Clinic Management System

Submitted to : Dr Amir Arsalan

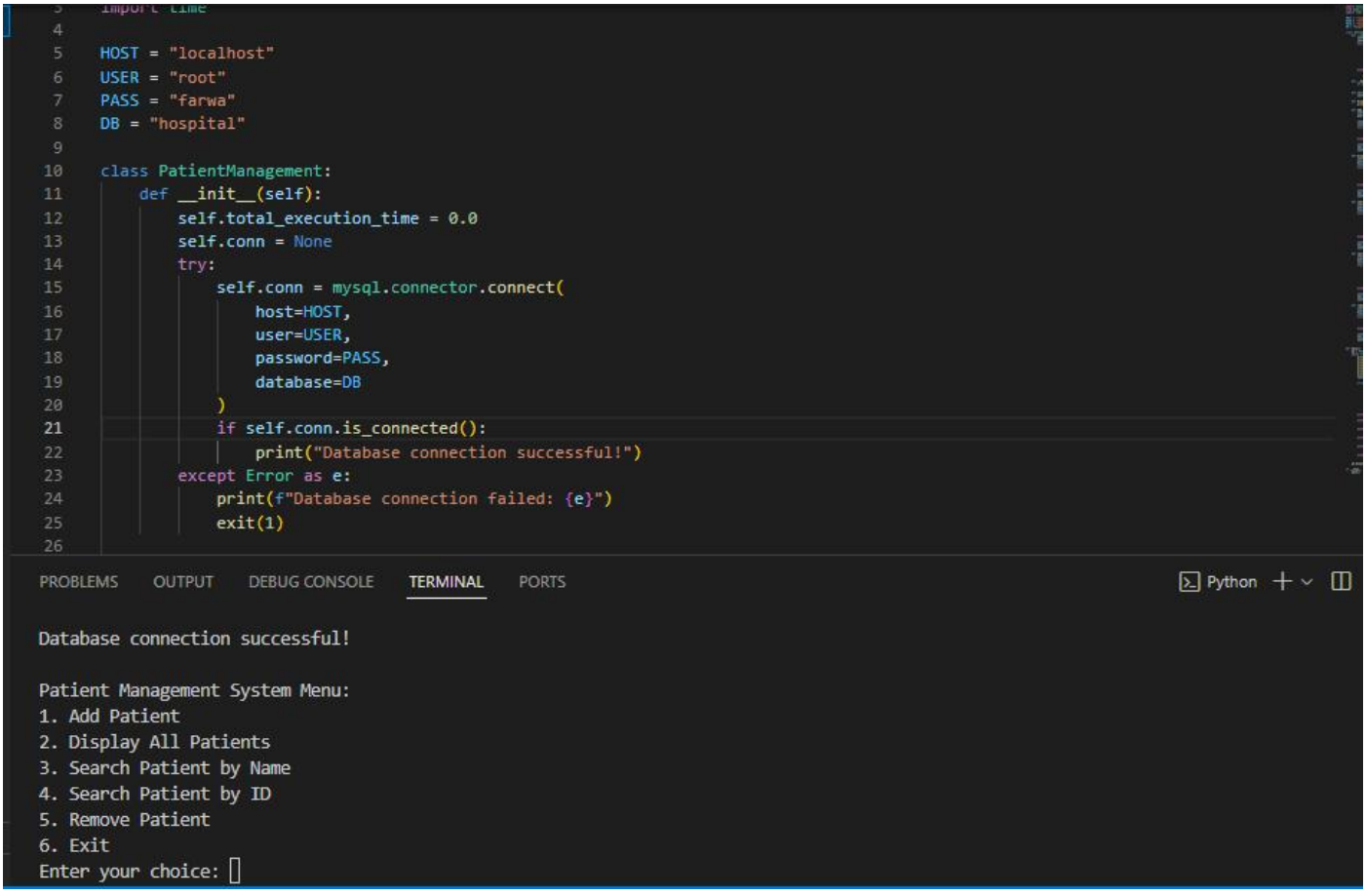
Submitted by :

