**# BuzzLine**

BuzzLine is a Full Stack Chatting App.

Uses Socket.io for real time communication and stores user details in encrypted format in Mongo DB Database.

**## Tech Stack**

**\*\*Client:\*\*** React JS

**\*\*Server:\*\*** Node JS, Express JS

**\*\*Database:\*\*** Mongo DB

**## Run Locally**

Clone the project

```bash

  git clone https://github.com/piyush-eon/mern-chat-app

```

Go to the project directory

```bash

  cd mern-chat-app

```

Install dependencies

```bash

  npm install

```

```bash

  cd frontend/

  npm install

```

Start the server

```bash

  npm run start

```

Start the Client

```bash

  //open now terminal

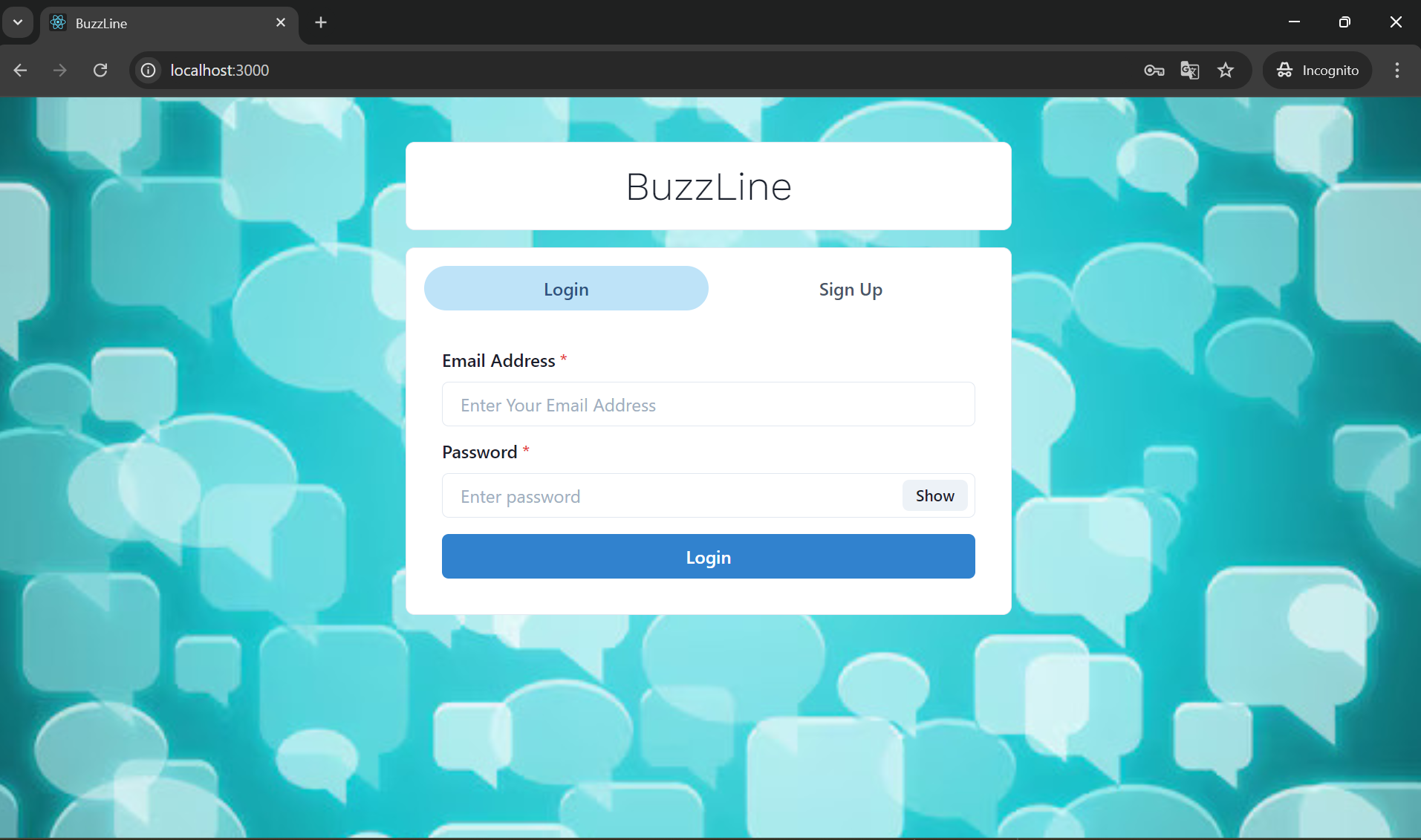
  cd frontend

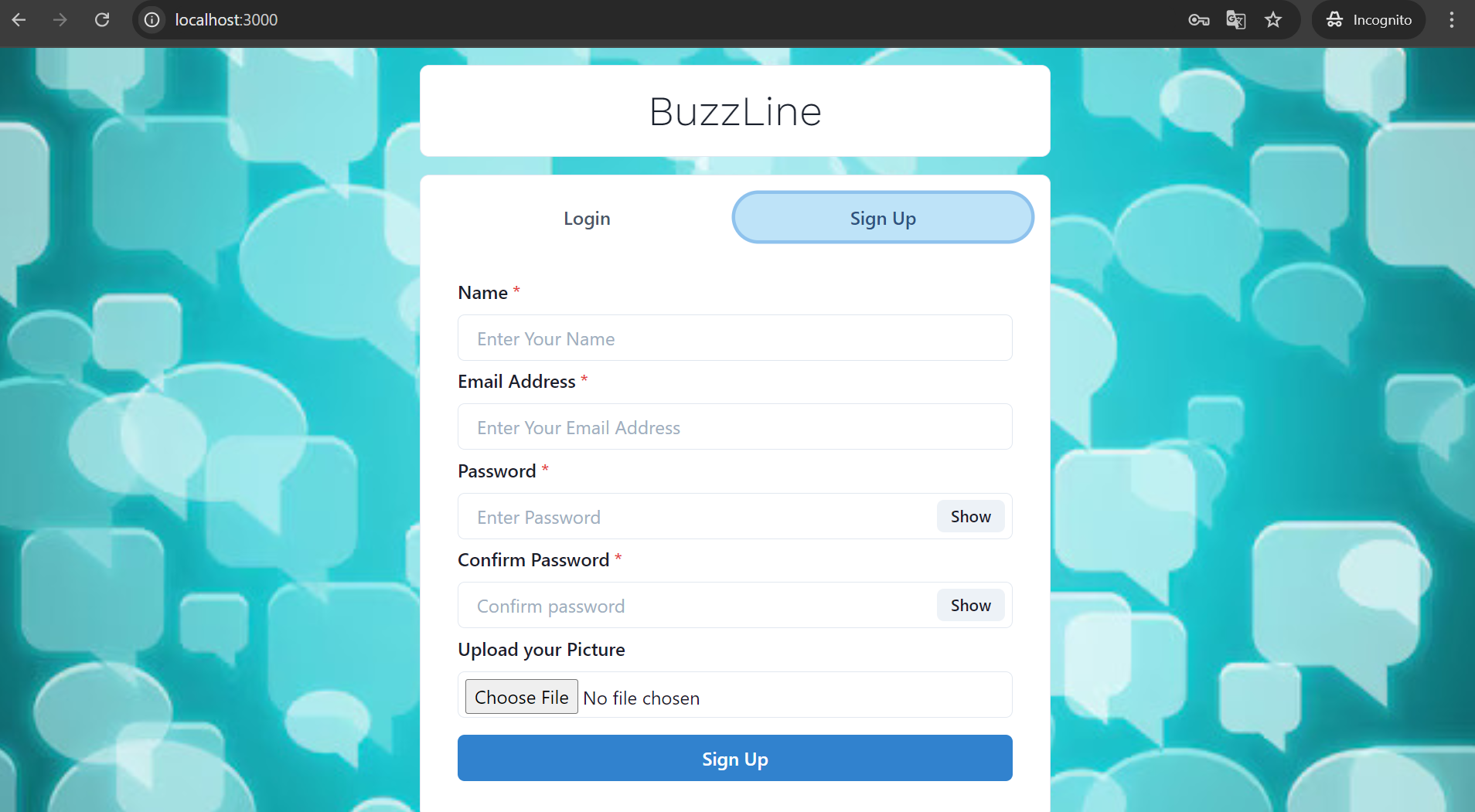
  npm start

