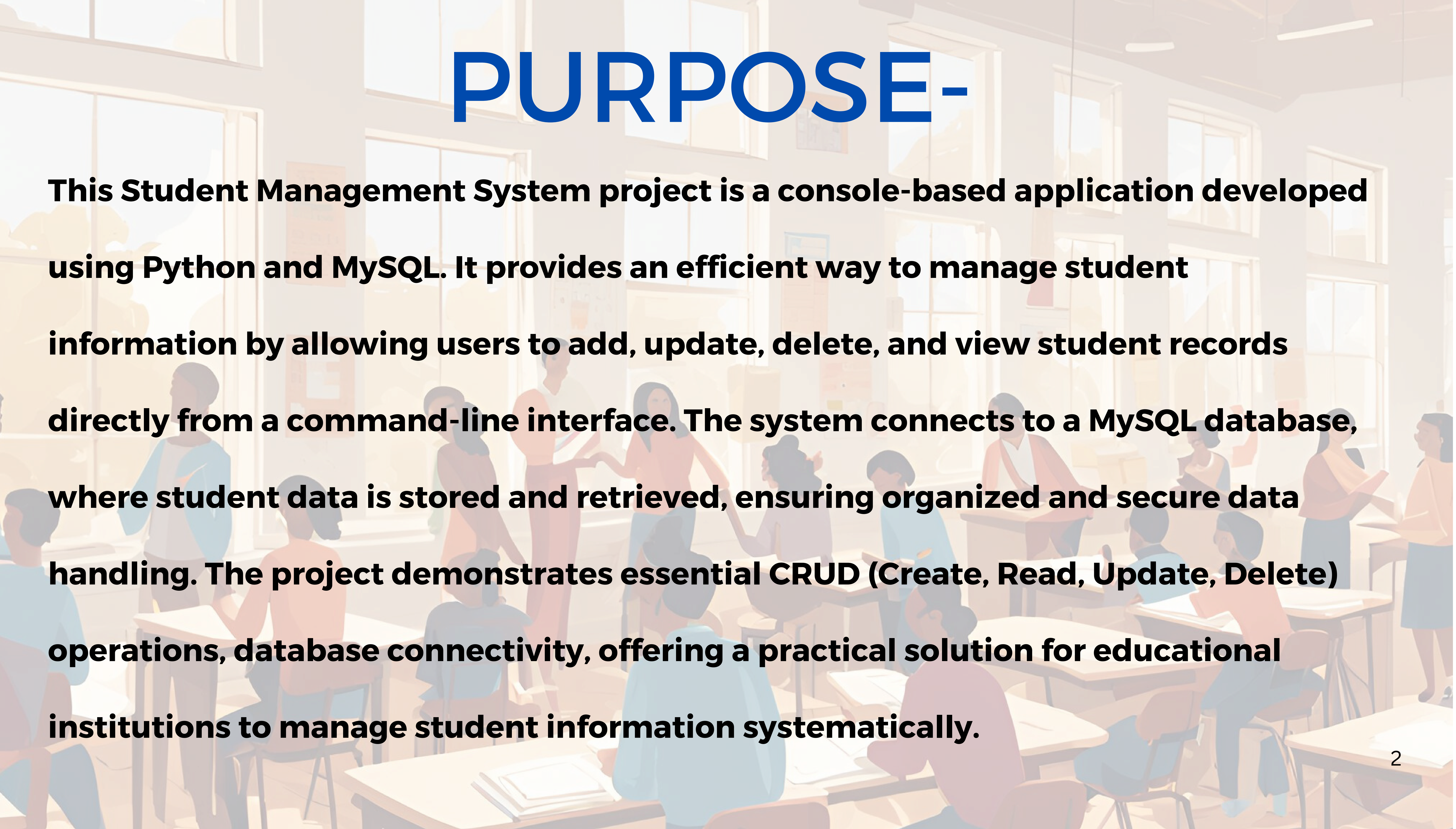
STUDENT MANAGEMENT SYSTEM

-CONSOLE BASED PROJECT



PURPOSE-

This Student Management System project is a console-based application developed using Python and MySQL. It provides an efficient way to manage student information by allowing users to add, update, delete, and view student records directly from a command-line interface. The system connects to a MySQL database, where student data is stored and retrieved, ensuring organized and secure data handling. The project demonstrates essential CRUD (Create, Read, Update, Delete) operations, database connectivity, offering a practical solution for educational institutions to manage student information systematically.

