Assignment 2

Individual Assignment
Due: 5pm Fri 28 September 2018

Introduction

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

MongoDB

The Node.js server storage will be changed from storing in a JSON file to a MongoDB database. The mongo database should store all of the user, group, and channel data as well as chat history.

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 update as users send messages. The channel should also show in realtime when users join and leave a channel.

User Authentication

If not already implemented in Assignment 1, you should add support for users entering a password. If the password does not match it should ask the user to login again.

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.

Testing

Tests should be implemented on the server to ensure that the routes function correctly.

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 divided 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