REPORT

PES UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

SUBJECT: DATABASE MANAGEMENT SYSTEM

FACULTY ADVISOR: DR. CHANDRASHEKHAR P CHAVAN

TOPIC: HOSPITAL MANAGEMENT SYSTEM

PRESENTED BY:

PRATYUSH JAISHANKAR- PES2UG21EC107

REPORT

Description:

The Hospital Management System is a comprehensive software solution designed to streamline the administrative and operational processes of a hospital or healthcare facility. The primary aim of this project is to develop a user-friendly system that automates various tasks, including patient management, appointment scheduling, medical record keeping, and staff management, to improve overall efficiency and patient care.

Objectives:

- To create a centralized system for managing patient records, appointments, and medical history.
- To facilitate efficient communication and coordination among healthcare professionals.
- To automate routine administrative tasks, such as billing and inventory management.
- To enhance patient experience by providing easy access to healthcare services and information.
- To ensure data security and confidentiality in compliance with healthcare regulations.

Tables Created:

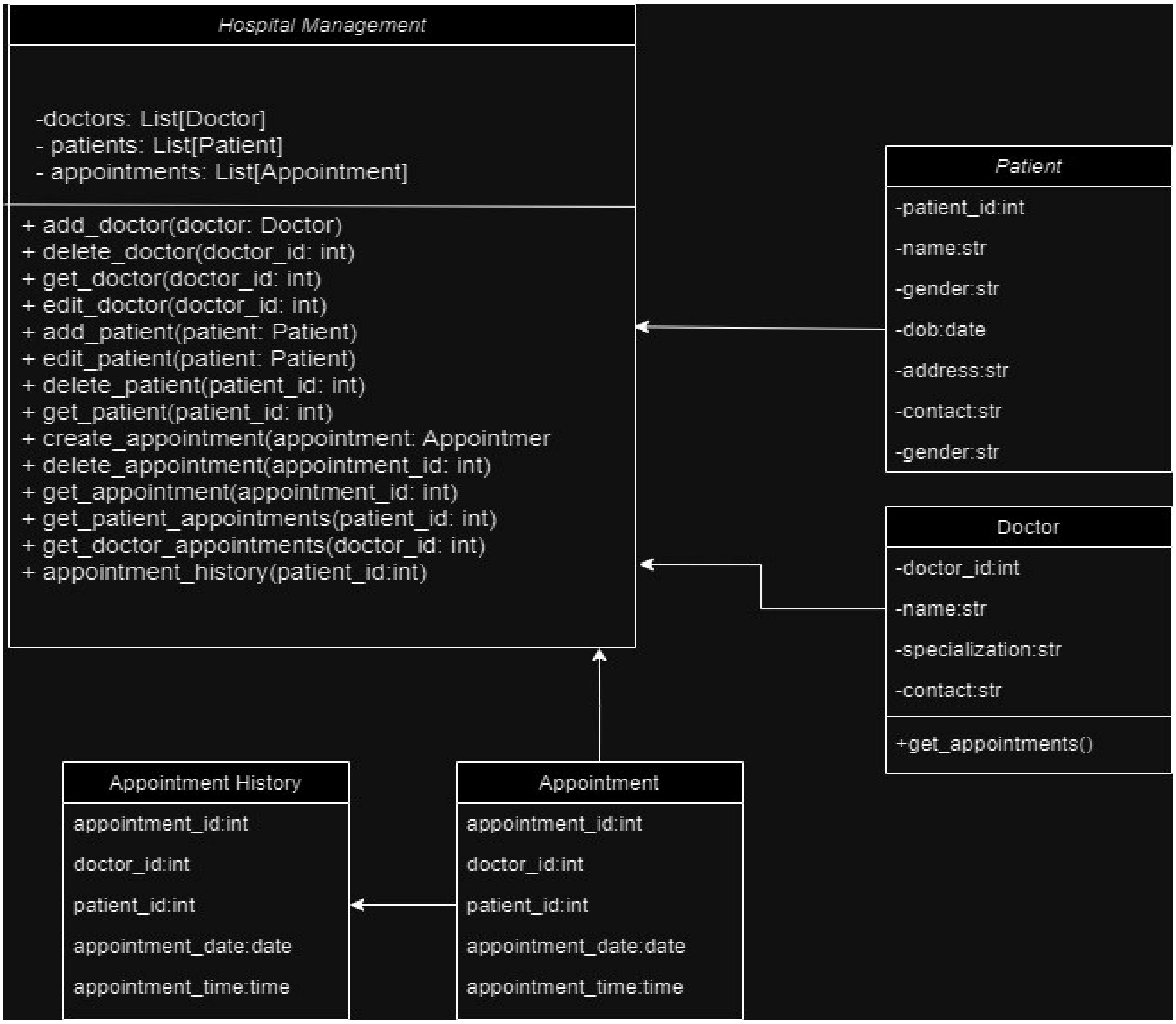
- Doctors: Stores information about doctors including their ID, name, specialization, and contact details.
- Patients: Contains details about patients such as their ID, name, gender, date of birth, address, and contact number.
- Appointments: Records appointments made between doctors and patients, including appointment ID, Doctor ID, patient ID, and appointment date.
- Appointment History: Keeps a history of all appointments made, including appointment ID, patient ID, Doctor ID, and appointment date, even after the appointment is deleted from the current appointments list.

