2024

Clinic Management System

1 OVERVIEW

The Clinic Management System is a Python-based application designed to help clinics manage patient records, schedule appointments, and handle daily operations efficiently. The system is divided into two modes: Admin Mode and User Mode. The Admin Mode allows authorized personnel to manage patient data and appointments, while the User Mode provides an interface for viewing patient information and reservations.

2 OBJECTIVE

The primary objective of this project is to provide a user-friendly system for managing clinic operations. The system simplifies the process of adding new patients, scheduling appointments, and managing existing records. By automating these tasks, the system reduces the workload on clinic staff and minimizes the chances of errors in scheduling and data entry.

3 TOOLS AND TECHNOLOGIES

- ➢ **Programming Language**: Python
- ➢ **Database: MySQL**
- ➢ **Database Connector**: mysql-connector-python
- ➢ **Python Libraries**:

- mysql-connector-python: For connecting to the MySQL database.
- time and datetime: For handling time and date-related functionalities (if needed).

- ➢ **Development Environment**: VS Code

4 USE CASE

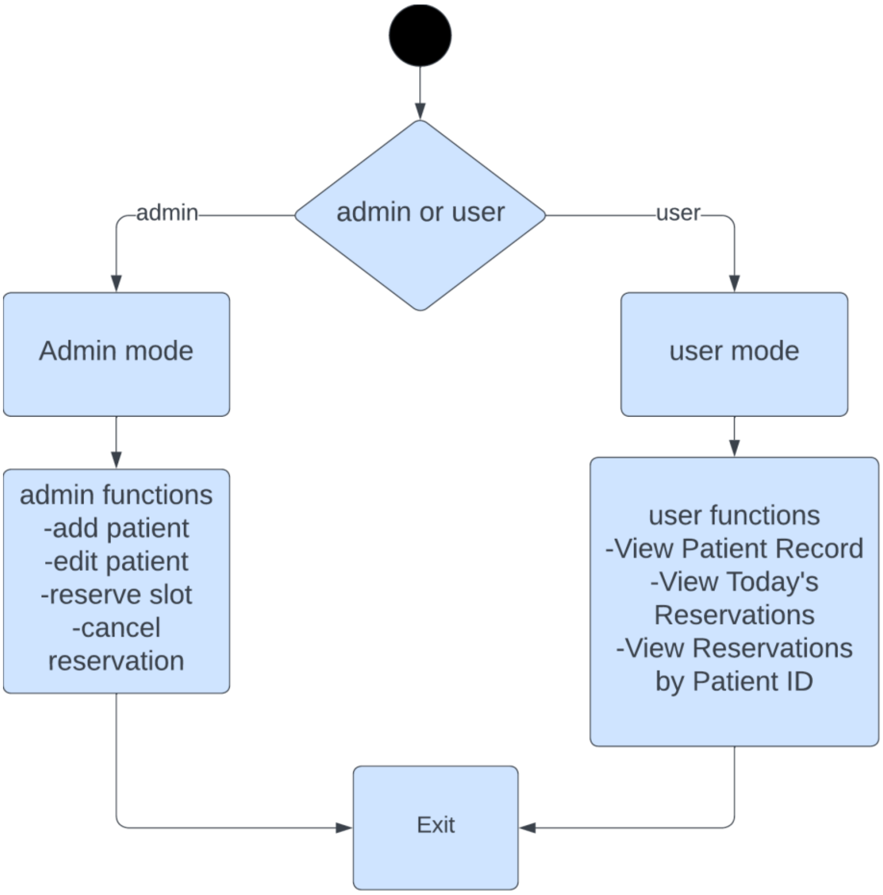
4.1 ADMIN MODE

- **Add Patient**: The admin can add new patient records, including details such as name, age, and gender.
- **Edit Patient**: The admin can edit existing patient records.
- **Reserve Slot**: The admin can reserve appointment slots for patients, ensuring that slots are not double-booked.
- **Cancel Reservation**: The admin can cancel existing reservations.

