Project Report: Implementing an Online Meeting Software

Introduction:

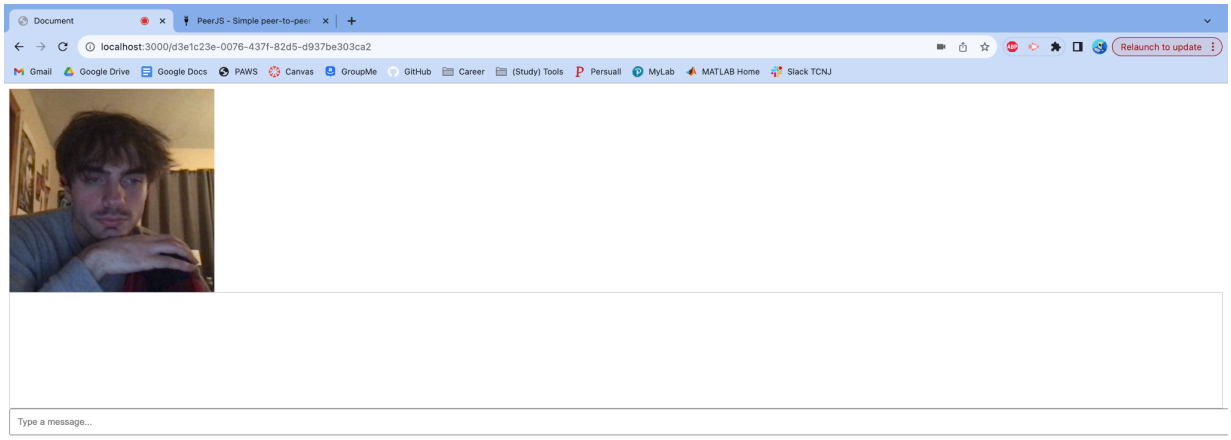
This project involved enhancing an existing online meeting software based on WebRTC to include a text chatting feature alongside its core video chatting functionality. The main goal was to implement the ability for people in the chat room to send and receive text messages in real-time without destroying the already existing video chat feature. To do this, the server-side, client-side, and front-end code required modifications.

Code Explanation:

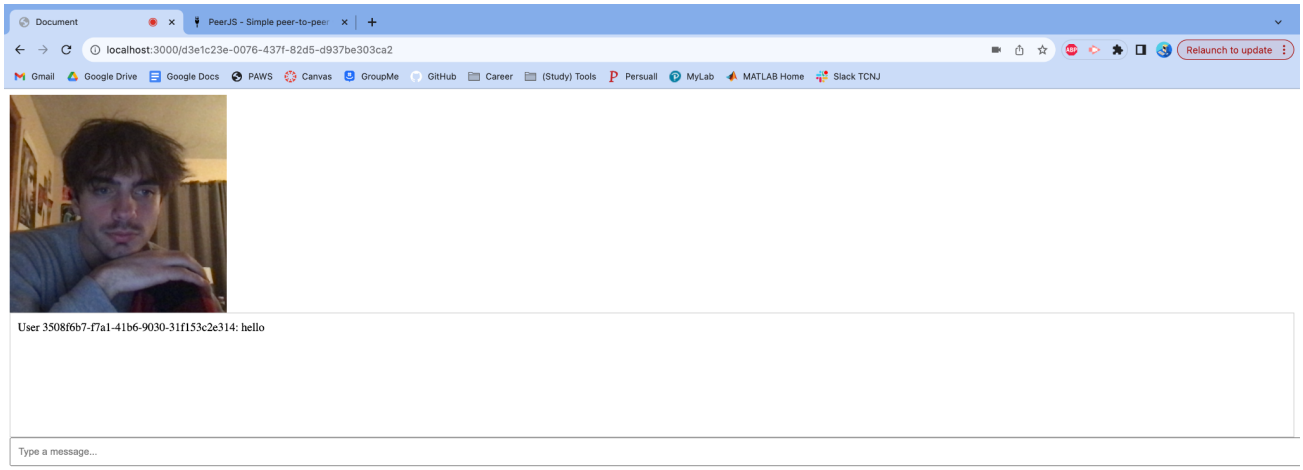
1. Server-Side Modifications (server.js):
    a. Added a new socket event listener socket.on('send-message', (message, roomId) => {...}) to the server script. This listener handles incoming text messages from clients and broadcasts them to other clients in the same chat room.
    b. The message parameter represents the text message sent by a user, and roomId helps in identifying which chat room the message should be broadcasted to.
    c. The implementation uses Socket.io's io.to(roomId).emit('message', message, userId) for message broadcasting.
2. Client-Side Modifications (script.js):
    a. Code to capture user input from a text field (message-input) and send this input to the server using socket.emit('send-message', messageInput.value, ROOM_ID).
    b. Added an event listener for the 'Enter' key to trigger the message send action.
    c. Implemented a socket event listener socket.on('message', (message, userId) => {...}) to receive broadcasted messages from the server.
    d. appendMessage(message, userId) function was created to append incoming messages to the message display area in the UI.
3. Front-End UI Updates (room.ejs):
    a. A div element with the ID message-container was added. This serves as the display area for the chat messages.
    b. The container was styled with a fixed height of 150px and an overflow-y property set to scroll, ensuring that it can display several messages in a scrollable format.
    c. A border and padding were added to make it look visually appealing

