Doordash Take Home Exercise - Chat App

I have created ChatApp using React and Redux. Detailed instructions for running the app are in the readMe file of the application

Features Implemented

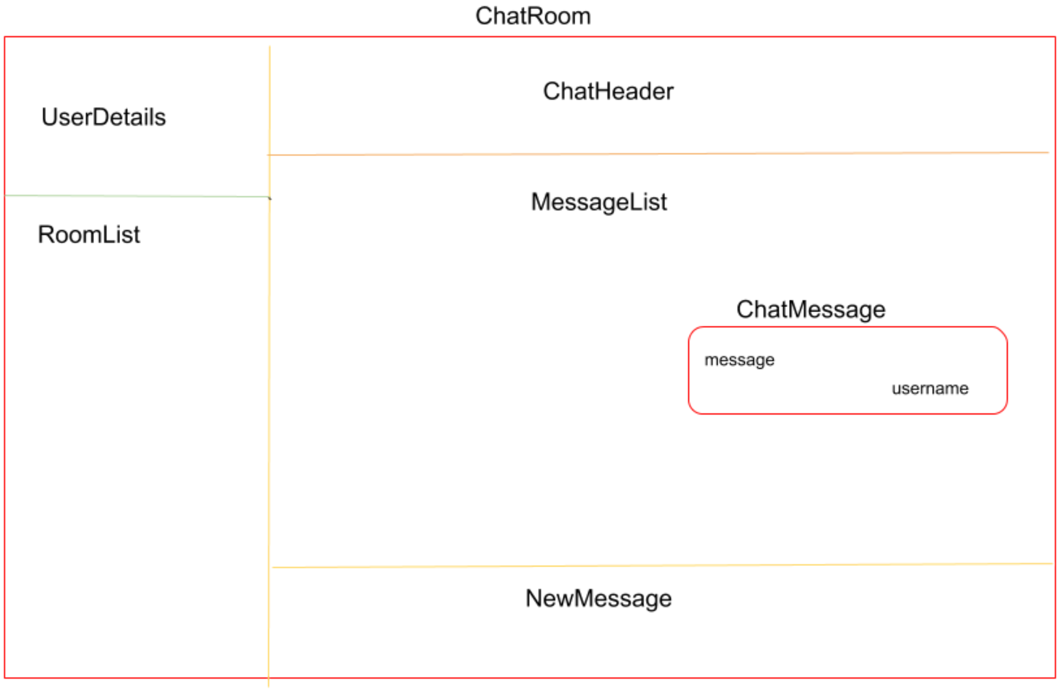
1. Multiple users are able to chat in real time using websockets
2. Messages persisted by calling backend server APIs
3. Scrolls the chat window to the bottom when new messages are added
4. Left panel fixed width while right panel stretches to accommodate
5. Notification for new unread messages in another room

Code Directory Structure

1. chat-app/ : This is the root directory
2. chat-app/src/components : This directory contains all the components
3. chat-app/src/redux/ : This directory contains the redux store, actions and reducers
4. chat-app/src//WebSocket.js: socket.io library, to enable instant communication of chat messages between the room
5. chat-app/WebSocketServer.js: This is websocket server, which listens to message sent by the client and appropriately broadcasts message to other connected clients
6. chat-app/server.js: This contains backend APIs (as provided in the assignment, has not been modified)
7. Config.js: This file contains the endpoints for the connecting to the backend APIs and web socket server
    1. Backend API server is running on port 8080
    2. Websocket Server is running on port 8000

Component break down

Each component has its own directory. The component directory contains an index.jsx file that holds the component definition and a corresponding style.css file



1. App
   Entry point in the app

2. ChatRoom
   This is the main component of the app holds all the logic and state management

3. UserDetails
   This component displays user name and time since the user is online. It is on the top-left section of the screen

4. RoomList
   This component displays a list of all chat rooms. It is on the left section of the screen, below UserDetails

5. ChatHeader

This component details of the currently selected room including room name and users in the room.

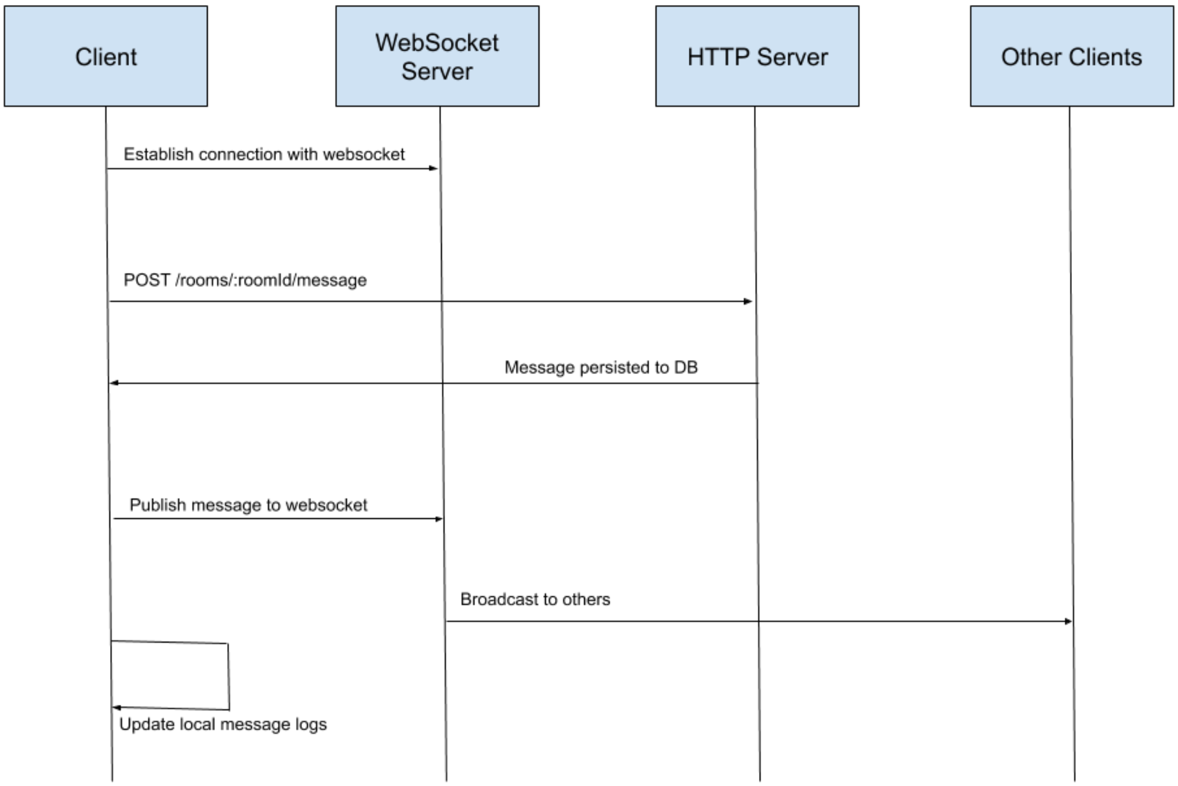
6. MessageList
   This component display list of chat message in selected chat room

7. NewMessage
   This component holds the text box and send button to send a new message

8. ChatMessage
   This component holds the chat message and user who sent the message .
   Incase if the message is sent by the current user, it will not have the username

Architecture

How message communication works with HTTPServer and WebSocketServer


Client
WebSocket Server
HTTP Server
Other Clients
Establish connection with websocket
POST /rooms/:roomId/message
Message persisted to DB
Publish message to websocket
Broadcast to others
Update local message logs

Please feel free to reach out if you have any questions. Thank you !