STUDENT MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted By

SOWMIYA J – 312322205173

SHALOM JUDI PUSHPAM L -312322205158

Of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

St. JOSEPH'S COLLEGE OF ENGINEERING

(An Autonomous Institution)

St. Joseph's Group of Institutions

OMR, Chennai 600 119

ANNA UNIVERSITY: CHENNAI

APRIL-2024

TABLE OF CONTENTS

ABSTRACT

In an era where data management is crucial for educational institutions, the Student Management System emerges as a vital tool for efficient organization and retrieval of student information. Developed using Python and SQL, this system provides a comprehensive platform for managing student details seamlessly.

This project aims to simplify the process of student data management by offering features such as adding, deleting, and searching student records. Through an intuitive user interface, educators and administrators can effortlessly update enrollment information, remove redundant records, and retrieve specific student details in real-time.

By leveraging the power of Python for backend processing and SQL for database management, our Student Management System ensures data accuracy, accessibility, and security. With its user-friendly design and robust functionality, this system enhances productivity and streamlines administrative tasks within educational institutions.

Join us in revolutionizing student data management with our innovative Student Management System, designed to meet the evolving needs of educational institutions in the digital age."

INTRODUCTION

Step into the world of our Student Management System, a technical marvel meticulously crafted with Python and SQL. This project epitomizes the fusion of advanced programming and database technologies to revolutionize student data management in educational institutions.

Engineered for efficiency and precision, our system offers a comprehensive suite of features tailored to streamline administrative workflows. Powered by Python's dynamic backend processing and SQL's robust database management, we've constructed a platform capable of handling complex data operations seamlessly.

Within this system, users will find a user-friendly interface facilitating tasks such as adding, deleting, and searching student details with unparalleled ease. Python's flexibility enables intricate backend processes to ensure optimal data handling, while SQL databases serve as reliable repositories for student records.