```

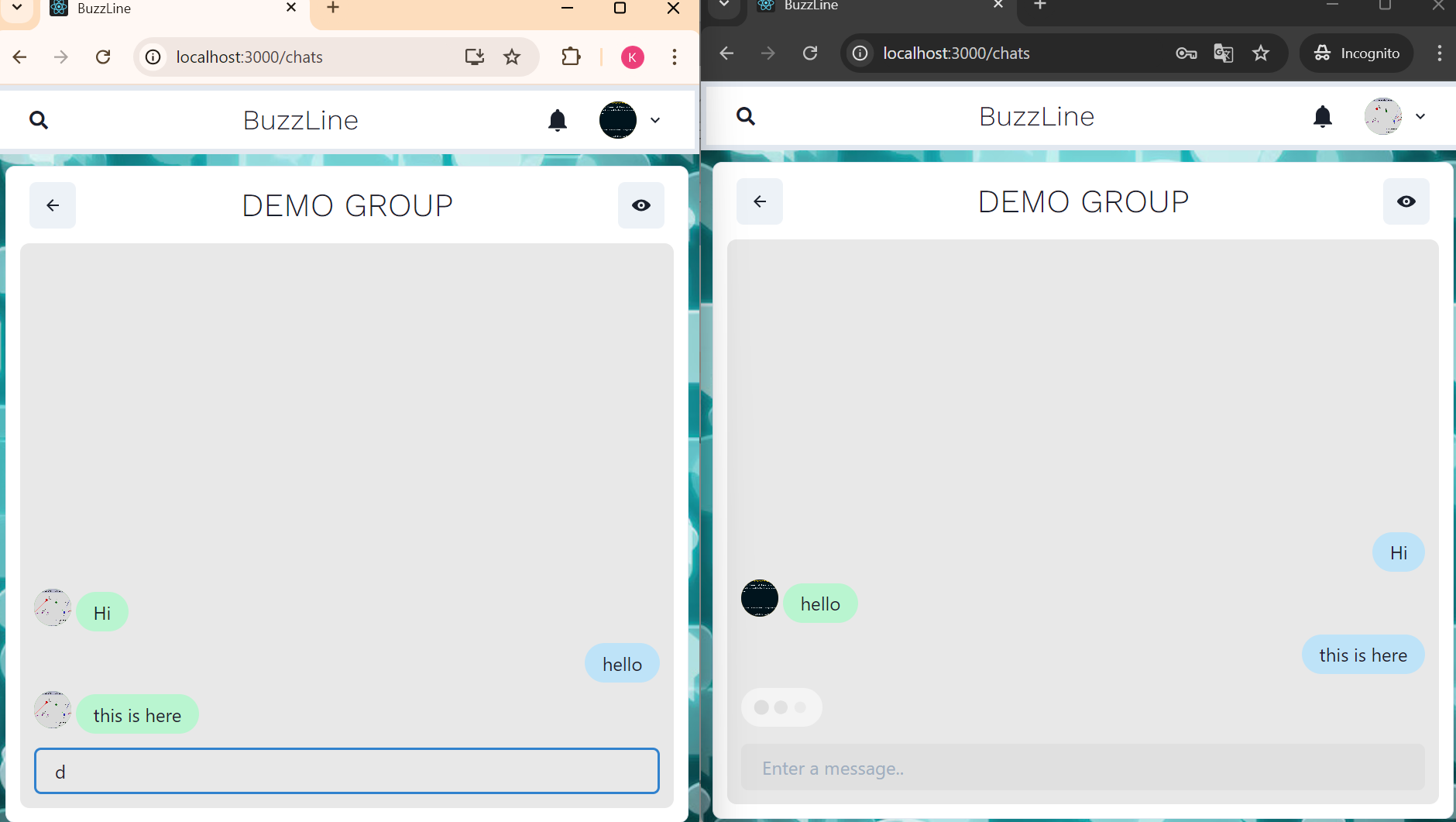
**# Features**

**### Authenticaton**

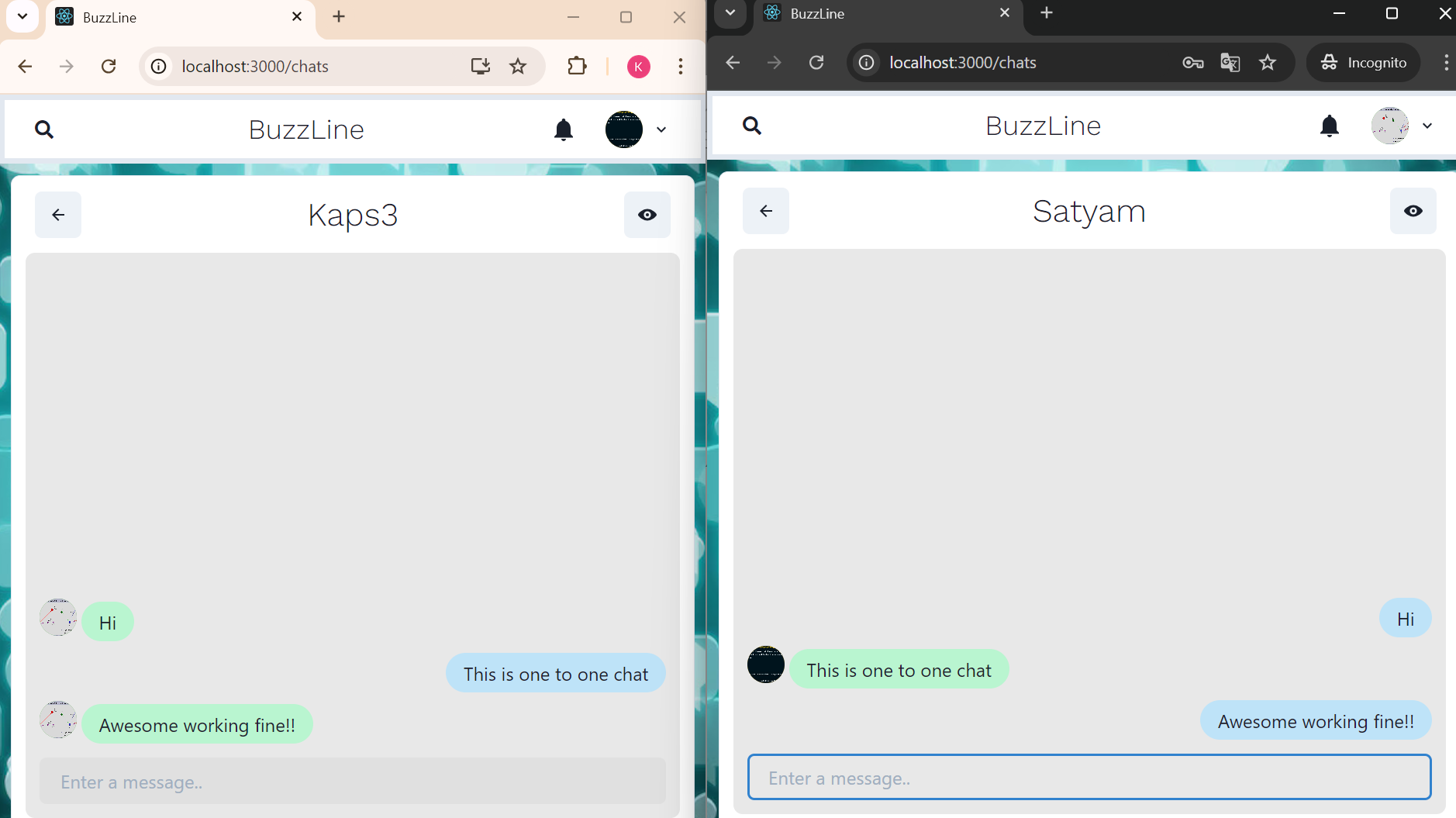




