APTECH COLLEGE OF COMPUTER EDUCATION

PROJECT DOCUMENTATION OF GROUP CHAT APPLICATION

BY AMARACHI OKEKE ASHLEY

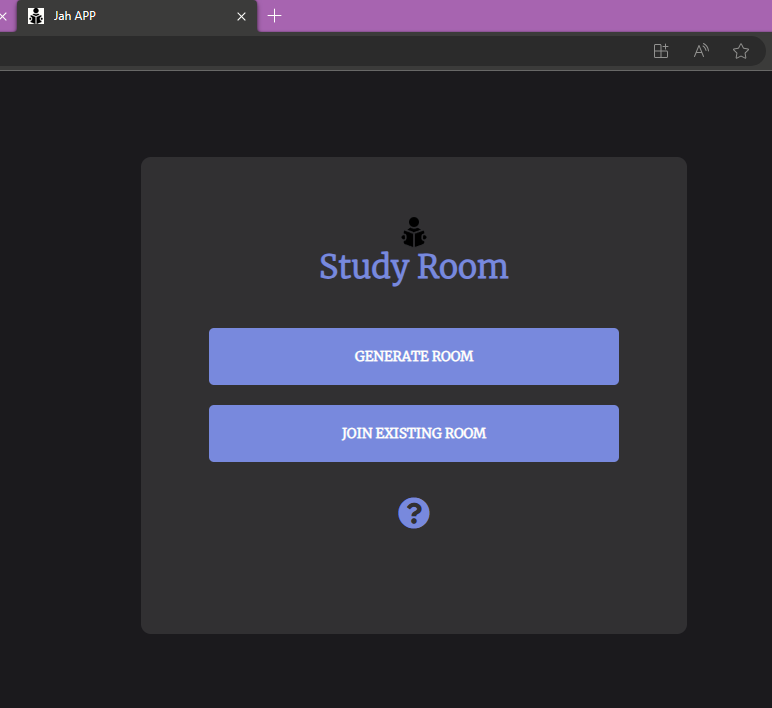
PROJECT STATEMENT: I was tasked with building a real-time chat application using a client-server architecture. The application should allow users to create chat rooms, join existing chat rooms, send messages, and receive messages instantly. The chat application should support multiple users simultaneously and maintain the order of messages within each chat room.

ABOUT PROPOSED GROUP CHAT APPLICATION (JAH APP)

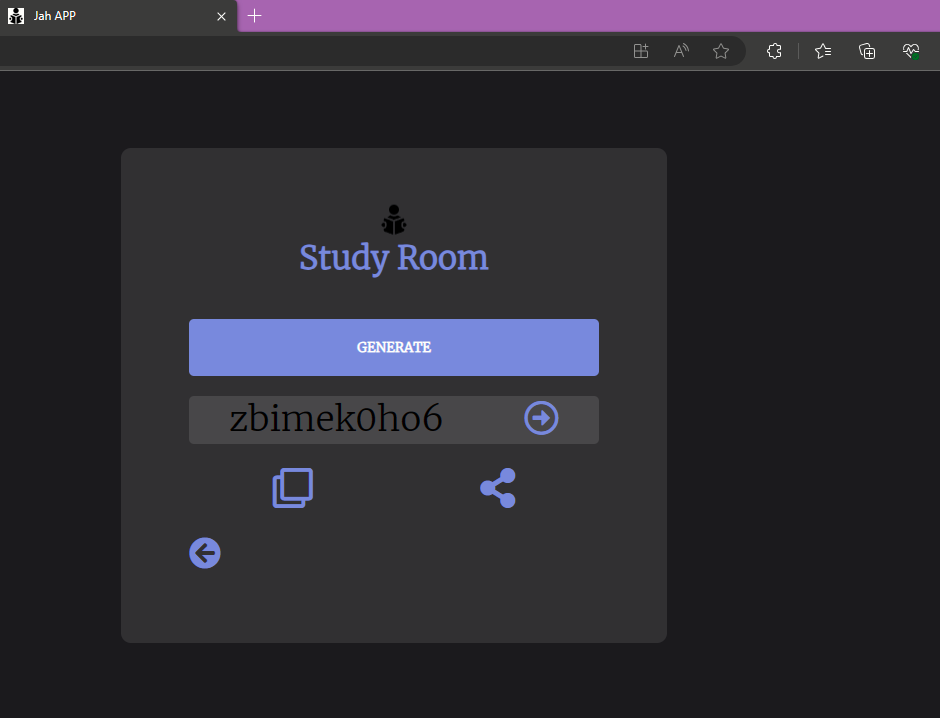
Jah App is a real-time group chatting application that supports features like room creation, room selection, message input, and message display. This application is scalable and can handle large number of concurrent users. Additionally, it allows users to see a list of active users in each chat room. When a user joins or leaves a chat room, the list of active users is updated in real-time for all users in that room and all messages sent by one user in a chat room are instantly received by all other users in the same room.

