**Real Time Chat Application: Chat – App**

This chat application offers a lively and easy-to-use messaging experience. After securely signing up and logging in, users can quickly find and chat with other online people using a simple search box. The app supports real-time text messaging, so conversations flow smoothly and instantly. Users can also share images and videos, making chats more engaging and fun. The application keeps users informed with notifications for any unread messages, so they never miss important updates. Each message shows the time it was sent, helping users keep track of their conversations. Users can update their profile details, like their username and profile picture, whenever they want. If users need to take a break or switch accounts, the logout feature ensures a safe exit while saving their information for the next time they log in.

In summary, this chat app combines instant messaging, multimedia sharing, and easy profile management to deliver a friendly and complete communication experience.

**Chat App Prototype Setup**

Clone the Repository

1. git clone <https://github.com/pratyush1602/IBY_ChatApp>
2. If Node.js is not installed on your device, you can download and install it from the official website: [Node.js Download](https://nodejs.org/en/download/package-manager).
3. Open two shells/Terminal in the Cloned Folder
4. In one terminal do: cd backend
5. In other terminal do : cd frontend
6. Do ‘ npm i ‘ in both the terminal to install all the compulsory modules ‘’ only run command ( npm i) ‘’
7. After that in backend terminal run the command (node index.js)
8. And in front end terminal run the command (npm run start)

Dependencies And Libraries Used in BackEnd:

* **bcrypt**: Used for hashing and securing passwords.
* **cookie-parser**: Parses cookies attached to the client request object.
* **cors**: Provides a middleware to enable Cross-Origin Resource Sharing.
* **dotenv**: Loads environment variables from a .env file into process.env.
* **express**: A web application framework for Node.js to build the API.
* **express-validator**: Provides middleware to validate and sanitize request data.
* **jsonwebtoken**: Used to generate and verify JSON Web Tokens for authentication.
* **mongoose**: An ODM (Object Data Modeling) library for MongoDB and Node.js.
* **socket.io**: Enables real-time, bidirectional communication between clients and servers.

DevDependencies:

* **nodemon**: A development utility that automatically restarts the server on file changes.

Why these Frontend Dependencies?

**bcrypt**

* **Purpose**: Hashes and secures passwords.
* **Reason**: Ensures that user passwords are stored securely in the database by hashing them, making it difficult for attackers to retrieve or misuse them.

**cookie-parser**

* **Purpose**: Parses cookies attached to the client request object.
* **Reason**: Extracts and makes cookies available on the request object, which is useful for handling session data and user authentication.

**cors**

* **Purpose**: Provides middleware to enable Cross-Origin Resource Sharing.
* **Reason**: Allows the backend to accept requests from different origins, which is essential for enabling communication between the frontend and backend hosted on different domains or ports.

**dotenv**

* **Purpose**: Loads environment variables from a .env file into process.env.
* **Reason**: Manages configuration settings, such as database credentials and API keys, in a secure and organized manner. This practice keeps sensitive information out of the source code.

**express**

* **Purpose**: A web application framework for Node.js to build the API.
* **Reason**: Provides a robust and flexible foundation for creating and managing routes, handling requests and responses, and integrating middleware.

**express-validator**

* **Purpose**: Provides middleware to validate and sanitize request data.
* **Reason**: Ensures that incoming data is valid and secure before processing it, helping to prevent common issues like malformed data and security vulnerabilities.

**jsonwebtoken**

* **Purpose**: Used to generate and verify JSON Web Tokens for authentication.
* **Reason**: Implements authentication and authorization mechanisms by issuing and validating tokens that verify user identity and permissions.

**mongoose**

* **Purpose**: An ODM (Object Data Modeling) library for MongoDB and Node.js.
* **Reason**: Simplifies interactions with MongoDB by providing a schema-based solution to model application data and manage database operations.

**socket.io**

* **Purpose**: Enables real-time, bidirectional communication between clients and servers.
* **Reason**: Facilitates real-time messaging and updates within the chat application, allowing users to send and receive messages instantly.

Dependencies And Libraries Used in FrontEnd:

* **@reduxjs/toolkit**: Provides a set of tools to simplify Redux development.
* **@testing-library/jest-dom**, **@testing-library/react**, **@testing-library/user-event**: Libraries for testing React components.
* **axios**: A promise-based HTTP client for making API requests.
* **dom**: Provides utility functions for DOM manipulation (not commonly used).
* **moment**: Library for date and time manipulation.
* **react**, **react-dom**: Core libraries for building user interfaces in React.
* **react-hot-toast**: Provides a way to display toast notifications.
* **react-icons**: Provides popular icons as React components.
* **react-redux**: Official React bindings for Redux.
* **react-router**, **react-router-dom**: Libraries for routing and navigation in React applications.
* **socket.io-client**: Client-side library for connecting to a Socket.IO server.
* **web-vitals**: Provides metrics for web performance.