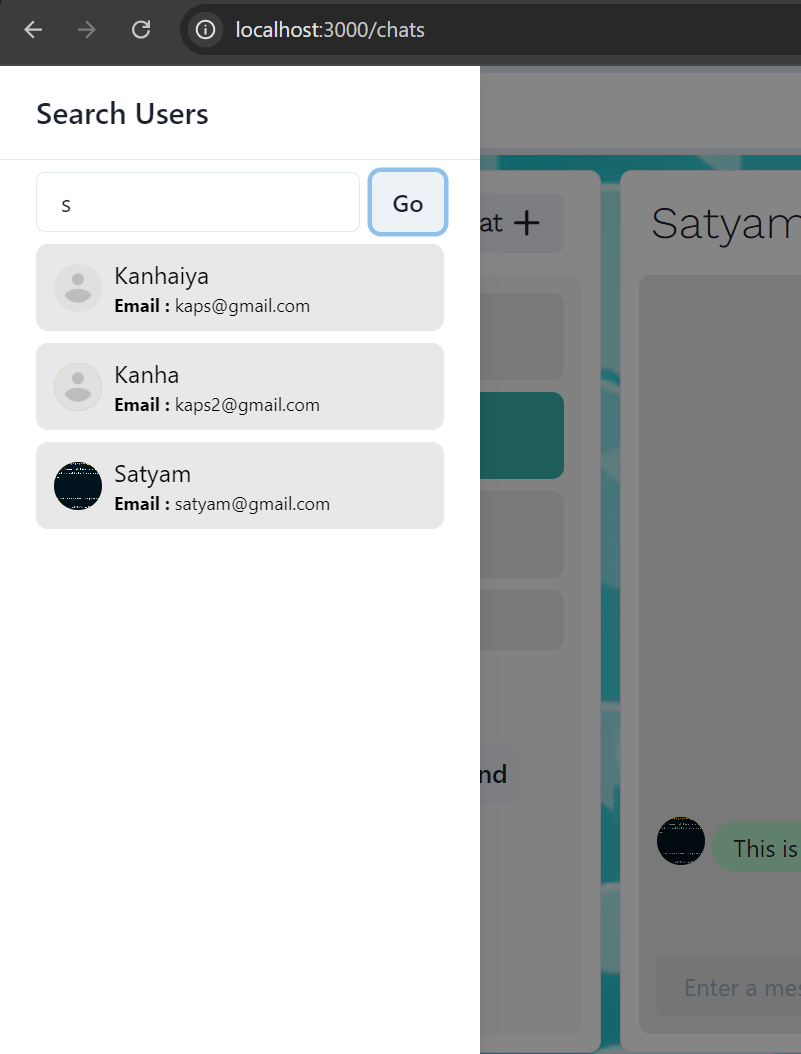
**### Real Time Chatting with Typing indicators**



