# Chat App

Initial Requirements:

1) Get basic chat working \*\* without \*\* user authentication or additional features

2) Add DB backend and simple user authentication (web only)

     Extra credit for JWT but don't delay your submission for extra credit

3) Clean up and beautify the user interface and UX

Technologies Used:

1. React – Front end framework.

2. NodeJS – Back end framework.

3. ExpressJS – RESTful API framework based on NodeJS.

4. Axios – Promise based HTTP client for NodeJS.

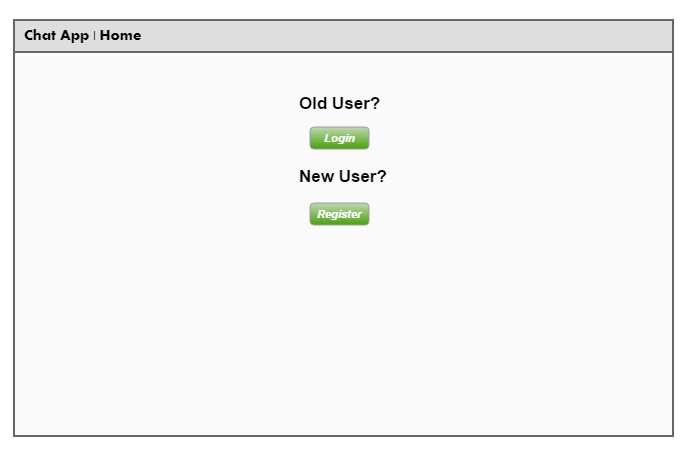
5. Socket.IO – Real time bidirectional event based communication.

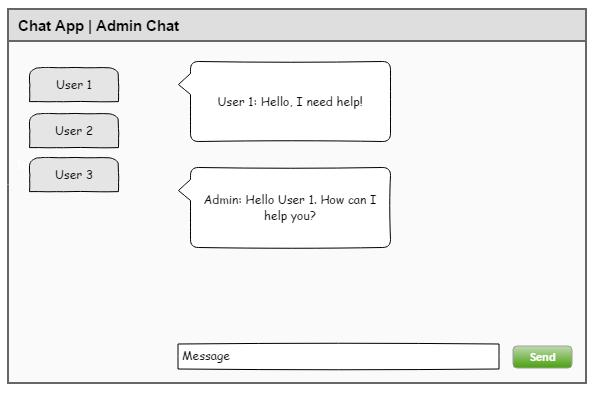
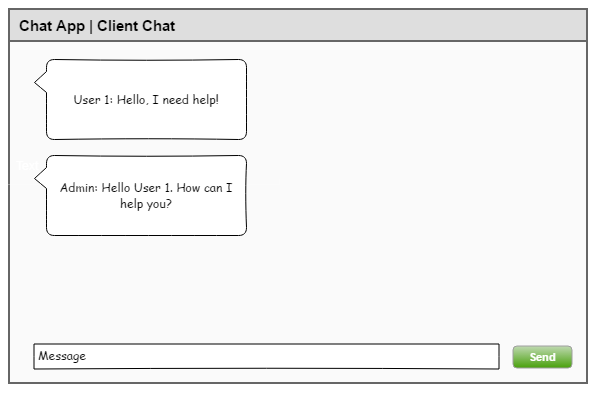
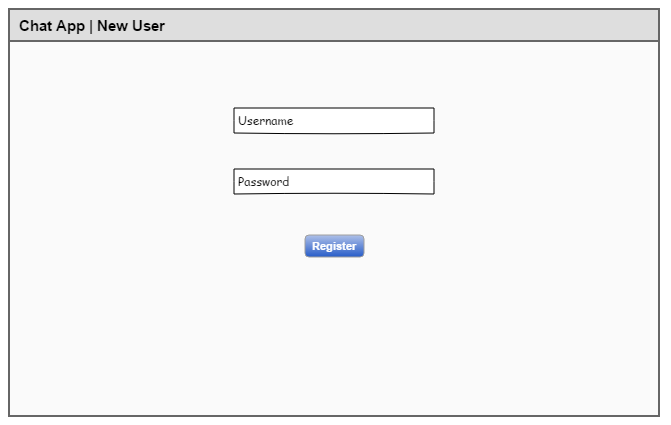
6. MongoDB – NOSQL Document based database.

