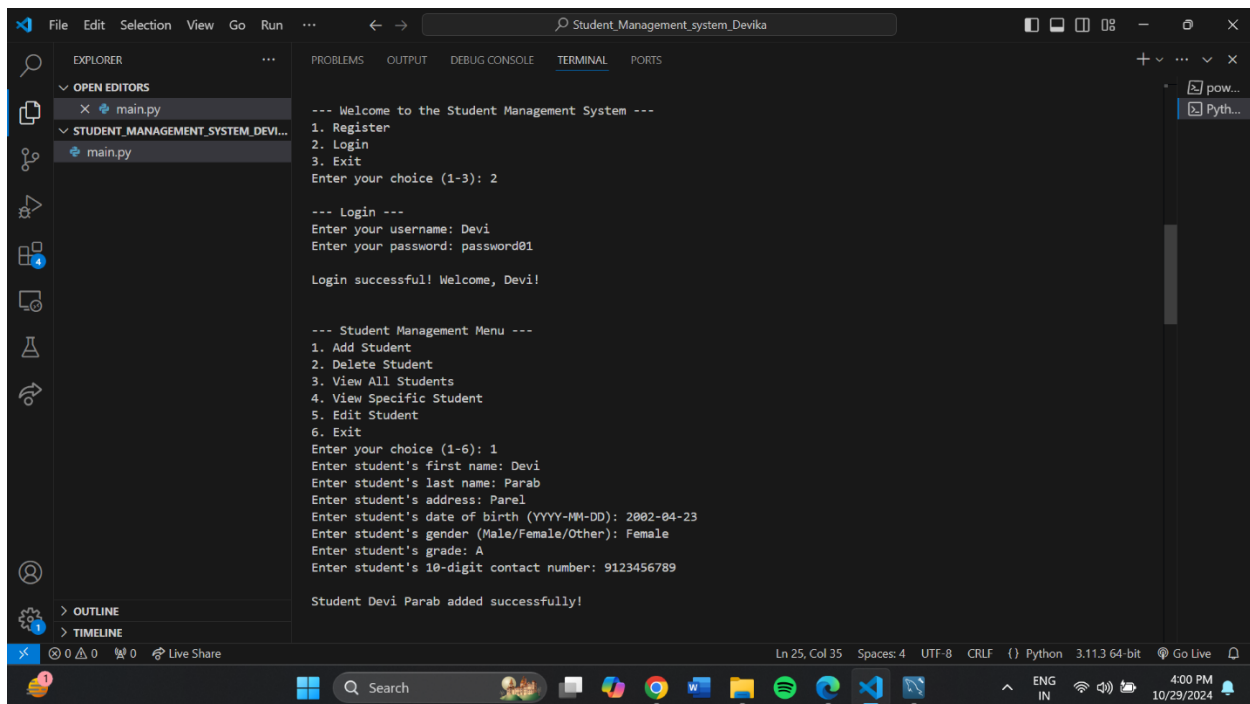


# Student Management System

Name – Devika Parab

Output & Implementation:



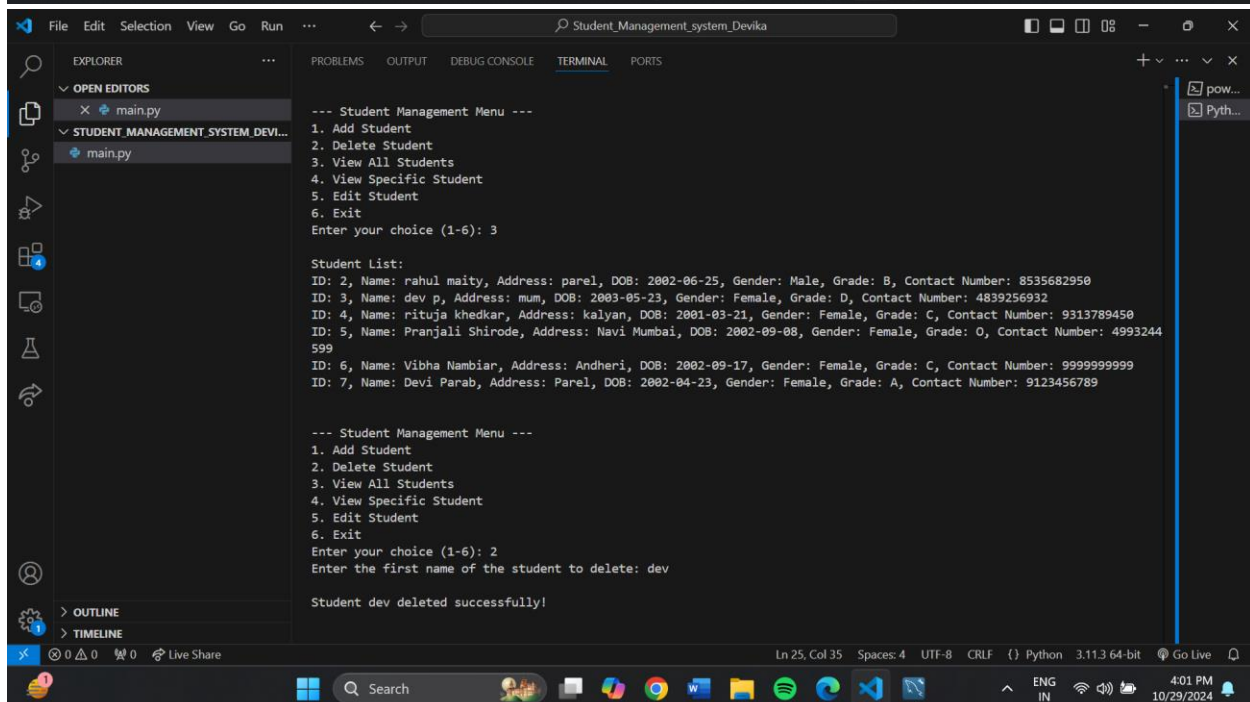
```
File Edit Selection View Go Run ... Student_Management_system_Devika
EXPLORER
  OPEN EDITORS
    main.py
  STUDENT_MANAGEMENT_SYSTEM_DEVIL...
    main.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
--- Welcome to the Student Management System ---
1. Register
2. Login
3. Exit
Enter your choice (1-3): 2

--- Login ---
Enter your username: Devi
Enter your password: password01

Login successful! Welcome, Devi!

--- Student Management Menu ---
1. Add Student
2. Delete Student
3. View All Students
4. View Specific Student
5. Edit Student
6. Exit
Enter your choice (1-6): 1
Enter student's first name: Devi
Enter student's last name: Parab
Enter student's address: Parel
Enter student's date of birth (YYYY-MM-DD): 2002-04-23
Enter student's gender (Male/Female/Other): Female
Enter student's grade: A
Enter student's 10-digit contact number: 9123456789

Student Devi Parab added successfully!
Ln 25, Col 35 Spaces: 4 UTF-8 CRLF () Python 3.11.3 64-bit Go Live
```



```
File Edit Selection View Go Run ... Student_Management_system_Devika
EXPLORER
  OPEN EDITORS
    main.py
  STUDENT_MANAGEMENT_SYSTEM_DEVIL...
    main.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
--- Student Management Menu ---
1. Add Student
2. Delete Student
3. View All Students
4. View Specific Student
5. Edit Student
6. Exit
Enter your choice (1-6): 3

Student List:
ID: 2, Name: rahul maity, Address: parel, DOB: 2002-06-25, Gender: Male, Grade: B, Contact Number: 8535682950
ID: 3, Name: dev p, Address: mum, DOB: 2003-05-23, Gender: Female, Grade: D, Contact Number: 4839256932
ID: 4, Name: rituja khedkar, Address: kalyan, DOB: 2001-03-21, Gender: Female, Grade: C, Contact Number: 9313789450
ID: 5, Name: Pranjali Shirode, Address: Navi Mumbai, DOB: 2002-09-08, Gender: Female, Grade: O, Contact Number: 4993244599
ID: 6, Name: Vibha Nambiar, Address: Andheri, DOB: 2002-09-17, Gender: Female, Grade: C, Contact Number: 9999999999
ID: 7, Name: Devi Parab, Address: Parel, DOB: 2002-04-23, Gender: Female, Grade: A, Contact Number: 9123456789

--- Student Management Menu ---
1. Add Student
2. Delete Student
3. View All Students
4. View Specific Student
5. Edit Student
6. Exit
Enter your choice (1-6): 2
Enter the first name of the student to delete: dev

Student dev deleted successfully!
Ln 25, Col 35 Spaces: 4 UTF-8 CRLF () Python 3.11.3 64-bit Go Live
```

File Edit Selection View Go Run ...

