DBMS Project: Student Portal

# Group members:

* Tasbiha Nasir (23k -0528)
* Zain Saqib (23k-0588)
* Madiha Aslam (23k -0846)

# Introduction

The current university student portal lacks usability and efficiency in handling academic information. To address these issues, this project aims to design and implement a modern Student Portal System that centralizes and streamlines essential academic services. The system will provide students and administrators with a reliable platform to manage course information, student records, attendance, GPA calculation, and related academic functions.  
  
The project will focus on building a Minimum Viable Product (MVP) that is both scalable and user-friendly, with room for future enhancements.

# Program Objectives

* Develop a centralized portal to access and manage student information.
* Provide course details, enrollment, and tracking functionality.
* Maintain student attendance records digitally.
* Automate GPA calculation and grade management.
* Ensure secure authentication and role-based access (students, faculty, admin).
* Create an easy-to-navigate user interface.
* Ensure data consistency and integrity through a well-designed relational database.

# Goals (MVP Scope)

* Store and retrieve student personal data.
* Manage course details and course registrations.
* Record and display attendance.
* Track grades and GPA calculations.
* Provide a simple dashboard for students and faculty.
* Build 10–15 normalized tables (e.g., Students, Courses, Enrollment, Attendance, Grades, Faculty, Departments, etc.).
* Allow basic CRUD operations (Create, Read, Update, Delete).

# Implementation Overview

## Frontend:

* React.js for building an interactive and responsive user interface.
* Bootstrap / Tailwind for styling components.
* Axios / Fetch API for communication with backend services.

## Backend:

* Python (Flask) for handling business logic and APIs.
* REST API architecture to connect frontend with backend.

## Database:

* Relational Database ( MySQL / MS SQL Server) for storing structured university data.
* Around 10–15 tables covering students, faculty, courses, departments, attendance, grades, etc.

**Security:**

* Authentication & Authorization.
* Role-Based Access Control.

## Additional Considerations (May or May not be used):

* Deployment on local servers
* AI Chat Gpt or Grok for generating queries for mass dummy data in tables