CDAC BANGALORE

PROJECT DOCUMENTATION

WEB BASED CHAT APPLICATION

Submitted By:

Akanksha Singh--200950181005

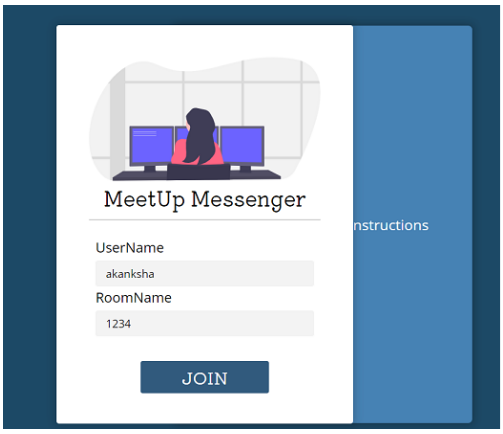
Anoop Singh--200950181011

Ashay Gore –200950181018

Shashank Pandey—200950181093

1.Project Functionality

- Main aim of the project is to allow a user to chat with another user or a group of users in real time.
- User does not need any security-based authentication credentials for logging in.
- User need to enter a unique username and chatroom id in order to create or join a chatroom.
    - If the user enters a new chatroom id , it will create a new chatroom.
    - If the user enters an existing chatroom id , it will enter into that chatroom.
- Emojis embedded within messages are supported.
- No message storage on server ,if room goes empty or any user refreshes the page then all messages vanish from UI .



2. **USE CASE**

2.1 PRESENTATION LAYER:

- When homepage is opened by the default routing path set by react router and there username and roomname is prompted.
- When user enters username and roomname and presses the login button then a localstorage variable "bool" is set true and react router directs the page with username and roomname embedded in url to chat component with help of "bool" and private routing.
- In chat component username and roomname is extracted from the URL and sent back to service layer and user gets the list of all connected users and becomes online.
- Whenever a user sends a message ,it goes with the socket event emitter to the service layer and the message is broadcast to the entire room with socket emitter.
- Whenever user refreshes the page all messages get lost as we are not storing messages to the server or client side
- Whenever user logs out then "bool" becomes false and the page is redirected to login page.

2.2 SERVICE LAYER:

- Whenever the chat component is mounted then client side socket emits the roomname and username to the server where server checks whether the room already exists or not and create the room according to need and adds the user to the room.
- When user sends the message client side emitter emits the message text with username to the server where within the listener same data is emitted to the whole room and every user receives the message.
- When user logs out or disconnect by any other mean then 'disconnect' socket listener on the server side emits the message to the whole room that particular user has left.

4.**PROBLEMS FACED DURING PROJECT:**

PROBLEM 1:

- Just by clicking on the JOIN button, without entering the user credentials, the chat application was redirecting the user to the chat dashboard.
  Redirected URL : http://localhost:3000/chat

RESOLUTION:

- We used PrivateRoute component with React router to resolve the above issue. If the user is logged in with the required credentials, go on and display the chat dashboard; otherwise, redirect the user to sign-in page again.

PROBLEM 2:

- We were facing problem while integrating the Login Page with the other components of the project.

RESOLUTION:

- Due to insufficient knowledge of Routing ,we did some POCs regarding React routes and we finally learnt how to integrate all the components with login page.

PROBLEM 3:

- We were facing issues with Socket.io concepts due to the off syllabus topic.

RESOLUTION

- We had to go through well with socket.io documentation to be well aware of socket methodology

5.LEARNINGS DURING THE PROJECT

- We learnt a number of technologies like Express ,Nodejs ,CSS, Reactjs and Socket.io.

- Despite the methodology used, there must be some amount of documentation available as resource should a team member leave or a new person is assigned to a role.

- We learnt how to use GITHUB which simplified the process of working with other people and made it easy to collaborate on project . It helped us to easily merge our changes in with the master branch of the project.

- We also used ASANA as our scrum board to organize tasks, assign specific tasks to team members, and track each task through its lifecycle.

Project Website Link:
https://60732428ac2621d92469d309--meetupmessenger.netlify.app/

GitHub Link:
https://github.com/ashaygore/webBasedChatApp/tree/master

Asana Link:
https://app.asana.com/0/1200017406834144/board