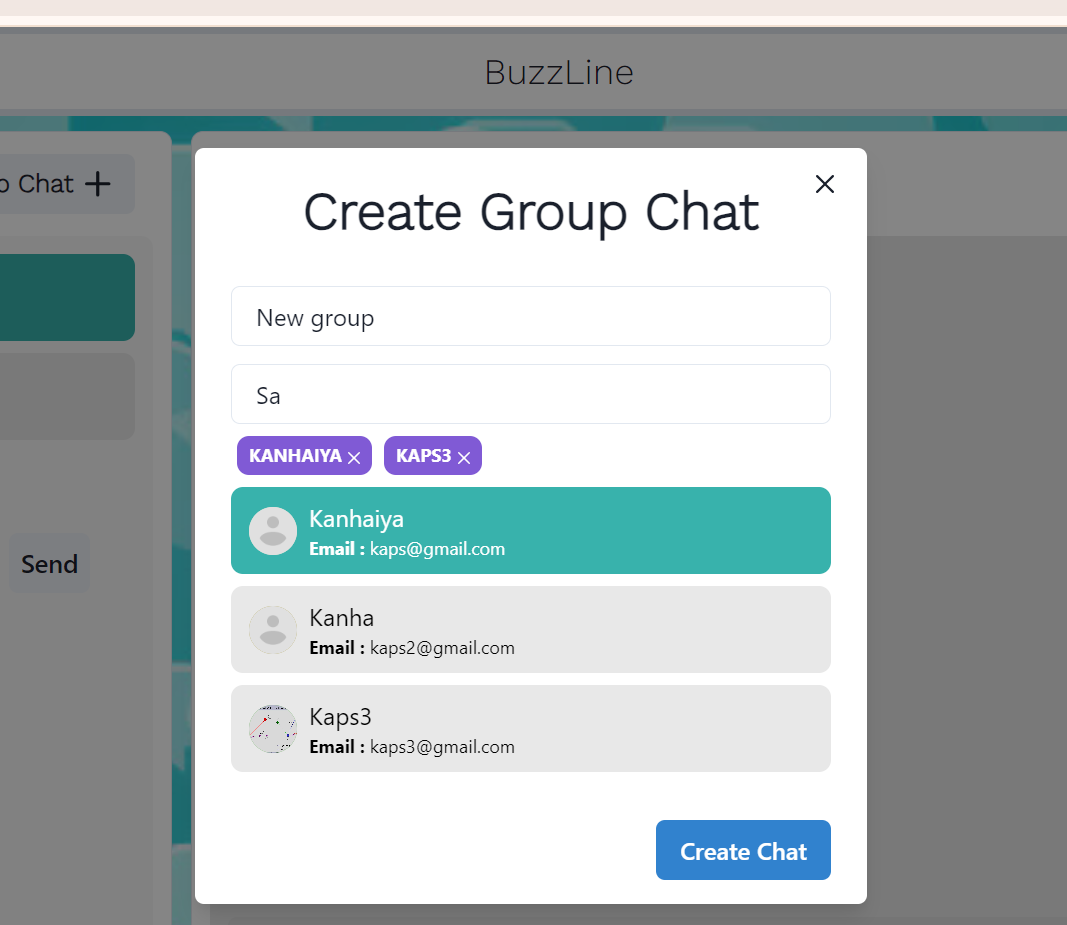
**### One to One chat**



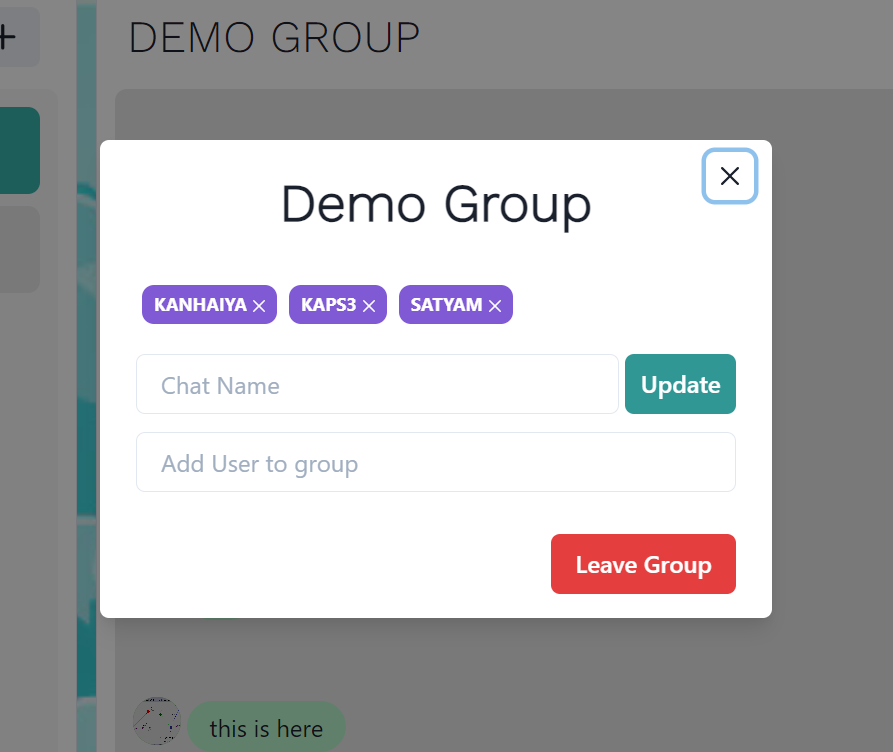
**### Search Users(If name and email contains searched keyword)**



