Final Project. CSE1

Activity: Building a CRUD REST API with MySQL, Testing, and XML/JSON Output

In this Final Project for CSE1, you will be creating a CRUD (Create, Read, Update, Delete) REST API for your chosen MySQL database (from our recent activity). The API will allow users to interact with the database and will act as an Interface to any client that understands JSON or XML. You will also set up tests to ensure the functionality of the API, and provide the option to format the API output as XML or JSON.

Instructions:

1. Setup your database: It will be assumed that you've installed a MySQL server with your chosen database in your system. Make sure it contains at least 20 records for testing.

2. Create a new GitHub repository for your project. Make sure to select the option to initialize the repository with a README file. Write down the installation procedures of your project here once you're done. Clone the repository to your local development environment.

3. Create a virtual environment for your project, and ignore it in your .gitignore file. Enter your virtual environment. Do not include the environment when you upload your project to GitHub.

4. Install libraries needed for this project such as (*flask, make_response, jsonify, request MySQL and requirements.txt)*.

5. Code your Flask REST API Web Application.

Grading Breakdown (60 Points Total)

1. **GitHub commit and push (multiple commits having different timestamps): 8 points** Push your local Git repository to the remote repository on GitHub. Search for Git best practices to have a basic idea of how to organize your commits.

2. **CRUD operations (must contain input validations and error handling): 16 points** Return the appropriate HTTP responses based on the success or failure of operations. Search for the correct header values when providing responses for APIs.

3. **CRUD Tests: 8 points** Write test cases to cover all CRUD operations for the API endpoints. Ensure that tests include different scenarios and edge cases.

4. **Formatting options (XML/JSON): 8 points** Modify API endpoints to check for a URI parameter (e.g., format) that specifies the desired output format (XML or JSON).

5. **Implement search functionality: 6 points** Allow users to search for records based on specific criteria.

6. **Add security mechanisms to secure the API endpoints (JWT or similar): 8 points** Search for JWT or similar authentication implementations.

7. **Documentation: 6 points** Update the README file in your GitHub repository to include: Project details, installation instructions, usage examples, API usage, any additional information useful to understand and run your project (See documentation of popular APIs for reference.)

8. **Video explainer**, demo the features in the video while explaining the technical aspects of its code (e.g why did you do it that way), the explainer has no points grade, but is required for this drill, submission without a video explainer means **No Grade**, features which are implemented in the code but without explanation in the video means No points. (You must be visible in the video. Explain your work as technical as possible)

SUBMISSION REQUIREMENTS:

- Public Google drive folder link containing the project (source code, ERD for db, SQL dump etc.)
- Video explainer (MP4 format) link (you can source this anywhere as long as it’s publicly accessible)
- Github URL that points to your project repository
- Provide all of the above links to a Google form provided in our Google classroom post (other forms of submission are prohibited).