

Student Management System

Student_management_system.py file

```
import mysql.connector
import hashlib
import csv

# Connect to the MySQL database
def connect_db():
    return mysql.connector.connect(
        host="localhost",
        user="root",
        password="G#ayatri#09",
        database="student_management",
        port=3307
    )

# Register a new user
def register_user(email, username, password, phone_no, role):
    db = connect_db()
    cursor = db.cursor()
    hashed_password = hashlib.sha256(password.encode()).hexdigest()
    sql = "INSERT INTO users (email, username, password, phone_no, role)
VALUES (%s, %s, %s, %s, %s)"
    val = (email, username, hashed_password, phone_no, role)
    try:
        cursor.execute(sql, val)
        db.commit()
        print("{} '{}' registered successfully!".format(role.capitalize(),
username))
    except mysql.connector.errors.IntegrityError:
        print("Username or email already exists.")
    cursor.close()
    db.close()

# Login user
def login_user(username, password):
    db = connect_db()
    cursor = db.cursor()
    hashed_password = hashlib.sha256(password.encode()).hexdigest()
    sql = "SELECT role FROM users WHERE username = %s AND password = %s"
```

```

val = (username, hashed_password)
cursor.execute(sql, val)
result = cursor.fetchone()
cursor.close()
db.close()
if result:
    print("Login successful! Logged in as
{}".format(result[0].capitalize()))
    return result[0]
else:
    print("Invalid username or password.")
    return None

# Add a student record
def add_student(name, age, contact, gender, email, dob, address):
    db = connect_db()
    cursor = db.cursor()
    sql = "INSERT INTO students (name, age, contact, gender, email, dob,
address) VALUES (%s, %s, %s, %s, %s, %s, %s)"
    val = (name, age, contact, gender, email, dob, address)
    cursor.execute(sql, val)
    db.commit()
    print("Student {} added successfully!".format(name))
    cursor.close()
    db.close()

# Display all student records
def display_students():
    db = connect_db()
    cursor = db.cursor()
    sql = "SELECT * FROM students"
    cursor.execute(sql)
    results = cursor.fetchall()
    if len(results) == 0:
        print("No students data found in the database.")
    else:
        print("Roll No | Name | Age | Contact | Gender | Email | DOB |
Address")
        print("-----")
        for row in results:
            print(f"{row[0]} | {row[1]} | {row[2]} | {row[3]} | {row[4]} |
{row[5]} | {row[6]} | {row[7]}")

```

```

        cursor.close()
        db.close()

# Search for a student by roll number
def search_student(roll_no):
    db = connect_db()
    cursor = db.cursor()
    sql = "SELECT * FROM students WHERE roll_no = %s"
    val = (roll_no,)
    cursor.execute(sql, val)
    result = cursor.fetchone()
    if result:
        print("Roll No | Name | Age | Contact | Gender | Email | DOB | Address")
        print("-----")
        print(f"{result[0]} | {result[1]} | {result[2]} | {result[3]} | {result[4]} | {result[5]} | {result[6]} | {result[7]}")
    else:
        print("Student not found.")
    cursor.close()
    db.close()

# Update a student record
def update_student(roll_no, name=None, age=None, contact=None, gender=None, email=None, dob=None, address=None):
    db = connect_db()
    cursor = db.cursor()
    updates = []
    params = []

    if name:
        updates.append("name = %s")
        params.append(name)
    if age:
        updates.append("age = %s")
        params.append(age)
    if contact:
        updates.append("contact = %s")
        params.append(contact)
    if gender:
        updates.append("gender = %s")
        params.append(gender)