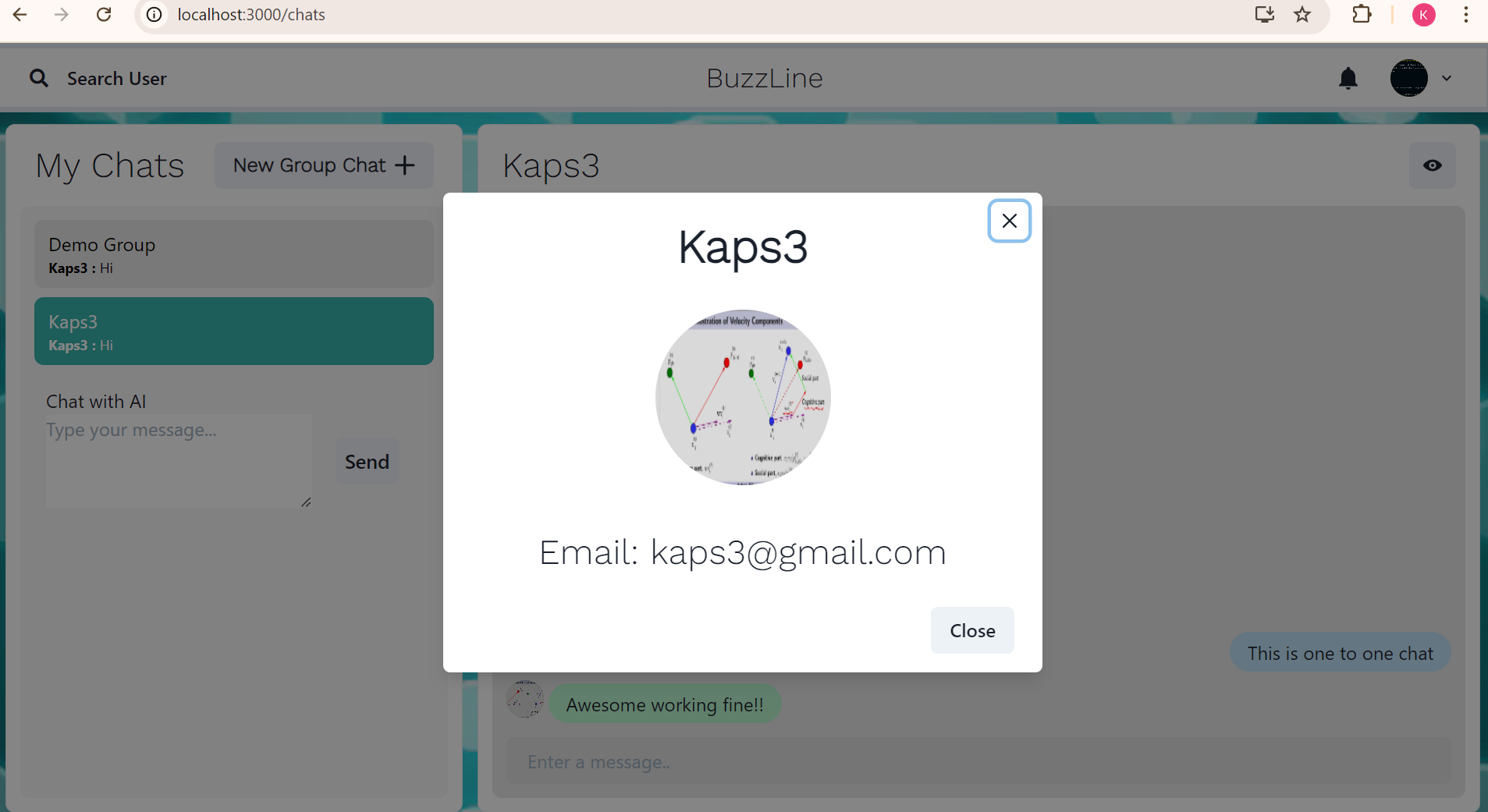
**### Create Group Chats**



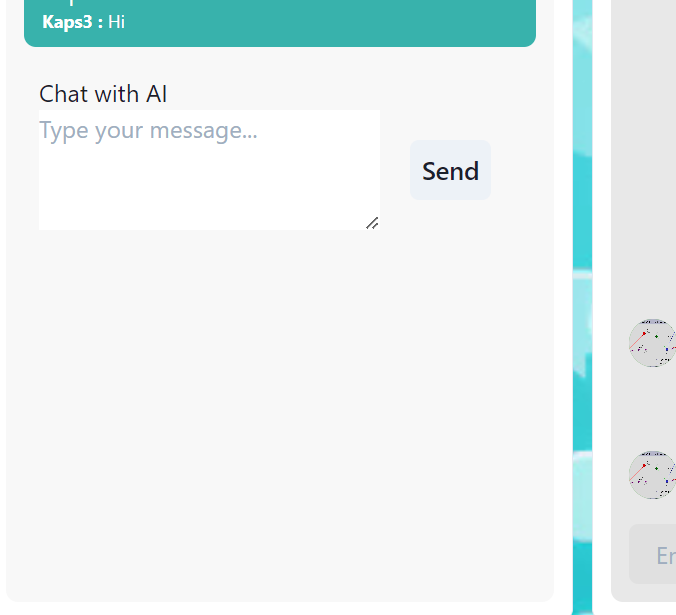
**### Add or Remove users from group**



**### View Other user Profile**



**### AI ChatBot using openAI**

****

\*\*For backend I have used following dependencies::

### 1. **\*\*bcryptjs\*\* (`^2.4.3`)**

- \*\*Purpose\*\*: Used for \*\*password hashing\*\* and verification.

