Assignment 1

Chat app

Name: Ritchie WIlls, Student number: s2967766

2018

Contents

[Introduction 2](#_Toc523598815)

[Git 2](#_Toc523598816)

[Git Commands: 2](#_Toc523598817)

[Data Structures 2](#_Toc523598818)

[Client and Server 3](#_Toc523598819)

[Npm install: 3](#_Toc523598820)

[Angular Architecture 3](#_Toc523598821)

[Angular commands: 4](#_Toc523598822)

# Introduction

This project is to design a chat system with a dashboard for the chat system by the end of development will allow the Angular client-side and node.js server-side to responded between each other with a socket pipe on the network. Allowing the end-user and the online client users to enter different chat channels and. Chat functionality will be added at a later state of time in part two and user authentication is not required at this point of development until part two.

# Git

Git is an open source software that provides a version control system for tracing changes in computer files and coordinating work on those files among multiple people editing the one project. With using Git, we can generate a Git repository that contains readme file at the start with information about how to install and run the other files with in the directory of Git. The Git repository was updated each day by using branches and then merging to the origin master to the head of the project to keep it all organized. On the local user end I used the Git init command to initialize the repository by using terminal commands.

Then Git clone command was used which copied the existing repository over the cloud service file into the local file. From working on the project for a while I updated the repository to keep a backup by terminal commands of Git add to add any altered files running around. Sometimes you get a merge conflicts from pushing the same file at once this was solved and removed from the Git repository.

## Git Commands:

* Git branch “Branch Name”
* Git checkout “Branch Name”
* Git add -A
* Git commit -m “updating”
* Git checkout master
* Git merge “Branch Name”
* Git push origin master

# Data Structures

The Data structures of the system has the additional group admin and super admin functionary for the user’s purpose when login into their account. A user object is identified by their username and email address but the email address will not be used to send emails to the user on webpage. Using a login function that take in two values of username and email this function checks if the input fields are empty and the input matches with the data on the node.js server if the user has successfully sign in the user’s information is stored into local-storage property and session-storage allowing access to the full webpage network. The session-storage will get cleared when the page session is destroyed when closing the webpage.

When user is logged in they can logout which will clears the local-storage and the session-storage by using a function called Logout that handles the clearing of data from the user. There are two admin types from the user object from the json file. The first is called super admin which can make fresh users by inputting username and email then pressing down on a button which is added to the node.js server and also can delete users too. The functions of create user which takes in two input fields returns a new user and delete users that would find the users identity deleted from the server-side. The other admin type is group which can also create new users but can’t any delete users from service-side but can add socket channels and remove socket channels from the chat system.

# Client and Server

REST API is an internet protocol that allows connection between the angular client and server-side of the node.js which transfer data across. This was central component so the front end of angular could update data on the server of the node.js side. Get request functions are key so the project can receive the API data and also can find the routing paths of files and send back to the client. And the other main function is the post functions are used to send data so it can be display it on the webpage. Npm package of express on the node.js side handles both of these functions for the server. On the client side a static directory called dist is made this will let the node.js server run the angular client on port 3000 or localhost:3000.

## Npm install:

* Npm init
* Npm body-parse
* Npm express
* Npm socket.io

# Angular Architecture

The angular architecture is quite an important part of the development as it’s the client-side of the project to connect with the node.js server and installing tools like npm install -g @angular/cli kept the client-side update to date. When creating the project at the start routing was added to the angular architecture so the user could travel around the webpage and navigate on the port of localhost:4200 with ng new “Project Name” –routing which create new folder with the routing files. Importing Router Module into the app.module.ts is very significant so the project doesn’t get error while running through multiple pages.

The app components that are necessary to be produced for this project were home, chat, account, login, not-found and menu which are stored in the source file in the angular file as they are important to make the webpage design look clear and user-friendly for the user in the final product. They were made by using the ng command of generate component “Component-Name” on command Prompt The services that needed to be made within the development were the user services and socket, services allowing the angular client to connect to the node.js server. Data binding is an important part of displaying values to the HTML the which will automatically update the value. Ng build was used multiple time to check for errors and bug on the server-side but also have the angular project build and ready to use.

## Angular commands:

* Ng add “Name” – routing
* Ng Build
* Ng Serve
* Ng g c “Component Name”
* Ng g service “Service Name”