```

```

if email:
    updates.append("email = %s")
    params.append(email)
if dob:
    updates.append("dob = %s")
    params.append(dob)
if address:
    updates.append("address = %s")
    params.append(address)

sql = f"UPDATE students SET {'', ' '.join(updates)} WHERE roll_no = %s"
params.append(roll_no)
cursor.execute(sql, params)
db.commit()
print(f"Student Roll No {roll_no} updated successfully!")
cursor.close()
db.close()

# Delete a student record
def delete_student(roll_no):
    db = connect_db()
    cursor = db.cursor()
    cursor.execute("DELETE FROM students WHERE roll_no = %s", (roll_no,))
    db.commit()
    print(f"Student Roll No {roll_no} deleted successfully!")
    cursor.close()
    db.close()

# Export all student data to CSV
def export_data():
    db = connect_db()
    cursor = db.cursor()
    sql="SELECT * FROM students"
    cursor.execute(sql)
    results = cursor.fetchall()
    with open("students_data.csv", "w", newline="") as file:
        writer = csv.writer(file)
        writer.writerow(["Roll No", "Name", "Age", "Contact", "Gender",
"Email", "DOB", "Address"])
        writer.writerows(results)
    print("Data exported to students_data.csv successfully!")
    cursor.close()
    db.close()

```

```

def reset_password(email, username):
    db = connect_db()
    cursor = db.cursor()
    sql = "SELECT email FROM users WHERE email = %s AND username = %s"
    val=(email, username)
    cursor.execute(sql, val)
    result = cursor.fetchone()

    if result:
        new_password = input("Enter your new password: ")
        confirm_password = input("Confirm your new password: ")

        if new_password == confirm_password:
            hashed_password =
hashlib.sha256(new_password.encode()).hexdigest()
            sql = "UPDATE users SET password = %s WHERE email = %s AND
username = %s"
            val = (hashed_password, email, username)
            cursor.execute(sql, val)
            db.commit()
            print("Password reset successful!")
        else:
            print("Passwords do not match.")
    else:
        print("Email and username do not match our records.")

    cursor.close()
    db.close()

# Admin menu
def admin_menu():
    while True:
        print("\nAdmin Menu")
        print("1. Add Student")
        print("2. Display All Students")
        print("3. Search Student by Roll No")
        print("4. Update Student by Roll No")
        print("5. Delete Student by Roll No")
        print("6. Export Data to CSV")
        print("7. Logout")

        choice = input("Enter your choice: ")

```

```

        if choice == '1':
            name = input("Enter student name: ")
            age = int(input("Enter student age: "))
            contact = input("Enter contact number: ")
            gender = input("Enter gender (M/F/Other): ")
            email = input("Enter email: ")
            dob = input("Enter date of birth (YYYY-MM-DD): ")
            address = input("Enter address: ")
            add_student(name, age, contact, gender, email, dob, address)
        elif choice == '2':
            display_students()
        elif choice == '3':
            roll_no = int(input("Enter roll number to search: "))
            search_student(roll_no)
        elif choice == '4':
            roll_no = int(input("Enter roll number to update: "))
            name = input("Enter new name (leave blank to skip): ")
            age = input("Enter new age (leave blank to skip): ")
            contact = input("Enter new contact (leave blank to skip): ")
            gender = input("Enter new gender (leave blank to skip): ")
            email = input("Enter new email (leave blank to skip): ")
            dob = input("Enter new date of birth (leave blank to skip): ")
            address = input("Enter new address (leave blank to skip): ")
            update_student(roll_no, name or None, int(age) if age else None,
contact or None, gender or None, email or None, dob or None, address or None)
        elif choice == '5':
            roll_no = int(input("Enter roll number to delete: "))
            delete_student(roll_no)
        elif choice == '6':
            export_data()
        elif choice == '7':
            break
        else:
            print("Invalid choice. Please try again.")

# Faculty menu
def faculty_menu():
    while True:
        print("\nFaculty Menu")
        print("1. Display All Students")
        print("2. Search Student")
        print("3. Export Data")
        print("4. Logout")

