Assignment -3

Client-Server Chat Room using REST web service

The chatApp is built using the REST web service. The apps server-side have been built in node.js using the express framework and socket.io web socket library. Here, the framework provides robust features to provide a fast and powerful router that maps router callback to specific HTTP request methods.

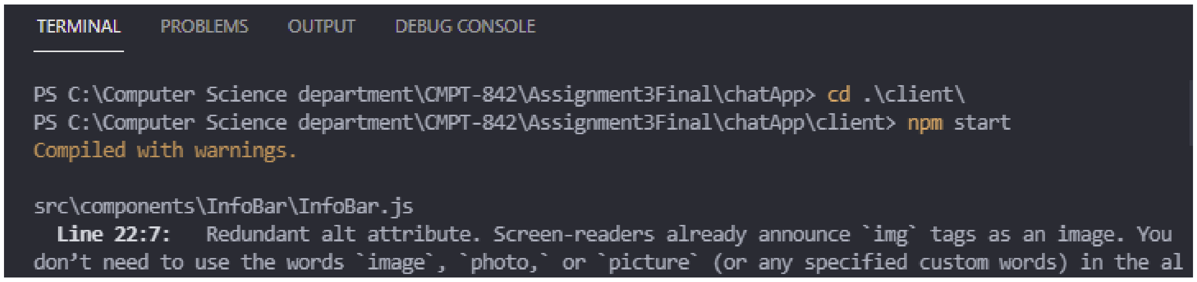
On the Client Side, the User can create a Room and Join the Room**. App.js** and **index.js** files route the path towards Join.js and client.js components as they are the main component. **Join.css** and **Client.css** contain all the CSS file for creating the form. Here A single map is created **Messages(<usermessage>,<userid>)** to handle all the messages being sent by the User

On the Server side. **Index.js** has all the REST web services, and Socket.io is used to emit the message to and from the User. Here, user.js contains all the CRUD operations, such as adding a user, removing a user, getting a User, and getting user room.

Here the node.js server is on port 5000 and react.js is on port 3000 so you will face a Cross-origin Resource Sharing (CORS) issue which can be handled either by intalling cors in the server or putting a proxy in the browser which is a temporary solution.

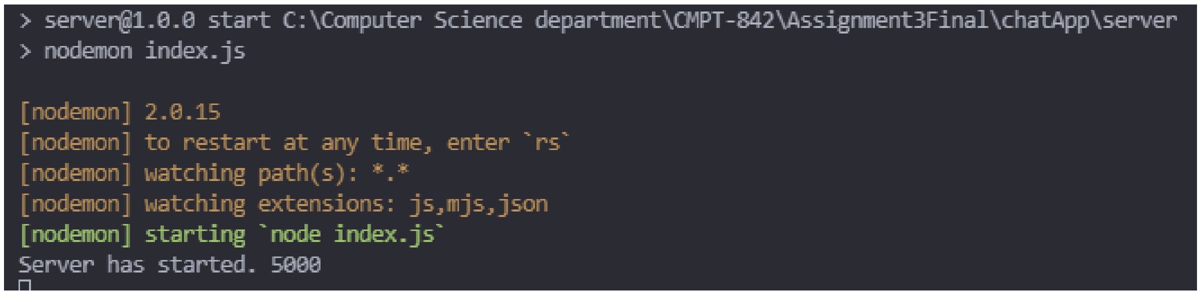
Steps to Run the chatApp.

1. Choose the preferred IDE to run the project. While building the project, VSCode IDE was used. Two terminals need to be created—one for running the client and the other one for running the server.
2. Here "**npm start** "will initiate the client and the server. It will look for scripts objects in package.json. To issue the command:- **npm start**, one has to be in the client and server director.



```
PS C:\Computer Science department\CMPT-842\Assignment3Final\chatApp> cd .\client\
PS C:\Computer Science department\CMPT-842\Assignment3Final\chatApp\client> npm start
Compiled with warnings.

src\components\InfoBar\InfoBar.js
  Line 22:7:   Redundant alt attribute. Screen-readers already announce `img` tags as an image. You
don't need to use the words `image`, `photo,` or `picture` (or any specified custom words) in the al
```