- **Syeda farwa batool (2022-BSE-071)**
- **Eman zai (2022-BSE-049)**
- **Areej intishad (2022-BSE-046)**

Abstract of the system:

The Patient Management System (PMS) is a software application designed to manage patient records in a hospital. It provides a user-friendly interface for healthcare administrators to efficiently handle patient data, including adding, displaying, searching (by name or ID), and removing patient records. The PMS aims to automate patient management processes, reducing manual effort and increasing efficiency. The system is connected to a MySQL database to securely store and retrieve patient information, while also tracking the total execution time for database operations to monitor performance.

VS Code connection with Mysql :



Code:

```
import mysql.connector
from mysql.connector import Error
import time

HOST = "localhost"
USER = "root"
PASS = "farwa"
DB = "hospital"
class PatientManagement:
    def __init__(self):
        self.total_execution_time = 0.0
        self.conn = None
        try:
            self.conn = mysql.connector.connect(
                host=HOST,
                user=USER,
                password=PASS,
                database=DB
            )
            if self.conn.is_connected():
                print("Database connection successful!")
        except Error as e:
            print(f"Database connection failed: {e}")
            exit(1)

    def __del__(self):
        if self.conn and self.conn.is_connected():
            self.conn.close()

    def increment_time_complexity(self, time_taken):
        self.total_execution_time += time_taken

    def display_time_complexity(self):
        print(f"Total execution time: {self.total_execution_time} seconds.")
```

```
    def add_patient(self, patient_id, first_name, last_name, age, gender, blood_group, contact, cnic, address):
        start_time = time.time()
        query = ("INSERT INTO patients (patient_id, first_name, last_name, age, gender, blood_group, contact, cnic, address) "
                 "VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)")
        values = (patient_id, first_name, last_name, age, gender, blood_group, contact, cnic, address)
        cursor = self.conn.cursor()
        try:
            cursor.execute(query, values)
            self.conn.commit()
            print("Patient added successfully.")
        except Error as e:
            print(f"Failed to add patient: {e}")
        cursor.close()
        end_time = time.time()
        self.increment_time_complexity(end_time - start_time)

    def display_patients(self):
        start_time = time.time()
        query = "SELECT * FROM patients"
        cursor = self.conn.cursor()
        try:
            cursor.execute(query)
            rows = cursor.fetchall()
            print("Patients in the system:")
            for row in rows:
                print(f"ID: {row[0]}, First Name: {row[1]}, Last Name: {row[2]}, Age: {row[3]}, Gender: {row[4]}, Blood Group: {row[5]}, Contact: {row[6]}, CNIC: {row[7]}, Address: {row[8]}")
        except Error as e:
            print(f"Failed to retrieve patients: {e}")
        cursor.close()
        end_time = time.time()
        self.increment_time_complexity(end_time - start_time)

    def search_patient_by_name(self, first_name):
        start_time = time.time()
```

```
        query = "SELECT * FROM patients WHERE first_name = %s"
        cursor = self.conn.cursor()
        try:
            cursor.execute(query, (first_name,))
            rows = cursor.fetchall()
            if rows:
                for row in rows:
                    print(f"Patient found - ID: {row[0]}, First Name: {row[1]}, Last Name: {row[2]}, Age: {row[3]}, Gender: {row[4]}, Blood Group: {row[5]}, Contact: {row[6]}, CNIC: {row[7]}, Address: {row[8]}")
            else:
                print("Patient not found.")
        except Error as e:
            print(f"Failed to search patient: {e}")
        cursor.close()
        end_time = time.time()
        self.increment_time_complexity(end_time - start_time)

    def search_patient_by_id(self, patient_id):
        start_time = time.time()
        query = "SELECT * FROM patients WHERE patient_id = %s"
        cursor = self.conn.cursor()
        try:
            cursor.execute(query, (patient_id,))
            rows = cursor.fetchall()
            if rows:
                for row in rows:
                    print(f"Patient found - ID: {row[0]}, First Name: {row[1]}, Last Name: {row[2]}, Age: {row[3]}, Gender: {row[4]}, Blood Group: {row[5]}, Contact: {row[6]}, CNIC: {row[7]}, Address: {row[8]}")
            else:
                print("Patient not found.")
        except Error as e:
            print(f"Failed to search patient: {e}")
        cursor.close()
        end_time = time.time()
        self.increment_time_complexity(end_time - start_time)
```

```
    def remove_patient(self, patient_id):
        start_time = time.time()
        query = "DELETE FROM patients WHERE patient_id = %s"
        cursor = self.conn.cursor()
        try:
            cursor.execute(query, (patient_id,))
            self.conn.commit()
            print("Patient removed successfully.")
        except Error as e:
            print(f"Failed to remove patient: {e}")
        cursor.close()
        end_time = time.time()
        self.increment_time_complexity(end_time - start_time)


def main():
    pm = PatientManagement()
    while True:
        print("\nPatient Management System Menu:")
        print("1. Add Patient")
        print("2. Display All Patients")
        print("3. Search Patient by Name")
        print("4. Search Patient by ID")
        print("5. Remove Patient")
        print("6. Exit")
        choice = int(input("Enter your choice: "))

        if choice == 1:
            patient_id = input("Enter Patient ID: ")
            first_name = input("Enter First Name: ")
            last_name = input("Enter Last Name: ")
            age = int(input("Enter Age: "))
            gender = input("Enter Gender: ")
            blood_group = input("Enter Blood Group: ")
            contact = input("Enter Contact: ")
            cnic = input("Enter CNIC: ")
```