- \*\*Why it's used\*\*: In modern applications, user passwords should never be stored in plain text. `bcryptjs` helps securely hash passwords before storing them in the database and allows you to later compare the hash with a user-provided password during login. It's a JavaScript implementation of bcrypt that works seamlessly in Node.js environments.

### 2. **\*\*colors\*\* (`^1.4.0`)**

- \*\*Purpose\*\*: Adds color to the terminal output (console logs).

- \*\*Why it's used\*\*: Makes console logging more visually distinguishable by adding colors, which can be helpful for \*\*debugging\*\* and \*\*development\*\*. For example, errors can be shown in red, while successful operations can be shown in green, improving readability during development.

### 3. **\*\*dotenv\*\* (`^9.0.2`)**

- \*\*Purpose\*\*: Loads environment variables from a `.env` file into `process.env`.

- \*\*Why it's used\*\*: Allows for secure and flexible \*\*environment configuration\*\* (like API keys, database URIs, and other sensitive information) without hardcoding them into the codebase. It's essential for differentiating between environments like development, staging, and production.

### 4. **\*\*express\*\* (`^4.17.1`)**

- \*\*Purpose\*\*: A web framework for Node.js.

- \*\*Why it's used\*\*: \*\*Express.js\*\* is a lightweight and flexible framework that simplifies the process of setting up web servers, routing requests, and handling middleware. It is the backbone of many Node.js applications due to its simplicity and extensibility.

### 5. **\*\*express-async-handler\*\* (`^1.1.4`)**

- \*\*Purpose\*\*: Simplifies error handling in asynchronous Express routes.

- \*\*Why it's used\*\*: When using async/await in Express routes, unhandled promise rejections can crash the server. `express-async-handler` provides a simple way to handle these errors without wrapping each route handler in try-catch blocks. This improves \*\*code readability\*\* and \*\*error management\*\*.

### 6. **\*\*jsonwebtoken\*\* (`^8.5.1`)**

- \*\*Purpose\*\*: For generating and verifying \*\*JWT (JSON Web Tokens)\*\*.

- \*\*Why it's used\*\*: JWT is widely used for \*\*authentication and authorization\*\*. It allows secure token-based user authentication. After a user logs in, a JWT is issued, which can be included in subsequent requests to authenticate the user without needing to store sessions on the server.

### 7. **\*\*mongoose\*\* (`^5.12.9`)**

- \*\*Purpose\*\*: An ODM (Object Data Modeling) library for MongoDB and Node.js.

- \*\*Why it's used\*\*: \*\*Mongoose\*\* allows you to model data in MongoDB with \*\*schemas\*\* and provides a rich API for querying and interacting with MongoDB collections. It also provides validation, hooks, and other utilities that simplify working with MongoDB in a structured manner.

### 8. **\*\*nodemon\*\* (`^2.0.22`)**

- \*\*Purpose\*\*: A tool that automatically restarts the Node.js server when file changes are detected.

- \*\*Why it's used\*\*: \*\*Nodemon\*\* improves developer experience by \*\*automatically restarting the server\*\* during development whenever file changes are made. This eliminates the need to manually stop and start the server every time changes are made, speeding up development workflow.

### 9. **\*\*openai\*\* (`^4.62.0`)**

- \*\*Purpose\*\*: The official Node.js client library for interacting with OpenAI APIs.

- \*\*Why it's used\*\*: It allows the application to integrate \*\*OpenAI's AI models\*\* (like GPT) for generating text, answering questions, or performing other AI-powered tasks. It's likely being used if the app includes features like an \*\*AI chatbot\*\* or content generation.

### 10. **\*\*socket.io\*\* (`^4.1.2`)**

- \*\*Purpose\*\*: Enables real-time, bidirectional communication between clients and servers using WebSockets.

- \*\*Why it's used\*\*: For implementing \*\*real-time messaging\*\* features like one-to-one chat, group chat, or live updates. \*\*Socket.io\*\* abstracts WebSockets to provide real-time functionality with automatic fallbacks to other transport methods if WebSockets are not supported by the client. It's essential for \*\*real-time updates\*\* like live chats, notifications, or live collaboration features.

\*\*Dependecies that I have I used in fronetend::

### 1. **\*\*@chakra-ui/icons\*\* (`^1.0.13`)**

- \*\*Purpose\*\*: Provides a collection of icon components from the Chakra UI framework.

- \*\*Why it's used\*\*: It offers an easy way to add customizable, scalable vector icons (SVGs) in React applications, which helps improve the UI's visual design and consistency.

### 2. **\*\*@chakra-ui/react\*\* (`^1.6.2`)**

- \*\*Purpose\*\*: A UI component library that provides accessible and customizable React components.

- \*\*Why it's used\*\*: Chakra UI simplifies building responsive and accessible UI layouts and components, making development faster and the UI more consistent.

### 3. **\*\*@emotion/react\*\* (`^11`)**

- \*\*Purpose\*\*: A library for writing CSS styles with JavaScript.

- \*\*Why it's used\*\*: Enables a \*\*CSS-in-JS\*\* approach, allowing you to style components dynamically based on state and props, leading to more maintainable styles in React applications.