Project Demo (snapshots of project functions):

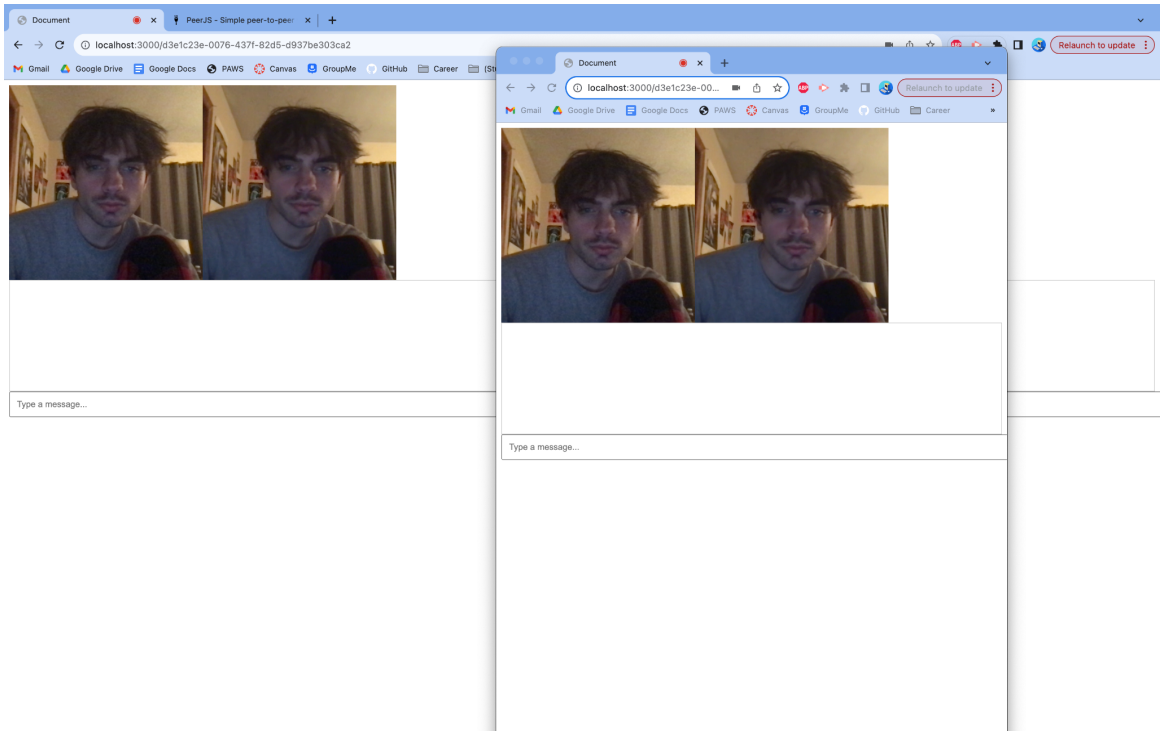
Default view when connecting to the server