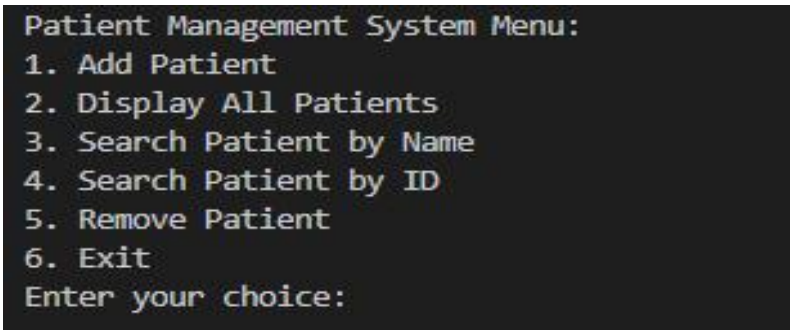
```
            address = input("Enter Address: ")
            pm.add_patient(patient_id, first_name, last_name, age, gender, blood_group, contact, cnic, address)
        elif choice == 2:
            pm.display_patients()
        elif choice == 3:
            first_name = input("Enter First Name to search: ")
            pm.search_patient_by_name(first_name)
        elif choice == 4:
            patient_id = input("Enter Patient ID to search: ")
            pm.search_patient_by_id(patient_id)
        elif choice == 5:
            patient_id = input("Enter Patient ID to remove: ")
            pm.remove_patient(patient_id)
        elif choice == 6:
            print("Exiting program.")
            break
        else:
            print("Invalid choice. Please try again.")

    pm.display_time_complexity()

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

Output:

1. Menu :



2. Add Patient

```
Enter your choice: 1
Enter Patient ID: 000
Enter First Name: ali
Enter Last Name: khan
Enter Age: 23
Enter Gender: male
Enter Blood Group: A
Enter Contact: 239494944
Enter CNIC: 2931233714
Enter Address: XYZ
Patient added successfully.
```

3. Display All Patients

```
Enter your choice: 2
Patients in the system:
ID: 000, First Name: ali, Last Name: khan, Age: 23, Gender: male, Blood Group: A, Contact: 239494944, CNIC: 2931233714, Address: XYZ
ID: 123, First Name: farwa, Last Name: mehdi, Age: 20, Gender: female, Blood Group: b+, Contact: 029392119292, CNIC: 293939292101, Address: ABC
```

4. Search Patient by Name