Student\_Management\_system\_Devika

EXPLORER

main.py

main.py

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
--- Student Management Menu ---
1. Add Student
2. Delete Student
3. View All Students
4. View Specific Student
5. Edit Student
6. Exit
Enter your choice (1-6): 4
Enter the first name of the student to view: Pranjali

Name: Pranjali Shirode, Address: Navi Mumbai, DOB: 2002-09-08, Gender: Female, Grade: O, Contact Number: 4993244599

--- Student Management Menu ---
1. Add Student
2. Delete Student
3. View All Students
4. View Specific Student
5. Edit Student
6. Exit
Enter your choice (1-6): 5
Enter the first name of the student to edit: Pranjali

Student found.
Enter new first name: Pranjali
Enter new last name: S
Enter new address: Pune
Enter new date of birth (YYYY-MM-DD): 2002-06-20
Enter new gender (Male/Female/Other): Female
Enter new grade: B
Enter new 10-digit contact number: 9999999999
```

Ln 25, Col 35 Spaces: 4 UTF-8 CRLF Python 3.11.3 64-bit Go Live

File Edit Selection View Go Run ...

Student\_Management\_system\_Devika

EXPLORER

main.py

main.py

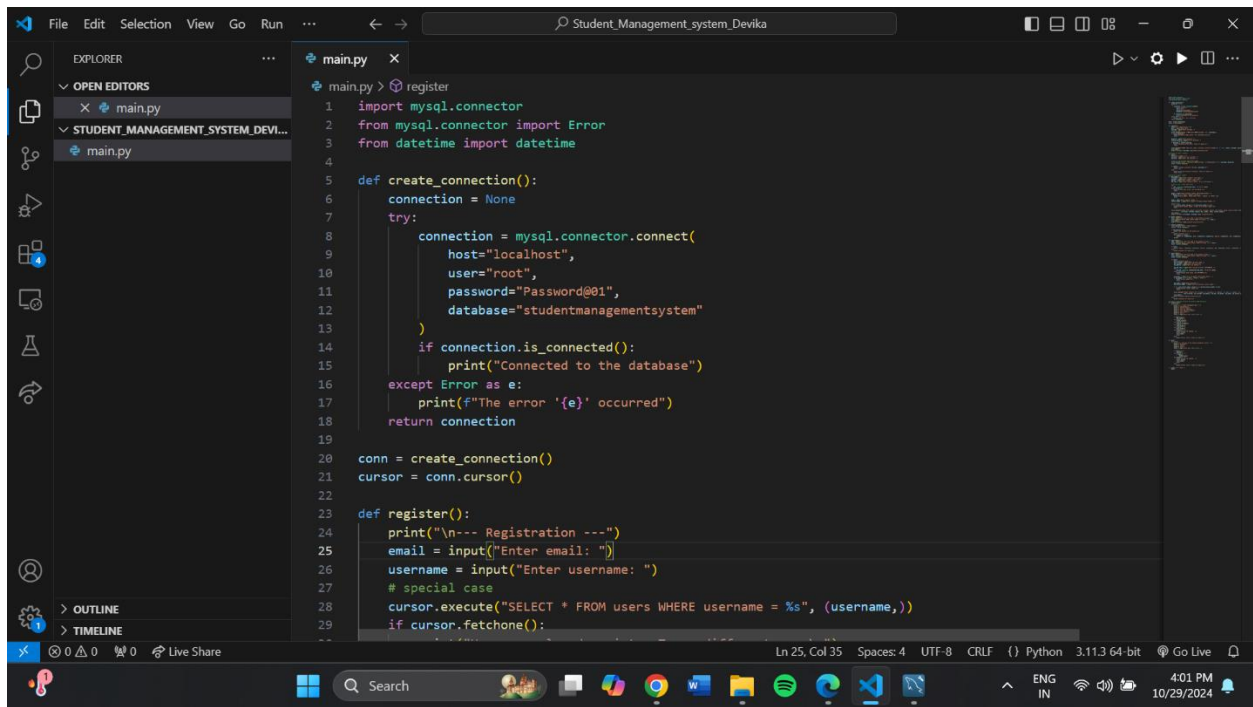
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Student updated successfully!

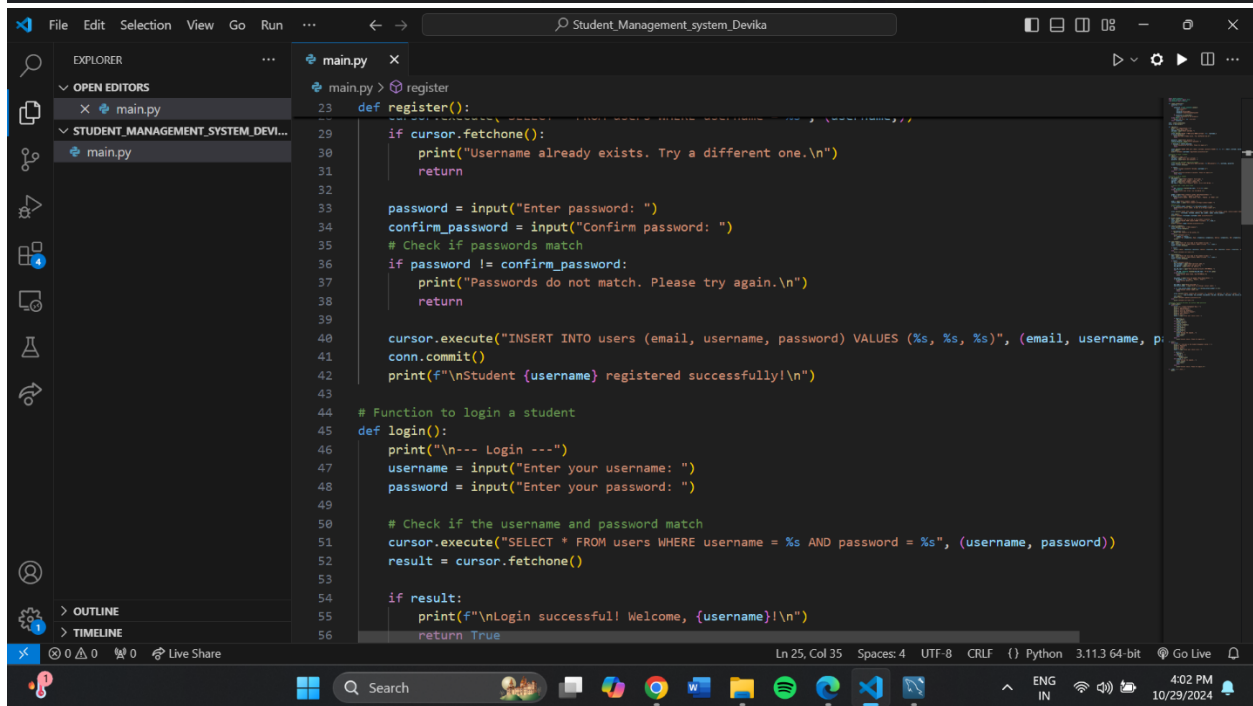
--- Student Management Menu ---
1. Add Student
2. Delete Student
3. View All Students
4. View Specific Student
5. Edit Student
6. Exit
Enter your choice (1-6): 6
Exiting the program...

--- Welcome to the Student Management System ---
1. Register
2. Login
3. Exit
Enter your choice (1-3):
```

