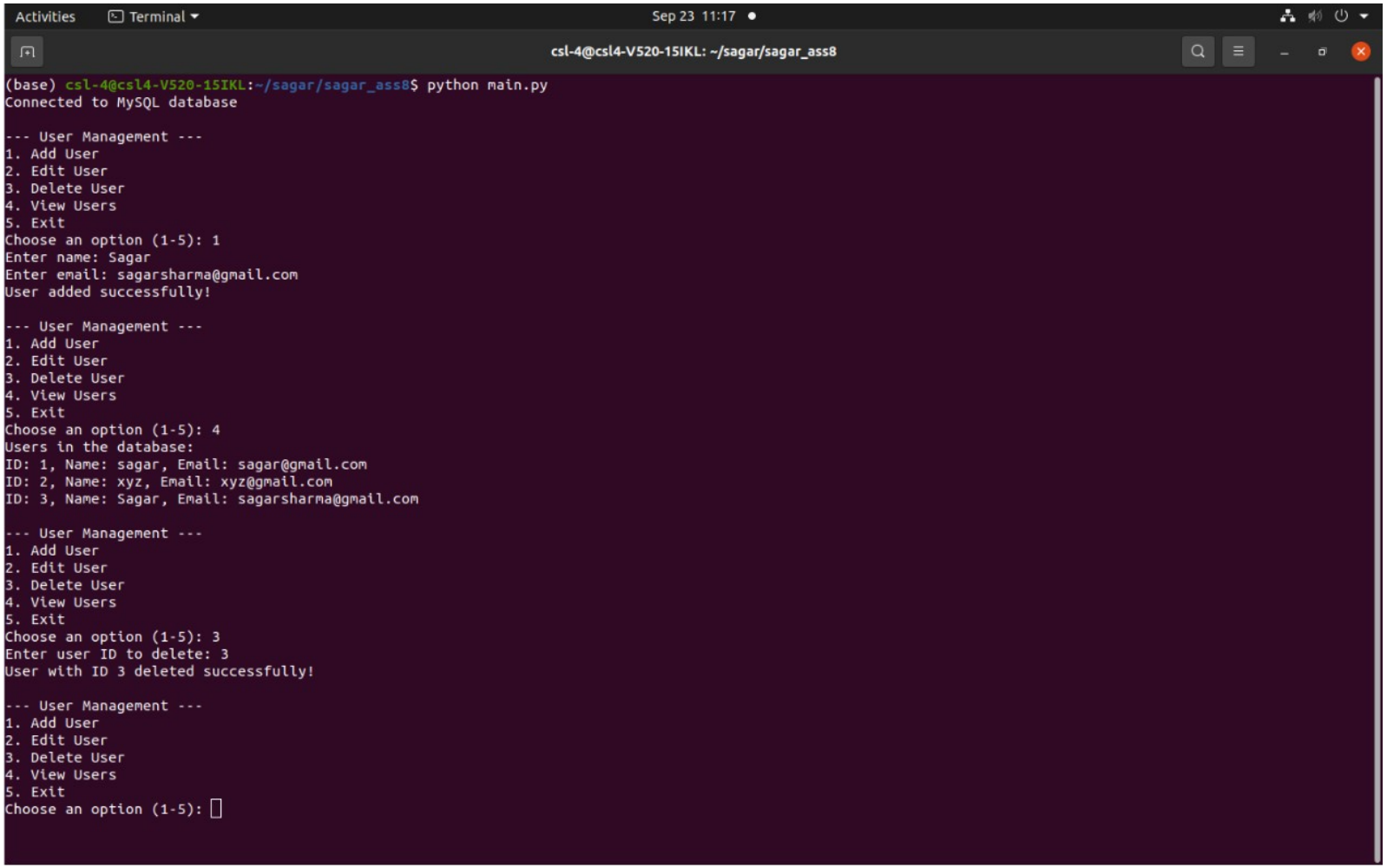
```
import mysql.connector


host = 'localhost'
user = 'root'
password = 'ait123'
database = 'sagar_ass8'

def connect_db():
    try:
        connection = mysql.connector.connect(
            host=host,
            user=user,
            password=password,
            database=database
        )
        if connection.is_connected():
            print("Connected to MySQL database")
        return connection
    except mysql.connector.Error as err:
        print(f"Error: {err}")
        return None


def add_user(connection):
    name = input("Enter name: ")
    email = input("Enter email: ")

    cursor = connection.cursor()
    query = "INSERT INTO users (name, email) VALUES (%s, %s)"
    cursor.execute(query, (name, email))
    connection.commit()

    print("User added successfully!")

def edit_user(connection):
    user_id = input("Enter user ID to edit: ")
    name = input("Enter new name: ")
    email = input("Enter new email: ")

    cursor = connection.cursor()
    query = "UPDATE users SET name = %s, email = %s WHERE id = %s"
    cursor.execute(query, (name, email, user_id))
    connection.commit()

    print(f"User with ID {user_id} updated successfully!")


def delete_user(connection):
    user_id = input("Enter user ID to delete: ")

    cursor = connection.cursor()
    query = "DELETE FROM users WHERE id = %s"
    cursor.execute(query, (user_id,))
    connection.commit()

    print(f"User with ID {user_id} deleted successfully!")
```

```
def view_users(connection):
    cursor = connection.cursor()
    query = "SELECT * FROM users"
    cursor.execute(query)

    users = cursor.fetchall()
    print("Users in the database:")
    for user in users:
