CA400 - Testing Documentation

Echo Labs

Created By Students: Munashe Lawrence Chifungo & Fiona Tuite

IDs: 16469764, 16389251

Supervisor: Monica Ward

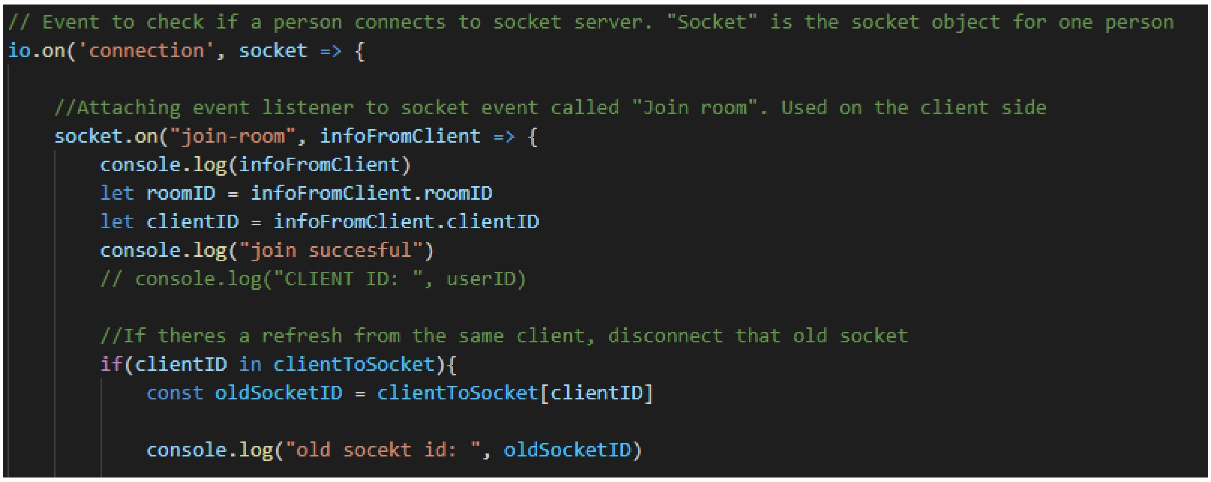
Testing Strategy

For our testing strategy we conducted ad hoc, integration testing and user testing. Our main focus was to get user testing done as our app is user focused.

Adhoc Testing

Adhoc testing was conducted at each level as the project progressed. This was mainly achieved by use of console logs to view the data and status of components as they were built and integrated. These logs can be found throughout our application code and can still be read in the browser when a user inspects the element of the code.

Examples of some of our ad hoc tests are as follows.



```
// Event to check if a person connects to socket server. "Socket" is the socket object for one person
io.on('connection', socket => {

    //Attaching event listener to socket event called "Join room". Used on the client side
    socket.on("join-room", infoFromClient => {
        console.log(infoFromClient)
        let roomID = infoFromClient.roomID
        let clientID = infoFromClient.clientID
        console.log("join succesful")
        // console.log("CLIENT ID: ", userID)

        //If theres a refresh from the same client, disconnect that old socket
        if(clientID in clientToSocket){
            const oldSocketID = clientToSocket[clientID]

            console.log("old sockekt id: ", oldSocketID)
```

Here at the beginning of the server file are logs that check that the correct data is being sent to the server by the client and that server is doing it’s checks properly. On connection a client sends it’s room ID and it’s user ID to the server in an object called infoFromClient. If the send was successful it logs that the user joined successfully. Once the user joins, the server checks if the client’s ID is already taking part in a lab session. This check collects the user's old socket ID and logs it so that we could examine and confirm that it was correct. That socket ID is then destroyed and replaced effectively removing the user from any other lab they may have been part of.

Integration Testing

Our project is written primarily in javascript and React. We chose React as it allows us to build dynamic modular web applications. Using react allowed us to section out the primary functions of our application into components. We built each component in isolation from the rest and once the ad hoc testing was complete we would integrate the component with the rest of the code base for testing or otherwise if it’s functionality was dependent on another component we would conduct component to component testing. A prime example of this is the development of the Room.js component.