DevDependencies:

* **tailwindcss**: A utility-first CSS framework for styling the frontend.

Why these Frontend Dependencies?

**@reduxjs/toolkit**

* Purpose: Simplifies Redux development by providing a set of tools and best practices for managing state.
* Reason: Redux is used for state management in the application, and @reduxjs/toolkit streamlines tasks such as creating slices and managing asynchronous actions.

**@testing-library/jest-dom, @testing-library/react, @testing-library/user-event**

* + Purpose: Testing utilities for React applications.
  + Reason: Ensures the reliability and correctness of UI components by providing utilities for rendering, interacting with, and asserting on React components in tests.

**axios**

* + Purpose: A promise-based HTTP client for making API requests.
  + Reason: Facilitates communication with the backend server, allowing the application to send and receive data through HTTP requests.

**dom**

* + Purpose: Provides utility functions for DOM manipulation.
  + Reason: Although not commonly used, it might have been included for specific DOM manipulation needs. In many cases, it could be replaced or omitted depending on the requirements.

**moment**

* + Purpose: A library for date and time manipulation.
  + Reason: Handles date and time formatting, parsing, and manipulation, which is useful for displaying timestamps in chat messages or managing scheduling features.

**react, react-dom**

* + Purpose: Core libraries for building user interfaces in React.
  + Reason: These libraries are essential for creating React components and rendering them to the DOM. They form the foundation of the user interface.

**react-hot-toast**

* + Purpose: Provides a way to display toast notifications.
  + Reason: Allows for user-friendly notifications for events such as message sent/received confirmations, error alerts, and other real-time feedback.

**react-icons**

* + Purpose: Provides popular icons as React components.
  + Reason: Enhances the user interface with icons, which are commonly used for buttons, menus, and other interactive elements.

**react-redux**

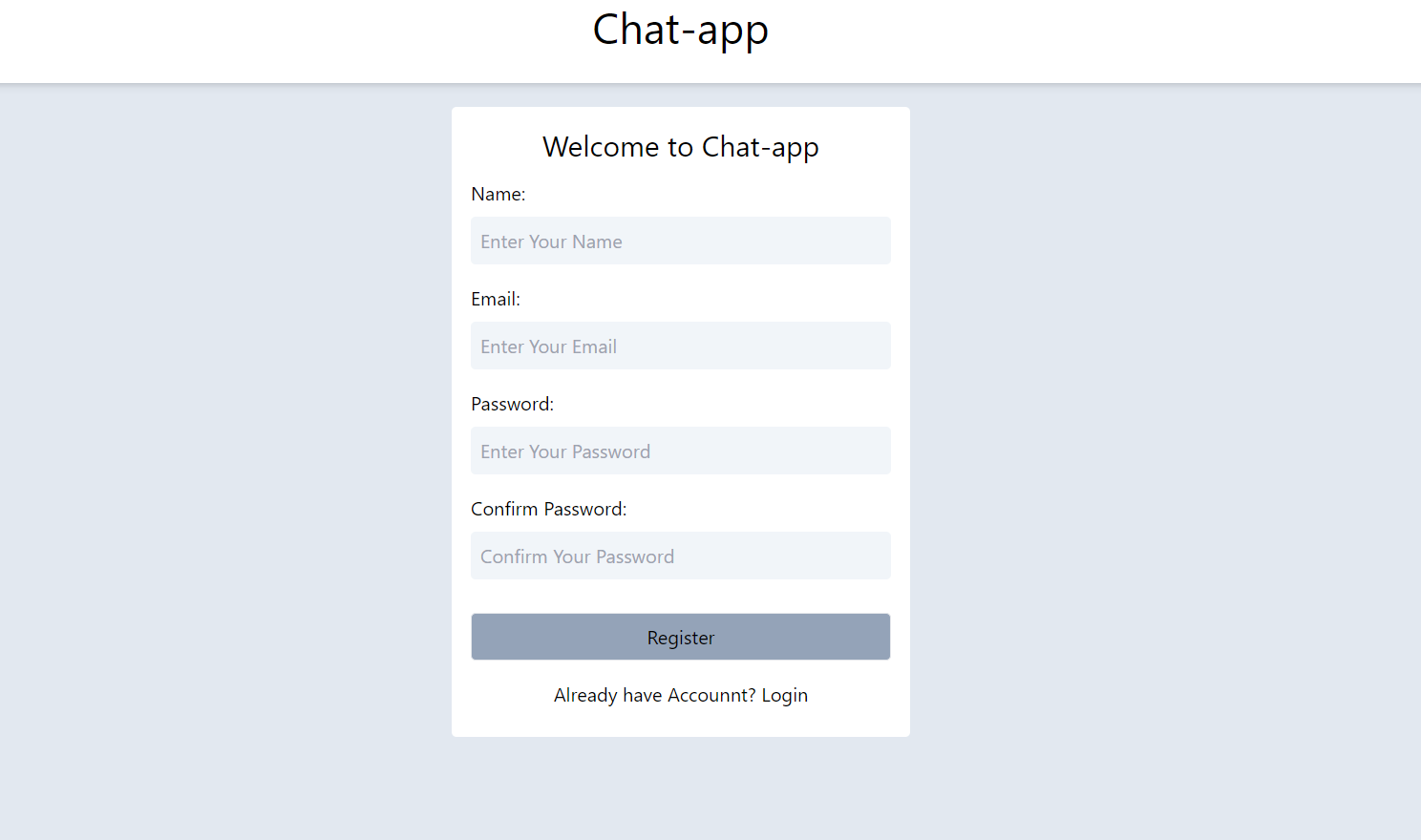
* + Purpose: Official React bindings for Redux.
  + Reason: Integrates Redux with React, enabling components to access and interact with the Redux store and state.

