WEB405 Project 3

Express Web Server with MongoDB Database

Create a two player game, where the computer is the opponent. You can use the same game as assignment 1 or create a new one. This time, the client will be the web browser. You will use Express and MongoDB on the server. You can use any `npm` modules you like to assist in your app.

Start of game

1. The player will connect to the game server using a web browser and the game will start. (Your app only needs to support the latest version of Firefox or Chrome.)
2. Many clients can connect to the same server but they will each play their own game.
    - since HTTP is "stateless" your client will need to send an identifier to the server each time. You can ask your player for a nickname when the game starts and use this. (This is not the best, since several users could use the same nickname, but it will simplify things for now.)

Coding

1. Name your server file `server.js`. This will be run with Node. You can name any other JavaScript files that you may need for your server as you like.
2. Your client (the web page) will also need to run javascript, call this file `client.js`. You can use any JavaScript/CSS libraries you like (such as jQuery, Skeleton, Bootstrap, etc.)
    - configure Express to server the static files to the web browser
3. Make your web app look good. Add CSS and graphics as needed. Free images of cards, dice, etc. are available online.
4. When you write your code, do the following:
    - Run Express on port 3000
    - Name your MongoDB database with your name or initials
5. You may use as much code from previous projects and assignments as you like.

Comment

Add a comment at the top of each file (after `"use strict";`) with your name, date, and the project name.

```
"use strict";
/*
*Name of My Game*
WEB405 Project 3
Author: *YOUR_NAME*
Date: *DATE*
*/
```

Also, include your name so that it is visible at the bottom of your game webpage.

Documentation

Include a `rules.html` file (linked on your game webpage) that includes the following:

1. Outlines the rules of the game (as you have implemented them)
2. Include one or more links to websites you used to research the rules
3. Note any rules that you ignore or modify in order to simplify your game
4. Make a note of any known bugs in your code

The CSS for this page should be a relative link so that it can be opened by double clicking the file, or clicking the link on the game page.

Submitting

1. Demonstrate your game to the class.
2. Submit a zip file including your server, client files (HTML, CSS, JavaScript), `packages.json` file and a `rules.html` file describing the game play. Include any other files such as custom modules. Do not include the `node_modules` folder.
3. Since you won't be submitting a database, make sure your code does not depend on any data existing in the database before the game starts.
4. You should test your submission by doing the following:
    1. unzip your submission file in a new location
    2. Make sure there is no `node_modules` folder
    3. Run `npm install` to install modules
    4. Start MongoDB with an empty `db` folder.
    5. Connect your browser to `http://localhost:3000` and make sure your app works as it should.

Everyone's code will be shared with the class. That way you can learn from each other.