**CONNECT.ME - CHATROOM APPLICATION**

**CSE 5306**

This project was implemented using Node JS.

**WHY NODE.js?**

* Node.js is javascript. So the same language can be used for front-end and back-end development.
* Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Reading and writing to network connections, filesystems and databases–all common tasks in web applications. All these tasks execute very fast in Node. Node allows to build faster and more scalable network applications capable of handling a huge number of simultaneous connections.
* The Socket.io library in Node js provides the functionality to create a server and listen to clients. It allows us to emit and receive reserved events like: connect, message, disconnect, join and leave along with custom events.
* Communication between client and server are done using HTTP protocols.

**SOFTWARE REQUIREMENTS:**

1. MySQL
2. Node.js

**HOW TO INSTALL?**

1. Download and install node.js from [www.nodejs.org](http://www.nodejs.org).
2. Download and extract the project file zip folder.
3. Run the database ConnectmeDump.sql file to setup the database.
4. Open the “*app.js*” file and update your MySql database username and password in line 49 and 50.
5. Run a command terminal and navigate to the extracted folder and execute the following commands:

* **npm install**

1. Once the installation is complete, type the following command in the same command prompt:

* **node app**

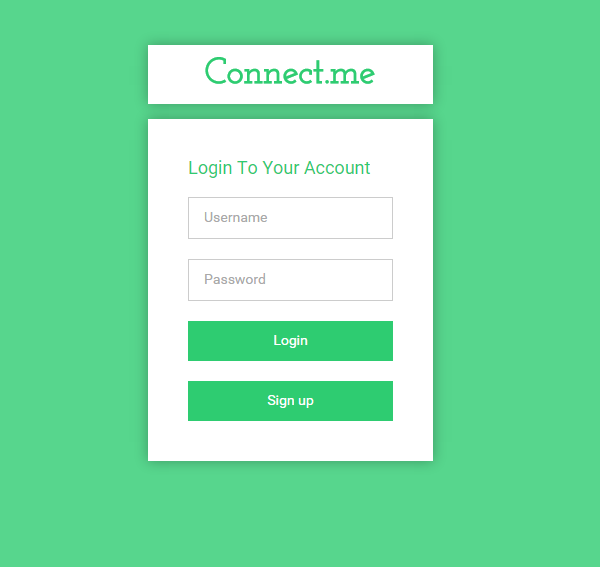
1. Go to your browser and open the following link: **http:/localhost:3000/**
2. The application is ready to use.

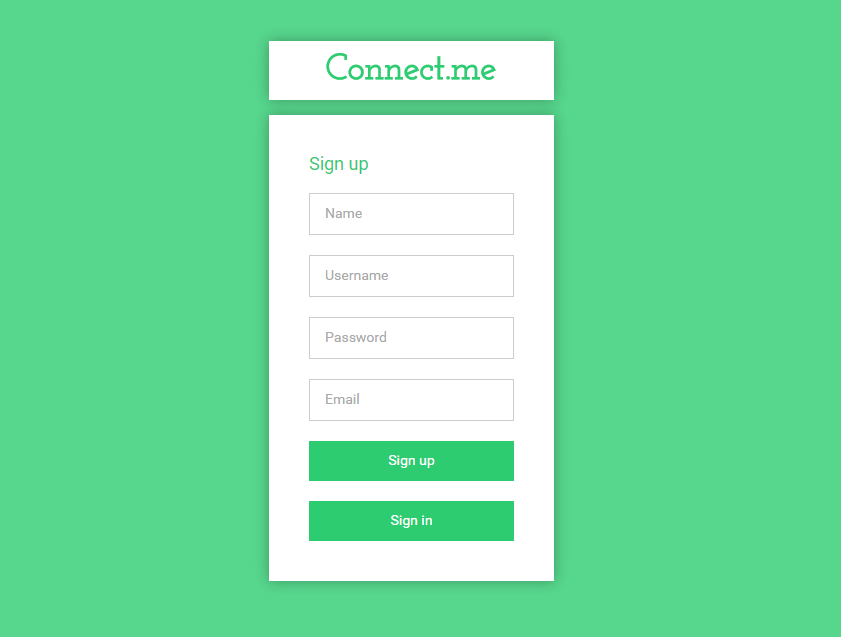
**HOW TO USE?**

1. Click on the Sign-up button and register yourself.
2. Once registered, it would redirect you to sign-in with your credentials.
3. Once authenticated, you would see the chat homepage.
4. Towards the right, you can see the available chat rooms.
5. Click on ‘+’ to create your own chatroom. In the modal, enter a unique chat room name and click on ‘create’.
6. A chatroom with the name defined by user is created and also added in the chat room list.
7. ‘Join’ must be clicked to join a chat room.
8. If any other user has joined the same chat room, a notification is displayed in the same chatroom window. For ex: ‘mary has joined the room’.
9. Similarly, if a person has left a chatroom, notification saying user has left the chatroom. For ex: ‘mary has left the room’.
10. User can leave a chat room by clicking on the ‘x’ sign on the chat room box.
11. User can also simultaneously chat with people in multiple chatrooms.
12. ‘Logout’ option allows the user to log out of the application. Once the user logs out, he will automatically be logged out from all of the chat rooms.