Room.js was developed in isolation from the rest of the components. Once it was in good working order we conducted component to component testing with the create room component and the room component. This was to test the transfer from the lobby to the code editing session.

Each component in the project went through a stage of development then integration with the rest of the system.

Additionally, after each integration a system test was conducted to ensure everything works well together

User Testing

For user testing we included a sample space of users (13 users in total) who were both current or ex students of computer science as well as users who didn’t know anything about computer science or programming but would have been interested in learning to code. The idea was to give some of the sample users the role of a tutor while the rest had the role of a student. Once the User testing was completed, the sample users filled out the survey questions below.

Echo Labs Questionnaire

*Required

1. I agree to participate in this research study. I understand the purpose and nature of this study and I am participating voluntarily. I understand that I can withdraw from the study at any time, without any penalty or consequences. * *

   Mark only one oval.

   ○ Yes
   ○ No

2. I grant permission for the data generated from this survey to be used by the researchers to improve the web application. * *

   Mark only one oval.

   ○ Yes
   ○ No

3. Did you participate as a Tutor or Student

   Mark only one oval.

   ○ Student
   ○ Tutor

4. Rate the User interface on a scale of 1-5

   Mark only one oval.

	1	2	3	4	5	
Very Bad	○	○	○	○	○	Excellent

5. Was joining a lab Room difficult in your opinion?

Mark only one oval.

○ Yes
○ No
○ Other: ________________

6. If answered yes, explain why?

____

____

____

____

____

7. If you were a Tutor, was creating a lab room difficult

Mark only one oval.

○ Yes
○ No

8. If creating a lab was difficult please explain why

____

____

____

____

____

9. Was it difficult to request help?

Mark only one oval.

○ Yes

○ No

10. If yes, please explain why?

11. If you were a Tutor, was it difficult to respond to help requests

Mark only one oval.

○ Yes

○ No

12. If you answered yes please explain why

13. How was the audio quality on the scale of 1-5

Mark only one oval.

	1	2	3	4	5	
Very Bad	○	○	○	○	○	Excellent

14. How was the overall quality of the call on a scale of 1-5

Mark only one oval.

	1	2	3	4	5	
Very Bad	○	○	○	○	○	Excellent

15. Would this web application be beneficial to your learning experience?

Mark only one oval.

○ Yes
○ No
○ Maybe

16. Was navigating through the application difficult?

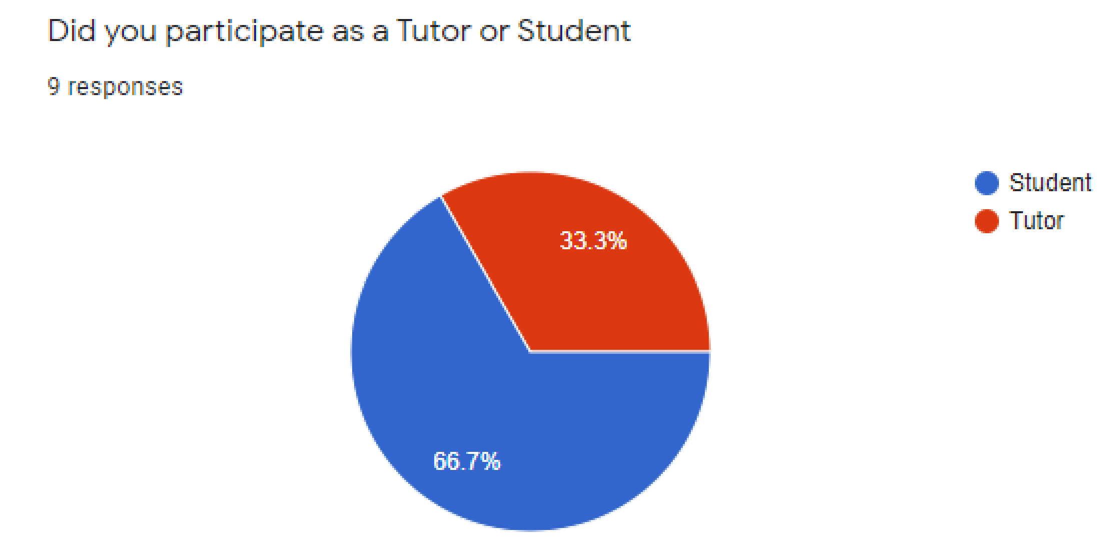
Mark only one oval.

