IT232 Final Project
Score: _____/100 v1.0

Link to assignment: https://classroom.github.com/a/R2bDGnnR

Specifications

Before beginning, accept the GitHub Classroom assignment and clone the assignment repository to your local computer using VS Code or the CLI. All files made/modified in this project should be made in your local repository.

Create a quiz application that allows users to take quizzes on various topics.

You will need to create a set of questions and answers for the quiz. You must create a class to represent questions, with properties to represent the question text, answer choices, correct answer, etc. You can make different classes for different types of questions (multiple choice, true/false, short answer, etc.).

You will need to use GET and POST requests to submit answers and navigate the website. When the user submits an answer, you can use a POST request to send the user's answer to the server. You can use a GET request to retrieve the next question from the server and display it to the user.

Questions must not be baked into the HTML and should be rendered into pages dynamically using PHP. For multiple choice questions, the answer choices should be randomized so that if the user takes the quiz again, the options for the questions appear in a different order each time.

You'll need to grade the quiz at the end and show the user their score. You can use PHP code to compare the user's selected answers with the correct answers and calculate the score.

You should have at least three pages for your quiz application. I suggest creating a landing/home page as the first page, a page dedicated to taking the quiz as the second page, and a third page to display the quiz results.

You must create at least one quiz with at least five questions. At least two questions must be multiple choice with at least three options for the user to choose from.

A header and footer should be used, each in separate files and included in your main pages. Your header must contain HTML for your navigational bar. Your footer can simply contain a short copyright string.

Programming Rules

Your submission must meet the following constraints:

- Strict types must be turned on.
- Each function and method should use type declarations for their parameter(s) and return type definitions (if applicable).
- Every class must have a constructor with appropriate parameters.
- Every property must use a type declaration.

- Client-side scripts are prohibited.
- Inputs via GET and POST must be sanitized.
- Any include files that a user will not directly see must have the extension `.inc`.

Evaluation

You will receive zero points if:

- You violate the academic honesty policy (sanctions can be greater than just a 0 for the assignment)
  - This includes using code from online or having code AI-generated
- Your code is not submitted to the assignment GitHub repository correctly
- You submit something other than a valid commit ID
- **If your name and South Hills username are not in the STUDENT_INFO.txt file**

You will lose points if:

- There are errors in your PHP
- Specifications are not followed
- You break any of the programming rules
- You do not follow the standard variable naming conventions for PHP
- You do not create and use a CSS file for styling

In addition to these evaluation guidelines, there is a rubric which you can find in the Canvas assignment.

To test your code, utilize the PHP Server extension that you should have already installed from a previous assignment. If you've forgotten how to use it, refer back to homework 1.

Submission

The submission for your project is the commit ID you want to be graded. You will submit it via the Canvas assignment.

If you've forgotten which commit ID to submit or how to find it, refer back to homework 1.

Late assignments will not be accepted.