Features:

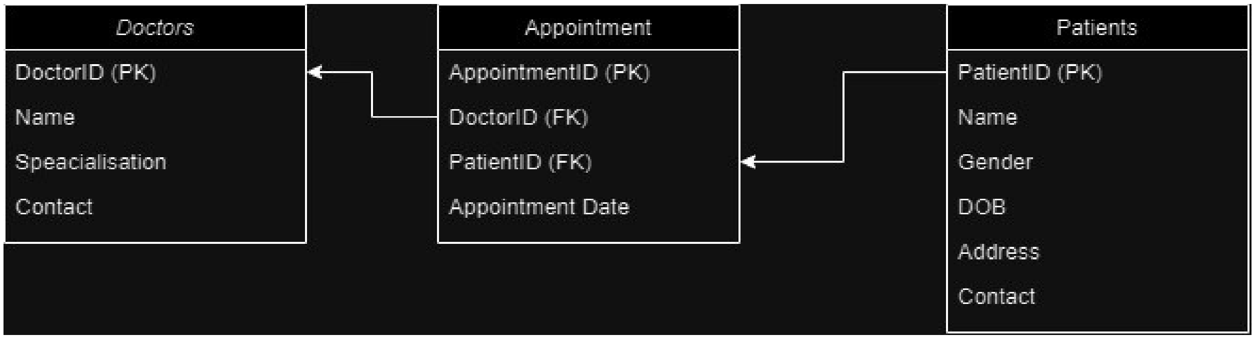
- Frontend: Develop a user-friendly interface using Streamlit for efficient interaction with the Hospital Management System.
- Patient Management: Manage patient records, including registration, admission, and discharge.
- Appointment Scheduling: Schedule appointments for patients with doctors.
- Medical Records Management: Maintain detailed medical histories, diagnoses, treatments, and test results for patients.
- Appointment History: Maintain a history of all appointments made, including patient ID, doctor ID, and appointment date.
- Search Functionality: Search for patients or doctors by ID or name.

- Edit Patient Information: Update patient details such as name, gender, date of birth, address, and phone number.
- Edit Doctor Information: Modify doctor details including name, specialization, and contact information.
- Delete Appointment Record: Allow deletion of appointment records while preserving appointment history.
- View Patient Appointments: Display a list of appointments for a selected patient.
- View Doctor Appointments: Show a list of appointments for a selected doctor.