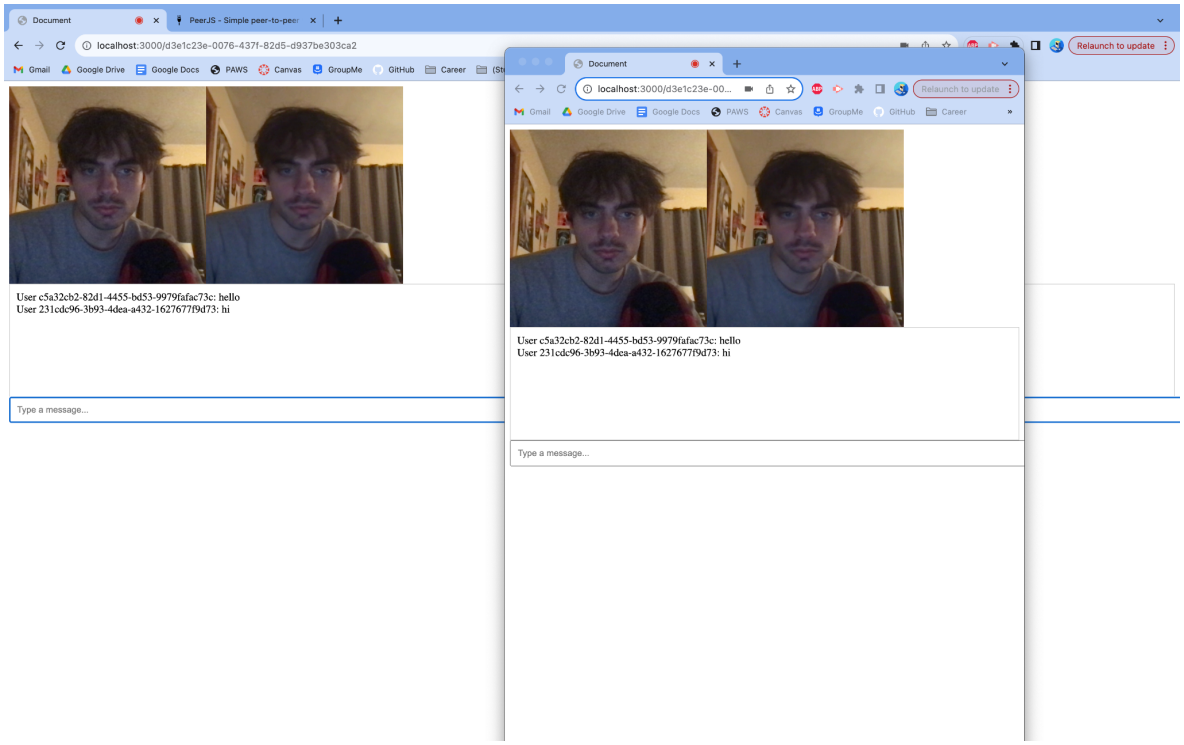
View when typing a message



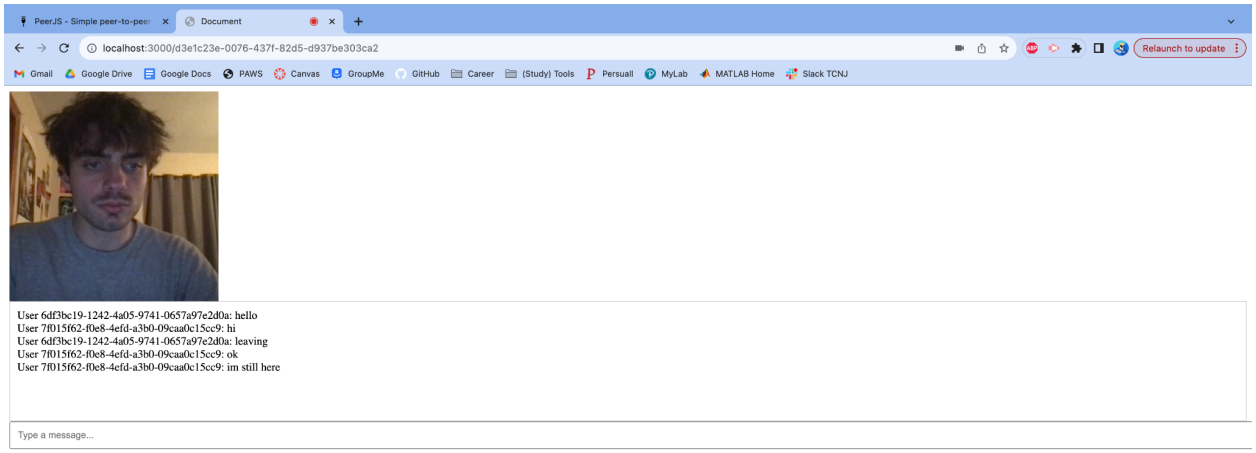
View when another user connects to the same chat room