○ Yes
○ No
○ Other: ______________________

17. If answered yes, explain why?

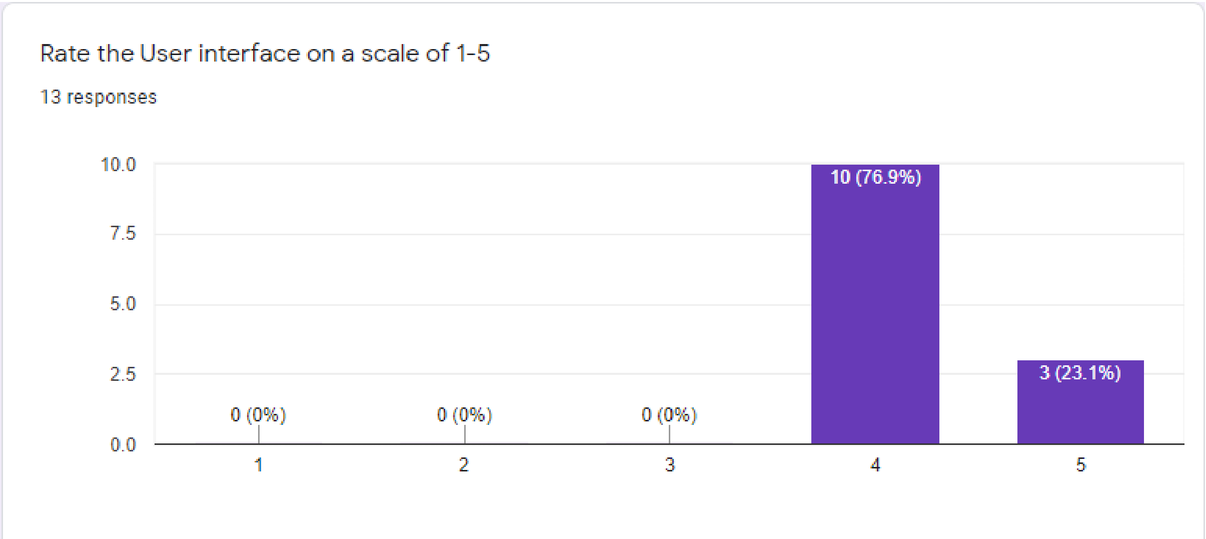
18. What are some changes you would make to the overall web application?