DATABASE-

1. Database: stud_management

This database is designed to store user information for authentication and detailed records for each student.

2.Tables

1. users Table
   - Purpose: Stores user credentials for authentication, allowing secure access to the Student Management System.
   - Fields:
     - username (VARCHAR(100), Primary Key): The unique identifier for each user.
     - password_hash (VARCHAR(64)): Stores the hashed password for secure login.
     - phone_number (VARCHAR(15)): Stores the user's phone number for contact or verification purposes.

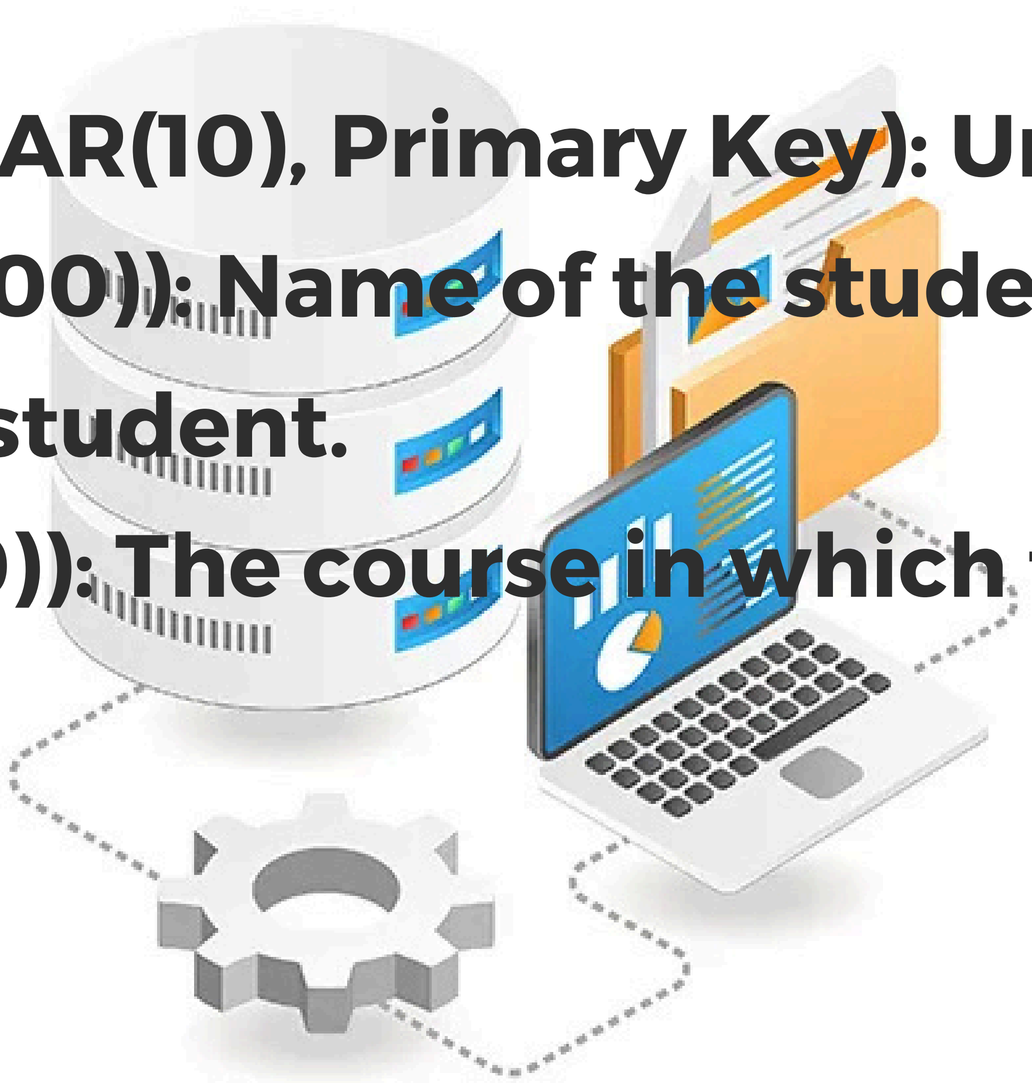
DATABASE-

2. s_students Table

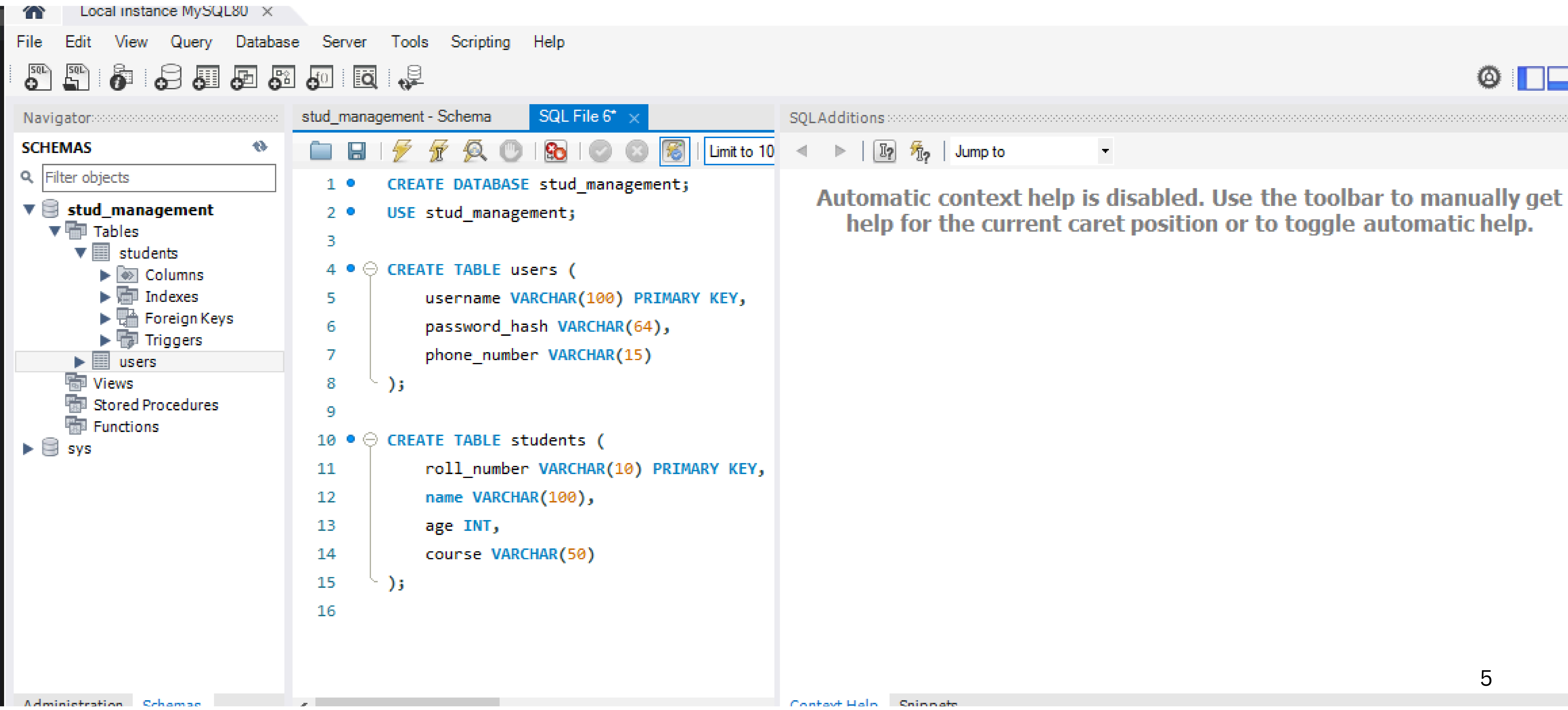
Purpose: Stores detailed records of students, including personal informat enrollment details.

