CS 465 Module Seven Assignment Guidelines and Rubric

Overview

You now have a working full stack project that serves both the public customer-facing side and the admin side of the application. Now it's time to add security functionality for secure login authentication. In order to complete the project, you will learn about and apply security features to the single-page application (SPA), such as producing secure web tokens for authentication.

Prompt

Follow the instructions in the **Security** section of the CS 465 Full Stack Guide and address the following:

- Develop a security protocol by adding a single layer of security to the SPA admin side, including an HTML login form for the admin site and logic to secure the administrative endpoint methods.
- Test the security protocol by testing the authentication with mock data and verify the new user class API endpoints using Postman.
- Incorporate the security protocol by refactoring the frontend and backend websites.
- Test the frontend website by logging in to ensure CRUD features are now enabled.

What to Submit

Submit the updated travlr.zip zipped file folder.

Please note: You will also "push" your local Git repository module7 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 the full stack project completion.

Module Seven Assignment Rubric

Criteria	Proficient (100%)	Needs Improvement (70%)	Not Evident (0%)	Value
Develop Security Protocol	Develops a security protocol for the single-page application to add login authentication with secure admin endpoints	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include ensuring security is fully implemented	Does not attempt criterion	40
Test Security Protocol	Tests the code to ensure full stack web application is secure and functioning	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include thoroughly testing all combinations of user interaction	Does not attempt criterion	10
Incorporate Security Protocol	Refactors the frontend and backend websites to incorporate the security protocol	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include secure elements appearing without logging in or API endpoints being able to be invoked without secure tokens	Does not attempt criterion	40
Test Frontend	Tests the frontend website by logging in to ensure CRUD features are now enabled	N/A	Does not attempt criterion	10
			Total:	100%