Class Diagram:


Hospital Management
-doctors: List[Doctor]
- patients: List[Patient]
- appointments: List[Appointment]
+ add_doctor(doctor: Doctor)
+ delete_doctor(doctor_id: int)
+ get_doctor(doctor_id: int)
+ edit_doctor(doctor_id: int)
+ add_patient(patient: Patient)
+ edit_patient(patient: Patient)
+ delete_patient(patient_id: int)
+ get_patient(patient_id: int)
+ create_appointment(appointment: Appointmer
+ delete_appointment(appointment_id: int)
+ get_appointment(appointment_id: int)
+ get_patient_appointments(patient_id: int)
+ get_doctor_appointments(doctor_id: int)
+ appointment_history(patient_id:int)
Patient
-patient_id:int
-name:str
-gender:str
-dob:date
-address:str
-contact:str
-gender:str
Doctor
-doctor_id:int
-name:str
-specialization:str
-contact:str
+get_appointments()
Appointment History
appointment_id:int
doctor_id:int
patient_id:int
appointment_date:date
appointment_time:time
Appointment
appointment_id:int
doctor_id:int
patient_id:int
appointment_date:date
appointment_time:time

ER Diagram:



SQL Code For Creating Tables:

```
CREATE TABLE Patients (
   PatientID INT PRIMARY KEY AUTO_INCREMENT,
   Name VARCHAR(255),
   Gender VARCHAR(10),
   DateOfBirth DATE,
   Address VARCHAR(255),
   PhoneNumber VARCHAR(20) UNIQUE
);
CREATE TABLE Doctors (
   DoctorID INT PRIMARY KEY AUTO_INCREMENT,
   Name VARCHAR(255),
   Specialization VARCHAR(255),
   ContactInformation VARCHAR(255)
);
CREATE TABLE Appointments (
   AppointmentID INT PRIMARY KEY AUTO_INCREMENT,
   PatientID INT,
   DoctorID INT,
   AppointmentDate DATE,
   AppointmentTime TIME,
   FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
   FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID)
);
CREATE TABLE AppointmentHistory (
   AppointmentID INT PRIMARY KEY,
   PatientID INT,
   DoctorID INT,
   AppointmentDate DATE,
   AppointmentTime TIME
```

SQL Code as Python Function for operations:

