

Student Information System

Project Report

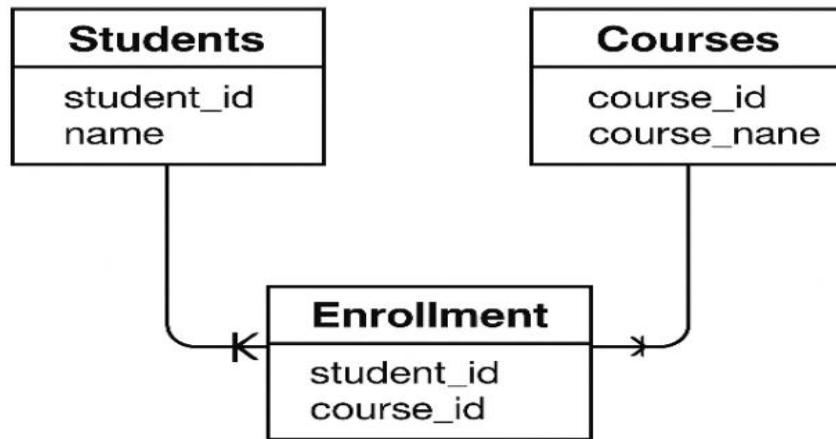
1. Introduction

The Student Information System is designed to manage student records, course listings, and enrollment data. Its purpose is to provide a simple and efficient way to store, retrieve, and update student information using a web-based interface.

2. Database Design

2.1 ER Diagram (Description)

The system consists of three main tables: Students, Courses, and Enrollment. Students can enroll in multiple courses, and each course can have many students, forming a many-to-many relationship represented by the Enrollment table.



2.2 Table Descriptions

- **students** – stores student personal details.
- **courses** – stores course names and descriptions.
- **enrollment** – serves as a junction table linking students and courses.

a. Students Table

Column	Type	Description
student_id	INT (PK)	Unique ID of each student
first_name	VARCHAR (50)	Student's first name
last_name	VARCHAR (50)	Student's birthdate
birthdate	DATE	Student's email address
email	VARCHAR (100)	Student's email address
contact_num	VARCHAR (20)	Student's contact number

b. Courses Table

Column	Type	Description
course_id	INT (PK)	Unique ID of each course
course_name	VARCHAR(100)	Name of the course
description	VARCHAR(255)	Short details about the course

c. Enrollment Table

Column	Type	Description
enrollment_id	INT (PK)	Unique enrollment record
student_id	INT (FK)	Links to a student
course_id	INT (FK)	Links to a course
enroll_date	DATE	When the student enrolled

2.3 Relationships

- One student can enroll in many courses.
- One course can have many enrolled students.
- The enrollment table connects student_id and course_id through foreign keys.

3. Web Interface: Key Pages and Functionalities

The system includes the following functional pages:

- Student List Page – displays all registered students with options to add, edit, and delete records.
- Course List Page – shows all courses offered.
- Add/Edit Forms – allows input and modification of student and course data.
- Enrollment Page – assigns students to courses.
- JavaScript validation and delete confirmation enhance user experience.
- CSS styling ensures a clean, responsive layout.

4. Challenges and Learning

Some challenges faced during the development included connecting the database to the PHP backend, structuring table relationships correctly, and ensuring the user interface remained clean and functional. Through this project, I learned how to build a multi-table relational database, validate form inputs, and create a functional front-end design using HTML, CSS, and JavaScript.