PYTHON CODING CHALLENGE

HOSPITAL MANAGEMENT SYSTEM

-Muskan Saxena (PGET)

Entity Package

patient.py

```
from util.DBConnUtil import DBConnection

class Patient(DBConnection):
    def __init__(self):
        super().__init__()
        self.patientid=0
        self.firstname=' '
        self.lastname=' '
        self.dateofbirth=' '
        self.gender=' '
        self.contactnumber=0
        self.address=' '
    #Setters

    def set_patientid(self,value):
        self.patientid=value

    def set_firstname(self, value):
        self.firstname = value

    def set_lastname(self, value):
        self.lastname = value

    def set_dateofbirth(self, value):
        self.dateofbirth = value

    def set_gender(self, value):
        self.gender = value

    def set_contactnumber(self, value):
        self.contactnumber = value

    def set_address(self, value):
        self.address = value

    #Getters

    def get_patientid(self):
        return self.patientid
```

```
    def get_firstname(self):
        return self.firstname

    def get_lastname(self):
        return self.lastname

    def get_dateofbirth(self):
        return self.dateofbirth

    def get_gender(self):
        return self.gender

    def get_contactnumber(self):
        return self.contactnumber

    def get_address(self):
        return self.address

    def str(self):
        return f'Patient ID:{self.patientid} First Name:{self.firstname}
Last Name:{self.lastname}\n ' \
               f'Date Of Birth:{self.dateofbirth} Gender:{self.gender}
Contact Number:{self.contactnumber} Address:{self.address}'
```

doctor.py

```
class Doctor(DBConnection):
    def __init__(self):
        super().__init__()
        self.doctorid=0
        self.firstname=' '
        self.lastname=' '
        self.specialization=' '
        self.contactnumber=0


    #Setters

    def set_doctorid(self,value):
        self.doctorid=value

    def set_firstname(self, value):
        self.firstname = value

    def set_lastname(self, value):
        self.lastname = value

    def set_specialization(self, value):
        self.specialization = value

    def set_contactnumber(self, value):
        self.contactnumber = value

    #Getters

    def get_doctoridid(self):
        return self.patientid
```

```
    def get_firstname(self):
        return self.firstname

    def get_lastname(self):
        return self.lastname

    def get_specialization(self):
        return self.specialization

    def get_contactnumber(self):
        return self.contactnumber

    def str(self):
        return f'Doctor ID:{self.patientid} First Name:{self.firstname}
Last Name:{self.lastname}\n ' \
               f'Specialization:{self.specialization} Contact
Number:{self.contactnumber}'
```

appointment.py

```
from entity.patient import Patient
from entity.doctor import Doctor

class Appointment(Patient, Doctor):
    def __init__(self):
        super().__init__()
        self.apppointmentid = 0
        self.patientid = 0
        self.doctorid = 0
        self.appointmentdate = ' '
        self.description = 0

    # Setters

    def set_apppointmentid (self, value):
        self.apppointmentid  = value

    def set_patientid(self, value):
        self.patientid = value

    def set_doctorid (self, value):
        self.doctorid  = value

    def set_appointmentdate(self, value):
        self.appointmentdate = value

    def set_description(self, value):
        self.description = value

    # Getters

    def get_apppointmentid (self):
        return self.apppointmentid

    def get_patientid(self):
        return self.patientid
```

```
    def get_doctorid(self):
        return self.doctorid

    def get_appointmentdate(self):
        return self.appointmentdate

    def get_description(self):
        return self.description

    def str(self):
        return f'Appointment ID:{self.apppointmentid} Patient
ID:{self.patientid} Doctor ID:{self.doctorid}\n ' \
               f'Appointment Date:{self.appointmentdate}
Description:{self.description}'
```