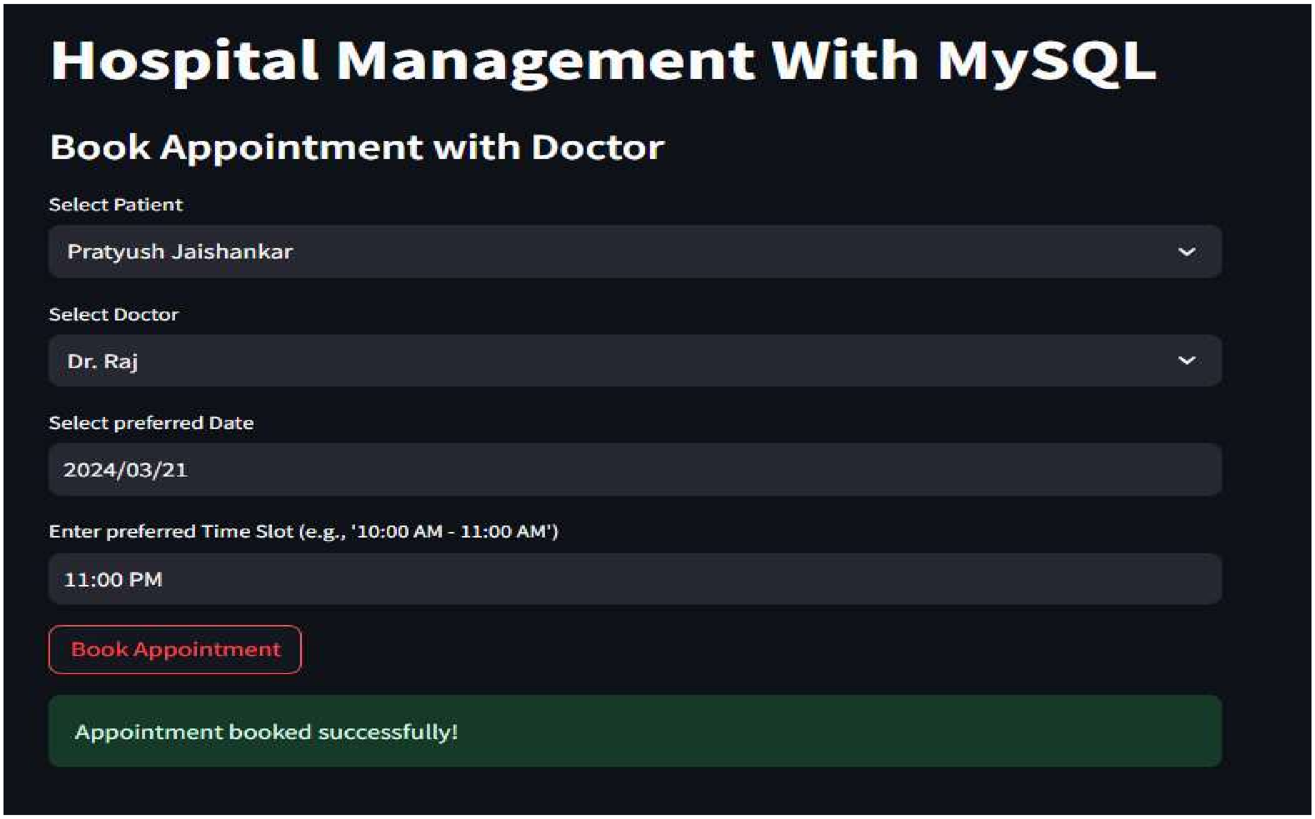
```
import mysql.connector
mydb = mysql.connector.connect(
   host="localhost",
   user="root",
   password="Password@123",
   database="Hospital_management"
)
mycursor=mydb.cursor()
print("Connection Established")

def delete_appointments():
   mycursor.execute

def get_available_doctors():
   mycursor.execute("SELECT * FROM Doctors")
   return mycursor.fetchall()

def get_available_patients():
   mycursor.execute("SELECT * FROM Patients")
```

```
    return mycursor.fetchall()

def get_patient_by_id(patient_id):
    mycursor.execute('SELECT * FROM Patients WHERE patientID=%s', (patient_id,))
    patient = mycursor.fetchone()
    return patient

def get_doctor_by_id(doctor_id):
    mycursor.execute('SELECT * FROM doctors WHERE doctorID=%s', (doctor_id,))
    doctor = mycursor.fetchone()
    return doctor

def get_doctor_by_name(doctor_name):
    mycursor.execute('SELECT * FROM doctors WHERE Name=%s', (doctor_name,))
    doctor = mycursor.fetchall()
    return doctor

def get_patient_by_name(patient_name):
    mycursor.execute('SELECT * FROM patients WHERE Name=%s', (patient_name,))
    patient = mycursor.fetchall()
    return patient

def update_doctor(doctor_id, new_name, new_specialization, new_contact):
    mycursor.execute('''UPDATE Doctors
                      SET name=%s, specialization=%s, contactinformation=%s
                      WHERE doctorid=%s''', (new_name, new_specialization, new_contact, doctor_id))
    mydb.commit()

# Function to update patient information
```

```
def update_patient(patient_id, name, gender, dob, address, phone):
