Mini Project 2: Web-based University Database Management

- **Submission Deadline:** March 31 by 11:59 PM
- **Presentation Deadline:** March 31 during class time
- **Note:** If you do not present, you will not receive a grade.

Student Learning Outcomes:

Upon successful completion of this project, students will be able to:

- Install and configure MariaDB/MySQL, Apache, and PHP.
- Understand relational database concepts.
- Write SQL queries for data retrieval from the provided university database schema (from Database System Concepts, Seventh Edition).
- Dynamically display database records in HTML through PHP scripts.
- Establish secure database connections using PHP.

Deliverables:

- **Setup Documentation**: Installation instructions for MariaDB/MySQL, Apache, and PHP.
- **Database Queries**: SQL scripts for data retrieval from the university database.
- **PHP Notebook (Colab)**: Google Colab notebook containing PHP scripts for database connection, query execution, and dynamic HTML generation.
- **Final Report**: Detailed summary of the project's objectives, methods, queries used, challenges encountered, solutions implemented, and overall outcomes.
- **Project Demonstration**: Presentation or live demo illustrating database querying capabilities via a web interface.

Resources:

- There is plenty of documentation and videos available on the web explaining how to carry out this project.

Here, and example: https://www.youtube.com/watch?v=S4fONO-t8G0

- You may use the assistance of an LLM, especially since this class is not focused on HTML or PHP programming. If you do use an LLM, please include the prompts used to generate your code.

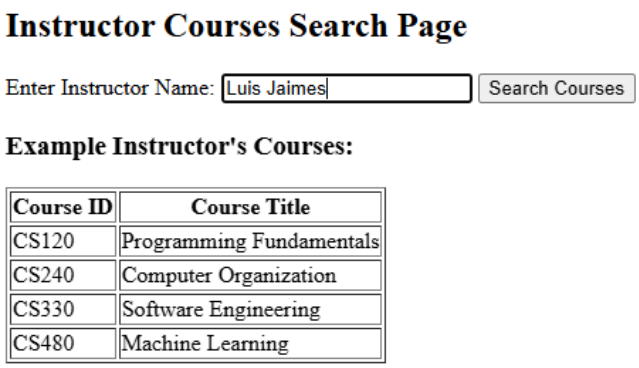
Project Stages:

1. **Environment Setup**:
   - Install and configure MariaDB/MySQL, Apache, and PHP.
2. **Database Exploration**:
   - Study and understand the university database schema.
   - Write and test SQL queries to retrieve relevant data.
3. **Web Interface Development**:
   - Develop PHP scripts to connect securely to the database.
   - Execute SQL queries and display results dynamically in HTML.
4. **Documentation and Reporting**:
   - Prepare detailed documentation of setup instructions, query scripts, PHP code implementations, and a comprehensive project report.
5. **Presentation and Demonstration**:
   - Conduct a presentation showcasing the web interface functionalities and the outcomes of executed queries.

Web Interface Examples:

- **Instructor Courses Search Page:**
  - Input Box: Enter instructor name
  - Button: "Search Courses"
  - Output Table Columns: Course ID | Course Title

Example:



- **Student Enrolled Courses Page:**
  - Input Box: Enter student name
  - Button: "Search Enrollments"
  - Output Table Columns: Course ID | Course Title

Example

Student Enrolled Courses Page

Enter Student Name: Mathew Perez Search Enrollments

Example Enrolled Courses:

Course ID	Course Title
ENG201	English Literature
MATH330	Linear Algebra
CS250	Operating Systems
ECON101	Introduction to Economics

Example

- **Department Course Listings Page:**
  - Input Box: Enter department name
  - Button: "Show Courses"
  - Output Table Columns: Course ID | Course Title

Computer Science Department Course Listings

Enter Department Name: Computer Science Show Courses

Example Computer Science Courses:

Course ID	Course Title
CS101	Introduction to Programming
CS215	Data Structures and Algorithms
CS320	Database Systems
CS410	Artificial Intelligence

Disclaimer: I used ChatGPT to generate the html code of the web interface examples.