Example Results from the User Testing Survey



Did you participate as a Tutor or Student

9 responses

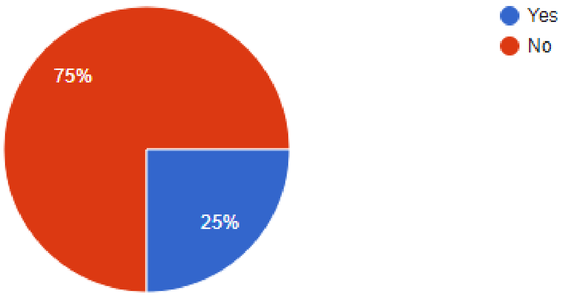


Rate the User interface on a scale of 1-5

13 responses

If you were a Tutor, was it difficult to respond to help requests

4 responses



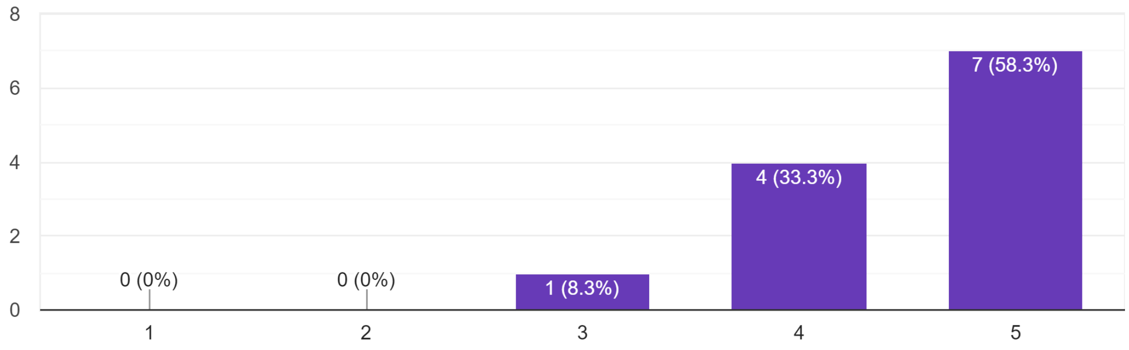
If you answered yes please explain why

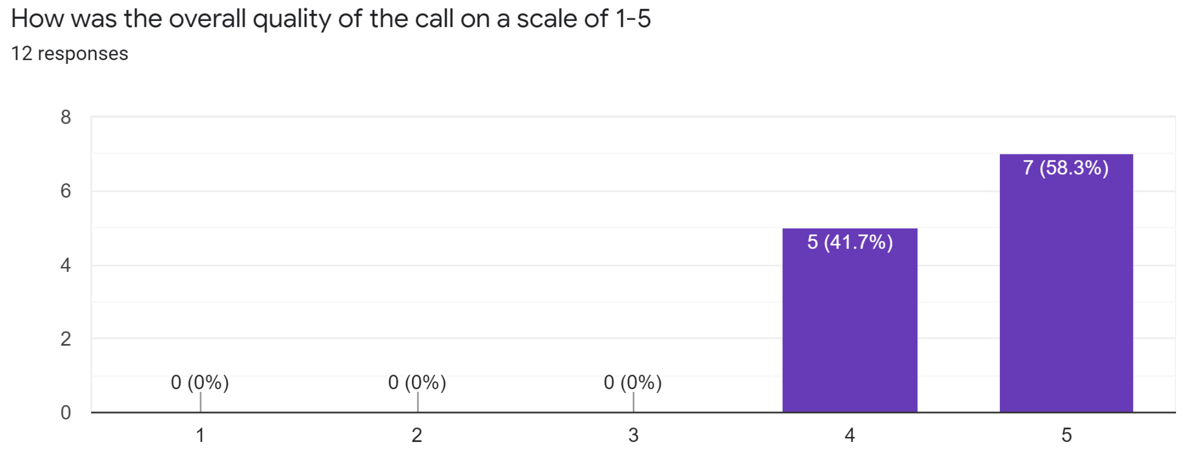
1 response

The initial request generated by my partner didn't work as intended, and we had to create a second request, which did work perfectly. Beyond that there were no issues.