    mycursor.execute('''UPDATE Patients
                    SET name=%s, gender=%s, DateOfBirth=%s, address=%s, phonenumber=%s
                    WHERE Patientid=%s''', (name, gender, dob, address, phone, patient_id))
    mydb.commit()


def book_appointment(patient_id,doctor_id,date_slot, time_slot):
    sql = "INSERT INTO appointments (PatientID, DoctorID, AppointmentDate, AppointmentTime) VALUES (%s, %s,%s, %s)"
    val = (patient_id, doctor_id,date_slot, time_slot)
    mycursor.execute(sql, val)
    mydb.commit()


    history_sql = "INSERT INTO AppointmentHist (AppointmentID, PatientID, DoctorID, AppointmentDate, AppointmentTime) VALUES (LAST_INSERT_ID(), %s, %s, %s, %s)"
    history_val = (patient_id, doctor_id, date_slot, time_slot)
    mycursor.execute(history_sql, history_val)
    mydb.commit()


def get_doctor_appointments(doctor_id):
    mycursor.execute("SELECT AppointmentDate, AppointmentTime FROM Appointments WHERE DoctorID = %s", (doctor_id,))
    return mycursor.fetchall()


def get_appointments_by_doctor(doctor_id):
    mycursor.execute('SELECT * FROM appointments WHERE DoctorID = %s', (doctor_id,))
    appointments = mycursor.fetchall()
    return appointments


# Function to delete appointment by appointment ID
```

```
def delete_appointment(appointment_id):
    mycursor.execute('DELETE FROM appointments WHERE AppointmentID = %s', (appointment_id,))
    mydb.commit()


def get_appointment_history(patient_id):
    sql = "SELECT * FROM AppointmentHist WHERE PatientID = %s"
    val = (patient_id,)
    mycursor.execute(sql, val)
    appointment_history = mycursor.fetchall()
    return appointment_history
```