4.2 USER MODE

- **View Patient Record**: Users can view detailed records of patients by entering their ID.
- **View Today's Reservations:** Users can view all reservations scheduled for the current day.
- **View Reservations by Patient ID**: Users can view all reservations associated with a specific patient.

5 FLOW CHART

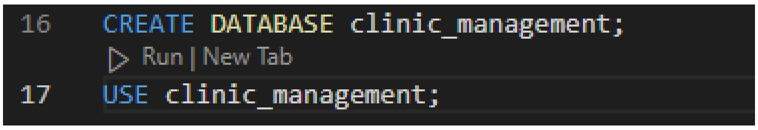


6 SET UP THE DATABASE

6.1 CREATE THE DATABASE:

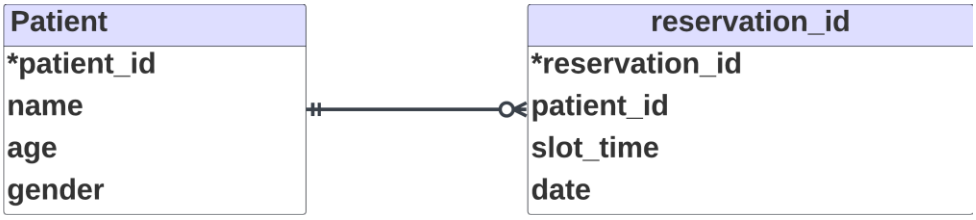
- Log into your MySQL server using the command line or a MySQL client (like MySQL Workbench).

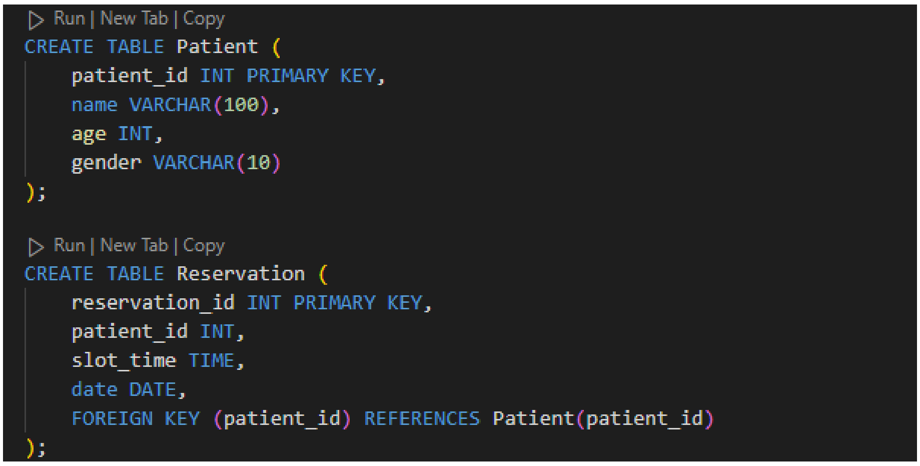
- create a new database for the clinic management system.



```
CREATE DATABASE clinic_management;
USE clinic_management;
```

6.2 CREATE THE TABLES:





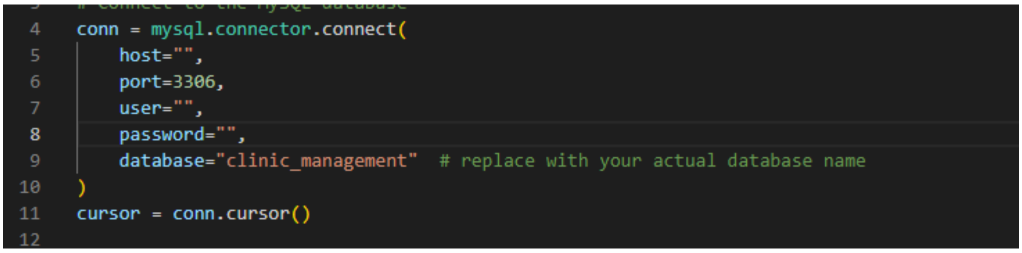
```sql
CREATE TABLE Patient (
    patient_id INT PRIMARY KEY,
    name VARCHAR(100),
    age INT,
    gender VARCHAR(10)
);

CREATE TABLE Reservation (
    reservation_id INT PRIMARY KEY,
    patient_id INT,
    slot_time TIME,
    date DATE,
    FOREIGN KEY (patient_id) REFERENCES Patient(patient_id)
);
```