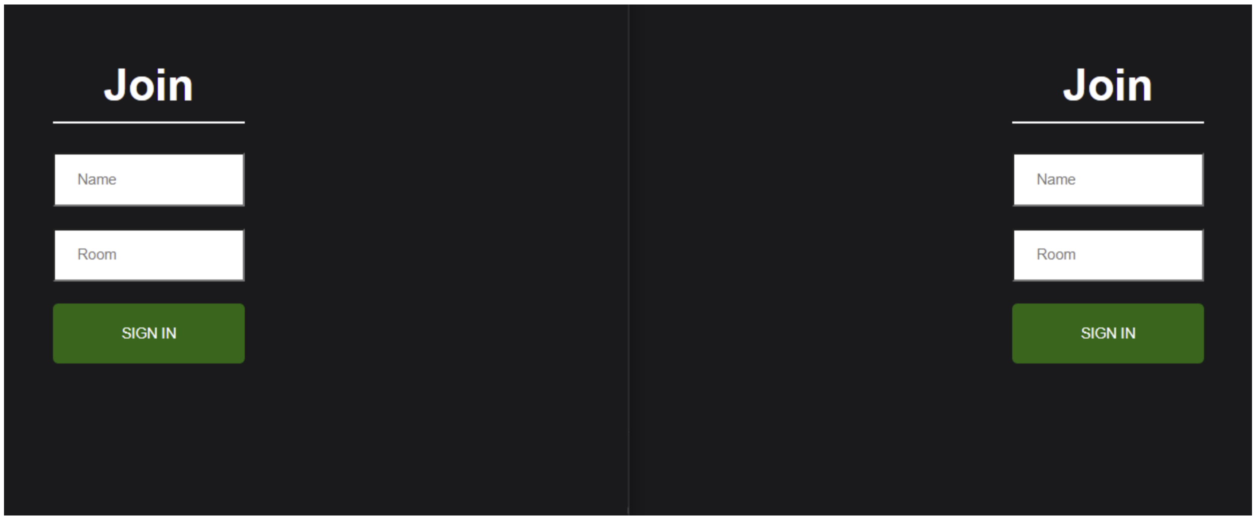


```
> server@1.0.0 start C:\Computer Science department\CMPT-842\Assignment3Final\chatApp\server
> nodemon index.js

[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Server has started. 5000
```

Fig: - Starting Client and Server.

3. After the client start, client will be hosted in localhost on port 3000, this is the landing page of the chapApp. To test the Chat application, we need to start two localhost in two tabs.

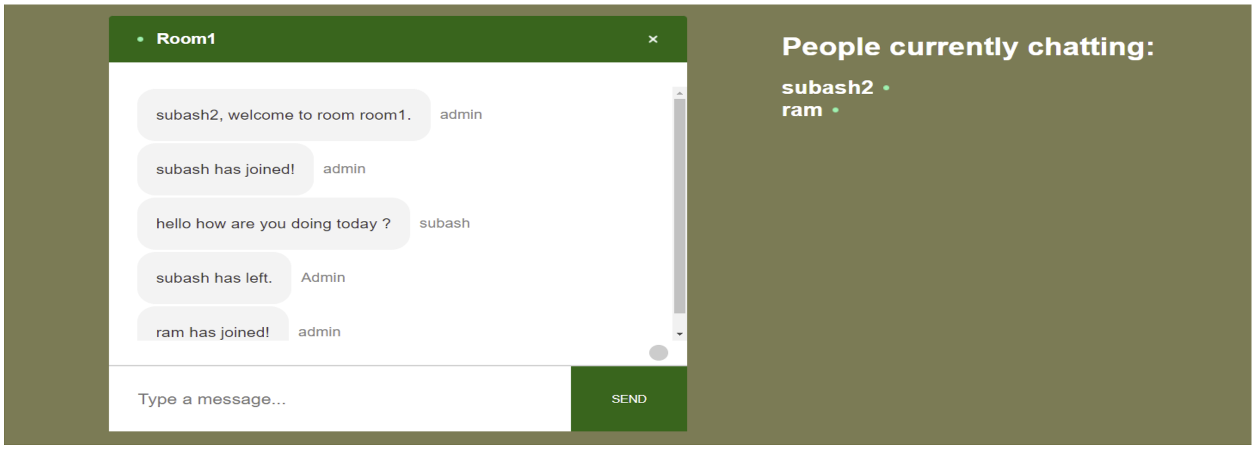


We can now signup and choose a Room.

4. To test the system, we need to choose the same room as the same room user can only chat for now in the system.

Let's choose Room1 to test chat between the clients.

5. The system will check if the username is the existing User or not. The application will display the number of users connected in the room. Here after connection a message will be created by the admin.