        print(f"ID: {user[0]}, Name: {user[1]}, Email: {user[2]}")


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

    while True:
        print("\n--- User Management ---")
        print("1. Add User")
        print("2. Edit User")
        print("3. Delete User")
        print("4. View Users")
        print("5. Exit")

        choice = input("Choose an option (1-5): ")

        if choice == '1':
            add_user(connection)
        elif choice == '2':
            edit_user(connection)
        elif choice == '3':
            delete_user(connection)
        elif choice == '4':
            view_users(connection)
        elif choice == '5':
            print("Exiting program.")
            break
        else:
            print("Invalid option. Please try again.")

    connection.close()

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


Activities
Terminal
Sep 23 11:17
csl-4@csl4-V520-15IKL: ~/sagar/sagar_ass8
(base) csl-4@csl4-V520-15IKL:~/sagar/sagar_ass8$ python main.py
Connected to MySQL database
--- User Management ---
1. Add User
2. Edit User
3. Delete User
4. View Users
5. Exit
Choose an option (1-5): 1
Enter name: Sagar
Enter email: sagarsharma@gmail.com
User added successfully!
--- User Management ---
1. Add User
2. Edit User
3. Delete User
4. View Users
5. Exit
Choose an option (1-5): 4
Users in the database:
ID: 1, Name: sagar, Email: sagar@gmail.com
ID: 2, Name: xyz, Email: xyz@gmail.com
ID: 3, Name: Sagar, Email: sagarsharma@gmail.com
--- User Management ---
1. Add User
2. Edit User
3. Delete User
4. View Users
5. Exit
Choose an option (1-5): 3
Enter user ID to delete: 3
User with ID 3 deleted successfully!
--- User Management ---
1. Add User
2. Edit User
3. Delete User
4. View Users
5. Exit
Choose an option (1-5):