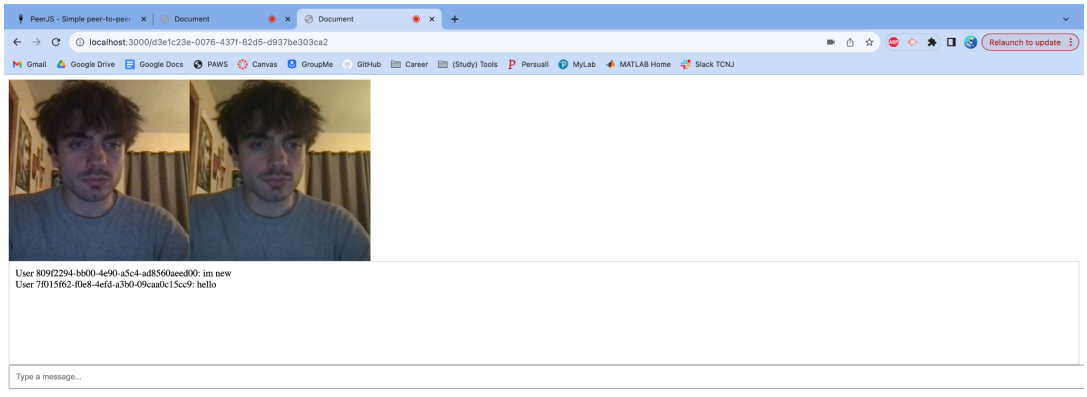
View when two users chat to each other in the same chat room



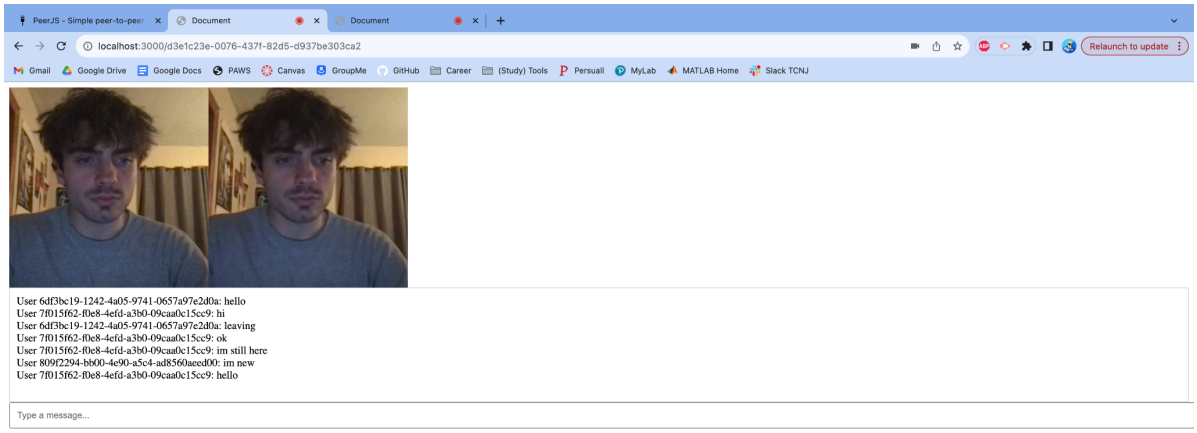
View when a connected user leaves the chat room



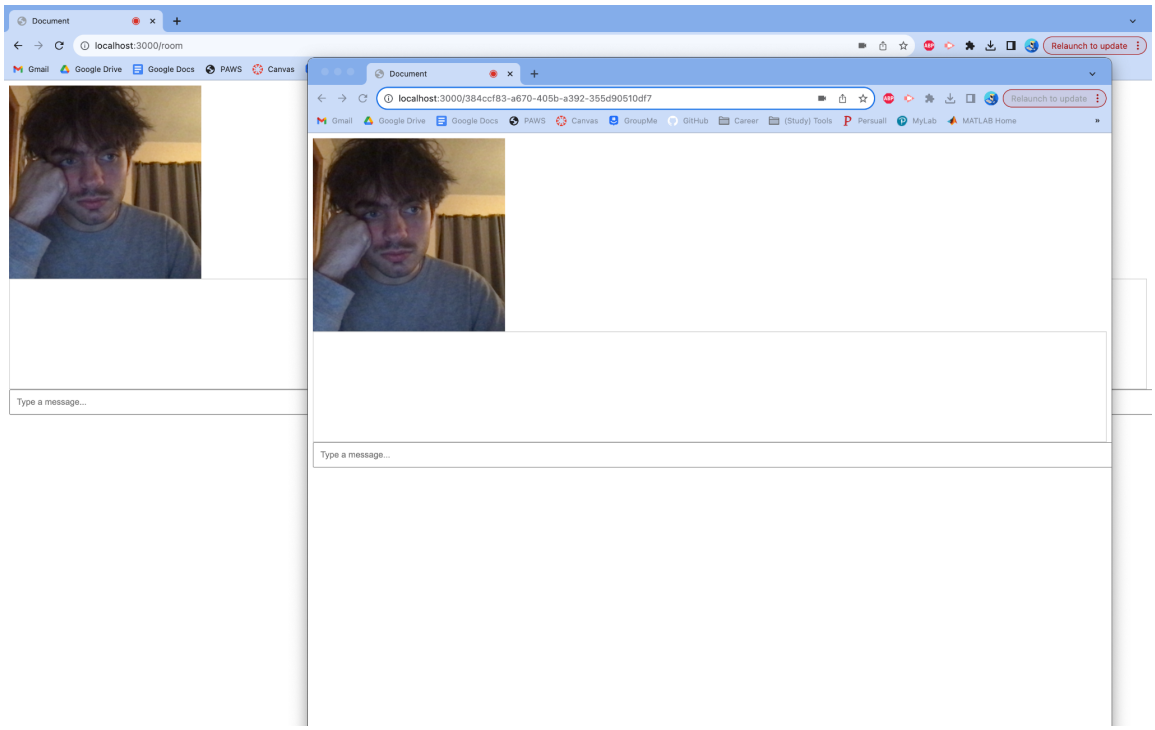
View when a new user connects to the same chat room (new user perspective)