How was the audio quality on the scale of 1-5

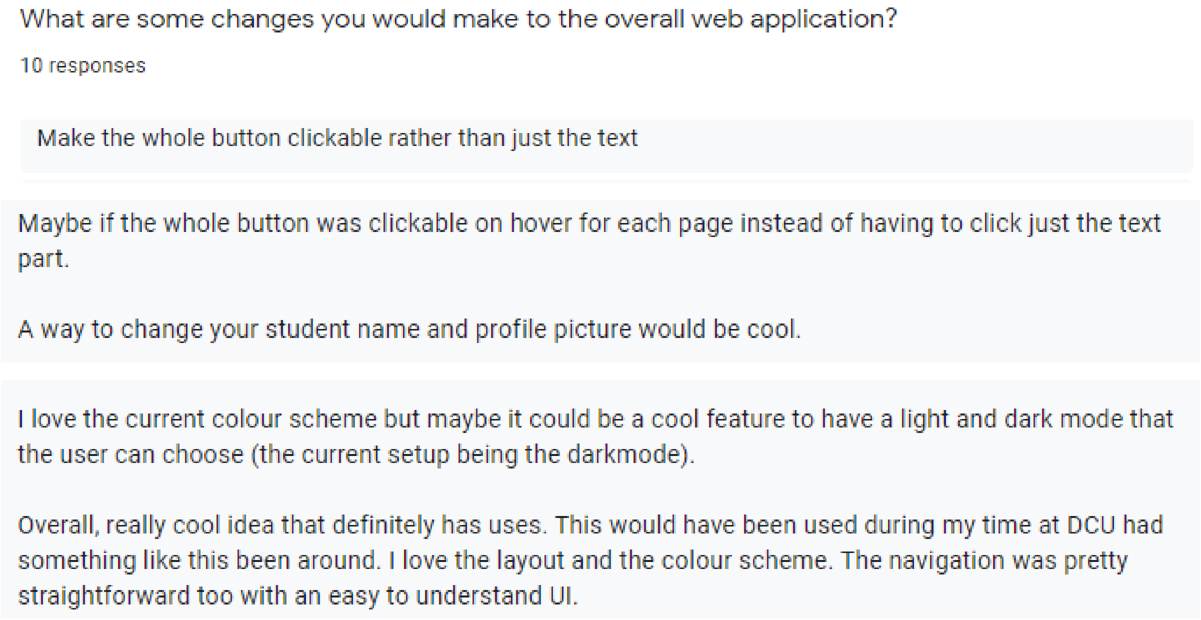
12 responses





How was the overall quality of the call on a scale of 1-5

12 responses



What are some changes you would make to the overall web application?

10 responses

Make the whole button clickable rather than just the text

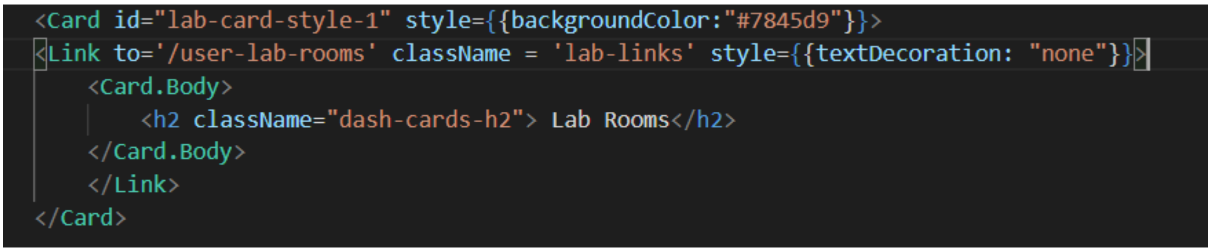
Maybe if the whole button was clickable on hover for each page instead of having to click just the text part.

A way to change your student name and profile picture would be cool.

I love the current colour scheme but maybe it could be a cool feature to have a light and dark mode that the user can choose (the current setup being the darkmode).

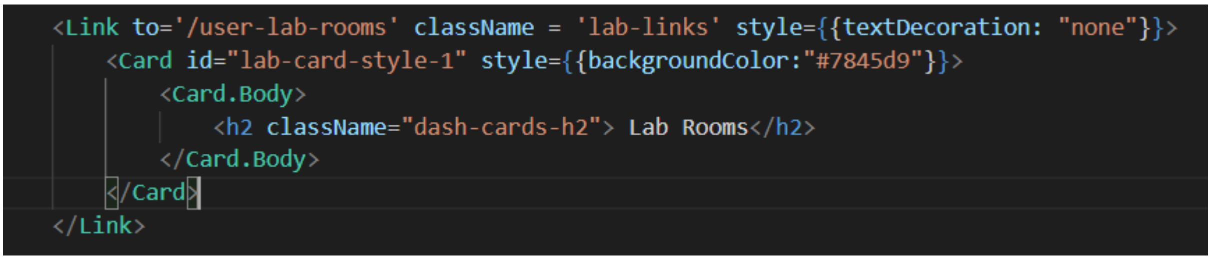
Overall, really cool idea that definitely has uses. This would have been used during my time at DCU had something like this been around. I love the layout and the colour scheme. The navigation was pretty straightforward too with an easy to understand UI.