Output:



Hospital Management With MySQL

Book Appointment with Doctor

Select Patient

Pratyush Jaishankar

Select Doctor

Dr. Raj

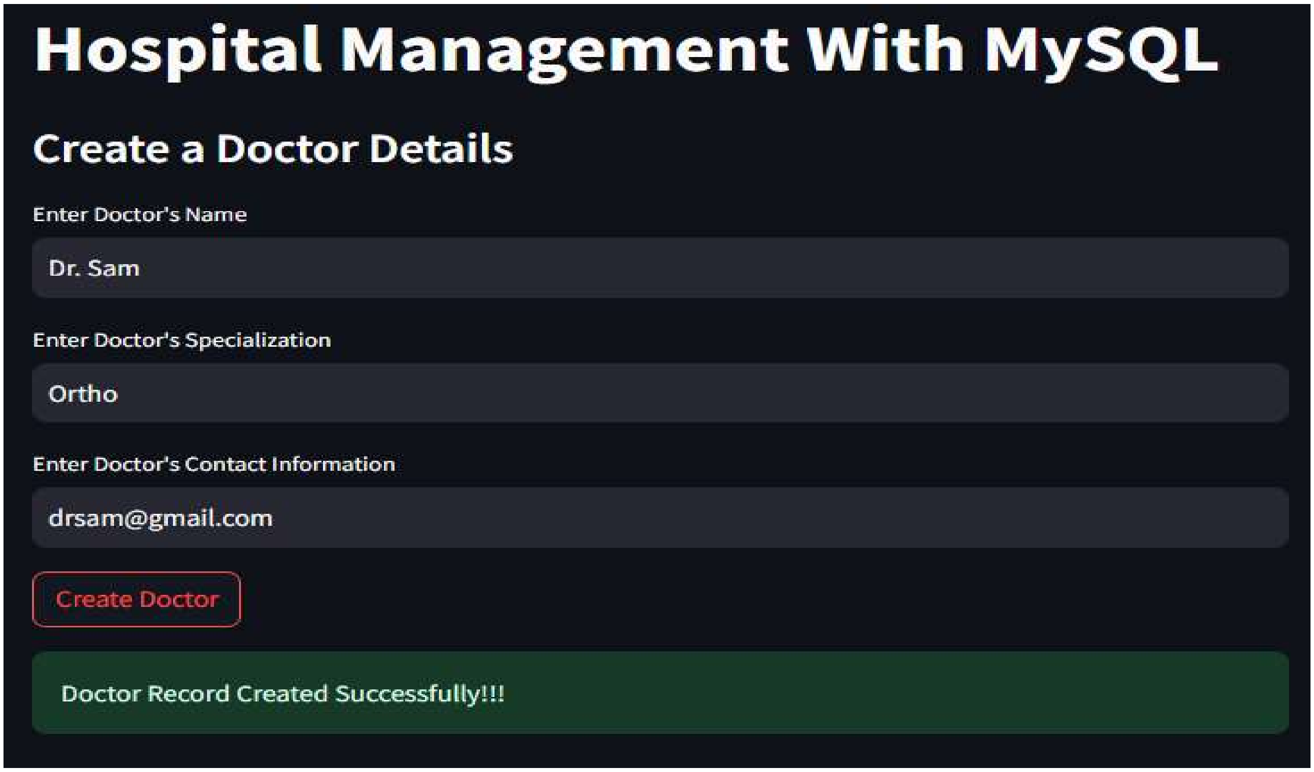
Select preferred Date

2024/03/21

Enter preferred Time Slot (e.g., '10:00 AM - 11:00 AM')

11:00 PM

Appointment booked successfully!