PROGRAM

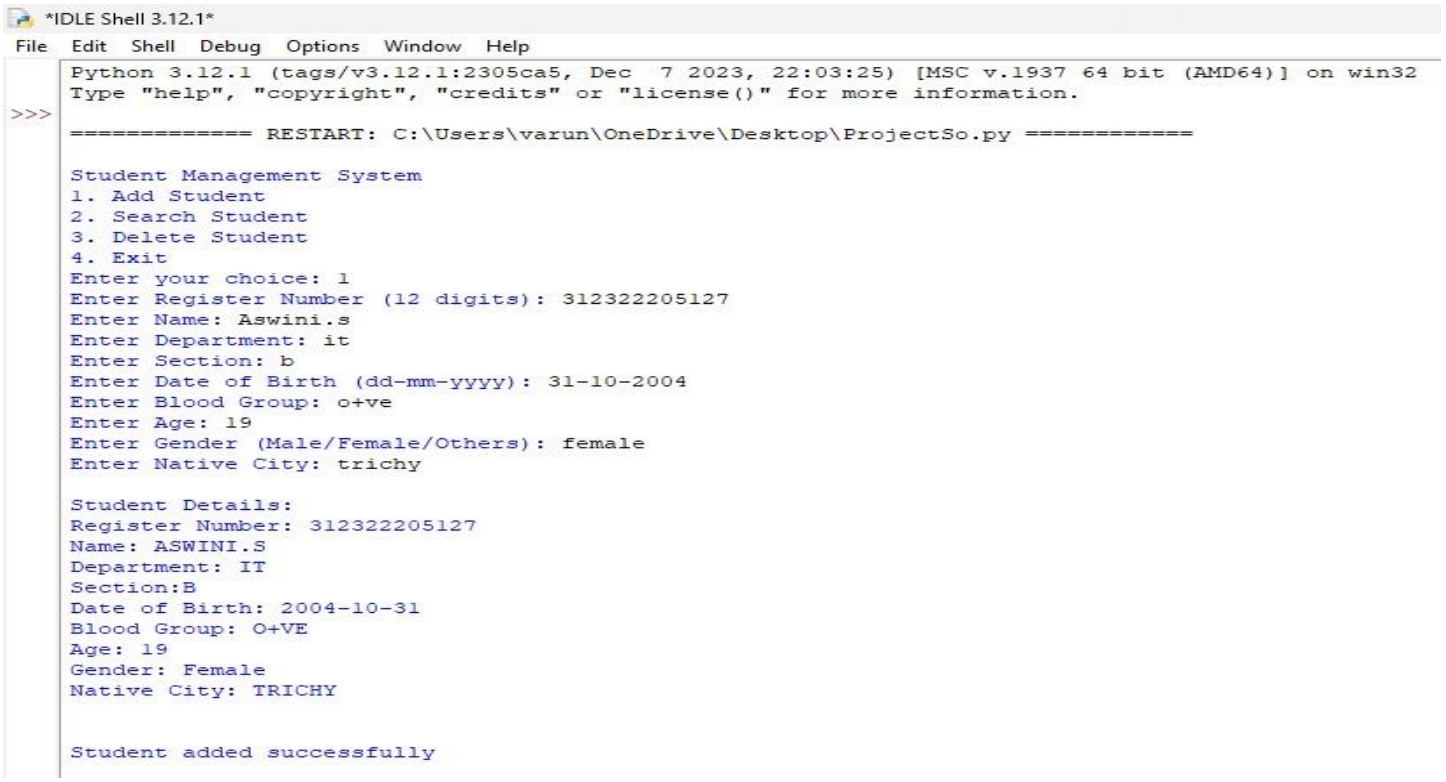
```
import mysql.connector
from datetime import datetime
from time import sleep

def tprint(string):
    for i in range (len(string)):
        print(string[i],end='')
        sleep(0.003)
    print()

# Connect to MySQL database
def connect_to_database():
    try:
        connection = mysql.connector.connect(
            host="localhost",
            user="varun",
            password="admin",
        )
        return connection
    except mysql.connector.Error as error:
        print("Error connecting to MySQL database:", error)
        return None

# Create student table if not exists
def create_student_table(cursor):
    cursor.execute("""CREATE TABLE IF NOT EXISTS students (
                        register_number BIGINT PRIMARY KEY,
                        name VARCHAR(255),
                        department VARCHAR(50),
                        section VARCHAR(10),
                        dob DATE,
                        blood_group VARCHAR(5),
                        age INT,
                        gender ENUM('Male', 'Female', 'Others'),
                        native_city VARCHAR(100)
                    )""")

# Add a new student
def add_student(cursor, connection):
    flag = 0
    register_number = int(input("Enter Register Number (12 digits): "))
    name = input("Enter Name: ").upper()
    department = input("Enter Department: ").upper()
    section = input("Enter Section: ").upper()
    dob_str = input("Enter Date of Birth (dd-mm-yyyy): ")

    try:
        # Parse the date string to a datetime object
        dob = datetime.strptime(dob_str, "%d-%m-%Y").date()
    except ValueError:
```

```
        print("Invalid date format. Please enter date in the format dd-mm-
yyyy.")
        return
    blood_group = input("Enter Blood Group: ").upper()
    age = int(input("Enter Age: "))
    gender = input("Enter Gender (Male/Female/Others): ").upper()
    native_city = input("Enter Native City: ").upper()

    # Insert into database
    try:
        cursor.execute("""INSERT INTO students
                       (register_number, name, department, section, dob,
blood_group, age, gender, native_city)
                       VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)""",
                    (register_number, name, department, section, dob,
blood_group, age, gender, native_city))
        connection.commit()
    except:
        print("\nStudent",register_number,"is already exists")
        flag = 1
    cursor.execute("SELECT * FROM students WHERE register_number = %s",
(register_number,))
    student = cursor.fetchone()
    if student:
        tprint("\nStudent Details:")
        tprint("Register Number: "+ str(student[0]))
        tprint("Name: " +student[1])
        tprint("Department: "+ student[2])
        tprint("Section:" + student[3])
        tprint("Date of Birth: "+ str(student[4]))
        tprint("Blood Group: "+ student[5])
        tprint("Age: "+ str(student[6]))
        tprint("Gender: "+ student[7])
        tprint("Native City: "+ student[8])
        print()
    if flag==0:
        print("\nStudent added successfully\n")

# Search for a student by register number
def search_student(cursor):
    register_number = int(input("Enter Register Number to search: "))
    cursor.execute("SELECT * FROM students WHERE register_number = %s",
(register_number,))
    student = cursor.fetchone()
    if student:
        tprint("\nStudent Details:")
        tprint("Register Number: "+ str(student[0]))
        tprint("Name: " +student[1])
        tprint("Department: "+ student[2])
        tprint("Section:" + student[3])
        tprint("Date of Birth: "+ str(student[4]))
        tprint("Blood Group: "+ student[5])
        tprint("Age: "+ str(student[6]))
        tprint("Gender: "+ student[7])
        tprint("Native City: "+ student[8])
        print()
    else:
```

