



COURSE MANAGEMENT SYSTEM

Report Prepared By

Maaz Saeed

Team Members

Ayzah Gardezi 2021656

Maaz Saeed 2021268

Warda Farhan 2021750

Guardians Of Data



SYNOPSIS & MILESTONES

Course Management System is a database project designed to provide a platform for educational purposes. CMS provides various features that will allow students, teachers and admin in an educational institute to manage processes of course administration, course creation, course deletion, course enrollment, attendance and grading.

The system is developed as a web app portable on different devices like mobile, tablet, desktop etc. With a user-friendly interface and an easy to navigate front-end and a robust back-end users can seamlessly interact with the system allowing them to manage their academic records throughout without any inconvenience. The system aims to streamline the academic process and enhance overall experience.

Admins can create and delete courses. Instructors can upload grades including assignments, quizzes and projects. Instructors can also upload attendance records of each student. Students can check their registered courses, view grades and assignments including projects, mids and finals. Students can also view their attendance for each subject they are enrolled in. Moreover the app has the option to view results with detailed breakdown of each semester.

1. Formulation

After brainstorming during several meetings and sessions where we discussed different problems and how a database would be a suitable solution for the problem. Eventually after our experiences with the current CMS of our institute, the need to develop a revamped and a remastered CMS was identified.

2. Problem Analysis

During this phase, we went through different CMS of other institutes to gain inspiration and ideas. Most of the requirements are something we ourselves had in our minds since the new CMS was introduced in our institute. We then further analysed the requirements of students, teachers and admins to make our app a Minimum Viable Product (MVP). Multiple sessions of brainstorming and discussion took place during this stage. We designed multiple prototype databases to ensure maximum requirement space is covered. Finally we formulated the functional requirements needed to make our app a Minimum Viable Product (MVP).

Technology Stack:

Front-End: HTML, CSS, JavaScript, BootStrap, SASS, Visual Studio Code

Back-End: Flask [Python Web Framework] with Flask-SQLAlchemy

Database: Postgres

GitHub Repo [url]: <https://github.com/MaazSaeed/LMS-DBMS>

3. Designing a Storage Solution

During this stage, the storage solution for the CMS was designed. With the use of ERD (Entity-Relationship-Diagram) to map the structure of the tables and data. The ERD conceptualised the relationship between different tables and entities. Furthermore, data constraints were applied to the tables for data integrity.

4. Analysis of the Presented Solution

At this level, normalisation was performed, to make the database ACIDic that is Atomicity, Consistency, Isolation and Durability. During this process larger tables were broken down to smaller ones and the relationships between the tables were reformed. Newer attributes were added and some removed as well further optimising the overall structure of the database and making it more manageable.

5. Implementation of the Solution

During this stage the actual development of the project took place, with the technology stack stated earlier in the second milestone. With the use of multiple programming languages HTML, CSS, JavaScript and Python. The front-end was developed using Bootstrap making it aesthetic and interactive. For the back-end Flask and FlaskSQLAlchemy were used to integrate the database with the front-end to develop the software.

6. Integration

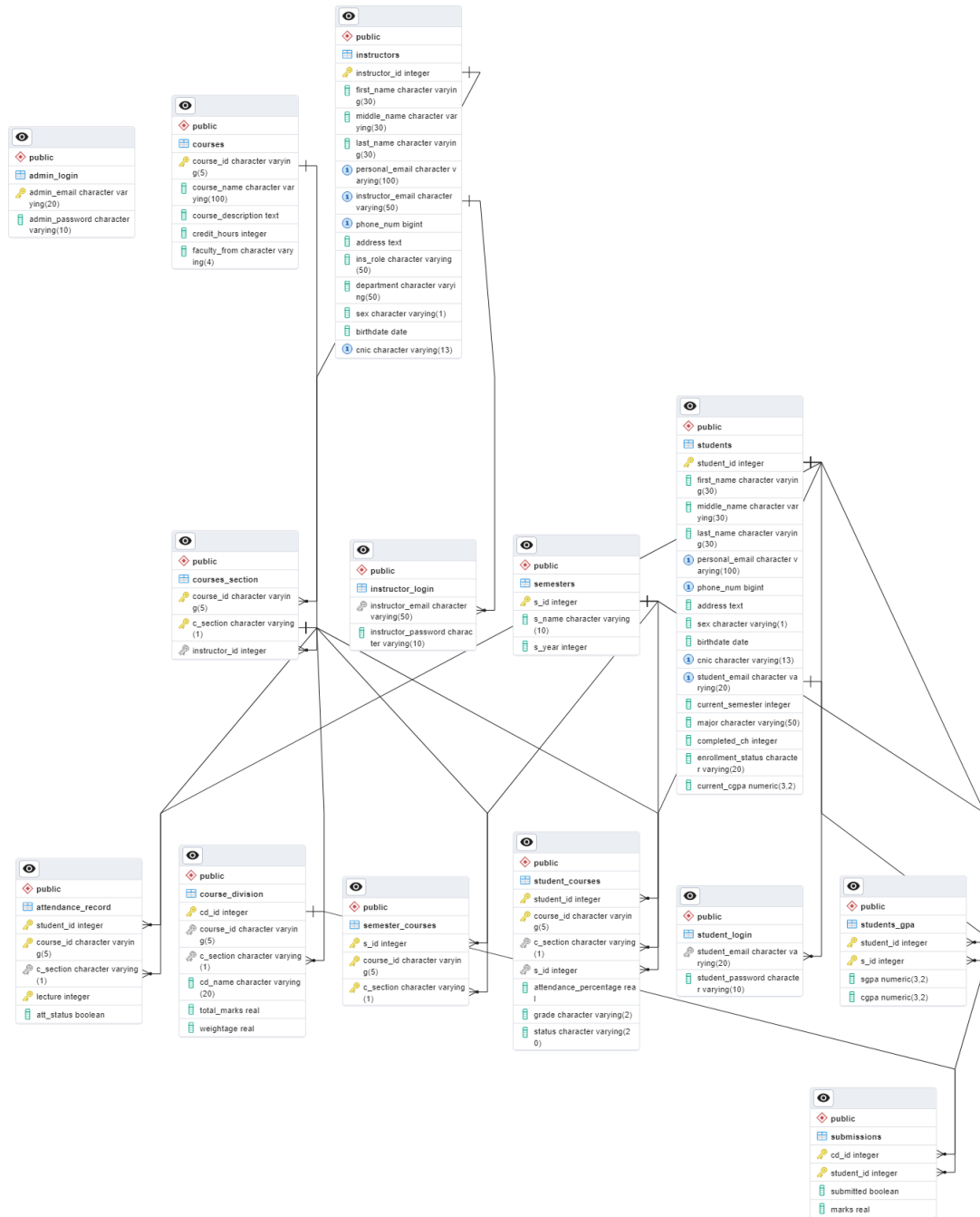
During this final phase, all the pieces of the puzzle were combined. The front-end from where the end users can interact with the application was linked to the powerful back-end which provided the protocol and logic, all of this connected with a database server where data is stored and retrieved from.

Conclusion

The project successfully caters to the need of a CMS in an educational institute for course management including but not limited to grades, attendance, enrollment etc. However there is still room for improvement, further optimizations and features to take the product into a deployable stage. But it addresses all the requirements to make it an MVP.



ERD





WORKLOAD DISTRIBUTION

Ayzah Gardezi & Warda Farhan

Backend and integration using PostgreSQL, and Flask and FlaskSQLAlchemy respectively

Maaz Saeed

Front-End using HTML, CSS, JavaScript and Bootstrap, Dummy Data using Mockaroo and Project Report