Ln 25, Col 35 Spaces: 4 UTF-8 CRLF Python 3.11.3 64-bit Go Live



```
1 import mysql.connector
2 from mysql.connector import Error
3 from datetime import datetime
4
5 def create_connection():
6     connection = None
7     try:
8         connection = mysql.connector.connect(
9             host="localhost",
10            user="root",
11            password="Password@01",
12            database="studentmanagementsystem"
13        )
14        if connection.is_connected():
15            print("Connected to the database")
16    except Error as e:
17        print(f"The error '{e}' occurred")
18    return connection
19
20 conn = create_connection()
21 cursor = conn.cursor()
22
23 def register():
24     print("\n--- Registration ---")
25     email = input("Enter email: ")
26     username = input("Enter username: ")
27     # special case
28     cursor.execute("SELECT * FROM users WHERE username = %s", (username,))
29     if cursor.fetchone():
```



```
29     if cursor.fetchone():
30         print("Username already exists. Try a different one.\n")
31         return
32
33     password = input("Enter password: ")
34     confirm_password = input("Confirm password: ")
35     # Check if passwords match
36     if password != confirm_password:
37         print("Passwords do not match. Please try again.\n")
38         return
39
40     cursor.execute("INSERT INTO users (email, username, password) VALUES (%s, %s, %s)", (email, username, password))
41     conn.commit()
42     print(f"\nStudent {username} registered successfully!\n")
43
44 # Function to login a student
45 def login():
46     print("\n--- Login ---")
47     username = input("Enter your username: ")
48     password = input("Enter your password: ")
49
50     # Check if the username and password match
51     cursor.execute("SELECT * FROM users WHERE username = %s AND password = %s", (username, password))
52     result = cursor.fetchone()
53
54     if result:
55         print(f"\nLogin successful! Welcome, {username}!\n")
56         return True
```

```
File Edit Selection View Go Run ... Student_Management_system_Devika
EXPLORER
  OPEN EDITORS
    main.py
  STUDENT_MANAGEMENT_SYSTEM_DEV...
    main.py
main.py
45 def login():
55     print(f"\nLogin successful! Welcome, {username}!\n")
56     return True
57 else:
58     print("\nInvalid username or password. Please try again.\n")
59     return False
60
61 # Function to add a student
62 def add_student():
63     firstname = input("Enter student's first name: ")
64     lastname = input("Enter student's last name: ")
65     address = input("Enter student's address: ")
66     dob_input = input("Enter student's date of birth (YYYY-MM-DD): ")
67
68     # special case - Check date format
69     try:
70         dob = datetime.strptime(dob_input, "%Y-%m-%d").date()
71     except ValueError:
72         print("Invalid date format. Use YYYY-MM-DD.\n")
73         return
74
75     gender = input("Enter student's gender (Male/Female/Other): ")
76     if gender not in ('Male', 'Female', 'Other'):
77         print("Invalid gender. Please enter 'Male', 'Female', or 'Other'.\n")
78         return
79
80     grade = input("Enter student's grade: ")
81     contact_number = input("Enter student's 10-digit contact number: ")
82
```

```
