312 Project Report

Frontend: Done by Kevin

The complete frontend is in the branch dev3 and inside the “project” folder.

We only use html, CSS, and JavaScript for the frontend. The frontend included a welcome page, a login page, register page, and a chat page.

First, when client open the welcome page at project/index.html, client will have two choice, if user done have an account, they need to go to the register page by clicking the “Sign up” button. If user already has an account, click the “Login” button.

When user click the “Sign up” button, it will redirect to the sign up page at project/register page/register\_main.html. User will input a username. password, and upload their profile picture. Once user click the submit button, page will redirect to the login page. When user input their username and password and click login, it will redirect to the chat page. (\*Because the server is not up yet, the page will not check the username and the password for now, this part should be done by the server. The redirection will use the local files for now.)

When user get into chat page, user can show and hide the chat window by clicking the “Show/Hide Chat Window” on the bottom right to show or hide the chat window. So, there can be more room for the user interaction for later. Because the server is not up yet, The JavaScript will add 10 users to the online list and simulate when user receives message from other user by calling the APIs automatically for now. When user receives a message, there will be a red dot next to the sender to notify the user. When user click it, it will see the messages.

To make the frontend code easier to use, and reduce the server part’s work, I wrote a few methods:

**In project/register/register\_main.js:**

**Onsend(\_username, \_password, \_img):** This method will be triggered when user click the submit button in the register page. It will get the inputted username, password, and image path in \_username, \_password, and \_img. The server part needs to handle this and sent it to the server and handle it.

**In project/login/login.js:**

**OnLogin(\_username, \_password):** This method will be triggered when user click the login button. It will get the inputted username, and password in \_username, and \_password. Server part needs to modify this function to send those information to the server and handle it.

In project/mainpage/main.js:

**AddMessage(\_username, \_msg):** This method is called when the user received message from the server. The server needs to send the information like the sender, and its message. When the client receives it, call this method, to add message to the frontend.

**AddUser(\_username, \_img):** This method is called when the server notify the user that someone is online. When user receives the online notification, call this method, with the parsed \_username, and the path for their profile image, then this method will add username and show its profile picture in the online list.

**RemoveUser(\_username):** Works similar to the AddUser method, but this only takes a username. When this is called, the \_username will be removed from the online list in the frontend.

**OnSend(\_to\_username, \_msg):** When the user click the send button, this method will be triggered. You can get the \_to\_username, and \_msg. When this method is called, it needs to send something to the server, then the server relay the message to the receiver. For now, the server is not up, so it just print out a message string now.