DAO Package

patientdao.py

```
from entity.patient import Patient

class PatientDao(Patient):
    def __init__(self):
        super().__init__()

    def perform_patient_actions(self):
        while True:
            print("(Patient) 1.CREATE 2.INSERT 3.UPDATE 4.DELETE 5.SELECT
0.EXIT")
            ch = int(input("Enter choice: "))
            if ch == 1:
                self.create_patient_table()
            elif ch == 2:
                print(self.add_patient())
            elif ch == 3:
                print(self.update_patient())
            elif ch == 4:
                print(self.delete_patient())
            elif ch == 5:
                self.select_patient()
            elif ch == 0:
                break
            else:
                print("Invalid choice")

    def create_patient_table(self):
        try:
            create_str = '''CREATE TABLE IF NOT EXISTS Patient (
                patientid INT PRIMARY KEY,
                firstname VARCHAR(255),
                lastname VARCHAR(255),
                dateofbirth DATE,
                gender VARCHAR(10),
```

```
                contactnumber VARCHAR(15),
                address VARCHAR(255))'''
            self.open()
            self.stmt.execute(create_str)
            self.close()
            print('Patient Table Created successfully.')
        except Exception as e:
            print(f"Error creating patient table: {e}")
    def add_patient(self):
        try:
            self.open()
            self.patientid = int(input('Enter Patient ID: '))
            self.firstname = input('Enter First Name: ')
            self.lastname = input('Enter Last Name: ')
            self.dateofbirth = input('Enter DOB: ')
            self.gender = input('Enter Gender: ')
            self.contactnumber = input('Enter Number: ')
            self.address = input('Enter address: ')

            data = [(self.patientid, self.firstname, self.lastname,
self.dateofbirth, self.gender, self.contactnumber,self.address)]
            insert_str = '''INSERT INTO Patient(patientid,
firstname,lastname, dateofbirth, gender, contactnumber, address)
                            VALUES(%s, %s, %s, %s, %s, %s, %s)'''
            self.stmt.executemany(insert_str, data)
            self.conn.commit()
            self.close()
            return True
        except Exception as e:
            return f"Error adding patient: {e}"


    def update_patient(self):
        try:
            self.open()
            patientid = int(input('Input Patient ID to be Updated: '))
            self.firstname = input('Enter First Name: ')
            self.lasnName = input('Enter Last Name: ')
            self.dateofbirth = input('Enter DOB: ')
            self.gender = input('Enter Gender: ')
            self.contactnumber = input('Enter Phone Number: ')
            self.address = input('Enter Address: ')

            data = [(self.firstname, self.lastname, self.dateofbirth,
self.gender, self.contactnumber, self.address,patientid)]
            update_str = '''UPDATE Patient SET firstname=%s, lastname=%s,
dateofbirth=%s, gender=%s,contactnumber=%s, address=%s
                            WHERE patientid = %s'''
            self.stmt.executemany(update_str, data)
            self.conn.commit()
            self.close()
            return True
        except Exception as e:
            return f"Error updating patient: {e}"


    def delete_patient(self):
        try:
            self.open()
            patientid = int(input('Input PatientID to be Deleted: '))
            delete_str = f'''DELETE FROM Patient WHERE patientId =
```

```
{patientid}'''
            self.stmt.execute(delete_str)
            self.conn.commit()
            self.close()
            return True
        except Exception as e:
            return f"Error deleting patient: {e}"

    def select_patient(self):
        try:
            select_str = '''SELECT * FROM Patient'''
            self.open()
            self.stmt.execute(select_str)
            records = self.stmt.fetchall()
            self.close()
            print('Records In Patient Table:')
            for i in records:
                print(i)
        except Exception as e:
            print(f"Error selecting Patient: {e}")
```