Fields:

1. roll_number (VARCHAR(10), Primary Key): Unique ID assigned to each st
2. s_name (VARCHAR(100)): Name of the student.
3. age (INT): Age of the student.
4. course (VARCHAR(50)): The course in which the student is enrolled



TABLES



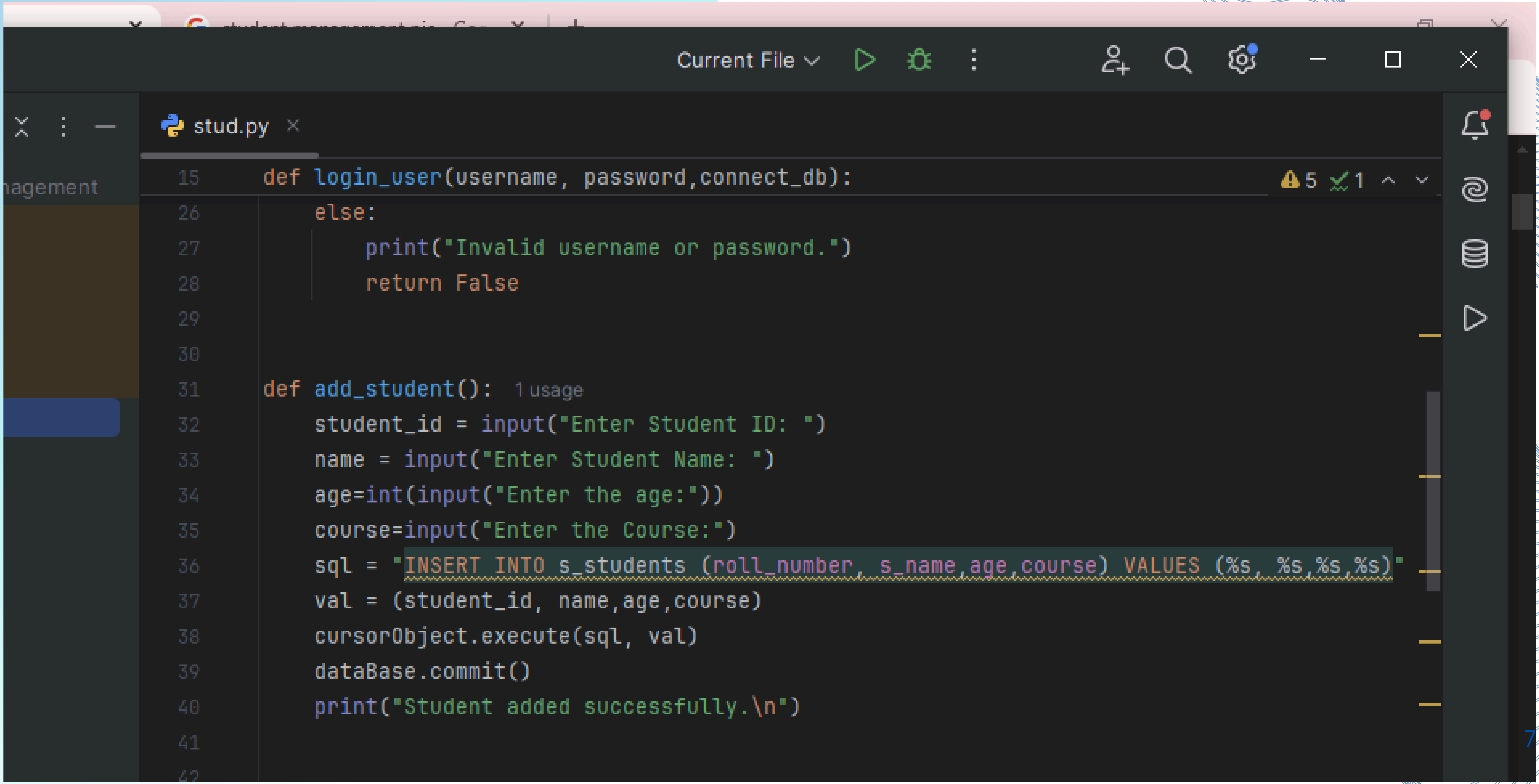
```sql
CREATE DATABASE stud_management;
USE stud_management;

CREATE TABLE users (
    username VARCHAR(100) PRIMARY KEY,
    password_hash VARCHAR(64),
    phone_number VARCHAR(15)
);

CREATE TABLE students (
