Documentation

Sadeed Ahmad

Chat App – Assignment 2

Software Frameworks

1. **Git**

I have used visual studio code terminal to initiate the project into git hub and then used the terminal to update the code from time to time as I made the progress. First, I initiated my local repository with git init and then added the files through git add. I pushed different commits into the GitHub as I made progress or changes in my code. The github address for this project is

<https://github.com/sadeedch/chatapp>

1. **Data structure**

The data structure used in the Angular application consists of both a MongoDB database and Angular's local variables stored in each component. The database MongoDB stores all the information about the users, channels, groups and the chat history. The database structure is as follows.

* The “userList” collection stores the user's username and password, their role within the chat application.
* The “groupList” collection has the information about the group names.
* The “channelList” collection stores the channel names, the groups they belong to, the owner of the group and the members which this channel have.
* The “messageList” stores the message details of the chat. It consists of the time of the chat message, user which sent the message, the body of the message and the channel which it belongs to.

The Angular application interacts with the MongoDB Database through routes which is further discussed in the rest of this document. These two collections are the main data objects in the Angular application to support user authentication, and groups that are used in the chat application itself.

1. **Responsibilities**

The responsibilities in this application have been divided between the client and the server though the use of node server which uses the MongoDB, sockets and http routes. The server is being used to receive and request the data through the routes.  This is mainly used to read and write to the MongoDB which includes information about the currently registered users, groups, and channels.

Through this system of division, Angular app can request only what data is needed at a particular time, thus reducing load times and improving performance of the chat application.

1. **Angular Architecture**

The Angular app is split into components, services, and interfaces. Each page consists of its own component, which includes data requested from the server. The service stores the functions to make requests to get or post data. This process of storing the function in the service allows components to use these functions instead of making the request in the components which will limit those functions to those components only. The routes are used to what components are shown on a page like login, account or chat and to get the information about chat through the sockets.io.

1. **NodeJS as server:**

I used nodejs as the server and all the code to start the server is put in the server folder and server.js file. This file was created with the command “ng new ChatApp”. To start the server, just have to go to server folder and then type the command node server.js. This will start the server on port 3000 as described in the program.

I used http package to create a new server because the http package has a create server method, this method has two arguments respectively are request and response.

In the server.js file, it has all the apis used to make the functions of this ChatApp.

The server file also includes dbAuth.js file which takes care of the database.

1. **Routes**

|  |  |  |  |
| --- | --- | --- | --- |
| Route | Parameters | Return Value | Purpose |
| /login | Input username,  Input password | User credentials true or false | To check if the user credentials exits in the database. If yes, then login the user. |
| /account | User email | Userdata empty or not | Check what type of account the user belong to. |
| /group |  | All Groups data | To retrieve all the group details |
| /channel | Channel name | Channel data | If the group have any channels, then display them along with the group name. |
| /messageList | Channel id | Chat data | Retrieve all the chat history for a particular channel. |