```
db.createCollection("student")
show collections
student

db.student.insertMany([
{ name: "Ram", rollNo: 101, marks: 85, CGPA: 9.2, branch: "Computer Science", nativeCity: "Delhi" },
{ name: "Sita", rollNo: 102, marks: 88, CGPA: 8.6, branch: "IT", nativeCity: "Mumbai" },
{ name: "Shyam", rollNo: 103, marks: 75, CGPA: 7.9, branch: "Electrical", nativeCity: "Kolkata" },
{ name: "Ravi", rollNo: 104, marks: 90, CGPA: 9.5, branch: "Computer Science", nativeCity: "Bangalore" },
{ name: "Priya", rollNo: 105, marks: 80, CGPA: 8.2, branch: "Electronics", nativeCity: "Chennai" },
{ name: "Amit", rollNo: 106, marks: 95, CGPA: 9.8, branch: "Computer Science", nativeCity: "Delhi" },
{ name: "Geeta", rollNo: 107, marks: 78, CGPA: 8.0, branch: "Civil", nativeCity: "Hyderabad" }])
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("68b68f3350e207eedee6292a"),
                ObjectId("68b68f3350e207eedee6292b"),
                ObjectId("68b68f3350e207eedee6292c"),
                ObjectId("68b68f3350e207eedee6292d"),
                ObjectId("68b68f3350e207eedee6292e"),
                ObjectId("68b68f3350e207eedee6292f"),
                ObjectId("68b68f3350e207eedee62930")
        ]
}

db.student.find().pretty()
{
        "_id" : ObjectId("68b68f3350e207eedee6292a"),
        "name" : "Ram",
        "rollNo" : 101,
        "marks" : 85,
        "CGPA" : 9.2,
        "branch" : "Computer Science",
        "nativeCity" : "Delhi"
}
{
        "_id" : ObjectId("68b68f3350e207eedee6292b"),
        "name" : "Sita",
        "rollNo" : 102,
        "marks" : 88,
        "CGPA" : 8.6,
        "branch" : "IT",
        "nativeCity" : "Mumbai"
}
{
        "_id" : ObjectId("68b68f3350e207eedee6292c"),
        "name" : "Shyam",
        "rollNo" : 103,
        "marks" : 75,
        "CGPA" : 7.9,
        "branch" : "Electrical",
        "nativeCity" : "Kolkata"
}
{
        "_id" : ObjectId("68b68f3350e207eedee6292d"),
        "name" : "Ravi",
        "rollNo" : 104,
```

```
        "marks" : 90,
        "CGPA" : 9.5,
        "branch" : "Computer Science",
        "nativeCity" : "Bangalore"
}
{
        "_id" : ObjectId("68b68f3350e207eedee6292e"),
        "name" : "Priya",
        "rollNo" : 105,
        "marks" : 80,
        "CGPA" : 8.2,
        "branch" : "Electronics",
        "nativeCity" : "Chennai"
}
{
        "_id" : ObjectId("68b68f3350e207eedee6292f"),
        "name" : "Amit",
        "rollNo" : 106,
        "marks" : 95,
        "CGPA" : 9.8,
        "branch" : "Computer Science",
        "nativeCity" : "Delhi"
}
{
        "_id" : ObjectId("68b68f3350e207eedee62930"),
        "name" : "Geeta",
        "rollNo" : 107,
        "marks" : 78,
        "CGPA" : 8,
        "branch" : "Civil",
        "nativeCity" : "Hyderabad"
}

db.student.updateOne(
{ name: "Ram" }, { $set: { marks: 90 } })
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

db.student.find({ $or: [{ CGPA: { $gte: 9 } }, { branch: "Computer Science" }] }).pretty()
{
        "_id" : ObjectId("68b68f3350e207eedee6292a"),
        "name" : "Ram",
        "rollNo" : 101,
        "marks" : 90,
        "CGPA" : 9.2,
        "branch" : "Computer Science",
        "nativeCity" : "Delhi"
}
{
        "_id" : ObjectId("68b68f3350e207eedee6292d"),
        "name" : "Ravi",
        "rollNo" : 104,
        "marks" : 90,
        "CGPA" : 9.5,
        "branch" : "Computer Science",
        "nativeCity" : "Bangalore"
}
{
```