Hospital Management With MySQL

Create a Doctor Details

Enter Doctor's Name

Dr. Sam

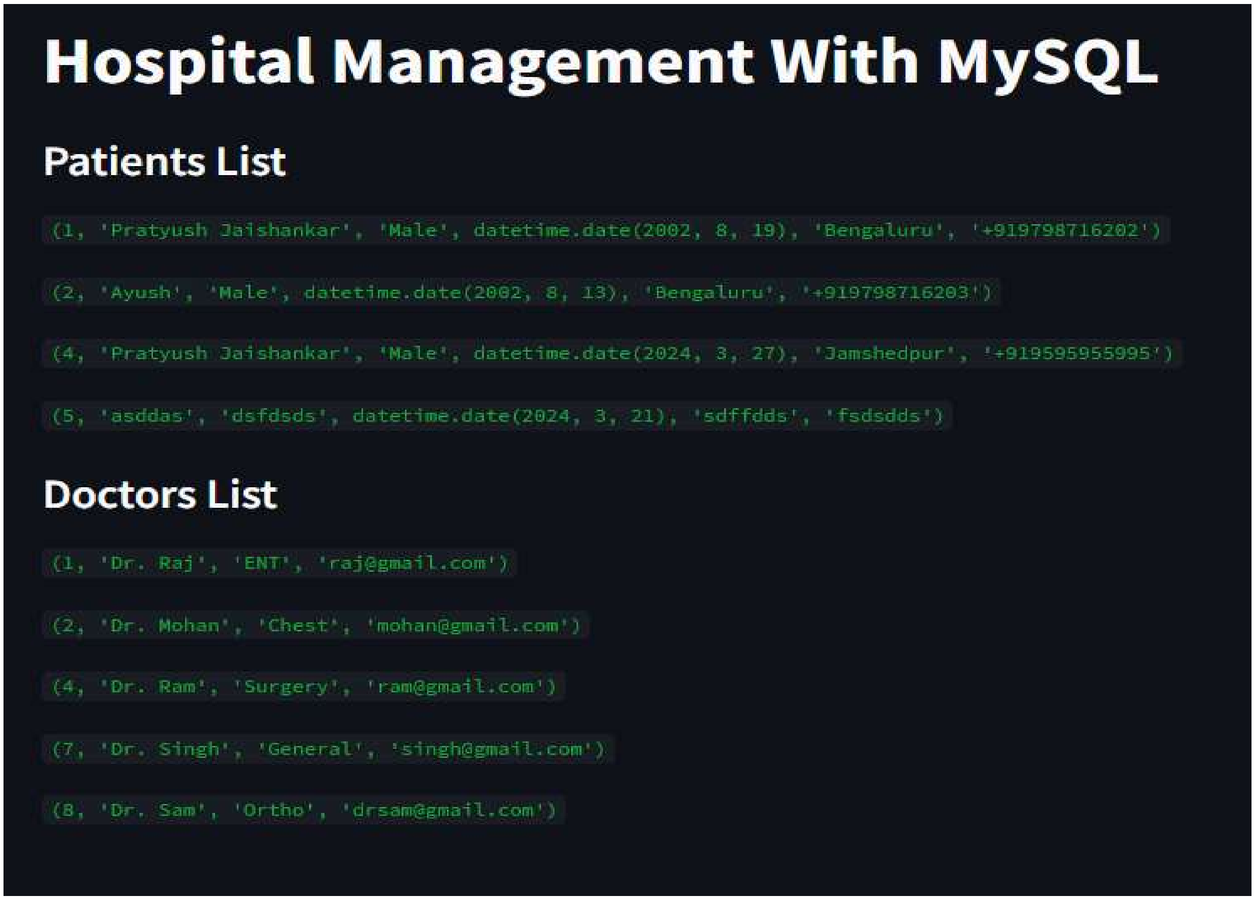
Enter Doctor's Specialization

Ortho

Enter Doctor's Contact Information

drsam@gmail.com

Doctor Record Created Successfully!!!