7. Jsonwebtoken – Implementation of JWT.

8. Mocha – Testing for NodeJS applications.

**Part I: Initial design and wireframes.**





**Part II: Installation Instructions**

1. Make sure you have NodeJS installed on your computer.
2. You will need a MongoDB database to store user credentials.
3. Install `yarn` by using the command `**npm install yarn –g**`
4. Download the project files and in the project root run the command `**npm install**`
5. Open the `config.js` file and put in your MongoDB credentials.
6. To run the app locally, use the command ‘**yarn dev**`
7. To run the test script, use the command `**npm test**`

Usage Instructions:

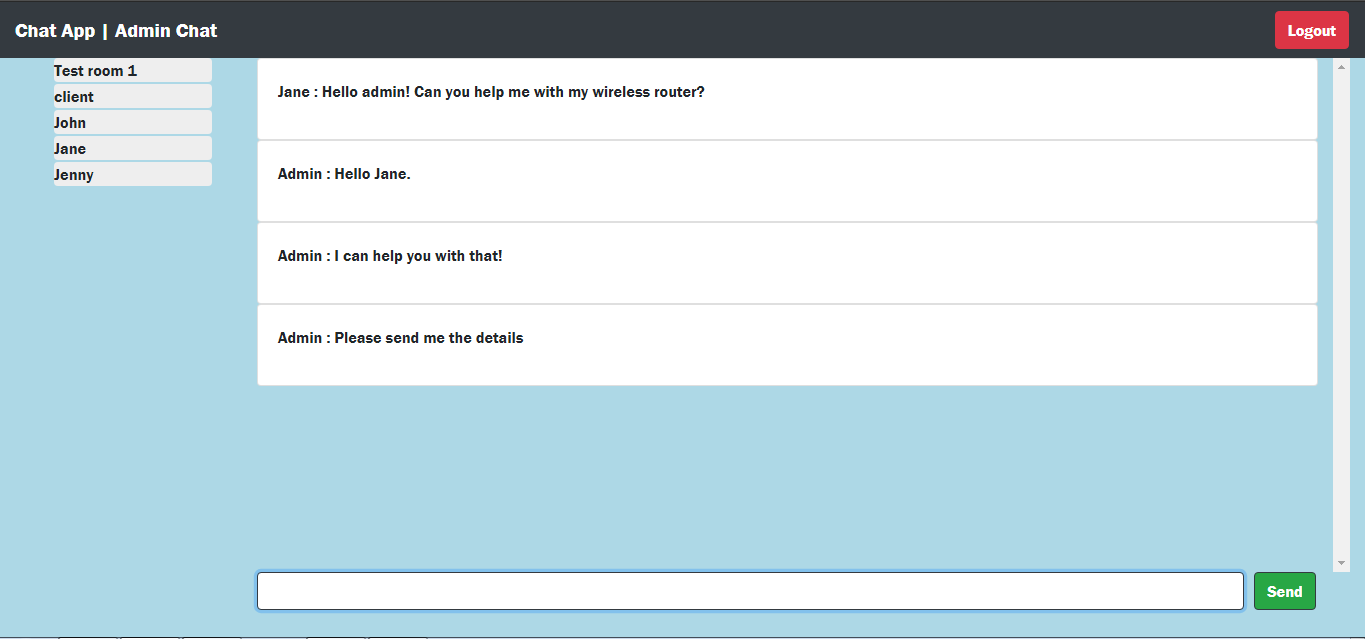
**Admin:**

1. To login as an Admin, use credentials- Username: admin, Password: password
2. The admin account is created automatically by the server.
3. The admin account cannot be created by anyone else, and a normal user will not be able to login as an admin.
4. When using the admin account, click on the user tabs on the left navigation bar to chat with that user.
5. When the user disconnects, the user tab will automatically fade out.

**User:**

1. To login, register as a user and use your credentials.
2. You will need to use a unique username to register.
3. Once logged in, use the text input to send messages to the admin.
4. Click the logout button to get out of the chat app.

**Screenshot:**

****