Exercise: Messaging App

Objective: Build a full-stack web application for sending and receiving messages.

Backend (Node.js)

Create a Node.js server that exposes RESTful APIs for the following operations:

• Register a new user (username and password). (Done)

• Authenticate users. (Done)

• Send a message to another user.

• Retrieve a list of messages for the authenticated user.gm

• Store everything in memory (no need to setup a db)

Frontend (React)

- Create a React application with the following components:

• LoginForm: Allow users to log in with their username and password.

• RegistrationForm: Allow users to register with a new username and password.

• ChatWindow: Display a chat where all users can write and receive messages.

• Use React Router to navigate between the login/registration page and the main chat interface.

- Connect the React frontend to the Node.js backend to perform user authentication and message retrieval.

- **Bonus:** Implement real-time updates for new messages using WebSocket or a similar technology. (refreshing the page to see messages is an acceptable solution)

Submission:

Share the source code via a GitHub repository.

Include a README file with instructions on how to run the application locally.

Evaluation Criteria:

• Code structure and organization.

• Proper usage of Node.js for backend development.

• Effective use of React for frontend development.

• User authentication implementation.

• Error handling and validation.

• UI/UX and overall application design.

• Bonus points for implementing real-time updates.