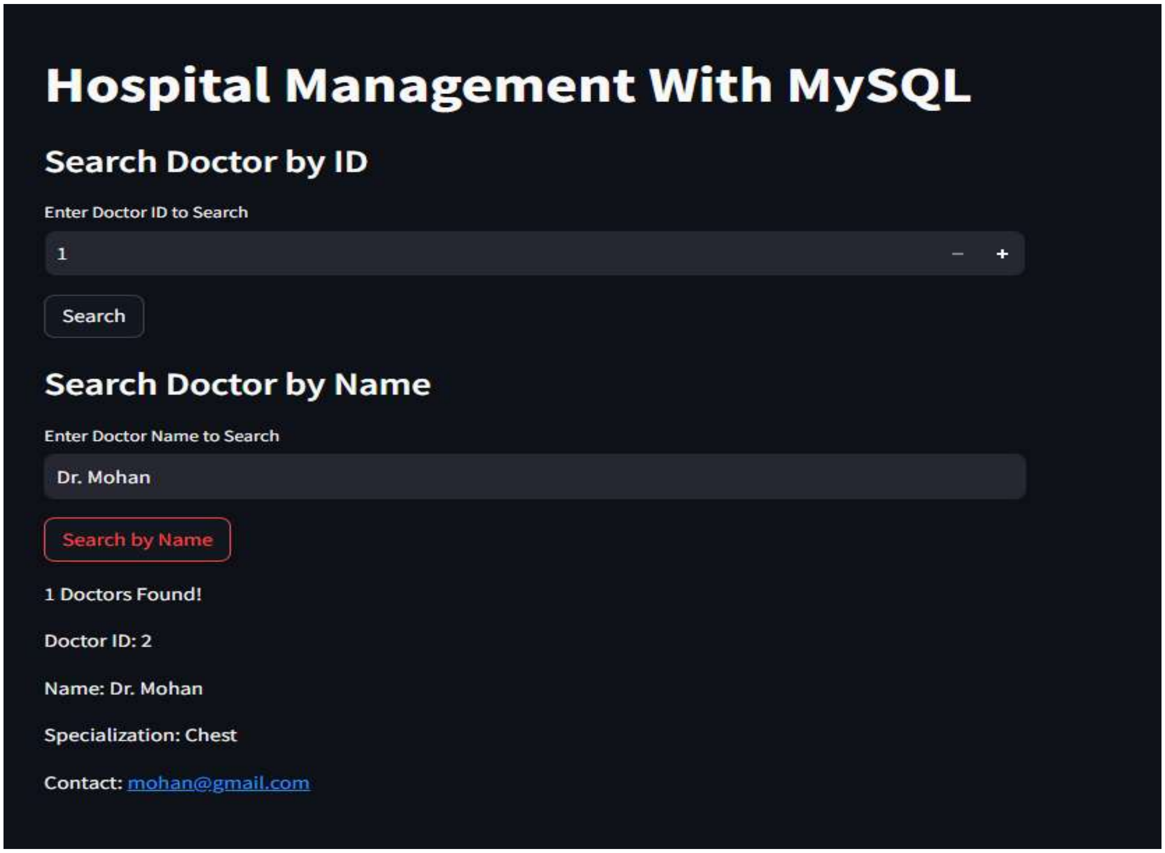
Hospital Management With MySQL

Patients List

```
(1, 'Pratyush Jaishankar', 'Male', datetime.date(2002, 8, 19), 'Bengaluru', '+919798716202')

(2, 'Ayush', 'Male', datetime.date(2002, 8, 13), 'Bengaluru', '+919798716203')

(4, 'Pratyush Jaishankar', 'Male', datetime.date(2024, 3, 27), 'Jamshedpur', '+919595955995')

(5, 'asddas', 'dsfdsds', datetime.date(2024, 3, 21), 'sdffdds', 'fsdsdds')
```

Doctors List

```
(1, 'Dr. Raj', 'ENT', 'raj@gmail.com')

(2, 'Dr. Mohan', 'Chest', 'mohan@gmail.com')

(4, 'Dr. Ram', 'Surgery', 'ram@gmail.com')

(7, 'Dr. Singh', 'General', 'singh@gmail.com')

(8, 'Dr. Sam', 'Ortho', 'drsam@gmail.com')
```



Hospital Management With MySQL

Search Doctor by ID

Enter Doctor ID to Search: 1

Search Doctor by Name

Enter Doctor Name to Search: Dr. Mohan

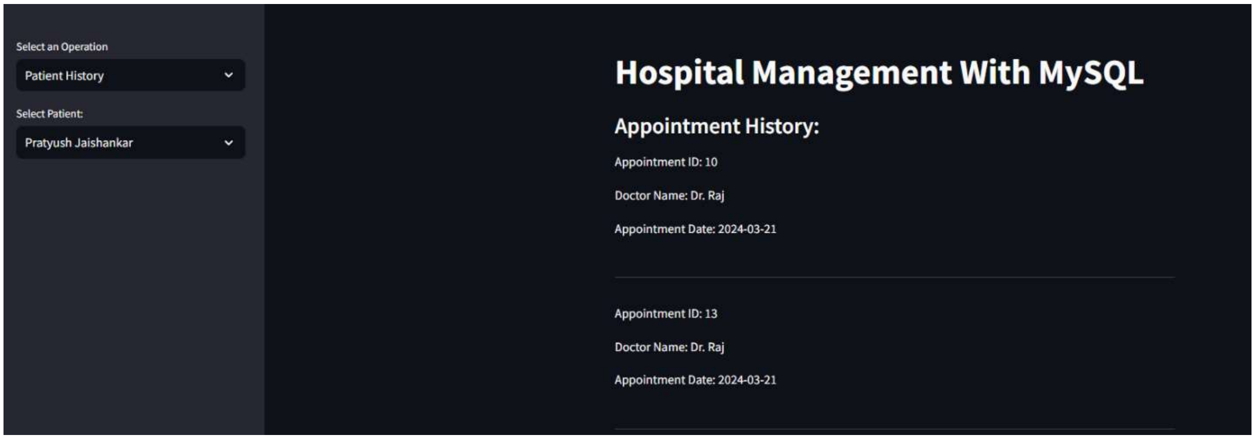
1 Doctors Found!

Doctor ID: 2

Name: Dr. Mohan

Specialization: Chest

Contact: mohan@gmail.com



Select an Operation: Patient History

Select Patient: Pratyush Jaishankar

Hospital Management With MySQL

Appointment History:

Appointment ID: 10

Doctor Name: Dr. Raj

Appointment Date: 2024-03-21

Appointment ID: 13

Doctor Name: Dr. Raj

Appointment Date: 2024-03-21