    roll_number VARCHAR(10) PRIMARY KEY,
    name VARCHAR(100),
    age INT,
    course VARCHAR(50)
);
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

PYTHON CODE: CORE FUNCTIONS

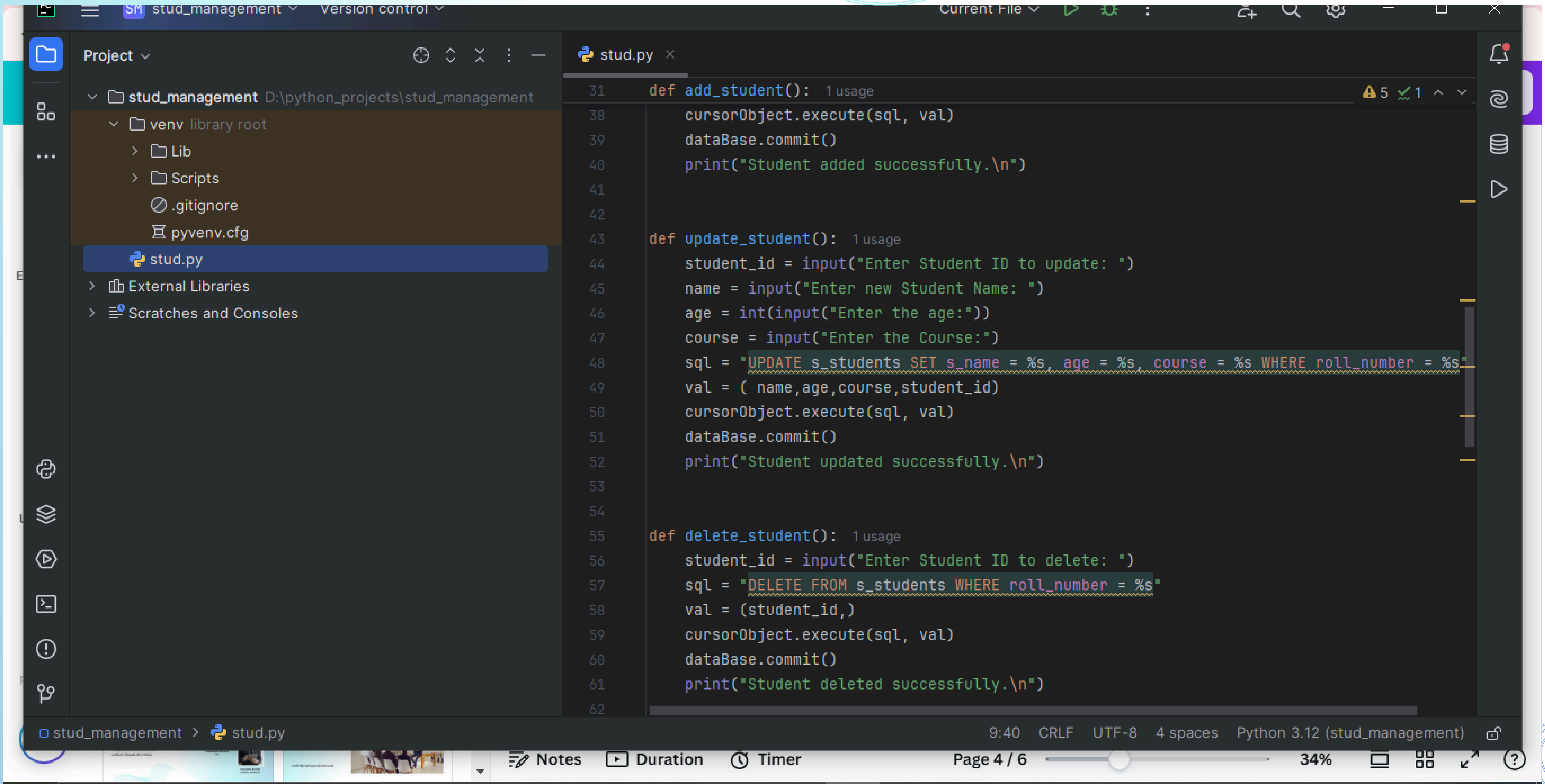
- **ADD STUDENT**: TAKES INPUT AND INSERTS DATA INTO S_STUDENTS.
- **UPDATE STUDENT**: MODIFIES EXISTING STUDENT DATA BY ID.
- **DELETE STUDENT**: DELETES A STUDENT'S RECORD BY ID.
- **VIEW STUDENTS**: RETRIEVES AND DISPLAYS ALL STUDENT RECORDS.

ADD STUDENT FUNCTION