File Edit Selection View Go Run ... Student_Management_system_Devika
EXPLORER
  OPEN EDITORS
    main.py
  STUDENT_MANAGEMENT_SYSTEM_DEV...
    main.py
main.py
62 def add_student():
83     #special case
84     if not (contact_number.isdigit() and len(contact_number) == 10):
85         print("Invalid contact number. It must be a 10-digit number.\n")
86         return
87
88     cursor.execute("INSERT INTO students (firstname, lastname, address, dob, gender, grade, contact_number)
89                     (firstname, lastname, address, dob, gender, grade, contact_number)")
90     conn.commit()
91     print(f"\nStudent {firstname} {lastname} added successfully!\n")
92
93 def delete_student():
94     name = input("Enter the first name of the student to delete: ")
95     cursor.execute("DELETE FROM students WHERE firstname = %s", (name,))
96     conn.commit()
97     print(f"\nStudent {name} deleted successfully!\n")
98
99 def view_all_students():
100     cursor.execute("SELECT * FROM students")
101     results = cursor.fetchall()
102
103     if len(results) == 0:
104         print("\nNo students in the system.\n")
105     else:
106         print("\nStudent List:")
107         for student in results:
108             print(f"ID: {student[0]}, Name: {student[1]} {student[2]}, Address: {student[3]}, DOB: {student[4]}, Gender: {student[5]}, Grade: {student[6]}, Contact: {student[7]}")
109         print()
110
```

```
File Edit Selection View Go Run ... Student_Management_system_Devika
EXPLORER
  OPEN EDITORS
    main.py
  STUDENT_MANAGEMENT_SYSTEM_DEV...
    main.py
  OUTLINE
  TIMELINE
  Live Share
Ln 25, Col 35 Spaces: 4 UTF-8 CRLF {} Python 3.11.3 64-bit Go Live 4:02 PM 10/29/2024

main.py
def view_student():
    name = input("Enter the first name of the student to view: ")
    cursor.execute("SELECT * FROM students WHERE firstname = %s", (name,))
    result = cursor.fetchone()

    if result:
        print(f"\nName: {result[1]} {result[2]}, Address: {result[3]}, DOB: {result[4]}, Gender: {result[5]}")
    else:
        print("\nStudent not found.\n")

def edit_student():
    name = input("Enter the first name of the student to edit: ")
    cursor.execute("SELECT * FROM students WHERE firstname = %s", (name,))
    result = cursor.fetchone()

    if result:
        print("\nStudent found.")
        new_firstname = input("Enter new first name: ")
        new_lastname = input("Enter new last name: ")
        new_address = input("Enter new address: ")

        new_dob_input = input("Enter new date of birth (YYYY-MM-DD): ")
        try:
            new_dob = datetime.strptime(new_dob_input, "%Y-%m-%d").date()
        except ValueError:
            print("Invalid date format. Use YYYY-MM-DD.\n")
            return

        new_gender = input("Enter new gender (Male/Female/Other): ")
```

```
File Edit Selection View Go Run ... Student_Management_system_Devika
EXPLORER
  OPEN EDITORS
    main.py
  STUDENT_MANAGEMENT_SYSTEM_DEV...
    main.py
  OUTLINE
  TIMELINE
  Live Share
Ln 25, Col 35 Spaces: 4 UTF-8 CRLF {} Python 3.11.3 64-bit Go Live 4:02 PM 10/29/2024

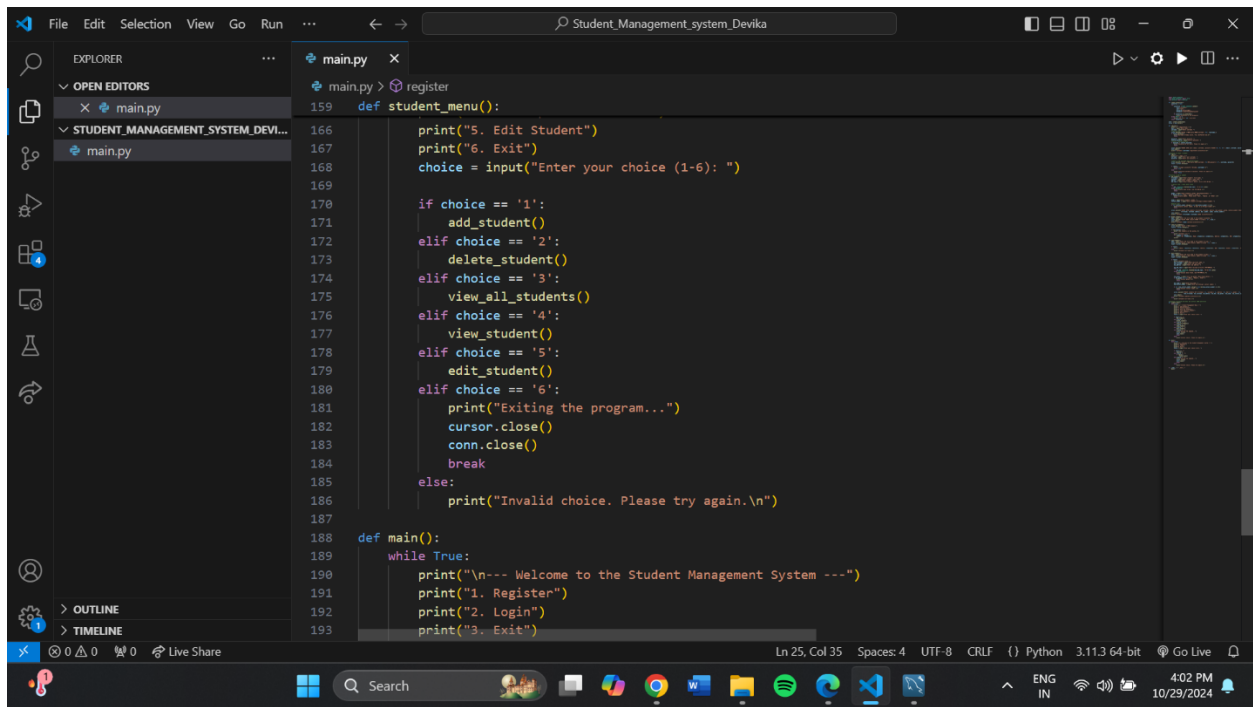
main.py
def edit_student():
    new_gender = input("Enter new gender (Male/Female/Other): ")
    if new_gender not in ('Male', 'Female', 'Other'):
