README

Charlie Bowler

1402402

Enterprise Web Coursework

How code is laid out:

File structure is made up of two main folders – app and public, as well as server.js

Server.js is used to start the application, and is used for the mongoDB.

App

This folder is where all the backend files are kept, with two main folders called models and routes.

Models are the js files for the chat and the user

Routes is where the api file is kept that links all the pages together

Public

This folder is where all the front-end files are kept, with another two folders called app and assets, where app is the application files and assets is where the assets are stored like the css and the js files and dependences.

The app folder is then divided into three folders, controllers which control the main functionality of the application, services is where the services and authentication are stored, and finally the views is where all the pages are stored.

This app is a racing simulator, in which each user created and logins to a profile, and then simulates races and gets time for each track.   
This is done in the form of each user selects a track to race, and then the user is given a time of how well they have done. This is then displayed on the leader board, and other users can race and beat the times.

There is also a chat system where different users can speak to others logged on, as well as a marketplace where users can use currency gained from races to spend on upgrades to improve their times – with the goal being to get the fastest times possible.

Functionality:

A Mongo Database is used as well as node to create and run the server. The application also allows for the user to register or login. When a user registers for the first time, their details are stored in the monodb. This displays the different fields required for a user, as well as having the password automatically hashed so unable to be viewed.

Also, bootstrap was used as the very basic layout for the website to save time and make sure all was working with the MongoDB

The application also allows the user to login, with correct authentication seeing if then user hasn’t inputted a certain field, or where the user cannot be found in the database.

Once the user is logged in, the user then has a token stored, in order for other pages to see which user is currently on the system. Finally, the user can then logout which gets rid of the token and logs the user off.

What doesn’t work

The chat system is very basic. All it currently does is display what the user types. There is currently no market place or leader board as there was a problem with the pages and time restrictions made it impossible to finish.

What would like to add

Obviously to have all the aspects working correctly like chat, marketplace and leader board, as well as the ability to actually compete with other users.

A better design of the actual application as well, as currently very basic was used.

Obviously, functionality doesn’t meet the required standards, as currently the only things that work are the login/logout systems, as well as register and being able to start and stop the database and node server.

Acknowledgments

There were various tutorials on YouTube used as guides for certain pages and functionality so full credit to those, as well as the lectures and labs for knowledge on how to use certain applications and dependencies.   
Bootstrap was also used as starting template for the index page of the application