STUDENT MANAGEMENT SYSTEM

PYTHON CODE:

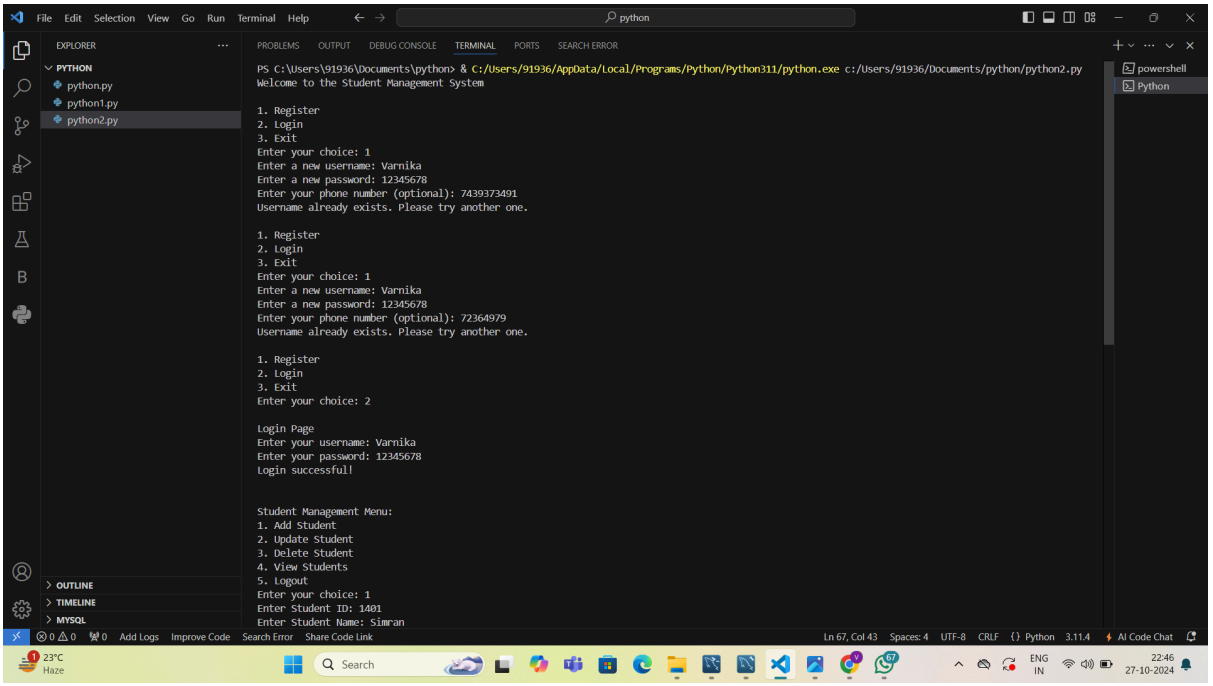
```
import mysql.connector
import bcrypt

# Database connection
dataBase = mysql.connector.connect(
    host="localhost",
    user="root2",
    passwd="amrit76688",
    database="stud_management",
    auth_plugin='mysql_native_password'
)

# Creating a cursor object
cursorObject = dataBase.cursor()

# Function to register a new user
def register_user():
    username = input("Enter a new username: ")
    password = input("Enter a new password: ")
    phone_number = input("Enter your phone number (optional): ")

    # Hash the password before storing it
    hashed_password = bcrypt.hashpw(password.encode('utf-8'),
bcrypt.gensalt())

    try:
        sql = "INSERT INTO users (username, password_hash,
phone_number) VALUES (%s, %s, %s)"
        val = (username, hashed_password, phone_number)
        cursorObject.execute(sql, val)
        dataBase.commit()
        print("User registered successfully!\n")
    except mysql.connector.IntegrityError:
        print("Username already exists. Please try another one.")

# Function to authenticate a user
def login_user():
```

