# Student Management System

## Project Description :

This project is a simple command-line Student Management System implemented in Python. It uses the built-in sqlite3 library to manage student data in a local SQLite database file named students.db .

The Student Management System is a console-based application designed to manage student records efficiently. The system allows the user to **add, view, search, update, and delete** student information. It also supports **persistent storage** using SQLite, so all data is saved in a database and can be accessed anytime.

## Project Structure:

student-management-system/

→student_management_sql.py

→ README.md

→students.db (Generated when the script is first run)

## Features:

The system provides the following core functionalities:

- **Add Student (1):** Allows inputting a new student's ID, name, age, course, and marks and storing them in the database.
- **View Students (2):** Fetches and displays all student records stored in the database.
- **Search Student (3):** Allows searching for students by name (supports partial matches).
- **Update Student (4):** Allows updating the name, age, course, or marks of an existing student using their Student ID.
- **Delete Student (5):** Removes a student record from the database using their Student ID.
- **Exit (6):** Closes the connection and terminates the application.

## Prerequisites:

- Python 3.x

The project uses only the Python standard library components, specifically the sqlite3 module.

## How to Run:

1. **Save the code:** Save the provided Python code as student_management_sql.py.
2. **Run the script:** Open your terminal or command prompt, navigate to the project directory, and run the script:

python student_management_sql.py

3. **Use the Menu:** The application will display a menu, and you can enter your choice (1- 6) to interact with the system.

===== Student Management System =====

1. Add Student

2. View Students

3. Search Student

4. Update Student

5. Delete Student

6. Exit

Enter your choice:

## Database Schema:

The script automatically creates the students table in the students.db file upon its first execution.

| Column Name | Data Type | Constraint | Description |
|---|---|---|---|
| **id** | TEXT | PRIMARY KEY | Unique Student Identifier (ID) |
| **name** | TEXT | NOT NULL | Student's Full Name |
| **age** | INTEGER | - | Student's Age |
| **course** | TEXT | - | Course Enrolled In |
| **marks** | REAL | - | Student's Marks (or GPA) |

## Code Highlights:

- **Database Connection:** The connection is established using sqlite3.connect('students.db').
- **Unique ID:** The id column is set as the PRIMARY KEY, ensuring no two students can have the same ID. The add_student function handles sqlite3.IntegrityError to catch duplicate IDs.
- **CRUD Operations:** The script implements all basic Create (Add), Read (View/Search), Update, and Delete operations using standard SQL commands (INSERT, SELECT, UPDATE, DELETE).
- **Parameterized Queries:** All SQL commands use ? placeholders for data values to prevent SQL Injection vulnerabilities.