```

```

        choice = input("Enter your choice: ")
        if choice == '1':
            display_students()
        elif choice == '2':
            roll_no = int(input("Enter roll number to search: "))
            search_student(roll_no)
        elif choice == '3':
            export_data()
        elif choice == '4':
            break
        else:
            print("Invalid choice. Please try again.")

# Student menu
def student_menu():
    while True:
        print("\nStudent Menu")
        print("1. View My Record")
        print("2. Logout")

        choice = input("Enter your choice: ")
        if choice == '1':
            roll_no = int(input("Enter roll number to view your record: "))
            search_student(roll_no)
        elif choice == '2':
            break
        else:
            print("Invalid choice. Please try again.")

def main():
    while True:
        print("\n
Management System                                Student
")
        print("1. Register")
        print("2. Login")
        print("3. Reset Password")
        print("4. Exit")

        choice = input("Enter your choice: ")

        if choice == '1':

```

```
        email = input("Enter email: ")
        username = input("Enter username: ")
        password = input("Enter password: ")
        phone_no = input("Enter phone number: ")
        role = input("Enter role (admin, faculty, student): ").lower()
        if role in ['admin', 'faculty', 'student']:
            register_user(email, username, password, phone_no, role)
        else:
            print("Invalid role.")

    elif choice == '2':
        username = input("Enter username: ")
        password = input("Enter password: ")
        role = login_user(username, password)
        if role == 'admin':
            admin_menu()
        elif role == 'faculty':
            faculty_menu()
        elif role == 'student':
            student_menu()

    elif choice == '3':
        email = input("Enter your registered email: ")
        username = input("Enter your username: ")
        reset_password(email, username)

    elif choice == '4':
        break

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


The screenshot shows the MySQL Workbench interface. The top toolbar includes icons for File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The 'Query' tab is active, displaying a SQL script for creating a database and two tables. The 'Navigator' pane on the left shows the 'student_management' database selected, with a tree view of its objects: Tables (students, users), Views, Stored Procedures, and Functions. The 'Administration' pane at the bottom left shows the 'Schemas' tab with 'student_management' selected. The 'Output' pane at the bottom right shows the 'Action Output' tab with a table of execution results.

SQL Script:

```

1 • CREATE DATABASE student_management;
2 • USE student_management;
3 • CREATE TABLE users (
4     id INT AUTO_INCREMENT PRIMARY KEY,
5     email VARCHAR(100) NOT NULL,
6     username VARCHAR(50) UNIQUE NOT NULL,
7     password VARCHAR(255) NOT NULL,
8     phone_no VARCHAR(15),
9     role ENUM('admin', 'faculty', 'student') NOT NULL
10 );
11 • CREATE TABLE students (
12     roll_no INT AUTO_INCREMENT PRIMARY KEY,
13     name VARCHAR(100) NOT NULL,
14     age INT NOT NULL,
15     contact VARCHAR(15),
16     gender ENUM('M', 'F', 'Other') NOT NULL,
17     email VARCHAR(100) UNIQUE,
18     dob DATE NOT NULL,
19     address TEXT
20 );

```

Action Output:

#	Time	Action	Message
✓ 1	08:25:25	CREATE DATABASE student_management	1 row(s) affected
✓ 2	08:25:29	USE student_management	0 row(s) affected
✓ 3	08:25:37	CREATE TABLE users (id INT AUTO_INCREMENT PRIMARY KEY, email VARCHAR(100) NOT NULL, ...	0 row(s) affected
✓ 4	08:25:43	CREATE TABLE students (roll_no INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT N...	0 row(s) affected

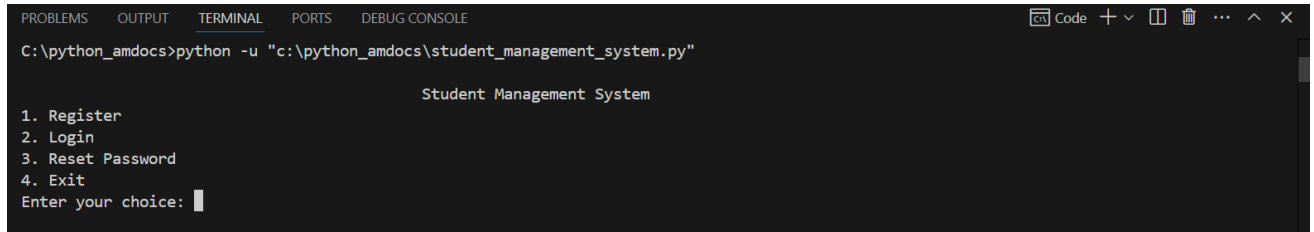
Result Grid						
Filter Rows: <input type="text"/> Edit: <input type="text"/> Export/Import: <input type="text"/> Wrap Cell Content: <input type="text"/>						
	id	email	username	password	phone_no	role
▶	1	admin123@gmail.com	admin123	fb340199f887da9955efd61c484994ed4794a02...	4567876578	admin
	2	gkpatil1@gmail.com	gayatri12	05d3302af2133d2c081034102879489bb2cbd6a...	5678768765	student
	3	simafac@gmail.com	Sima 123@	6e43ade0a604f9ca10143e0f617f6c02e4a96a8...	5678987687	faculty
✱	NULL	NULL	NULL	NULL	NULL	NULL

Users Table

[illegible]

Students Table

Output

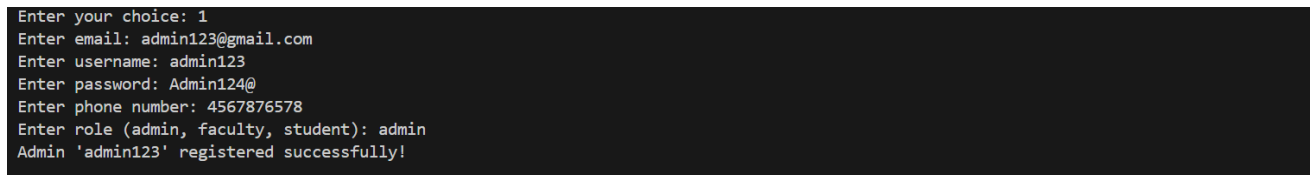