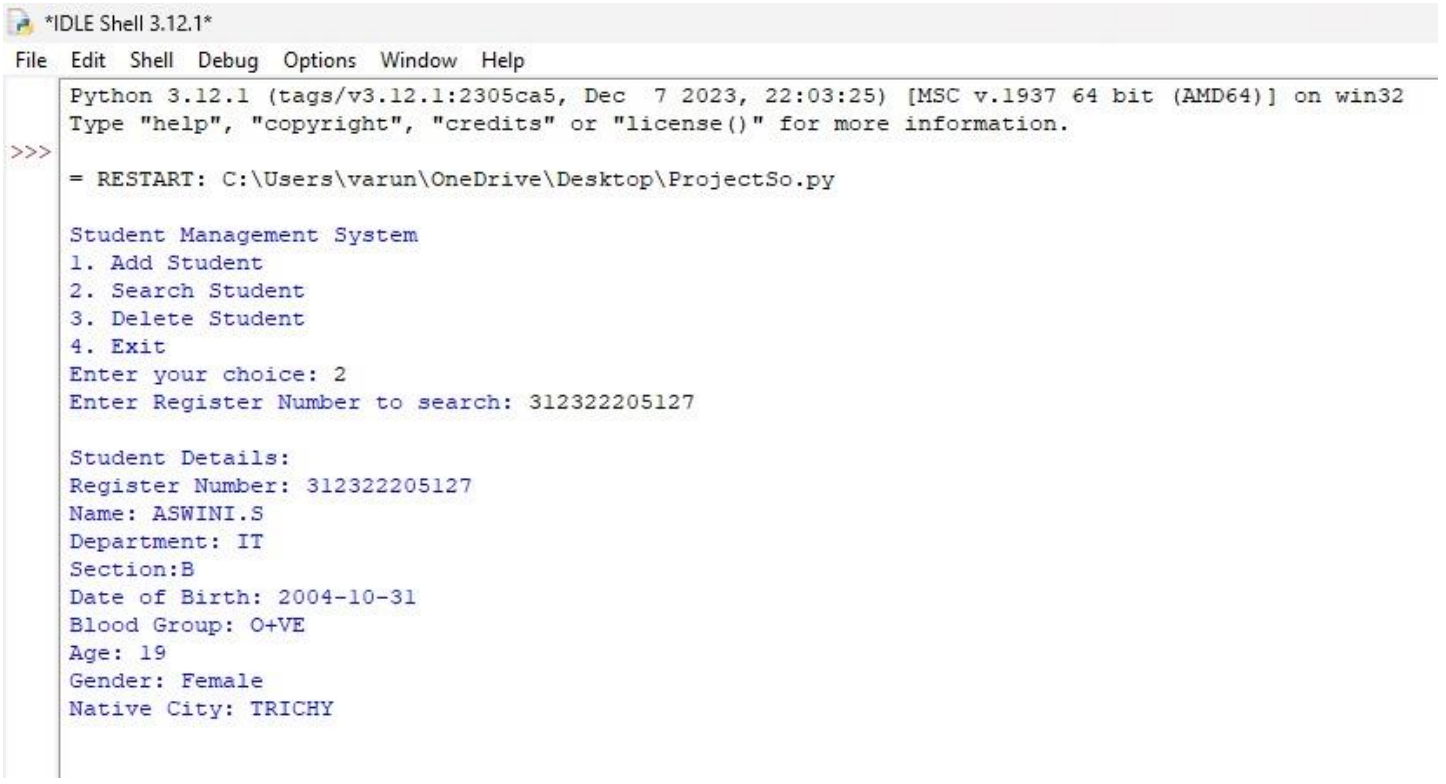
```
        print("Student",register_number,"not found")
        print()

# Delete a student by register number
def delete_student(cursor, connection):
    register_number = int(input("Enter Register Number to delete: "))
    cursor.execute("DELETE FROM students WHERE register_number = %s",
(register_number,))
    connection.commit()
    print("\nStudent",register_number,"deleted successfully\n")

def main():
    connection = connect_to_database()
    if not connection:
        return

    cursor = connection.cursor()
    cursor.execute("create database IF NOT EXISTS student_management;")
    cursor.execute("use student_management;")

    # Create student table if not exists
    create_student_table(cursor)

    while True:
        tprint("\nStudent Management System")
        tprint("1. Add Student")
        tprint("2. Search Student")
        tprint("3. Delete Student")
        tprint("4. Exit")
        choice = input("Enter your choice: ")

        if choice == '1':
            add_student(cursor, connection)
        elif choice == '2':
            search_student(cursor)
        elif choice == '3':
            delete_student(cursor, connection)
        elif choice == '4':
            break
        else:
            tprint("Invalid choice. Please enter again.")

    cursor.close()
    connection.close()
    tprint("CLOSED")

if __name__ == "__main__":
    main()
```

RESULT

i) ADDS STUDENT DETAILS



ii) SEARCH STUDENT DETAILS



iii) DELETE STUDENT DETAILS

```
*IDLE Shell 3.12.1*
File  Edit  Shell  Debug  Options  Window  Help
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\varun\OneDrive\Desktop\ProjectSo.py ==========

Student Management System
1. Add Student
2. Search Student
3. Delete Student
4. Exit
Enter your choice: 1
Enter Register Number (12 digits): 312322205127
Enter Name: Aswini.s
Enter Department: it
Enter Section: b
Enter Date of Birth (dd-mm-yyyy): 31-10-2004
Enter Blood Group: o+ve
Enter Age: 19
Enter Gender (Male/Female/Others): female
Enter Native City: trichy

Student Details:
Register Number: 312322205127
Name: ASWINI.S
Department: IT
Section:B
Date of Birth: 2004-10-31
Blood Group: O+VE
Age: 19
Gender: Female
Native City: TRICHY


Student added successfully
```

CONCLUSION

"In conclusion, our Student Management System stands as a testament to the power of technical innovation in educational data management. By harnessing the capabilities of Python and SQL, we've created a platform that not only streamlines administrative tasks but also enhances data accuracy and accessibility.

As we reflect on our journey, it's evident that the synergy between advanced programming techniques and robust database management has paved the way for a more efficient and effective system. Moving forward, we remain committed to pushing the boundaries of technological advancement in the realm of educational administration.

With our Student Management System, we strive to empower educators and administrators with the tools they need to thrive in an increasingly digital landscape.

THANK YOU