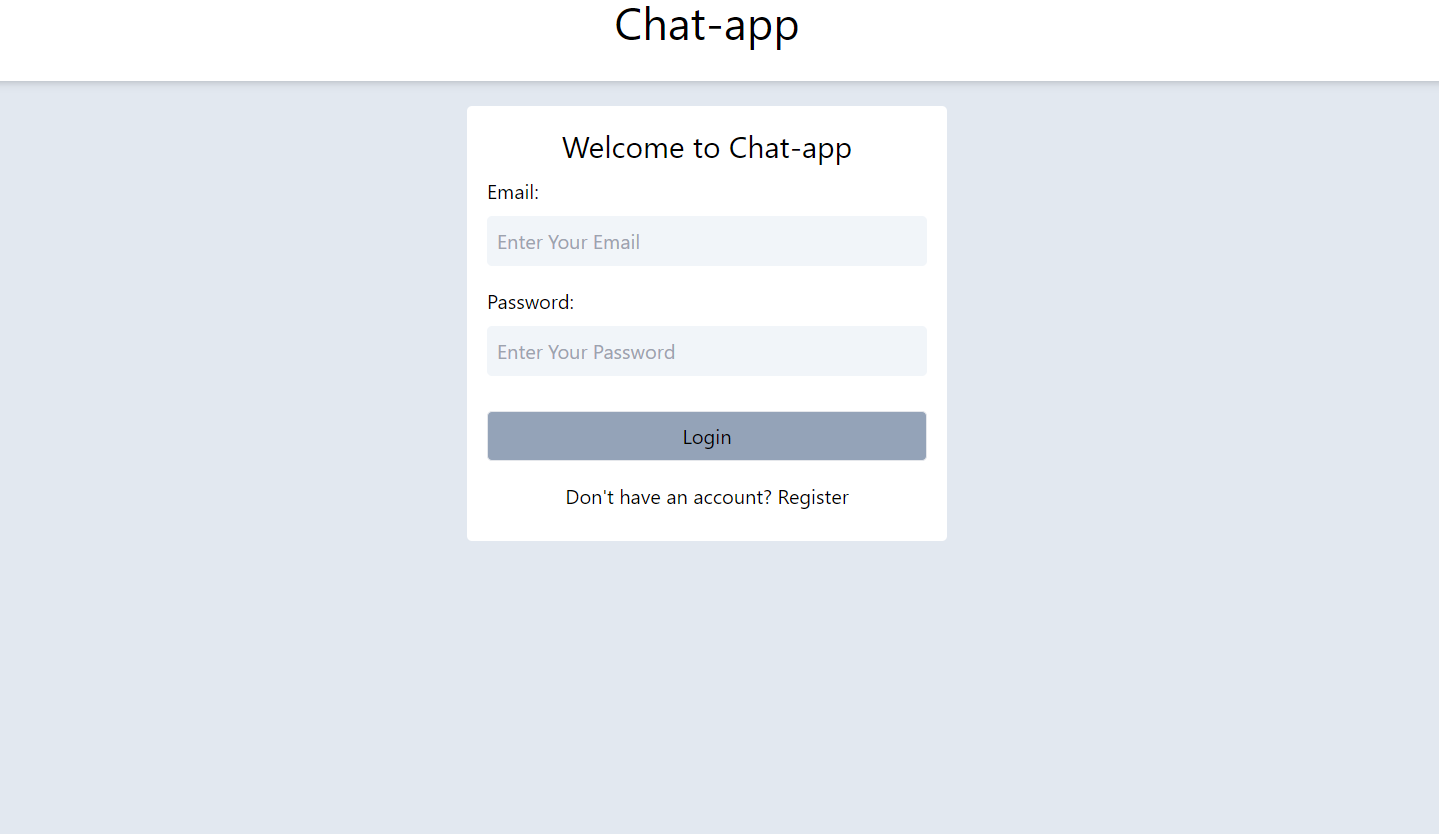
**react-router, react-router-dom**

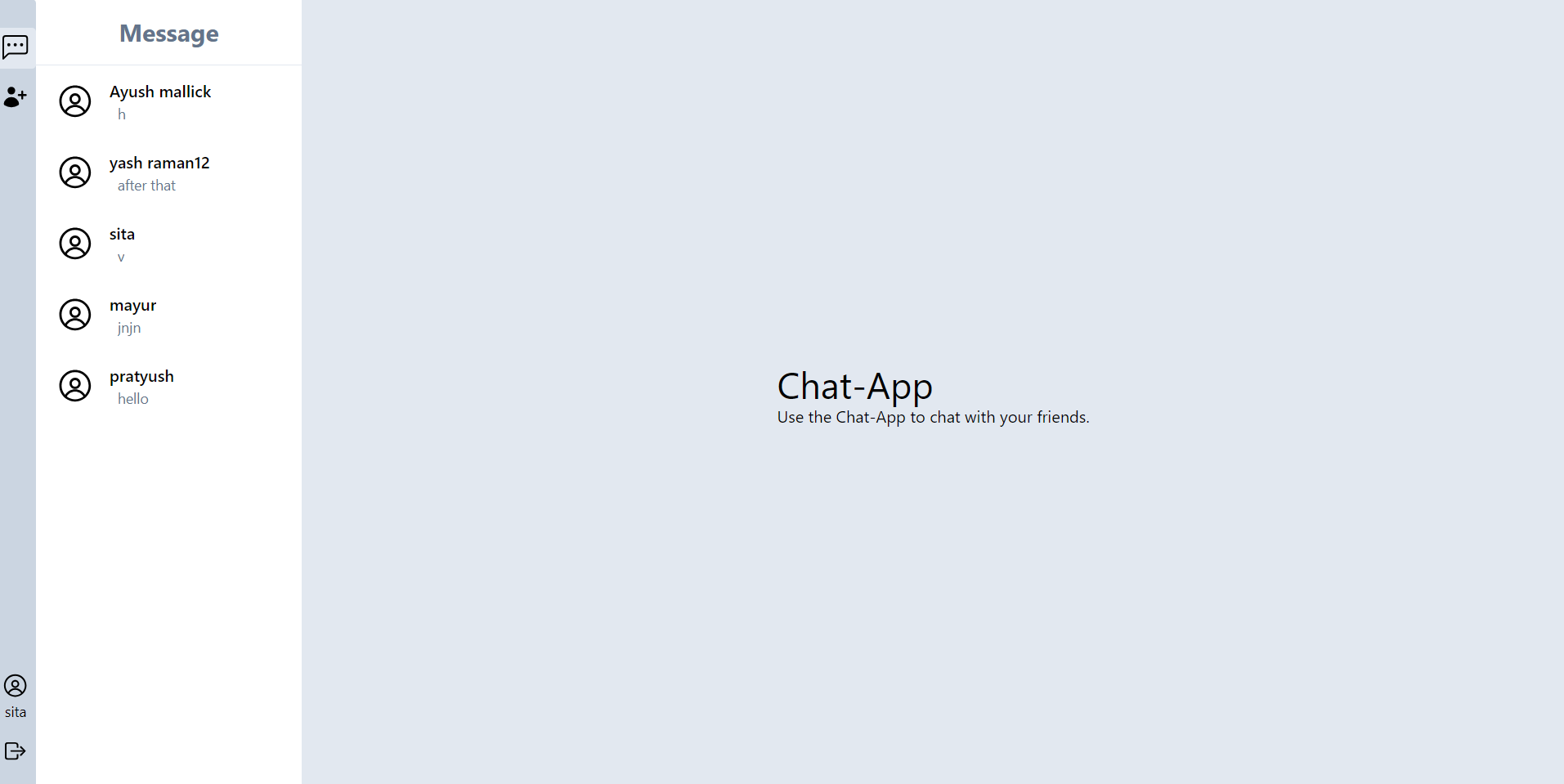
* + Purpose: Libraries for routing and navigation in React applications.
  + Reason: Manages navigation between different views or pages in the application, such as chat windows, user profiles, and login pages.

