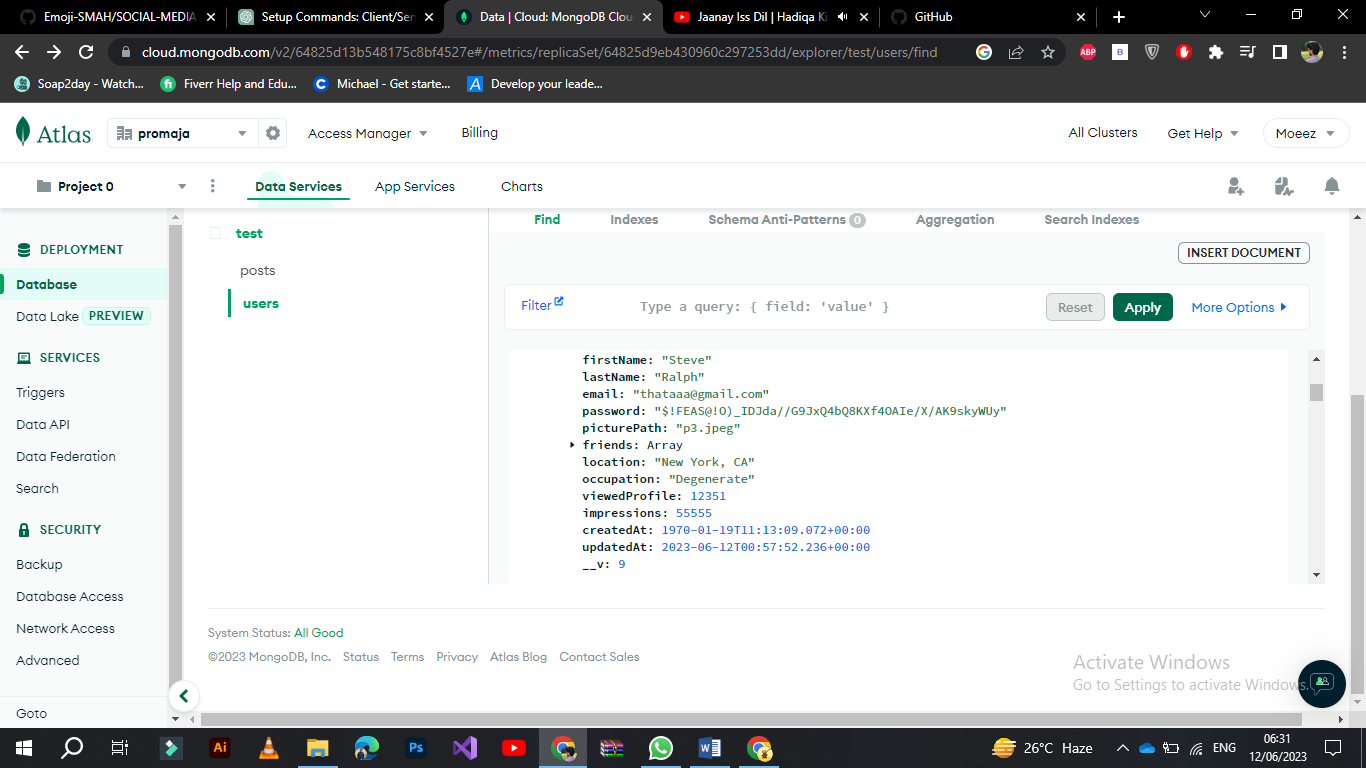
**Figure: Add Feed & Feed**



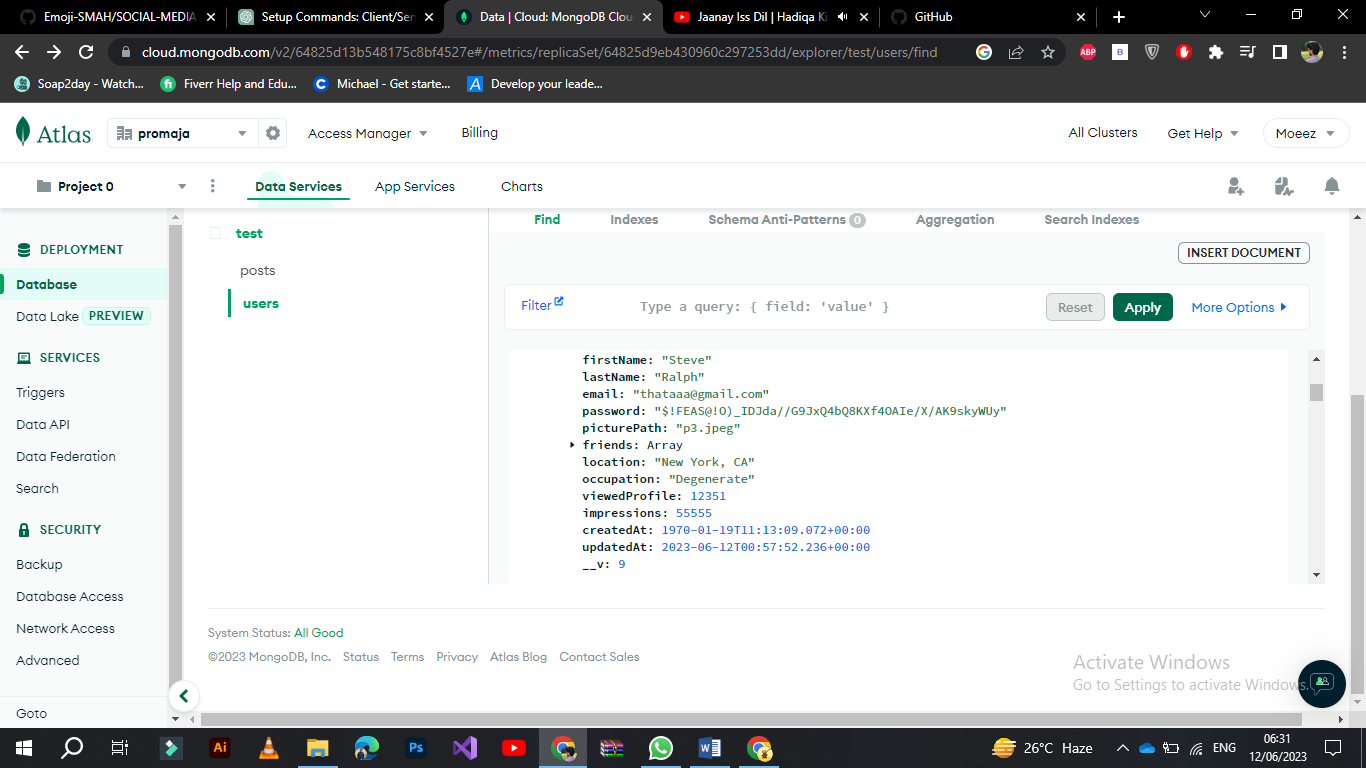
**Figure: Profile Details**



**Figure: Friend List**



**Figure: Database**



**Figure: Database Curd**

**12. Conclusion**

The University Social Media Platform offers a powerful tool for fostering communication and collaboration within the university community. With its comprehensive features, user-friendly interface, and scalable architecture, the platform serves as a centralized hub for students, faculty, and staff to connect, share information, and engage in meaningful interactions. By leveraging the MERN stack and various technologies, the platform provides a robust foundation for further development and customization to meet the specific needs of any university.

**THE END**