doctordao.py

```
from entity.doctor import Doctor

class DoctorDao(Doctor):
    def __init__(self):
        super().__init__()

    def perform_doctor_actions(self):
        while True:
            print("(Doctor) 1.CREATE 2.INSERT 3.UPDATE 4.DELETE 5.SELECT
0.EXIT")
            ch = int(input("Enter choice: "))
            if ch == 1:
                self.create_doctor_table()
            elif ch == 2:
                print(self.add_doctor())
            elif ch == 3:
                print(self.update_doctor())
            elif ch == 4:
                print(self.delete_doctor())
            elif ch == 5:
                self.select_doctor()
            elif ch == 0:
                break
            else:
                print("Invalid choice")

    def create_doctor_table(self):
        try:
            create_str = '''CREATE TABLE IF NOT EXISTS Doctor (
                doctorid INT  PRIMARY KEY,
                firstname VARCHAR(255),
                lastname VARCHAR(255),
                specialization VARCHAR(255),
                contactnumber VARCHAR(15))'''
            self.open()
```

```
            self.stmt.execute(create_str)
            self.close()
            print('Doctor Table Created successfully.')
        except Exception as e:
            print(f"Error creating Doctor table: {e}")



    def add_doctor(self):
        try:
            self.open()
            self.doctorid = int(input('Enter Doctor ID: '))
            self.firstname = input('Enter First Name: ')
            self.lastname = input('Enter Last Name: ')
            self.specialization = input('Specialization: ')
            self.contactnumber = input('Enter Number: ')


            data = [(self.doctorid, self.firstname, self.lastname,
self.specialization, self.contactnumber)]
            insert_str = '''INSERT INTO Doctor(doctorid,
firstname,lastname, specialization, contactnumber)
                            VALUES(%s, %s, %s, %s, %s)'''
            self.stmt.executemany(insert_str, data)
            self.conn.commit()
            self.close()
            return True
        except Exception as e:
            return f"Error adding doctor: {e}"

    def update_doctor(self):
        try:
            self.open()
            doctorid = int(input('Input Doctor ID to be Updated: '))
            self.firstname = input('Enter First Name: ')
            self.lastname = input('Enter Last Name: ')
            self.specialization = input('Enter Specialization: ')
            self.contactnumber = input('Enter Phone Number: ')

            data = [(self.firstname, self.lastname, self.specialization,
self.contactnumber, doctorid)]
            update_str = '''UPDATE Doctor SET firstName=%s, lastName=%s,
specialization=%s, contactnumber=%s
                            WHERE doctorid = %s'''
            self.stmt.executemany(update_str, data)
            self.conn.commit()
            self.close()
            return True
        except Exception as e:
            return f"Error updating patient: {e}"

    def delete_doctor(self):
        try:
            self.open()
            doctorid = int(input('Input Doctor ID to be Deleted: '))
            delete_str = f'''DELETE FROM Doctor WHERE doctorid =
{doctorid}'''
            self.stmt.execute(delete_str)
            self.conn.commit()
            self.close()
            return True
```

```
        except Exception as e:
            return f"Error deleting doctor: {e}"

    def select_doctor(self):
        try:
            self.open()
            select_str = '''SELECT * FROM Doctor'''
            self.stmt.execute(select_str)
            records= self.stmt.fetchall()
            self.close()
            print('Records In Doctor Table:')
            for i in records:
                print(i)

        except Exception as e:
            print(f"Error selecting doctor: {e}")
```