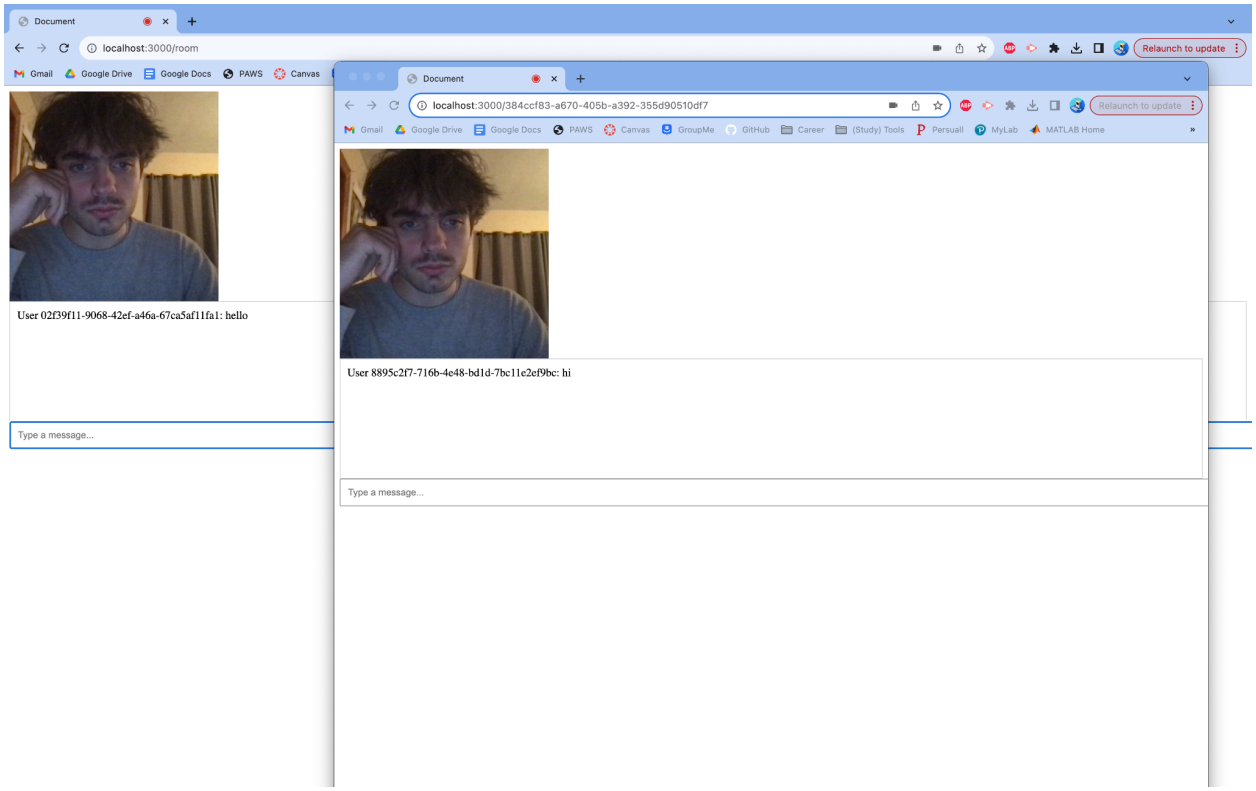
View when a new user connects to the same chat room (already existing user perspective)



View when two users joined separate chat rooms



View when two users use the text chat in separate chat rooms



Project Limitations:

- When two users connect to the same chat room, it will not ALWAYS dynamically load
    - If a second user joins a chat room and does not automatically appear, you must refresh the pages until it does work
    - I was unable to figure out why this is happening, I rewatched the entire video and made sure I did everything right and it still happened