```
Enter your choice: 3
Enter First Name to search: farwa
Patient found - ID: 123, First Name: farwa, Last Name: mehdi, Age: 20, Gender: female, Blood Group: b+, Contact: 029392119292, CNIC: 29393929210
1, Address: ABC
```

5. Search Patient by ID

```
Enter your choice: 4
Enter Patient ID to search: 123
Patient found - ID: 123, First Name: farwa, Last Name: mehdi, Age: 20, Gender: female, Blood Group: b+, Contact: 029392119292, CNIC: 29393929210
1, Address: ABC
```

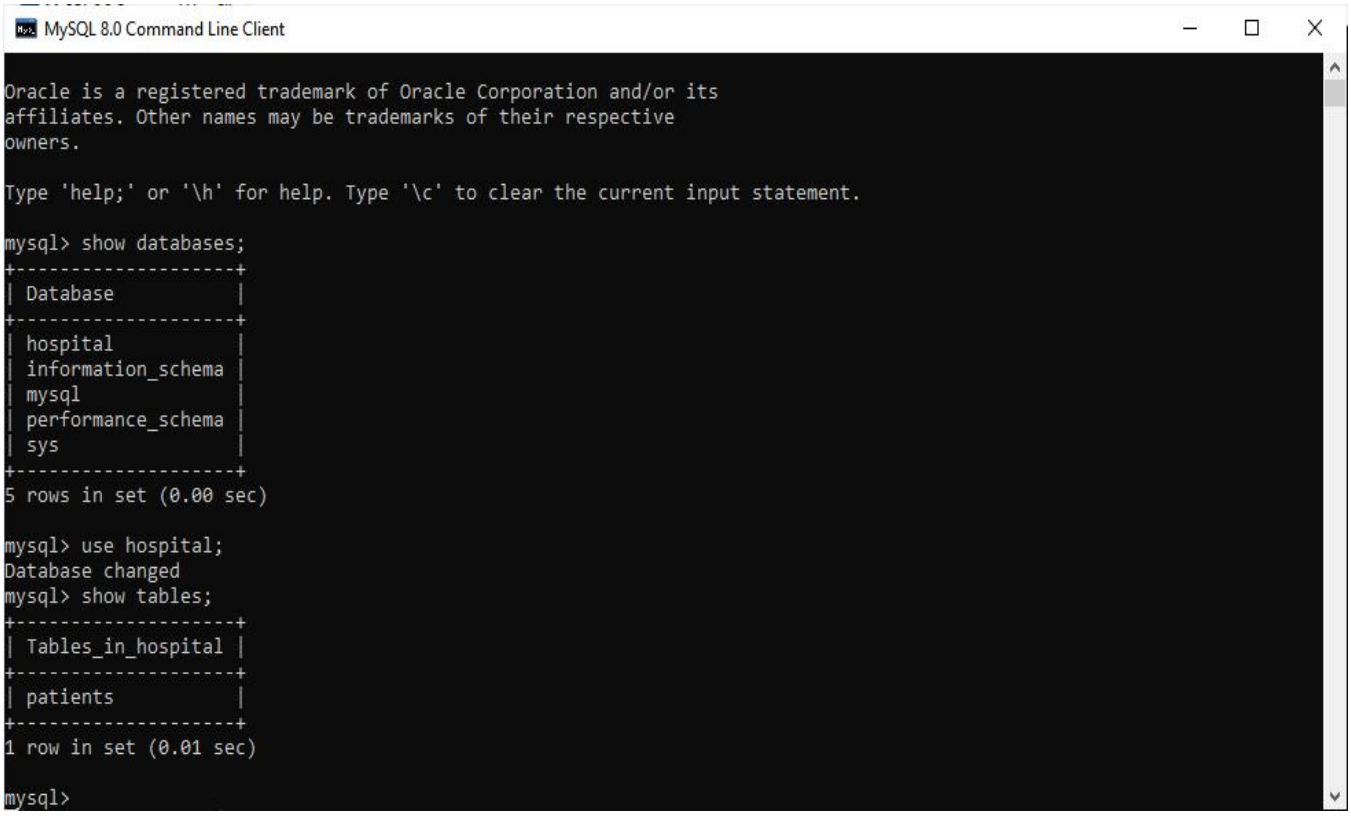
6. Remove Patient

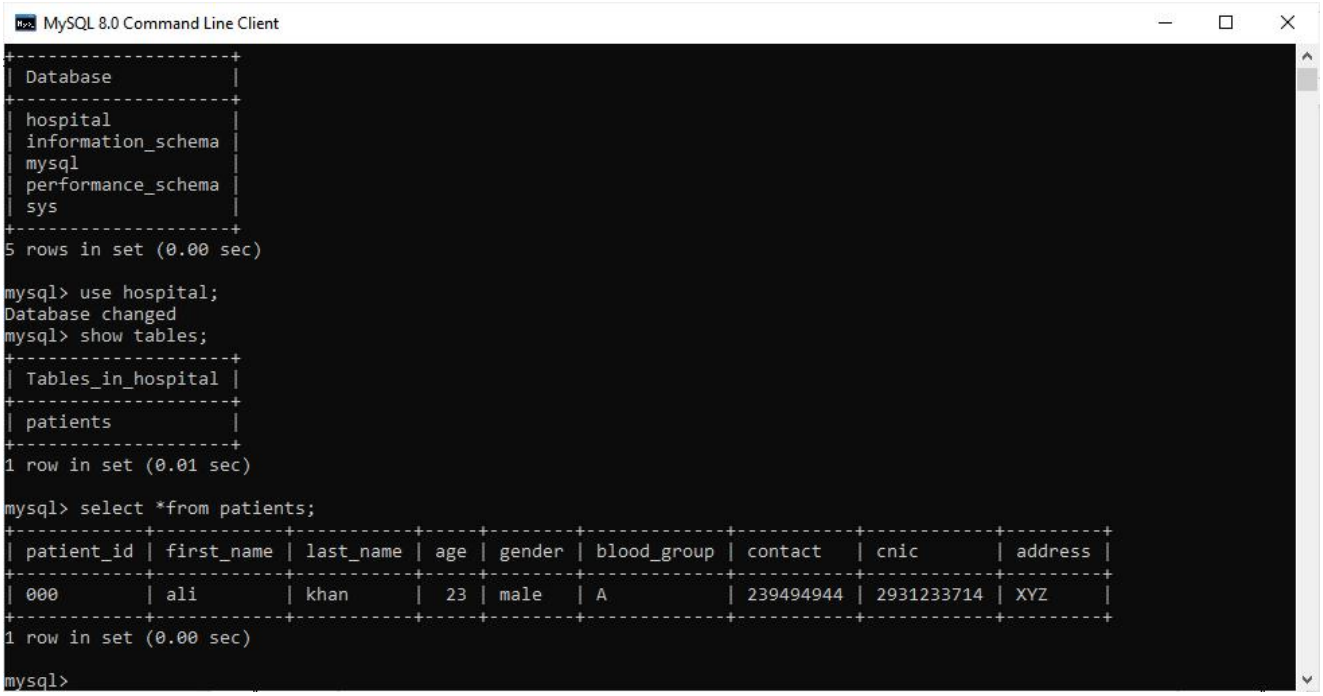
```
Enter your choice: 5
Enter Patient ID to remove: 123
Patient removed successfully.
```

7. Exit

```
Enter your choice: 6
Exiting program.
Total execution time: 0.029170989990234375 seconds.
PS C:\Users\hp\Downloads\Farwa SDA Project\Farwa SDA Project>
```

Added Patient Record in MYSQL:





Time complexity:

```
Patient Management System Menu:
1. Add Patient
2. Display All Patients
3. Search Patient by Name
4. Search Patient by ID
5. Remove Patient
6. Exit
Enter your choice: 6
Exiting program.
Total execution time: 0.029170989990234375 seconds.
PS C:\Users\hp\Downloads\Farwa SDA Project\Farwa SDA Project>
```