```
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
C:\python_amdocs>python -u "c:\python_amdocs\student_management_system.py"

Student Management System

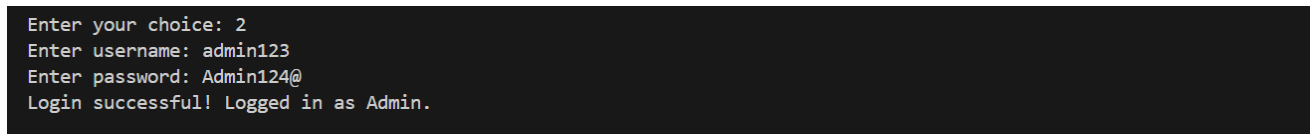
1. Register
2. Login
3. Reset Password
4. Exit
Enter your choice: █
```

Fig 01 : Student Management System



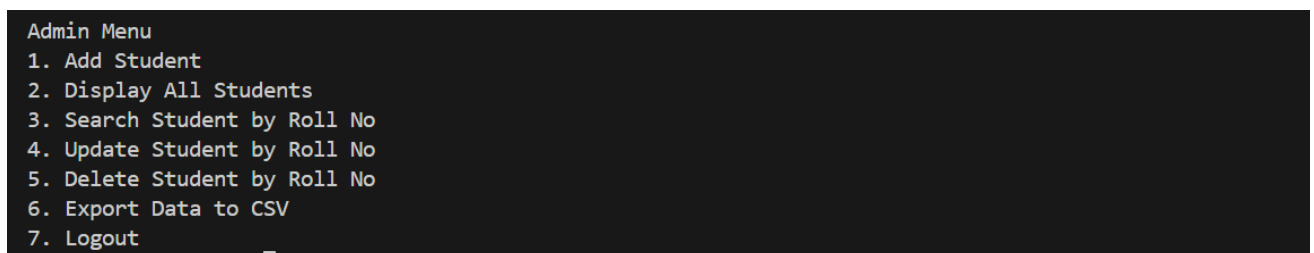
```
Enter your choice: 1
Enter email: admin123@gmail.com
Enter username: admin123
Enter password: Admin124@
Enter phone number: 4567876578
Enter role (admin, faculty, student): admin
Admin 'admin123' registered successfully!
```

Fig 02 : Admin Registration



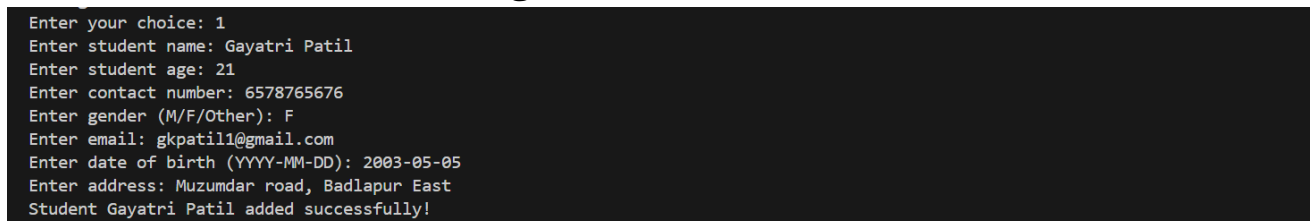
```
Enter your choice: 2
Enter username: admin123
Enter password: Admin124@
Login successful! Logged in as Admin.
```

Fig 03: Admin Login



