Web Development – Mr. Turner

Project – Login and Registration

**Project Overview**

Create a simple user login and registration page.

**The Assignment**

Create a text file that contains the following information for several users:

* Username
* Password
* First Name
* Last Name

This file can be built in any string format or you can create the users as JSON objects (if using JSON objects, remember to put all properties and their values in double quotes).

Build a simple form that allows the user to enter in a username and password.

***Add a link to a registration page.***

When the form is submitted, the form should send the data, as post data, to a welcome page. This request will go through a Nodejs webserver file.

When the webserver receives the request, it will check to see if login information has been sent. If there is login information, it will go into the text file with all of the usernames and passwords to see if the login is correct.

1. Read in the text file.
2. Parse the string data into readable users.
3. Find the username/password combination that matches the one entered.

If a match is found, return the first and last name data to the welcome page. It should say, “Hello FirstName LastName”.

If there is no match, it should say “Username or Password is incorrect.”

***On the registration page, create a form that allows the user to enter his or her first name, last name, and to choose a username and password.***

***When the user submits the form and that data goes to the web server, it should process the data as follows:***

1. ***Check the data file and make sure that the user name does not exist already. If it does, then the server should send back an error message and return the user to the registration page.***
2. ***If the username does not exist, save the user’s information to the data file and return to the login page.***

**Programming Hints**

* You can’t do anything until you’ve checked to see whether or not the user is legitimate, so don’t bother reading in the user text file asynchronously (refer back to early Nodejs code to see how to read in synchronously).
* Don’t overcomplicate the server page. Beyond reading in the data and accessing the POST data, checking the client’s entry against the data is just basic Javascript.
* If you understand the templates that we’ve been using since beginning Nodejs, you should be able to adapt them to this project.
* Take your time.
* Consult with your peers.
* Ask me questions when you have them.

**Writing Data**

There are two important functions belonging to the ***fs*** package that will allow you to write data.

writeFile(filename, outputText, function (err))

The writeFile function will open up and completely overwrite an entire file. If the file does not already exist, it will create the file. writeFile has 3 parameters.

* **filename** - a string containing the name of the file.
* **outputText -** the text being written to the file.
* **function (err)** - a callback function that handles error data.

appendFile(filename, outputText, function (err))

The appendFile function will open and add text to the end of a file. If the file does not exist, it will create the file. appendFile has the same 3 parameters as writeFile.

There are also the synchronized versions of these two functions (writeFileSynch and appendFileSynch). Each of these functions works like their asynchronous counterparts, but there is no callback function parameter.