Users pointed out that in order to click a link/button you would have to specifically click on the text on the inside. This caused trouble for some users to navigate to the next page/component. In order to fix this issue, we investigated what was happening for that issue to occur:



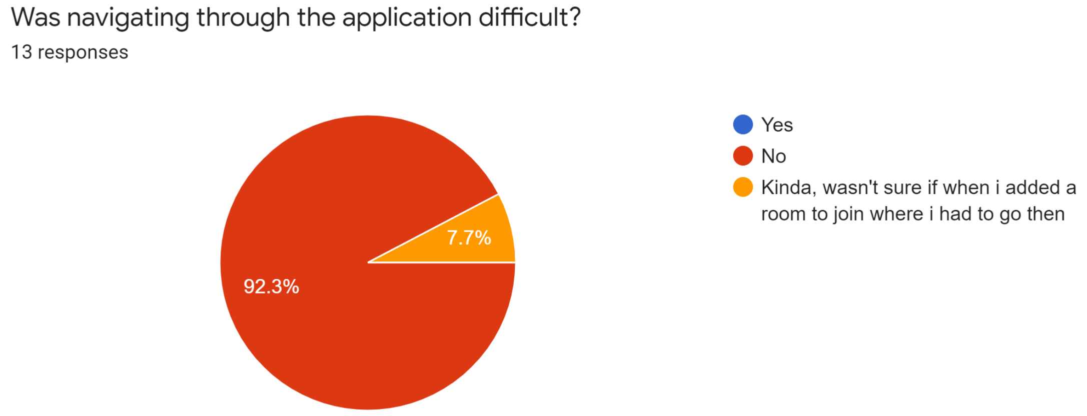
```
<Card id="lab-card-style-1" style={{backgroundColor:"#7845d9"}}>
<Link to='/user-lab-rooms' className = 'lab-links' style={{textDecoration: "none"}}>
    <Card.Body>
        <h2 className="dash-cards-h2"> Lab Rooms</h2>
    </Card.Body>
    </Link>
</Card>
```

In order to fix this issue we moved Link from within the card body to outside the card itself.



```
<Link to='/user-lab-rooms' className = 'lab-links' style={{textDecoration: "none"}}>
    <Card id="lab-card-style-1" style={{backgroundColor:"#7845d9"}}>
        <Card.Body>
            <h2 className="dash-cards-h2"> Lab Rooms</h2>
        </Card.Body>
    </Card>
</Link>
```

Another issue users encountered was the fact that they didn't really understand where to go after they Joined a lab room:

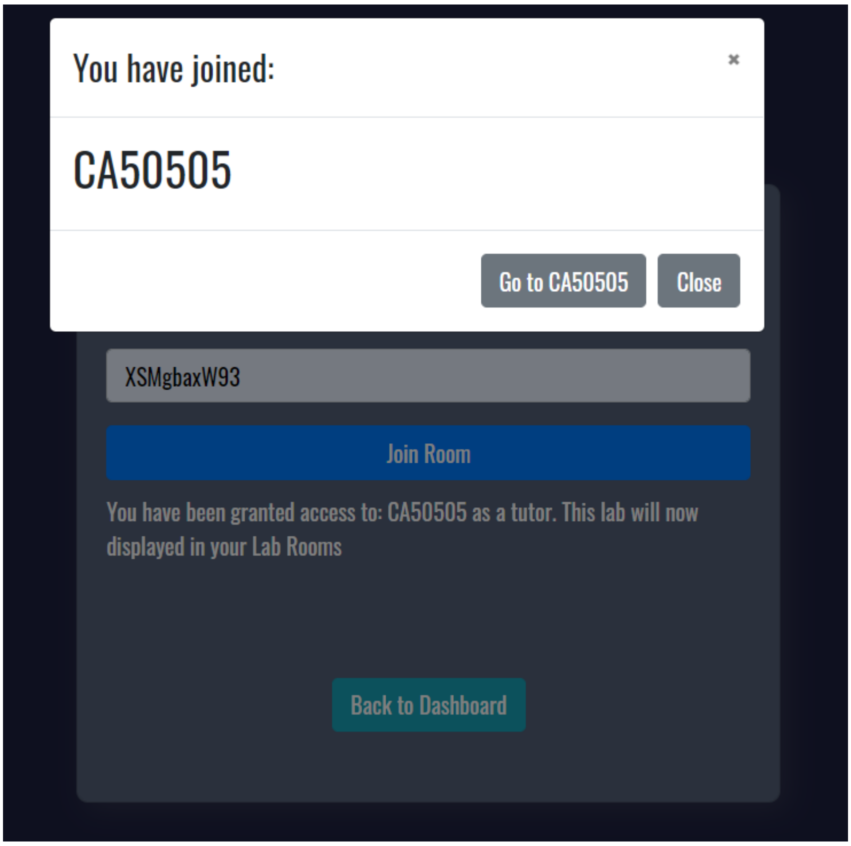


Was navigating through the application difficult?

13 responses

The initial User Interface displayed a message to the user saying for example : “You have been made a tutor for `Insert lab name here` , This lab has been displayed in your user lab rooms.

In order to give users a sense of direction, we implemented a pop up when the user clicked on the ‘Join’ button.



You have joined:

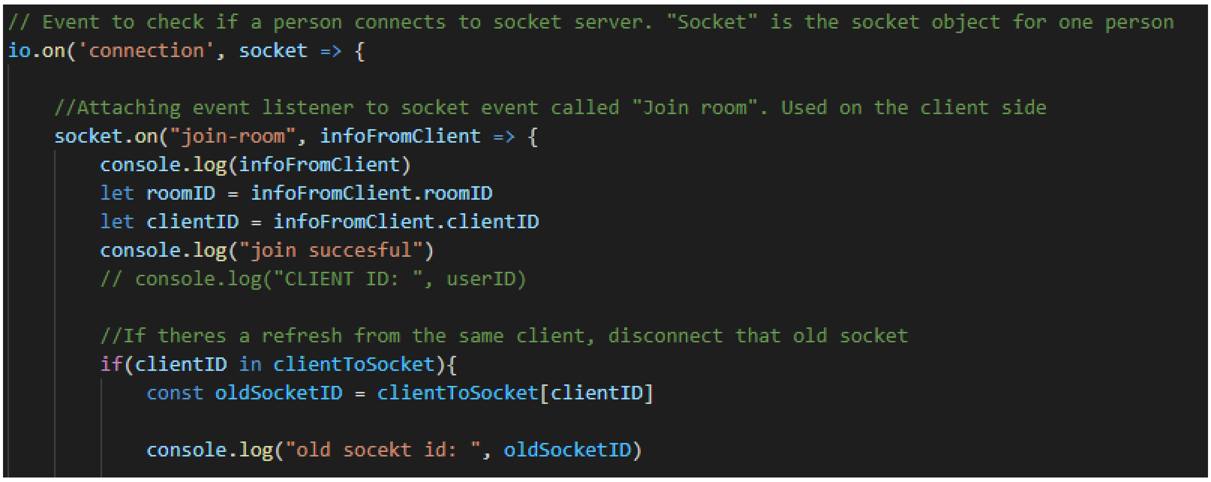
CA50505

XSMgbaxW93

You have been granted access to: CA50505 as a tutor. This lab will now displayed in your Lab Rooms

This pop up allows the user to directly go to the specific lab they just joined.

Users also experienced the issue of duplication, when they refreshed the editing session page. Refreshing would create a new socket ID and that would result in the same user appearing twice in the session. This was solved by checking the user presence in the lab against the usersID not their socket ID. The following code solved that issue.



```
// Event to check if a person connects to socket server. "Socket" is the socket object for one person
io.on('connection', socket => {

    //Attaching event listener to socket event called "Join room". Used on the client side
    socket.on("join-room", infoFromClient => {
        console.log(infoFromClient)
        let roomID = infoFromClient.roomID
        let clientID = infoFromClient.clientID
        console.log("join succesful")
        // console.log("CLIENT ID: ", userID)

        //If theres a refresh from the same client, disconnect that old socket
        if(clientID in clientToSocket){
            const oldSocketID = clientToSocket[clientID]

            console.log("old sockekt id: ", oldSocketID)
```