        print("Invalid gender.\n")
        return

    new_grade = input("Enter new grade: ")
    new_contact_number = input("Enter new 10-digit contact number: ")

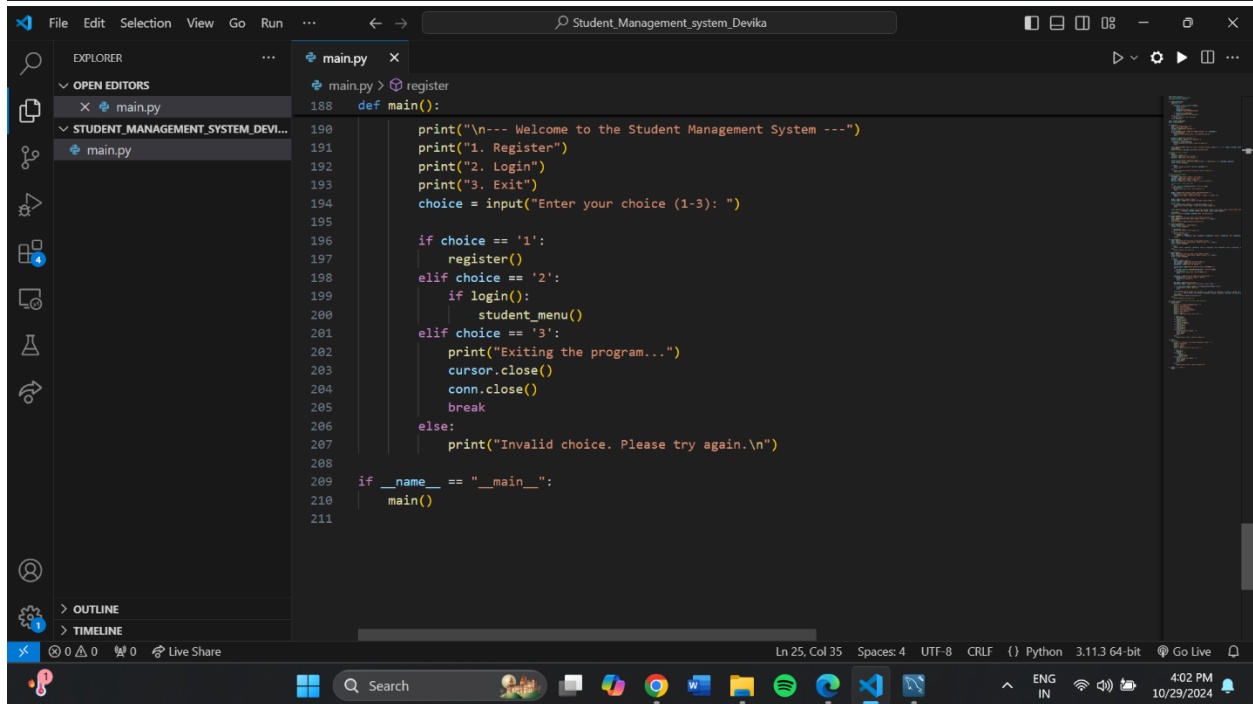
    if not (new_contact_number.isdigit() and len(new_contact_number) == 10):
        print("Invalid contact number.\n")
        return

    cursor.execute("UPDATE students SET firstname = %s, lastname = %s, address = %s, dob = %s, gender = %s, grade = %s, contact_number = %s",
                  (new_firstname, new_lastname, new_address, new_dob, new_gender, new_grade, new_contact_number))
    conn.commit()
    print("\nStudent updated successfully!\n")
    else:
        print("\nStudent not found.\n")

# Function to display the menu and perform CRUD operations
def student_menu():
    while True:
        print("\n--- Student Management Menu ---")
        print("1. Add Student")
        print("2. Delete Student")
        print("3. View All Students")
        print("4. View Specific Student")
        print("5. Edit Student")
```



```
File Edit Selection View Go Run ... Student_Management_system_Devika
EXPLORER
main.py
STUDENT_MANAGEMENT_SYSTEM_DEVIL...
main.py
main.py
159 def student_menu():
160     print("5. Edit Student")
161     print("6. Exit")
162     choice = input("Enter your choice (1-6): ")
163
164     if choice == '1':
165         add_student()
166     elif choice == '2':
167         delete_student()
168     elif choice == '3':
169         view_all_students()
170     elif choice == '4':
171         view_student()
172     elif choice == '5':
173         edit_student()
174     elif choice == '6':
175         print("Exiting the program...")
176         cursor.close()
177         conn.close()
178         break
179     else:
180         print("Invalid choice. Please try again.\n")
181
182 def main():
183     while True:
184         print("\n--- Welcome to the Student Management System ---")
185         print("1. Register")
186         print("2. Login")
187         print("3. Exit")
188         choice = input("Enter your choice (1-3): ")
189
190         if choice == '1':
191             register()
192         elif choice == '2':
193             login()
194             student_menu()
195         elif choice == '3':
196             print("Exiting the program...")
197             cursor.close()
198             conn.close()
199             break
200         else:
201             print("Invalid choice. Please try again.\n")
202
203 if __name__ == "__main__":
204     main()
205
206
```



```
File Edit Selection View Go Run ... Student_Management_system_Devika
EXPLORER
main.py
STUDENT_MANAGEMENT_SYSTEM_DEVIL...
main.py
main.py
188 def main():
189     print("\n--- Welcome to the Student Management System ---")
190     print("1. Register")
191     print("2. Login")
192     print("3. Exit")
193     choice = input("Enter your choice (1-3): ")
194
195     if choice == '1':
196         register()
197     elif choice == '2':
198         login()
199         student_menu()
200     elif choice == '3':
201         print("Exiting the program...")
202         cursor.close()
203         conn.close()
204         break
205     else:
206         print("Invalid choice. Please try again.\n")
207
208 if __name__ == "__main__":
209     main()
210
211
```