```
        "_id" : ObjectId("68b68f3350e207eedee6292f"),
        "name" : "Amit",
        "rollNo" : 106,
        "marks" : 95,
        "CGPA" : 9.8,
        "branch" : "Computer Science",
        "nativeCity" : "Delhi"
}

db.student.find({ branch: { $ne: "IT" } }).pretty()
{
        "_id" : ObjectId("68b68f3350e207eedee6292a"),
        "name" : "Ram",
        "rollNo" : 101,
        "marks" : 90,
        "CGPA" : 9.2,
        "branch" : "Computer Science",
        "nativeCity" : "Delhi"
}
{
        "_id" : ObjectId("68b68f3350e207eedee6292c"),
        "name" : "Shyam",
        "rollNo" : 103,
        "marks" : 75,
        "CGPA" : 7.9,
        "branch" : "Electrical",
        "nativeCity" : "Kolkata"
}
{
        "_id" : ObjectId("68b68f3350e207eedee6292d"),
        "name" : "Ravi",
        "rollNo" : 104,
        "marks" : 90,
        "CGPA" : 9.5,
        "branch" : "Computer Science",
        "nativeCity" : "Bangalore"
}
{
        "_id" : ObjectId("68b68f3350e207eedee6292e"),
        "name" : "Priya",
        "rollNo" : 105,
        "marks" : 80,
        "CGPA" : 8.2,
        "branch" : "Electronics",
        "nativeCity" : "Chennai"
}
{
        "_id" : ObjectId("68b68f3350e207eedee6292f"),
        "name" : "Amit",
        "rollNo" : 106,
        "marks" : 95,
        "CGPA" : 9.8,
        "branch" : "Computer Science",
        "nativeCity" : "Delhi"
}
{
        "_id" : ObjectId("68b68f3350e207eedee62930"),
```

```
        "name" : "Geeta",
        "rollNo" : 107,
        "marks" : 78,
        "CGPA" : 8,
        "branch" : "Civil",
        "nativeCity" : "Hyderabad"
}

db.student.find({ $nor: [{ CGPA: { $eq: 9 } }, { branch: "Computer Science" }]}).pretty()
{
        "_id" : ObjectId("68b68f3350e207eedee6292b"),
        "name" : "Sita",
        "rollNo" : 102,
        "marks" : 88,
        "CGPA" : 8.6,
        "branch" : "IT",
        "nativeCity" : "Mumbai"
}
{
        "_id" : ObjectId("68b68f3350e207eedee6292c"),
        "name" : "Shyam",
        "rollNo" : 103,
        "marks" : 75,
        "CGPA" : 7.9,
        "branch" : "Electrical",
        "nativeCity" : "Kolkata"
}
{
        "_id" : ObjectId("68b68f3350e207eedee6292e"),
        "name" : "Priya",
        "rollNo" : 105,
        "marks" : 80,
        "CGPA" : 8.2,
        "branch" : "Electronics",
        "nativeCity" : "Chennai"
}
{
        "_id" : ObjectId("68b68f3350e207eedee62930"),
        "name" : "Geeta",
        "rollNo" : 107,
        "marks" : 78,
        "CGPA" : 8,
        "branch" : "Civil",
        "nativeCity" : "Hyderabad"
}

db.student.save({ name: "Nisha", rollNo: 108, marks: 88, CGPA: 8.1, branch: "Computer Science",
nativeCity: "Mumbai" })
WriteResult({ "nInserted" : 1 })
db.student.deleteOne({ name: "Sita" })
{ "acknowledged" : true, "deletedCount" : 1 }
db.student.find({ branch: "Computer Science" }).sort({ CGPA: -1 }).limit(1).pretty()
{
        "_id" : ObjectId("68b68f3350e207eedee6292f"),
        "name" : "Amit",
        "rollNo" : 106,
        "marks" : 95,
```

```
        "CGPA" : 9.8,
        "branch" : "Computer Science",
        "nativeCity" : "Delhi"
}
```