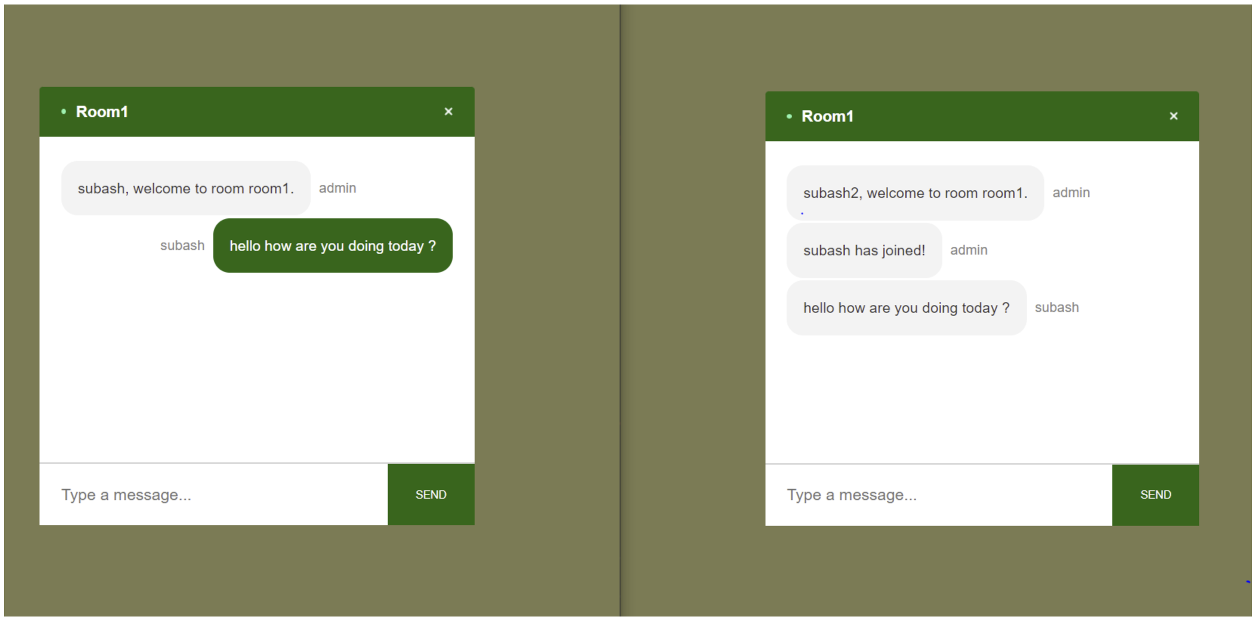


subash2, welcome to room room1. admin
subash has joined! admin
hello how are you doing today ? subash
subash has left. Admin
ram has joined! admin

People currently chatting:
subash2
ram

Fig: - Chat Room visual of Room 1 for user Subash2

6. Now User can chat between each other and enjoy the chatApp.


Room1

subash, welcome to room room1. admin

subash hello how are you doing today ?

Type a message... SEND

Room1

subash2, welcome to room room1. admin

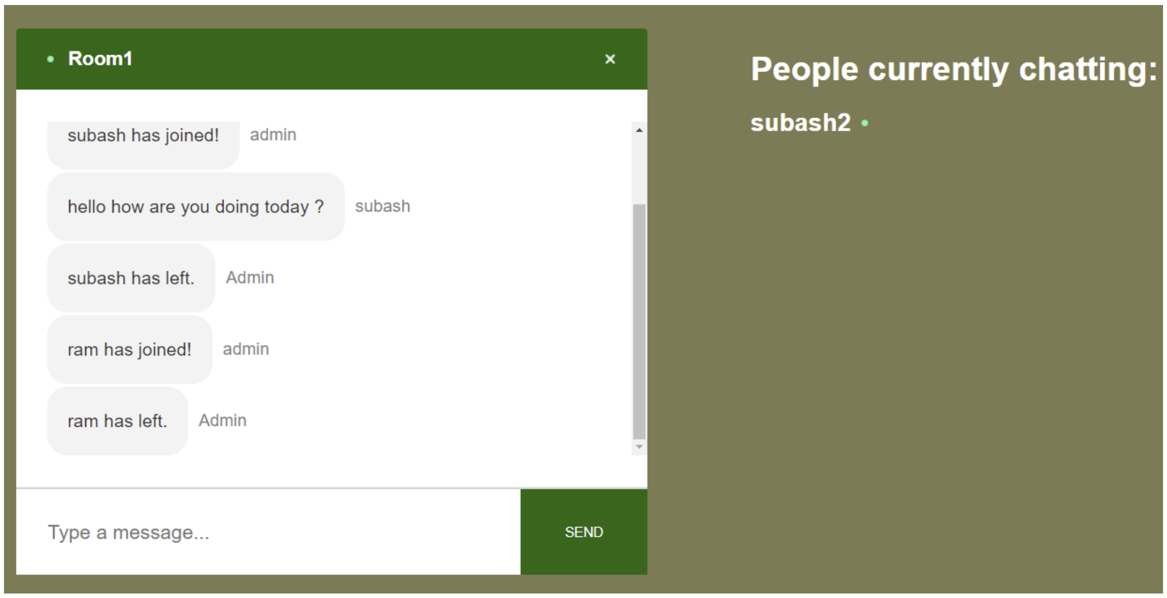
subash has joined! admin

hello how are you doing today ? subash

Type a message... SEND

Fig: - User1 and User2 are chatting.

7. If a new user adds in the room all the users will be notified and the new User will be greeted by the admin, and is if the User disconnects, it will also be notified and will be removed from the group.


Room1

subash has joined! admin

hello how are you doing today ? subash

subash has left. Admin

ram has joined! admin

ram has left. Admin

Type a message... SEND

People currently chatting:

subash2

Fig: User getting notified if someone leaves the chat room.