Database Management Systems
Project 3: Database Integration on Web

Vikas Thammanna Gowda

April 1, 2025

1 Project Overview

In this project, students will design and implement a lightweight Learning Management System (LMS) to be deployed locally. The LMS will manage course enrollment, course materials, and grade records for an instructor teaching four courses: **DAT-210, DAT-410, CSI-300-01, and CSI-300-02**. Notably, **CSI-300-01 and CSI-300-02** are different sections of the same course, and students must not be allowed to enroll in both sections simultaneously.

1.1 Learning Objectives

1. **Database Schema Design:** Students will create tables and relationships in a relational database (SQLite) for storing student information, course information, and enrollment records.
2. **SQL Query Proficiency:** Students will write SQL statements (CRUD operations) for data insertion, retrieval, updates, and deletions.
3. **Access Control and Constraints:** Students will enforce business rules (e.g., restricting enrollment in both sections of CSI-300) using constraints and application-level validation.
4. **Front-End Integration:** Students will connect a React user interface to the database via a suitable back-end.
5. **Practical System Deployment:** Though local, students will learn the fundamentals of a full-stack app deployment, including setting up and running the **SQLite** database.

1.2 Expected Outcomes

By the end of this project, students should be able to:

- **Design and Implement a Database:** Create normalized tables to store user, course, enrollment, and grading data.
- **Use SQL Effectively:** Manage enrollment, and store grades using SQL CRUD operations.
- **Build a Basic Full-Stack Application:** Merge React (front end), SQLite (database), and a backend framework to create a functional LMS.
- **Enforce Business Rules:** Apply constraints to ensure valid student enrollment (i.e., avoid dual enrollment in CSI-300-01 and CSI-300-02).
- **Provide Documentation:** Deliver a guide on system design, setup instructions, and usage details.

1.3 Technology Stack

- **Front end:** React
- **Database:** SQLite
- **Server side:** Node.js

2 Minimum Project Requirements

- Stores courses (course prefix, course number, class room, start time) information.
- Stores students (first name, last name, email, major, graduating year) information.
- Stores grading (2 quiz, 2 project, final exam) information.
- Upload some course materials for any one course.
- A student can be enrolled in multiple courses, but **cannot** be enrolled in **both** CSI-300-01 **and** CSI-300-02.
- A valid sign-in is **required** for enrollment by the instructor.

3 Submission Details

Intermediate Submission: Due 04/13/2025

1. Group submission (30%):

- MySQL code containing the create table statements and few insert value statements for the **normalized** database.

Final Submission: Due 04/27/2025

1. Group submission (35% + 10%):

- Source code in a compressed format.
- A user guide (.pdf) for setup, enrolling a student, and using instructor features.

2. Individual submission (25%): Answer the following with minimum 600 words (.pdf)

- Your contributions.
- How did the coursework help you with the project?
- What additional knowledge did you need to gain to complete this project? Explain.
- How challenging is it to apply the theoretical knowledge of database design to practical implementation?
- Your experience interacting with AI to assist with the code generation.

Academic Integrity Policy

- You are allowed to use AI to assist you with the coding.
- The use of AI tools (e.g., ChatGPT, Bard, etc.) to generate or assist in writing the user guide and/or individual contributions is strictly prohibited. Any detected use of AI tools for report generation will result in an automatic **F grade** for the project.
- Proper citations and references must be provided for any external sources consulted. Plagiarism will not be tolerated and will be subject to academic penalties.