CS 465 Module Three Assignment Guidelines and Rubric

Overview

In this module, you will complete customer-facing web pages based on customer requirements provided in a wireframe document, adapting it to the software constraints and MEAN stack architecture. You will take the static HTML site and move it to dynamic JSON using Handlebars (HBS).

Specifically, in this assignment, you will use Visual Studio Code (VS Code) to edit the HBS templates and JavaScript within the Express framework to implement Model View Controller (MVC)-style functionality.

Prompt

Continue developing the full stack web application for Travlr Getaways. Follow the **Static HTML to Templates with JSON** section of the CS 465 Full Stack Guide carefully to successfully complete the following tasks:

- Develop and run a customer-facing frontend that meets client software requirements.
- Dynamically render web pages by utilizing the templating engine, Handlebars (HBS).
- Utilize routes and views to align with the client's software requirements using Model View Controller (MVC).
- Test to ensure that static HTML transitions to dynamic JSON data and renders properly.

What to Submit

Submit a zipped file folder which includes the updated Travlr website (travlr.zip) and the trips.json file.

Please note: You will "commit" or "push" your local Git repository module3 branch to your GitHub repository. This procedure will become second nature to you as you move through the course modules. In the end, you will have a full stack web application with branches that represent key stages of building a full stack application leading up to full project completion.

Module Three Assignment Rubric

Criteria	Proficient (100%)	Needs Improvement (70%)	Not Evident (0%)	Value
Architecture	Develops and runs a customer-facing frontend that meets client software requirements	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include addressing code errors	Does not attempt criterion	25
Views: Rendering	Dynamically renders web pages by utilizing a templating engine	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include the web pages' ability to dynamically render data	Does not attempt criterion	30
Routes: MVC Controllers	Utilizes routes and views to align with the client's software requirements using Model View Controller	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include utilizing partials or view fragments in common UI elements	Does not attempt criterion	30
Testing	Ensures that static HTML transitions to dynamic JSON data and renders properly	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include data not displaying correctly or rendering problems	Does not attempt criterion	15
			Total:	100%