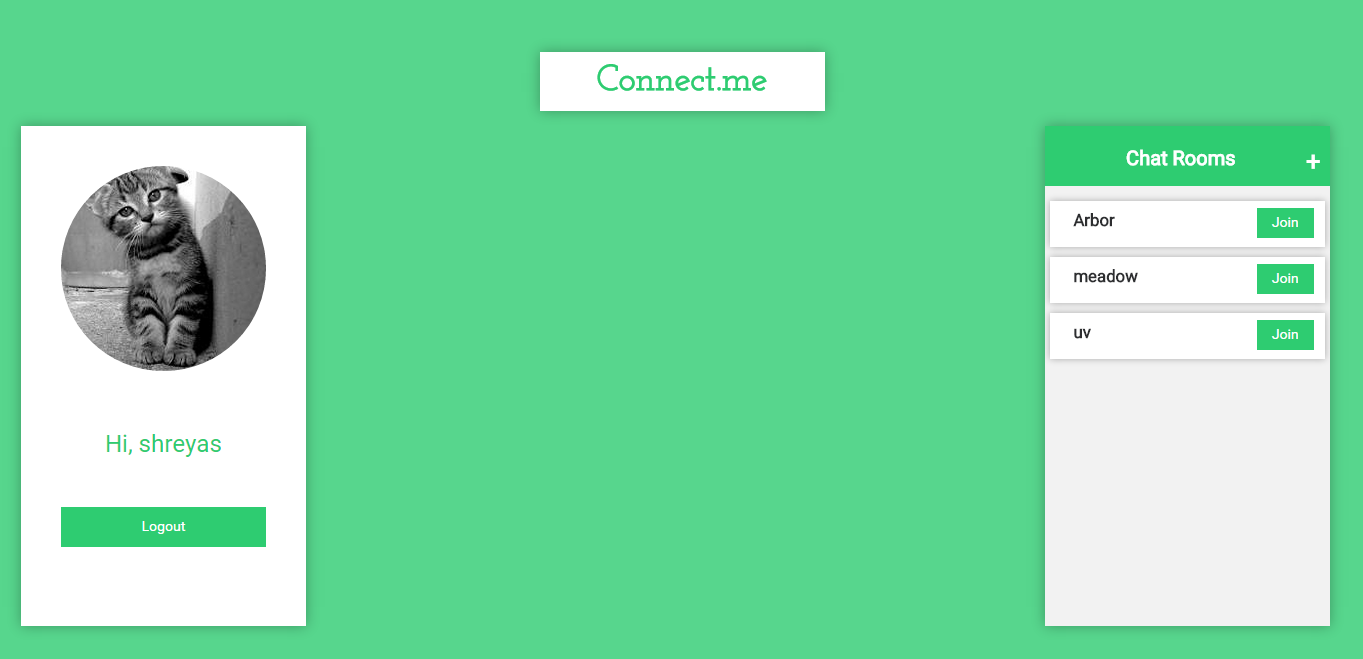
**SCREENSHOTS**

**Login screen**

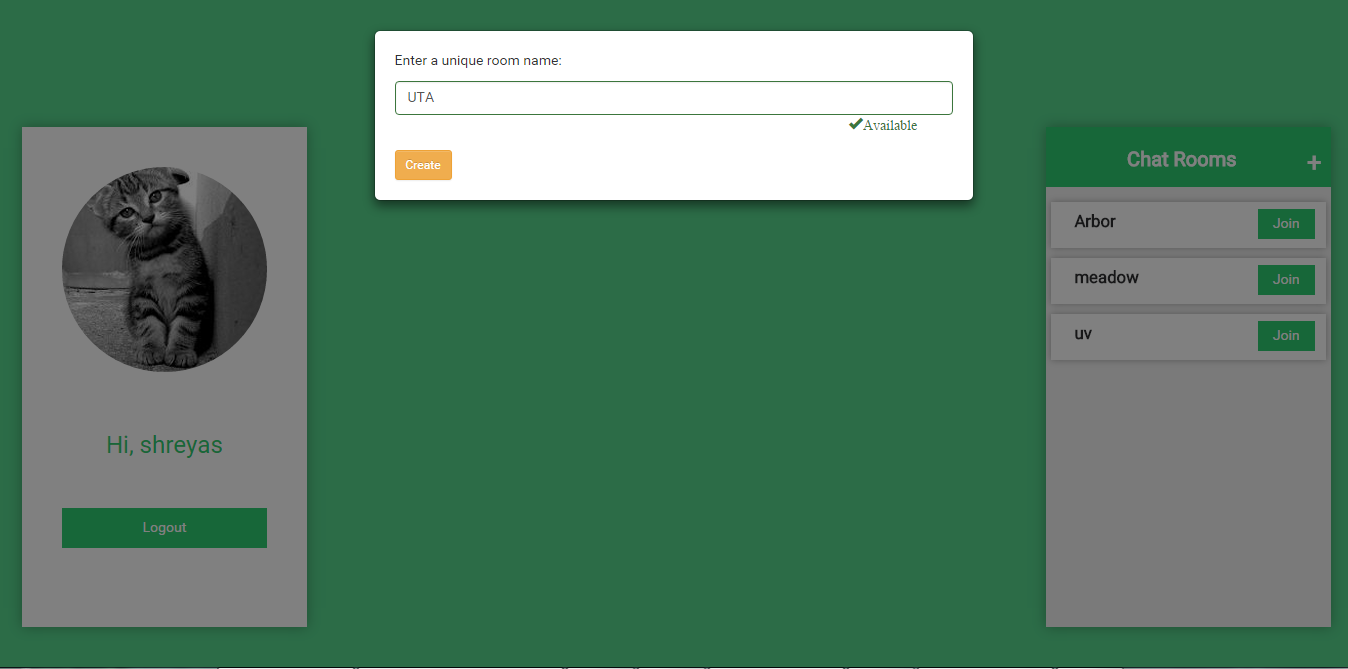
****

**Sign up page**

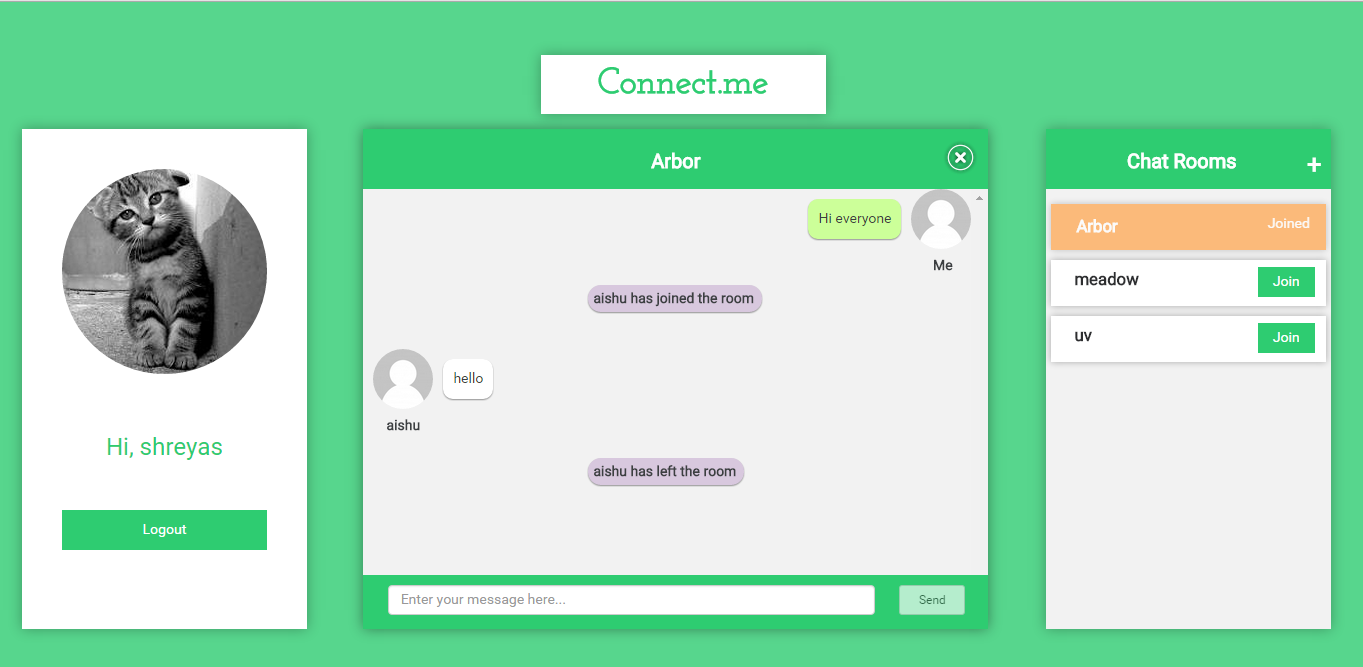
**Home page**

****

**Create chatroom dialog box**

****

**Chatroom Window**

****

**Reference:**

* <https://vijayannadi.wordpress.com/tutorials/sample-chat-app-using-nodejs-socketio/>
* <http://www.sitepoint.com/using-node-mysql-javascript-client/>
* <http://www.designsave.net/2014/08/responsive-html5-login-forms-free.html>
* <http://tutorialzine.com/2014/03/nodejs-private-webchat/>
* <https://www.pubnub.com/socket.io/>
* <https://nodesource.com/blog/understanding-socketio>
* <http://socket.io/docs/>
* <https://www.google.com/fonts/>
* <http://getbootstrap.com/components/>