

Circle Open Chat: Writeup

Members: Daran “Nut” Thawornwattanapol 6380413
Rasika “Gloria” Aramvejanan 6380540
Chayapol “Airbus” Bunnag 6380518

Project name: Circle OpenChat — **Code name:** Don’t think, just do

How the project works:

In our chat web server, we divided it into 4 pages: Register, Login, Home, and Chat. When the client enters our web server, they have to register themselves in the register page(/register) by entering their information: username and password. These data will be backed up into our database, and the client will be allowed to log in to the Login page(/login), then they will be redirected to the homepage. On the home page, we provide a ‘+ join commu’ button which allows users (we will call our client as a user after this) to join or create the server by putting in the commu ID, and if the server does not exist, the web server will automatically create a new one. This data will link to the database once the user joins the new server, and they will be shown on the navigation bar on the left side of the web. At this moment, users can start to chat with their friends on the server by clicking the name of the server on the list. The web server will connect with the database to call the chat list of the server and display it on the Chat page(/chat). We design the chat bubble on the Chat page to be labeled with username, user’s role, and timestamp. If the chat bubble belongs to the user, we emphasize it with green color, while the others are in mustard yellow.

Findings:

As this is almost everyone’s first time in our team in frontend and backend web development. We learnt how to start the work and how to choose the framework for our project like Rocket and Vuetify. For frontend development, we have to use Vue to create GUI, JS to connect it with data as well as backend stuff, and CSS to style it up and each extra function from Vuetify APIs in the components. For backend development, we learned how to use Rocket, which is Rust framework, and how to implement an API in Rust, and how to connect to a database using diesel. Nevertheless, we studied how to connect between the frontend and backend of the web server. After we have learnt Rust from this course, we discovered how Rust integrates in many projects. Lastly, we found how to program and develop as a team. We were always used to programming individually, however, in our real life, many digital careers required us to work as a team and develop the web server on GitHub. Using Github makes our work much easier to communicate between frontend and backend developers and see how our project develops.

Summary:

In this project, we have set the project's planned features (mentioned in README.md/ project proposal.) And what we successfully achieved are starting the framework of our chat server project, designing and implementing the GUI of the web server, the ability to send and receive messages back and forth between users, and creating and joining the server. However, what we tried to achieve but were not completed are setting a role, removing people from the server, persistent data in a room, and tagging people via a message.