```python
def login_user(username, password,connect_db):
    else:
        print("Invalid username or password.")
        return False


def add_student():  1 usage
    student_id = input("Enter Student ID: ")
    name = input("Enter Student Name: ")
    age=int(input("Enter the age:"))
    course=input("Enter the Course:")
    sql = "INSERT INTO s_students (roll_number, s_name,age,course) VALUES (%s, %s,%s,%s)"
    val = (student_id, name,age,course)
    cursorObject.execute(sql, val)
    dataBase.commit()
    print("Student added successfully.\n")
```

UPDATEFUNCTION-



stud_management
Version control
Current File
Project
stud_management D:\python_projects\stud_management
venv library root
Lib
Scripts
.gitignore
pyvenv.cfg
stud.py
External Libraries
Scratches and Consoles
stud_management > stud.py
9:40 CRLF UTF-8 4 spaces Python 3.12 (stud_management)
Notes Duration Timer Page 4 / 6 34%

```python
def add_student():  1 usage
    cursorObject.execute(sql, val)
    dataBase.commit()
    print("Student added successfully.\n")


def update_student():  1 usage
    student_id = input("Enter Student ID to update: ")
    name = input("Enter new Student Name: ")
    age = int(input("Enter the age:"))
    course = input("Enter the Course:")
    sql = "UPDATE s_students SET s_name = %s, age = %s, course = %s WHERE roll_number = %s"
    val = ( name,age,course,student_id)
    cursorObject.execute(sql, val)
    dataBase.commit()
    print("Student updated successfully.\n")


def delete_student():  1 usage
    student_id = input("Enter Student ID to delete: ")
    sql = "DELETE FROM s_students WHERE roll_number = %s"
    val = (student_id,)
    cursorObject.execute(sql, val)
    dataBase.commit()
    print("Student deleted successfully.\n")
```

CODE OVERVIEW-

This Python script implements a console-based Student Management System with MySQL database integration. It establishes a connection using mysql.connector and authenticates users via a login function that validates username and password from the users table. Once authenticated, the system enables CRUD operations on student records stored in the s_students table. Specifically, it uses parameterized SQL queries in functions for adding, updating, deleting, and retrieving student records, each using SQL commands (INSERT, UPDATE, DELETE, SELECT) and committing changes via dataBase.commit() to ensure ACID compliance. The main function contains an interactive loop presenting a menu, where user input dictates which operation to execute. Finally, the database connection is closed upon program termination to release resources securely.

THANK YOU!

AAYUSHI SHARMA