|  |  |
| --- | --- |
| INTERFACE | HOW INTERFACE WAS IMPLEMENTED AND DEPLOYED |
| FRONT END(CLIENT) | Using Reactjs, I built a responsive front-end interface for the chat application. The interface contains features like room creation, room selection, message input, and message display |
| BACK END(SERVER) | Using Nodejs to implement the server-side logic, The server handles incoming connections, manage chat rooms, and facilitate message exchange between users. |
| WEBSOCKETS | Using WebSocket to implement real-time communication between clients and the server. Messages sent by one user in a chat room are instantly received by all other users in the same room. |

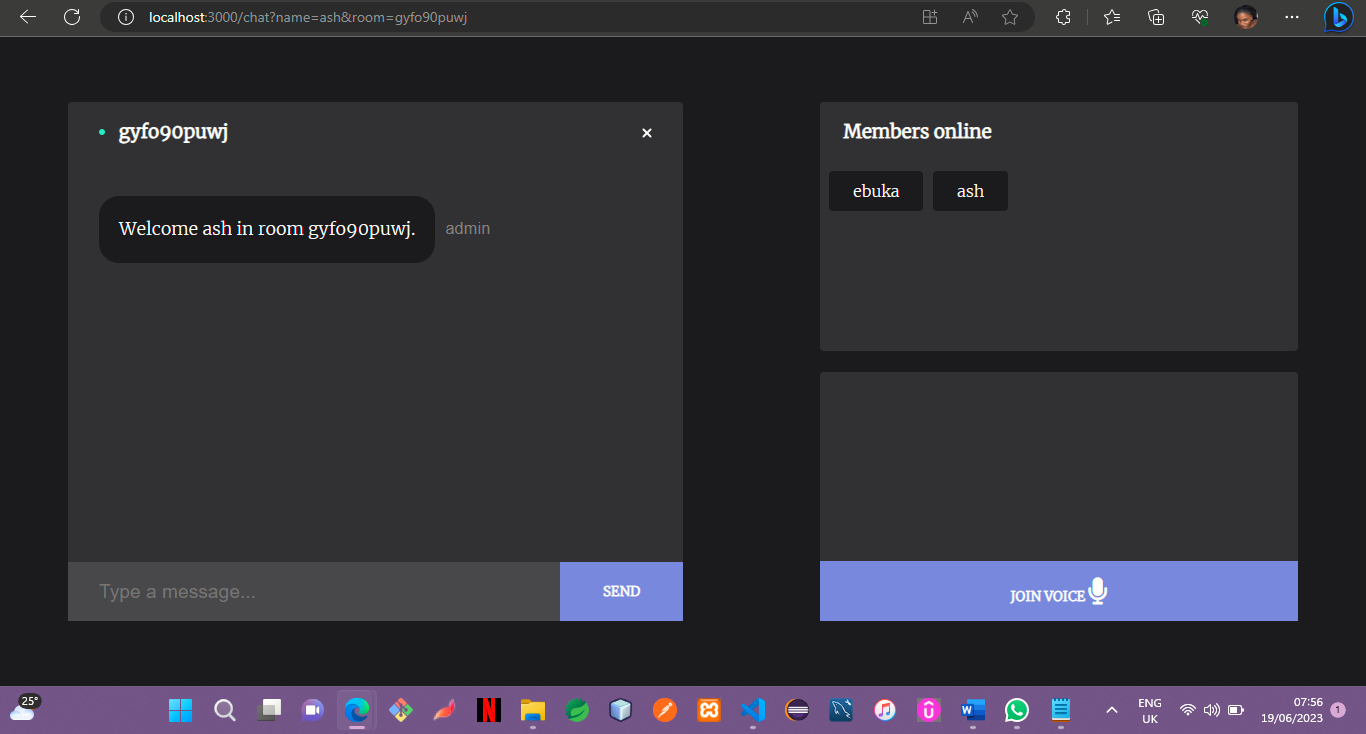
OVERVIEW OF HOW IT WORKS USING IMAGES



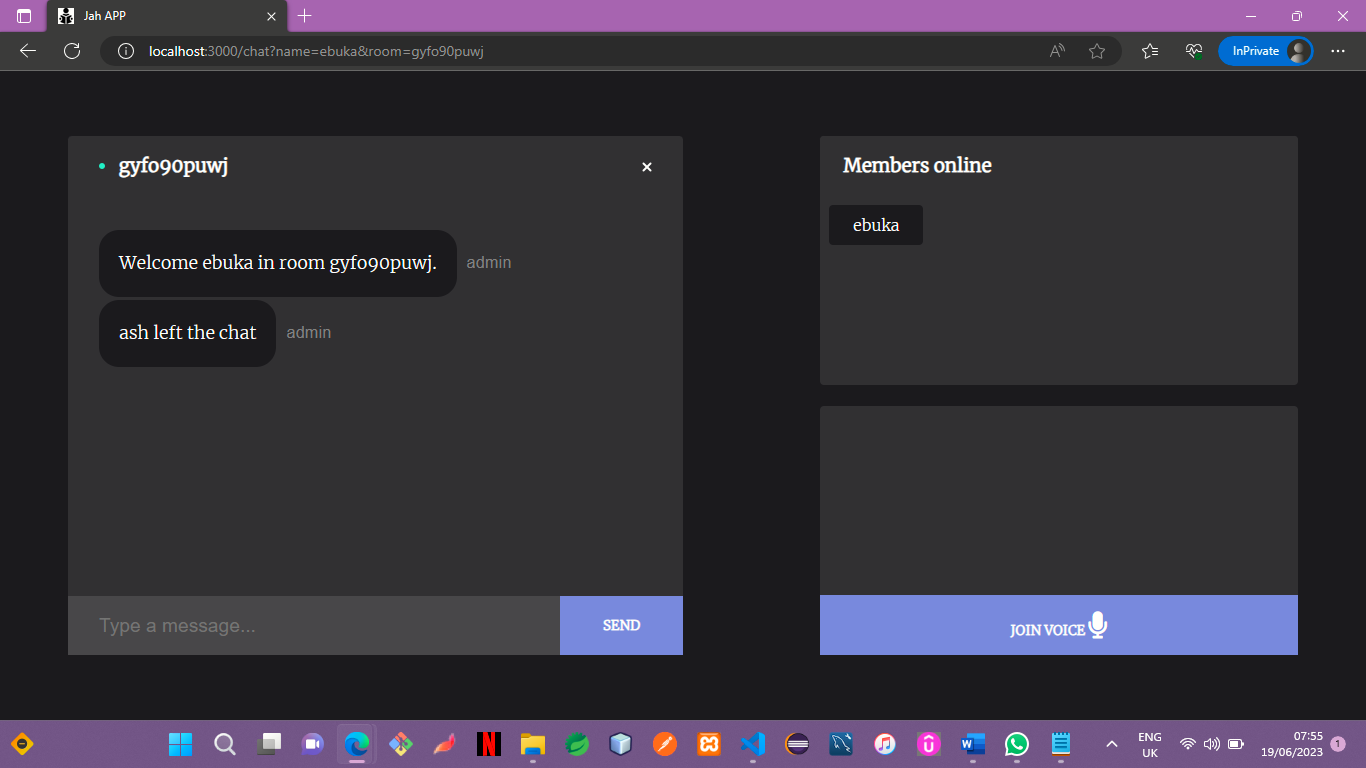
LANDING PAGE



HOW TO GENERATE A GROUP



HOW TO JOIN AN EXISTING GROUP



WHAT DISPLAYS WHEN SOMEONE JOINS OR LEAVES A CHAT

HOW TO RUN CODE ON TERMINAL: npm i

npm start