CS 465 Module Five Assignment Guidelines and Rubric

Overview

After successfully wiring up the database, you will create a second application, app_api, to separate the RESTful endpoints. This approach is known as Separation of Concerns, which amounts to separating an application into logical layers or segments to reduce complexity, minimize the scope and effort of future changes, and reduce the chance of defects going undetected. This refactoring allows for three different clients—Angular.js, Express backend, and any external callers (applications)—to use the API when integrating Angular into your application.

In this assignment, you will ensure the database and API are wired to the frontend and complete testing to check routes using the methods FIND to locate items in a table and GET to retrieve a file.

Prompt

Refactor the database access code from being embedded within the backend website logic to a separate top-level site accessed at the endpoint "/API". Follow the instructions in the **RESTful API Architecture** section of the CS 465 Full Stack Guide and perform the following:

- **Establish Routes:** Establish routes for the API endpoints. Be sure to standardize naming conventions for routes and parameters to return specific trips.
- **MVC and REST API:** Refactor the MVC app server controller model code for use with the app API REST endpoints using Mongoose.
- **Testing:** Test the RESTful API application using Postman.
    - Check the routes and retrieval logic of specific product requests using the Mongoose functionality FIND method.
    - Verify that errors are handled correctly by returning HTTP status codes.
    - Verify that JSON data for a requested individual trip and a collection of all trips are returned.

What to Submit

After completing your work for this assignment, submit the updated travlr.zip zipped file folder, which includes the trips.json file.

Please note: You will also "push" your local Git repository module5 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 Five Assignment Rubric

Criteria	Proficient (100%)	Needs Improvement (70%)	Not Evident (0%)	Value
Establish Routes	Establishes routes for the endpoints and maps them to controller methods correctly	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include endpoint following RESTful best practices or appropriate use of models	Does not attempt criterion	30
MVC and REST API	Refactors the MVC app server controller model code for use with the app API REST endpoints using Mongoose	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include the Express app server controller being able to request and process JSON from the API correctly	Does not attempt criterion	60
Testing	Checks routes and retrieval of specific product requests by testing the REST API	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include refactoring the code to obtain better results	Does not attempt criterion	10
			Total:	100%