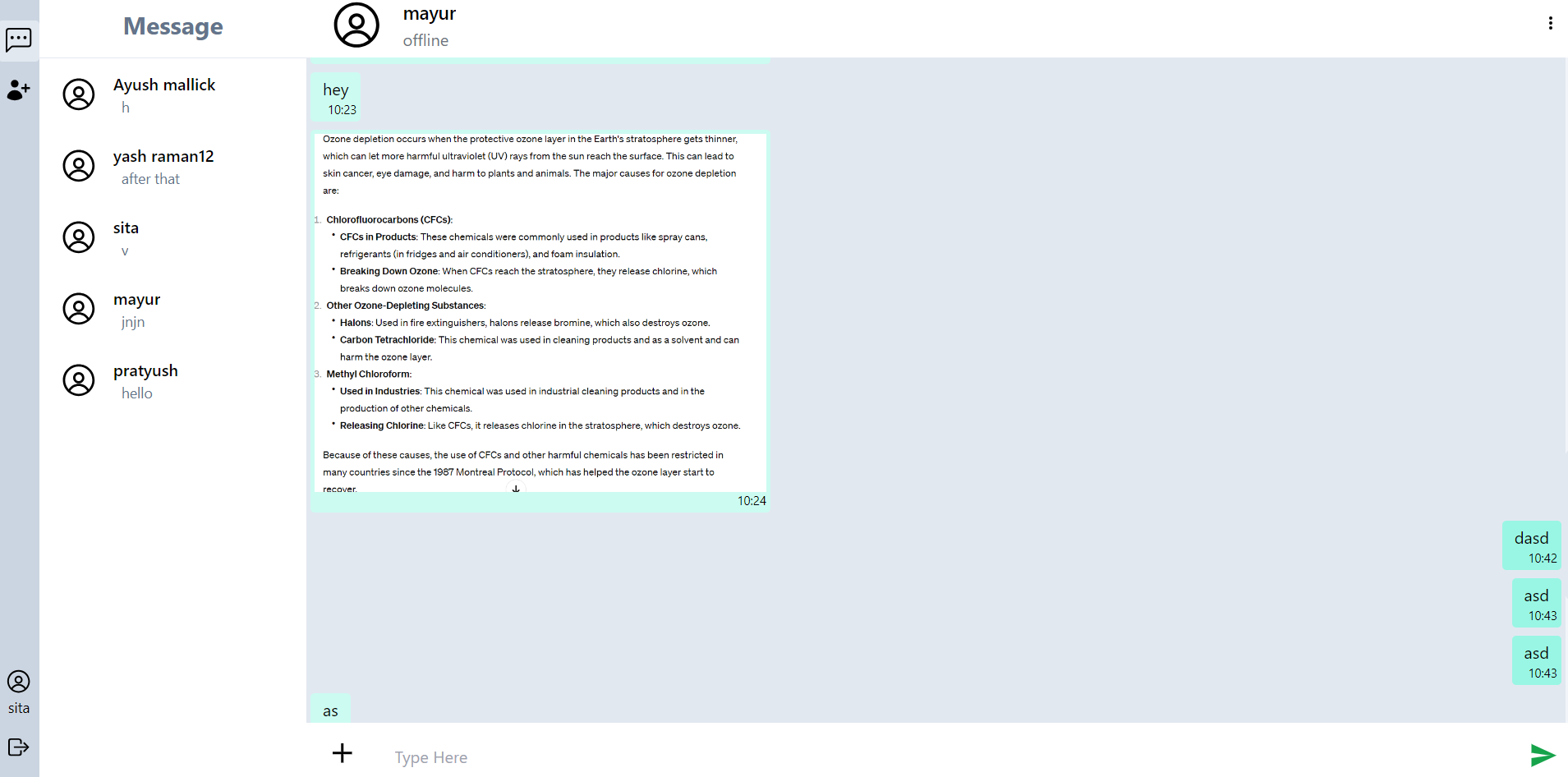
**socket.io-client**

* + Purpose: Client-side library for connecting to a Socket.IO server.
  + Reason: Enables real-time communication between the client and server, allowing for instant messaging and updates.

Images .





****