7 CONFIGURATION

7.1 DATABASE CONNECTION:

Open the functions.py file and configure the database connection by updating the following details with your MySQL credentials:



```python
# connect to the MySQL database
conn = mysql.connector.connect(
    host="",
    port=3306,
    user="",
    password="",
    database="clinic_management"  # replace with your actual database name
)
cursor = conn.cursor()
```

7.2 INSTALL REQUIRED PYTHON PACKAGES:

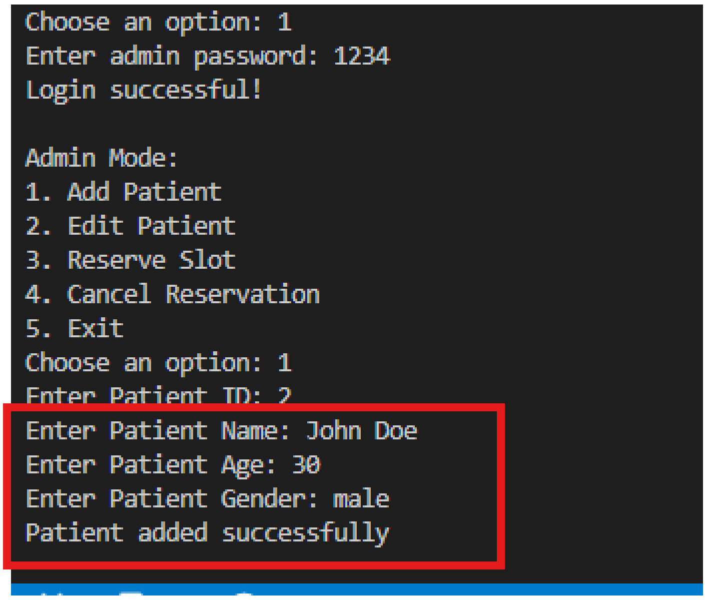
Install the MySQL connector for Python < pip install mysql-connector-python >

8 EXAMPLE SCENARIOS

8.1 ADDING A PATIENT (ADMIN MODE):

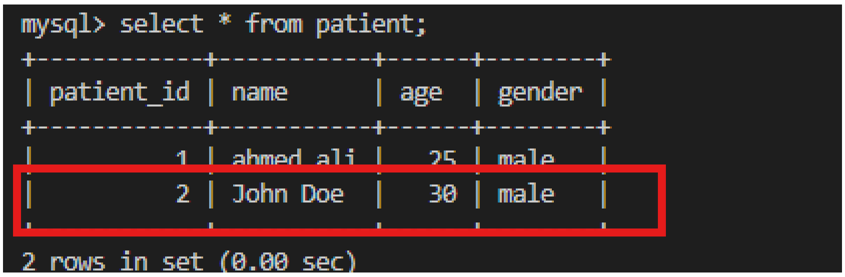
A new patient, John Doe, visits the clinic for the first time.

1. Log into Admin Mode.
2. Select "Add Patient" from the menu.
3. Enter John Doe’s details (e.g., Name: John Doe, Age: 30, Gender: Male).
4. Confirm the addition.



```
Choose an option: 1
Enter admin password: 1234
Login successful!

Admin Mode:
1. Add Patient
2. Edit Patient
3. Reserve Slot
4. Cancel Reservation
5. Exit
Choose an option: 1
Enter Patient ID: 2
Enter Patient Name: John Doe
Enter Patient Age: 30
Enter Patient Gender: male
Patient added successfully
```



```
mysql> select * from patient;
+------------+-----------+------+--------+
| patient_id | name      | age  | gender |
+------------+-----------+------+--------+
|          1 | ahmed ali |   25 | male   |
|          2 | John Doe  |   30 | male   |
+------------+-----------+------+--------+
2 rows in set (0.00 sec)
```

8.2 RESERVING AN APPOINTMENT SLOT (ADMIN MODE):

The admin needs to schedule an appointment for John Doe on 2023-12-01 from 14:00 to 14:30.