### 4. **\*\*@emotion/styled\*\* (`^11`)**

- \*\*Purpose\*\*: A library for creating styled components.

- \*\*Why it's used\*\*: Offers a powerful way to style components in React, combining the flexibility of JavaScript with the syntax of CSS, making component-specific styles more modular and easier to maintain.

### 5. **\*\*@testing-library/jest-dom\*\* (`^5.11.4`)**

- \*\*Purpose\*\*: Provides custom DOM element matchers for Jest testing.

- \*\*Why it's used\*\*: Enhances \*\*unit testing\*\* by making assertions on DOM nodes more readable and expressive (e.g., checking if an element is visible or has a specific class).

### 6. **\*\*@testing-library/react\*\* (`^11.1.0`)**

- \*\*Purpose\*\*: A React-specific testing library for rendering and interacting with UI components.

- \*\*Why it's used\*\*: It helps simulate user interactions and test the behavior of React components in a more accessible and user-centric way.

### 7. **\*\*@testing-library/user-event\*\* (`^12.1.10`)**

- \*\*Purpose\*\*: Simulates user interactions with the DOM (like clicks, typing, etc.) during tests.

- \*\*Why it's used\*\*: Used alongside \*\*React Testing Library\*\* to emulate complex user behavior during testing, ensuring that components behave correctly in real-world scenarios.

### 8. **\*\*axios\*\* (`^0.21.1`)**

- \*\*Purpose\*\*: A promise-based HTTP client for making API requests.

- \*\*Why it's used\*\*: Facilitates communication with a backend by simplifying HTTP requests such as GET, POST, PUT, and DELETE, and handling asynchronous API calls effectively.

### 9. **\*\*chat-app\*\***

- \*\*Purpose\*\*: Local file system dependency for a custom chat application module.

- \*\*Why it's used\*\*: Likely points to a custom-built chat module that is being developed locally and used in the project.

### 10. **\*\*framer-motion\*\* (`^4`)**

- \*\*Purpose\*\*: A library for creating smooth animations in React.

- \*\*Why it's used\*\*: Enables complex UI animations with ease, improving user experience through smooth transitions, gesture-based interactions, and dynamic animations.

### 11. **\*\*react\*\* (`^17.0.2`)**

- \*\*Purpose\*\*: The core React library for building user interfaces.

- \*\*Why it's used\*\*: React is the fundamental library for creating components and managing state in this application.

### 12. **\*\*react-chips\*\* (`^0.8.0`)**

- \*\*Purpose\*\*: A React component for adding chips (tags) UI elements.

- \*\*Why it's used\*\*: It provides a ready-made component for allowing users to select or input tags, improving the UI for cases like entering multiple keywords or categories.

### 13. **\*\*react-dom\*\* (`^17.0.2`)**

- \*\*Purpose\*\*: The package that provides DOM-specific methods to React.

- \*\*Why it's used\*\*: It is required to render React components into the DOM and handle updates efficiently.

### 14. **\*\*react-lottie\*\* (`^1.2.3`)**

- \*\*Purpose\*\*: A library for adding Lottie animations to React apps.

- \*\*Why it's used\*\*: It enables the integration of lightweight, scalable animations, commonly used for loading screens, enhancing visual appeal.

### 15. **\*\*react-notification-badge\*\* (`^1.5.1`)**

- \*\*Purpose\*\*: A library for adding notification badges in React apps.

- \*\*Why it's used\*\*: Provides a simple, customizable badge component for displaying notifications or counts, enhancing the UX by drawing attention to important areas.

### 16. **\*\*react-router-dom\*\* (`^5.2.0`)**

- \*\*Purpose\*\*: A library for handling client-side routing in React.

- \*\*Why it's used\*\*: Allows for navigation between different pages or views in a single-page application (SPA), enabling dynamic and efficient routing in the app.

### 17. **\*\*react-scripts\*\* (`4.0.3`)**

- \*\*Purpose\*\*: Scripts and configuration for running and building Create React App projects.

- \*\*Why it's used\*\*: Bundles important tools like Webpack and Babel for developing React applications without needing to configure them manually.

### 18. **\*\*react-scrollable-feed\*\* (`^1.3.1`)**

- \*\*Purpose\*\*: Provides an auto-scrolling feed for React apps.

- \*\*Why it's used\*\*: Useful for creating chat interfaces or notification feeds that automatically scroll when new content is added.

### 19. **\*\*socket.io-client\*\* (`^4.1.2`)**

- \*\*Purpose\*\*: A library for real-time, bidirectional communication between client and server.

- \*\*Why it's used\*\*: Allows the app to enable real-time features like live chat or updates by connecting to a \*\*Socket.io\*\* server.

### 20. **\*\*web-vitals\*\* (`^1.0.1`)**

- \*\*Purpose\*\*: A tool for measuring essential web performance metrics.

- \*\*Why it's used\*\*: Helps track and improve the performance of the app by measuring metrics like \*\*First Contentful Paint (FCP)\*\*, \*\*Largest Contentful Paint (LCP)\*\*, and \*\*Cumulative Layout Shift (CLS)\*\*, which are crucial for user experience.