DMIT2008 Assessment 2 Part A

Task: Employee Manager Login and Registration.
Due: Midnight on Sunday February 28th, 2021
Weight: 15%

Assessment Objective
In class we have been working on the employee manger site. The end goal is to have a simple client application running on Heroku that has a registration form, a login form and a dashboard that displays data from a custom API. For this build you are going to complete 3 tasks and an optional bonus task.

Task Register Post Route /register 5 (5 marks)
Simple form that has three inputs username email and password.These inputs are the data we need to create a new account. The form data will be validated on the server before sent and on the server using express validator. Once the form data is validated server side you will add the data to the users.json file. Give each new user a unique ID using the uuid module. You can find the link to the module in the section called required modules. Once the registration process is completed respond with a redirect to the login page.

Task Login Post Route /login 5 (5 marks)
The login form has two inputs, one for the user's email and the other for their password. The form data will be validated on the client before being sent to the login post route to be validated client side. If the email and password return true for authentication the redirect the response to the dashboard.ejs file.

Task Dashboard Get Route /dashboard (2 marks)
Before you start, move the ***dashboard.html*** page to the views folder and rename it to ***dashboard.ejs***. We want this file only to be accessible if the user is authenticated. This is a protected route and we will need to add express sessions in order to add a session variable to test when a user of the site tries to access the /dashboard route. The dashboard should only be accessible to users who are logged in and authenticated.

Bonus Task: /api/v1/users Get Route (5 marks)
Create a page called users.html and add it to the public folder. Style it in accordance with the other pages. Add a link on the signup index and login pages to this page. Use one of the other pages as a template for this page. You just need the navigation and the header elements.
Attached a javascript file named users.js to the page. Add this file to the public/js folder. The script will fetch a list of all users from the /api/v1/users endpoint. You will create the endpoint using the app.get() method inside the server/index.js file. The endpoint will return the data from the users.json file as JSON and your client code will display the data on the users.html page.

Task Deployment (3 marks)
project deployed on github..
Git repo is deployed to Heroku.
Heroku app has a ***semantic name***, and not the random generated one.

NOTE: Heroku will not persist your data between opening and closing the browser. Do not worry about this. You would need to add a database layer to the app to save the data between user sessions.

Project Setup
Use the folder structure that we have created in class. Keep the route paths names to **/login /register /dashboard** and if you are doing the bonus task the route will be **/users**

Git Repo
Setup a git repo for your project. You will be using the repo to manage your application code and to deploy to netlify. Do not forget to include a .gitignore file in your project and make sure you only upload the files that are required by your project to git.

All projects must contain a Readme.md.
This file should contain a description of your app and what it is used for as well as how to use it.
There should be instructions on how to install your app in development and in production.
You should list any special tooling you use and provide links to the tools and documentation.
Provide a link to a working live demo of your app
**** Do not include the node_modules/ directory in your git repo ****

Package.json
Initialize the project using npm.
Add a script with the command "start":"node server/index.js"

Procfile
Make sure to create a procfile and include the command web: npm start

Code Format and Comments
Your code and comments should follow and adhere to JavaScript best practices. Use the Airbnb style guide as a reference.

Modules
For this project you will need to install the following modules in addition to the modules installed in class.

express-validator:
https://www.npmjs.com/package/express-validator

express-session:
https://www.npmjs.com/package/express-session

uuid:
https://www.npmjs.com/package/uuid

cors:
https://www.npmjs.com/package/cors

Submission
Add a ***link*** to moodle to your github repo for your project.
Add a ***link*** to moodle to your app deployed to heroku.
****Late submission will not be graded****

MARKING KEY

Tasks	Grade	Received
Register Route Validation completed client side with error handling. Validation completed server side with error handling Data is written to the users.json file in correct format. Redirect to login	**1** **2** **1** **1**	**/5**
Login Route Validate username and password client side. Validate username and password server side. Authentication redirect to login. Custom authentication middleware. Session management variable for authenticated users.	**1** **1** **1** **1** **1**	**/5**
Dashboard Route Dashboard route is protected. Dashboard renders an ejs template.	**1** **1**	**/2**
Bonus Page displays the unique id, username, password, and email for each user in the users.json file.	**5**	**/5**
Submission Project readme is available on git is well formatted and contains a good overview of how to use and install your project. Code is well committed and formatted. All functionality completed (minus the bonus). App is deployed to Heroku using a semantic name. All of these must be completed to get full marks.	**3**	**/3**

Total /20
Grade