

Assignment Phase 2

Individual Assignment

Due: 8am Thurs 10 October 2024

Introduction

For Assignment Phase 2 you will extend the functionality of Assignment Phase 1 based on what has been covered in the second half of the course. Assignment Phase 2 will store data using MongoDB, provide chat functionality using sockets, and support uploading and display of images.

MongoDB

The Node.js server storage for chatting channel and chatting history will be changed from storing in JSON files to a MongoDB database. The Node.js server should store the user and group data

Sockets

Sockets will be used to support chat communication in a channel. When the user logs in they can choose from the list of groups they are in. They can then choose a particular channel to communicate in. When they are in a channel the chat history should be updated as users send messages. The channel should also be shown in realtime when users join and leave a channel.

Image support

The chat system should allow users to specify a profile image (i.e., avatar). The profile image should be displayed in the chat history alongside their username for messages that they posted. The chat system should also support sending images as a chat message and will appear to all users viewing the chat. Image storage on the server can be as files in a specified directory, with the path to the file stored in the mongo database.

Video support

The chat system should allow users to have video chat. For this purpose, more APIs can be used in both the Angular and server side. For example, PeerJS can be used in Angular side and Peer server can be implemented on your sever side. For demonstrating the function of video chat, you may upload and run your app on elf.

Testing

On the server side, tests should be implemented to ensure that the routes function correctly. On the angular side, both unit and e2e tests should be implemented.

Git

Git must be used during the development of the chat system. We recommend that you use GitHub and share the repository with your marker. You will be marked on frequent updates to the repository and the usage of git features.

Documentation

Documentation of your implementation is required. You will need to provide the following:

- Describe the organisation of your Git repository and how you used it during the development of your solution
- Description of data structures used in the client and server to represent the various entities, e.g.: users, groups, channels, etc.
- A description of how you divide the responsibilities between client and server (you are encouraged to have the server provide a REST API which returns JSON in addition to a static directory)
- A list of routes, parameters, return values, and purpose
- Angular architecture: components, services, models, routes
- Describe the details of the interaction between client and server by indicating how the files and global vars in server side will be changed and how the display of each angular component page will be updated.