```
Admin Menu
1. Add Student
2. Display All Students
3. Search Student by Roll No
4. Update Student by Roll No
5. Delete Student by Roll No
6. Export Data to CSV
7. Logout
```

Fig 04 : Admin Menu



```
Enter your choice: 1
Enter student name: Gayatri Patil
Enter student age: 21
Enter contact number: 6578765676
Enter gender (M/F/Other): F
Enter email: gkpatil1@gmail.com
Enter date of birth (YYYY-MM-DD): 2003-05-05
Enter address: Muzumdar road, Badlapur East
Student Gayatri Patil added successfully!
```

Fig 05 : Added Student Details

```

Enter your choice: 2
Roll No | Name | Age | Contact | Gender | Email | DOB | Address
-----
1 | Gayatri Patil | 21 | 6578765676 | F | gkpatil1@gmail.com | 2003-05-05 | Muzumdar road, Badlapur East
2 | Suhana Chaudhari | 22 | 6574389264 | F | Suhana3@gmail.com | 2002-08-01 | Shivaji Nagar, Pune
3 | Rohan More | 23 | 6574543454 | M | RohM1@gmail.com | 2002-03-21 | Juhu Tara Road, Santacruz, Mumbai
4 | Yash Patil | 22 | 8798968769 | M | yash2@gmail.com | 2002-03-08 | Cosmos, Near Destination Centre, Pune
5 | Madhuri Patel | 21 | 6787988767 | F | Madhp@gmail.com | 2003-01-05 | Cybercity, Magarpatta, Pune

```

Fig 06 : Displayed All Students Details

```

Enter your choice: 3
Enter roll number to search: 4
Roll No | Name | Age | Contact | Gender | Email | DOB | Address
-----
4 | Yash Patil | 22 | 8798968769 | M | yash2@gmail.com | 2002-03-08 | Cosmos, Near Destination Centre, Pune

```

Fig 07 : Fetched Student Details by Roll No.

```

Enter your choice: 4
Enter roll number to update: 3
Enter new name (leave blank to skip):
Enter new age (leave blank to skip):
Enter new contact (leave blank to skip): 4565435423
Enter new gender (leave blank to skip):
Enter new email (leave blank to skip):
Enter new date of birth (leave blank to skip):
Enter new address (leave blank to skip):
Student Roll No 3 updated successfully!

```

Fig 08 : Updating Student Details by Roll No.