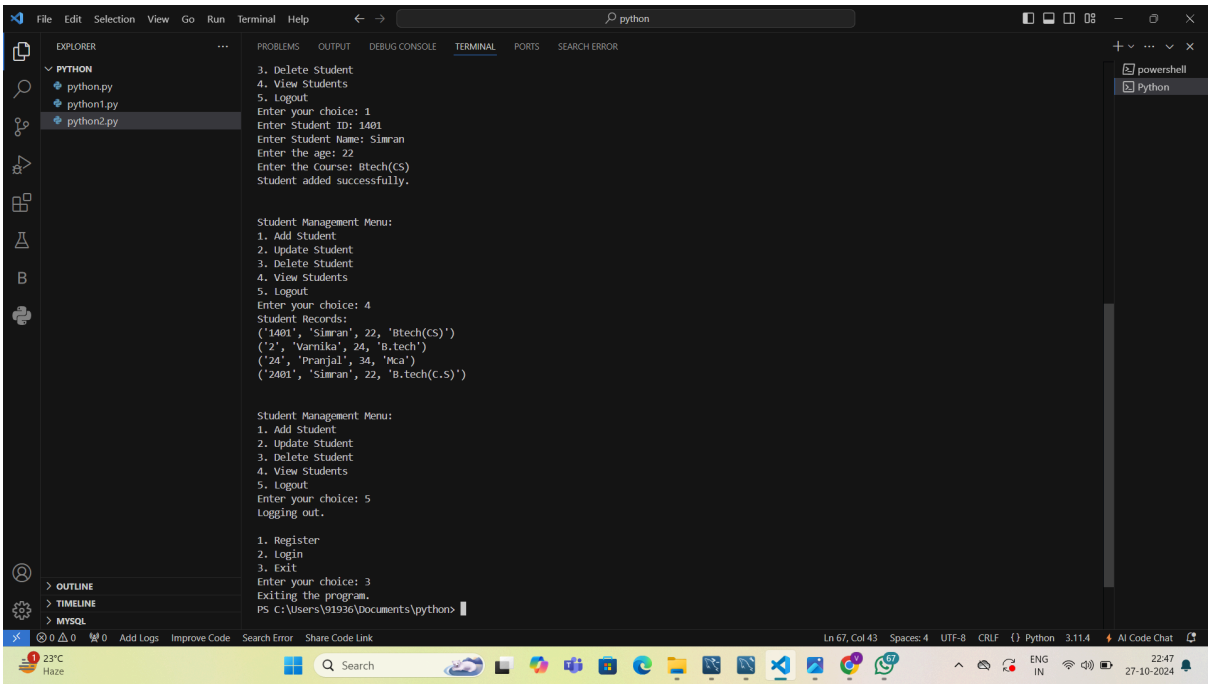
```
    print("\nLogin Page")
    username = input("Enter your username: ")
    password = input("Enter your password: ")

    cursorObject.execute("SELECT password_hash FROM users WHERE
username = %s", (username,))
    result = cursorObject.fetchone()

    if result and bcrypt.checkpw(password.encode('utf-8'),
result[0].encode('utf-8')):
        print("Login successful!\n")
        return True
    else:
        print("Invalid username or password.")
        return False

# Function to add a student
def add_student():
    student_id = input("Enter Student ID: ")
     # Check if the student ID already exists
    cursorObject.execute("SELECT * FROM s_students WHERE roll_number =
%s", (student_id,))
    existing_student = cursorObject.fetchone()

    if existing_student:
        print(f"Student ID {student_id} already exists. Please use a
different ID.")
        return  # Exit the function if the ID already exists
    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")

# Function to update a student
def update_student():
    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")

# Function to delete a student
def delete_student():
    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")

# Function to view all students
def view_students():
    query = "SELECT * FROM s_students"
    cursorObject.execute(query)
    myresult = cursorObject.fetchall()
    print("Student Records:")
    for x in myresult:
        print(x)
    print()

# Main function with login and registration options
def main():
    print("Welcome to the Student Management System")

    while True:
        print("\n1. Register")
        print("2. Login")
        print("3. Exit")

        choice = input("Enter your choice: ")

        if choice == '1':
            register_user()
        elif choice == '2':
            # Display the login page and authenticate the user
            if login_user():
```

```
                # Main CRUD operations menu after successful login
                while True:
                    print("\nStudent Management Menu:")
                    print("1. Add Student")
                    print("2. Update Student")
                    print("3. Delete Student")
                    print("4. View Students")
                    print("5. Logout")

                    sub_choice = input("Enter your choice: ")

                    if sub_choice == '1':
                        add_student()  # This calls the add_student
function
                    elif sub_choice == '2':
                        update_student()
                    elif sub_choice == '3':
                        delete_student()
                    elif sub_choice == '4':
                        view_students()
                    elif sub_choice == '5':
                        print("Logging out.")
                        break
                    else:
                        print("Invalid choice. Please try again.")
            else:
                print("Access denied.")

        elif choice == '3':
            print("Exiting the program.")
            break
        else:
            print("Invalid choice. Please try again.")

    # Closing database connection
    dataBase.close()

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

OUTPUT :


PS C:\Users\91936\Documents\python> & C:/Users/91936/AppData/Local/Programs/Python/Python311/python.exe c:/Users/91936/Documents/python/python2.py
Welcome to the Student Management System
1. Register
2. Login
3. Exit
Enter your choice: 1
Enter a new username: Varnika
Enter a new password: 12345678
Enter your phone number (optional): 7439373491
Username already exists. Please try another one.
1. Register
2. Login
3. Exit
Enter your choice: 1
Enter a new username: Varnika
Enter a new password: 12345678
Enter your phone number (optional): 72364979
Username already exists. Please try another one.
1. Register
2. Login
3. Exit
Enter your choice: 2
Login Page
Enter your username: Varnika
Enter your password: 12345678
Login successful!
Student Management Menu:
1. Add Student
2. Update Student
3. Delete Student
4. View Students
5. Logout
Enter your choice: 1
Enter Student ID: 1401
Enter Student Name: Simran


3. Delete Student
4. View Students
5. Logout
Enter your choice: 1
Enter Student ID: 1401
Enter Student Name: Simran
Enter the age: 22
Enter the Course: Btech(CS)
Student added successfully.
Student Management Menu:
1. Add Student
2. Update Student
3. Delete Student
4. View Students
5. Logout
Enter your choice: 4
Student Records:
('1401', 'Simran', 22, 'Btech(CS)')
('2', 'Varnika', 24, 'B.tech')
('24', 'Pranjal', 34, 'Mca')
('2401', 'Simran', 22, 'B.tech(C.S)')
Student Management Menu:
1. Add Student
2. Update Student
3. Delete Student
4. View Students
5. Logout
Enter your choice: 5
Logging out.
1. Register
2. Login
3. Exit
Enter your choice: 3
Exiting the program.
PS C:\Users\91936\Documents\python>

DATABASE :

```
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 s_students (
    roll_number VARCHAR(10) PRIMARY KEY,
    s_name VARCHAR(100),
    age INT,
    course VARCHAR(50)
    );
```

OUTPUT: