Chatlog

GitHub Repository Link:

https://github.com/GeekyAndy/1972435_AnudeepReddyPatlola_TCSMEANStackTraining.git

The project files are present in the Chatlog folder in the above mentioned GitHub repository.

Documentation:

This report is in regard to the Phase 3 End Project named Chatlog, as a part of the MEAN Stack Training program.

Aim: To create an online chat portal and store the messages in the database.

Technologies and Tools Used:

- Visual Studio Code
- Node JS
- Express JS
- Socket.io
- GitHub
- MongoDB

Report:

In this project I initially created subfolders by the names **controller, model, public** and **router** where in each subfolder has one file each. The **app.js** file is created outside the folders, which is used to create and establish the connection with the database. We install the **express mongoose** and **body-parser** modules using the command **npm install express body-parser mongoose** after using the **npm init** command to create the package.json file. We also install **socket.io** module using the command **npm install socket.io.** In the same app.js file, we provide the details of port number and the database name which we create on successful execution of the files.

I then created a **chat.model.js** file in the model folder, where I created the schema for the database. In order to use this schema in other file, I export this schema variable using **exports** which is imported using **.require()** statement in other files. In the **chat.controller.js** file, I provided a connection to the template present in the **page.js** file in the public folder.

The **chat.router.js** file in the router folder is created and stores the get and post methods of the functionalities on the website using **router.get()** and **router.post()** functions. The main functionality lies in the socket.io connections in the controller and app.js pages and also in the page.js file where we use the tag **<script src="/socket.io/socket.io.js"></script>** and also transfers messages using **socket.emit()** function. The data is represented in the form of tables on retrieval as Logs of the messages being exchanged. The same can be viewed in the terminal using the **.find()** command to show the successful connection and storage of data on MongoDB with the website using socket.io module.