```

Enter your choice: 3
Enter roll number to search: 3
Roll No | Name | Age | Contact | Gender | Email | DOB | Address
-----
3 | Rohan More | 23 | 4565435423 | M | RohM1@gmail.com | 2002-03-21 | Juhu Tara Road, Santacruz, Mumbai

```

Fig 09 : Updated Student Details

```

Enter your choice: 5
Enter roll number to delete: 2
Student Roll No 2 deleted successfully!

```

Fig 10 : Deleted Student Details

```

Enter your choice: 2
Roll No | Name | Age | Contact | Gender | Email | DOB | Address
-----
1 | Gayatri Patil | 21 | 6578765676 | F | gkpatil1@gmail.com | 2003-05-05 | Muzumdar road, Badlapur East
3 | Rohan More | 23 | 4565435423 | M | RohM1@gmail.com | 2002-03-21 | Juhu Tara Road, Santacruz, Mumbai
4 | Yash Patil | 22 | 8798968769 | M | yash2@gmail.com | 2002-03-08 | Cosmos, Near Destination Centre, Pune
5 | Madhuri Patel | 21 | 6787988767 | F | Madhp@gmail.com | 2003-01-05 | Cybercity, Magarpatta, Pune

```

Fig 11 : Updated Students Details after Deletion

```

Enter your choice: 6
Data exported to students_data.csv successfully!

```

Fig 12 : Exported Student Data in csv format

```
student_management_system.py  students_data.csv X
students_data.csv > data
1 Roll No,Name,Age,Contact,Gender,Email,DOB,Address
2 1,Gayatri Patil,21,6578765676,F,gkpatil1@gmail.com,2003-05-05,"Muzumdar road, Badlapur East"
3 3,Rohan More,23,4565435423,M,RohM1@gmail.com,2002-03-21,"Juhu Tara Road, Santacruz, Mumbai"
4 4,Yash Patil,22,8798968769,M,yash2@gmail.com,2002-03-08,"Cosmos, Near Destination Centre, Pune"
5 5,Madhuri Patel,21,6787988767,F,Madhp@gmail.com,2003-01-05,"Cybercity, Magarpatta, Pune"
6
```

Fig 13 : csv file

```
Enter your choice: 1
Enter email: gkpatil1@gmail.com
Enter username: gayatri12
Enter password: Gkp12@
Enter phone number: 5678768765
Enter role (admin, faculty, student): student
Student 'gayatri12' registered successfully!
```

Fig 14 : Student Registration

```
Enter your choice: 2
Enter username: gayatri12
Enter password: Gkp12@
Login successful! Logged in as Student.
```

Fig 15 : Student Login

```
Student Menu
1. View My Record
2. Logout
```

Fig 16 : Student Menu

```
Enter your choice: 1
Enter roll number to view your record: 1
Roll No | Name | Age | Contact | Gender | Email | DOB | Address
-----
1 | Gayatri Patil | 21 | 6578765676 | F | gkpatil1@gmail.com | 2003-05-05 | Muzumdar road, Badlapur East
```

Fig 17 : Display Student Record

```
Enter your choice: 1
Enter email: simafac@gmail.com
Enter username: Sima123@
Enter password: facsima12
Enter phone number: 5678987687
Enter role (admin, faculty, student): faculty
Faculty 'Sima123@' registered successfully!
```

Fig 18 : Faculty Registration

```
Enter your choice: 2
Enter username: Sima123@
Enter password: facsima12
Login successful! Logged in as Faculty.
```

Fig 19 : Faculty Login

```
Faculty Menu
1. Display All Students
2. Search Student
3. Export Data
4. Logout
```

Fig 20 : Faculty Menu

```
Enter your choice: 3
Enter your registered email: gkpatil1@gmail.com
Enter your username: gayatri12
Enter your new password: Gayatrip12@
Confirm your new password: Gayatrip12@
Password reset successful!
```

Fig 21 : Reset Password Functionality

```
Enter your choice: 3
Enter your registered email: admin@gmail.com
Enter your username: admin
Email and username do not match our records.
```

Fig 22 : Error in Resetting the Password

```
Enter your choice: 1
Enter email: admin123@gmail.com
Enter username: admin123
Enter password: Admin124@
Enter phone number: 4567876578
Enter role (admin, faculty, student): admin
Username or email already exists.
```

Fig 23 : Ensuring No data redundancy.