# Workbench :

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

student\_system

studentmanagementsystem

Tables

students

Columns

Indexes

Foreign Keys

Triggers

users

Columns

Indexes

Foreign Keys

Triggers

Views

Administration Schemas

Information

Table: students

Columns:

id int(11) P

firstname varchar(50)

lastname varchar(50)

address text

dob date

gender enum('M', 'F')

grade varchar(10)

contact\_number varchar(20)

SQL Editor

1 • SELECT \* FROM studentmanagementsystem.students;

Result Grid

#	id	firstname	lastname	address	dob	gender	grade	contact_number
1	2	rahul	malvi	parel	2002-06-25	Male	B	8535682950
2	3	dev	p	mum	2003-05-23	Female	D	4839256932
3	4	rituja	khedkar	kalyan	2001-03-21	Female	C	9313789450
4	5	Pranjal	Shirode	Navi Mumbai	2002-09-08	Female	O	4993244599
5	6	Vibha	Nambiar	Andheri	2002-09-17	Female	C	9999999999

students 1 x

Apply Revert

Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	12:10:02	SELECT * FROM studentmanagementsystem.users LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
2	12:10:41	SELECT * FROM studentmanagementsystem.students LIMIT 0, 1000	5 row(s) returned	0.016 sec / 0.000 sec

ENG IN 4:03 PM 10/29/2024