Final Exam Drill. CSE1

Activity: Building a REST API with MySQL, Applying Unit Testing, and integrating JSON in data exchange

In this hands-on activity, you will develop a REST API for a chosen system, integrating database operations & unit testing. This project emphasizes practical application of web services, database management, and software testing principles.

Instructions:

1. Visit Databaseanswers.org Mirror to choose a system, clone this repository, add your entries like in the example, request a Pull Request (PR). System titles will be first-come-first-serve.
2. Wait for approval, if denied it means your chosen system has very small scope, or not appropriate for the project. (hint: ERD table with at least 5 interconnected entities)
3. Once PR is accepted, start by creating a document like this to simplify the system, then proceed with succeeding tasks below:

Technical Requirements:

1. Database Setup (6 points)
   - Create ERD for your chosen system
   - Implement database schema
   - Populate with sample data (minimum 25 records per table)
     Use sites like filldb.info, or use faker library
   - Minimum of 4 tables (minimized scope from large System ERD, see sample)
     Max of 10 tables
   - Include proper relationships and constraints
2. API Development (8 points)
   - Implement using Flask framework
   - Create endpoints based on user stories (see example)
   - Follow REST Architecture conventions
   - Include CRUD operations with:
     - Input validation
     - Error handling
     - Appropriate HTTP status codes
     - Support JSON response format
3. Testing (6 points)
   - Write unit tests using Pytest
   - Implement mock database connections
   - Achieve minimum 80% code coverage
   - Include edge case testing

4. Security (3 points)
   - Implement JWT authentication
   - Add request validation
   - Implement role-based access control

5. Documentation (3 points)
   - Create a README.md, see sample README.md

6. Version Control (2 points)
   - Code must be committed to Github/Gitlab

- Use the conventional commits format (see ________)
- Do NOT submit your project as a single bulk commit with no development history (-7 points penalty). Your repository must demonstrate genuine project development through regular commits showing incremental progress, feature additions, and bug fixes. Repositories containing only one initial commit with the complete project will be considered as lacking proper version control practices and will incur the full point deduction

7. Deployment & video (2 points)
   - Deploy to PythonAnywhere.com
   - Ensure all endpoints are accessible
   - Provide API documentation in your readme.md

Failing to complete any of the following results in project not accepted:

- Video presentation (10-15 minutes)
- Working deployment on PythonAnywhere.com
- All core CRUD operations functional
- Minimum 80% test coverage

Minimum Requirements to Pass

1. Score at least 70% of total points
2. All required submissions completed
3. Working deployment
4. Minimum 80% test coverage
5. Core CRUD operations implemented
6. No plagiarism

SUBMISSION REQUIREMENTS:

1. ERD Analysis Document
2. GitHub repository URL of the System
3. PythonAnywhere deployment URL
4. Video presentation (10-15 minutes)
   - Code walkthrough
   - Database design explanation
   - API demonstration
   - Test execution
5. Database dump file