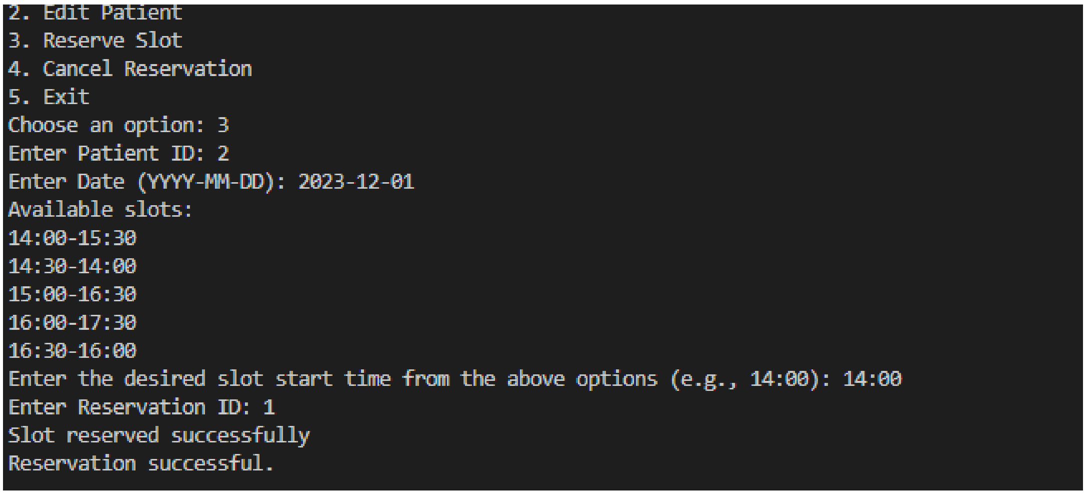
1. Log into Admin Mode.

2. Select "Reserve Slot" from the menu.

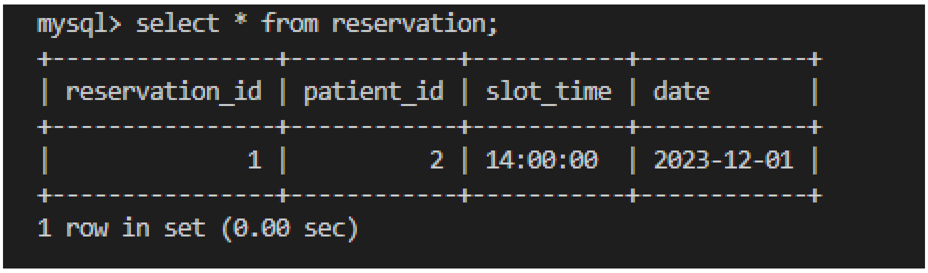
3. Enter John Doe’s patient ID and the desired date.

4. Choose the 14:00-14:30 time slot.

5. Confirm the reservation.



```
2. Edit Patient
3. Reserve Slot
4. Cancel Reservation
5. Exit
Choose an option: 3
Enter Patient ID: 2
Enter Date (YYYY-MM-DD): 2023-12-01
Available slots:
14:00-15:30
14:30-14:00
15:00-16:30
16:00-17:30
16:30-16:00
Enter the desired slot start time from the above options (e.g., 14:00): 14:00
Enter Reservation ID: 1
Slot reserved successfully
Reservation successful.
```



```
mysql> select * from reservation;
+----------------+------------+-----------+------------+
| reservation_id | patient_id | slot_time | date       |
+----------------+------------+-----------+------------+
|              1 |          2 | 14:00:00  | 2023-12-01 |
+----------------+------------+-----------+------------+
1 row in set (0.00 sec)
```