appointmentdao.py

```
from entity.appointment import Appointment

class AppointmentDao(Appointment):
    def __init__(self):
        super().__init__()

    def perform_appointment_actions(self):
        while True:
            print("(Appointment) 1.CREATE 2.INSERT 3.UPDATE 4.DELETE
5.SELECT 0.EXIT")
            ch = int(input("Enter choice: "))
            if ch == 1:
                self.create_appointment_table()
            elif ch == 2:
                print(self.add_appointment())
            elif ch == 3:
                print(self.update_appointment())
            elif ch == 4:
                print(self.delete_appointment())
            elif ch == 5:
                self.select_appointment()
            elif ch == 0:
                break
            else:
                print("Invalid choice")

    def create_appointment_table(self):
        try:
            create_str = '''CREATE TABLE IF NOT EXISTS Appointment (
                appointmentid INT  PRIMARY KEY,
                patientid INT,
                doctorid INT,
                appointmentdate DATE,
                description VARCHAR(255),
                FOREIGN KEY(patientid) REFERENCES Patient(patientid) ON
DELETE CASCADE ON UPDATE CASCADE,
                FOREIGN KEY(doctorid) REFERENCES Doctor(doctorid) ON DELETE
CASCADE ON UPDATE CASCADE)'''
            self.open()
```

```
            self.stmt.execute(create_str)
            self.close()
            print('Appointment Table Created successfully.')
        except Exception as e:
            print(f"Error creating Appointment table: {e}")

    def add_appointment(self):
        try:
            self.open()
            self.appointmentid = int(input('Enter Appointment ID: '))
            self.patientid = int(input('Enter Patient ID: '))
            self.doctorid = int(input('Enter Doctor ID: '))
            self.appointmentdate = input('Enter Appointment Date: ')
            self.description= input('Enter Description: ')

            data = [(self.appointmentid, self.patientid, self.doctorid,
self.appointmentdate, self.description)]
            insert_str = '''INSERT INTO Appointment(appointmentid,
patientid, doctorid,appointmentdate,description)
                            VALUES(%s, %s, %s, %s, %s)'''
            self.stmt.executemany(insert_str, data)
            self.conn.commit()
            self.close()
            return True
        except Exception as e:
            return f"Error adding Appointment: {e}"

    def update_appointment(self):
        try:
            self.open()
            appointmentid = int(input('Input Appointment ID to be Updated:
'))
            self.patientid = int(input('Enter Patient ID: '))
            self.doctorid = int(input('Enter Doctor ID: '))
            self.appointmentdate = input('Enter Appointment Date: ')
            self.description = input('EnterDescription: ')

            data = [(self.patientid, self.doctorid, self.appointmentdate,
self.description, appointmentid)]
            update_str = '''UPDATE Appointment SET patientid=%s,
doctorid=%s, appointmentdate=%s, description=%s
                            WHERE appointmentid = %s'''
            self.stmt.executemany(update_str, data)
            self.conn.commit()
            self.close()
            return True
        except Exception as e:
            return f"Error updating Appointment: {e}"

    def delete_appointment(self):
        try:
            self.open()
            appointmentid = int(input('Input Appointment ID to be Deleted:
'))
            delete_str = f'''DELETE FROM Appointment WHERE appointmentid =
{appointmentid}'''
            self.stmt.execute(delete_str)
            self.conn.commit()
            self.close()
            return True
        except Exception as e:
```

```
            return f"Error deleting Appointment: {e}"

    def select_appointment(self):
        try:
            select_str = '''SELECT * FROM Appointment'''
            self.open()
            self.stmt.execute(select_str)
            records = self.stmt.fetchall()
            self.close()
            print('Records In Appointment Table:')
            for i in records:
                print(i)
        except Exception as e:
            print(f"Error selecting Appointment: {e}")
```

ihospitalservice.py

```
from dao.appointmentdao import AppointmentDao
from dao.patientdao import PatientDao
from exception.patientnumbernotfound import PatientNumberNotFound

class  HospitalServiceImpl(AppointmentDao,PatientDao):

#Get Appointment by ID
    def getAppointmentById(self, appointmentid):
       print('Enter Appointment ID to get Information: ')
       try:
          self.open()
          select_str=f'''SELECT * FROM Appointment WHERE
appointmentid={appointmentid}'''
          self.stmt.execute(select_str)
          records = self.stmt.fetchall()
          self.close()
          return records

       except Exception as e:
          print(e)

# Get Appointment for Patient
    def getAppointmentsForPatient(self, patientid):

       print('Enter Patient ID to get Information: ')
       try:
          self.open()
          select_str = f'''SELECT * FROM Appointment WHERE
patientid={patientid}'''
          self.stmt.execute(select_str)
          records = self.stmt.fetchall()
          self.close()
          return records
       except PatientNumberNotFound as e:
          return e
       except Exception as e:
          print(e)

#Get Appointment For Doctor
    def getAppointmentsForDoctor(self,doctorid):
```

```
        print('Enter Appointment ID to get Information: ')
        try:
            self.open()
            select_str = f'''SELECT * FROM Appointment WHERE
doctorid={doctorid}'''
            self.stmt.execute(select_str)
            records = self.stmt.fetchall()
            self.close()
            return records

        except Exception as e:
            print(e)

    def cancelAppointment(self,appointmentid):
        print('Enter Appointment ID to cancel appointment: ')
        try:
            self.open()
            select_str = f'''DELETE  FROM  Appointment WHERE
appointmentid={appointmentid}'''
            self.stmt.execute(select_str)
            self.close()
            return True

        except Exception as e:
            print(e)
            return False
```

Exception Package

patientnumbernotfound.py

```
class PatientNumberNotFound(Exception):
    def __init__(self):
        super().__init__(f'Patient ID not found')
```

Main Package

main.py

```
from util.DBConnUtil import DBConnection

def main():

    dbconnection = DBConnection()

    try:
        dbconnection.open()
        print("--Database Is Connected:--")
    except Exception as e:
        print(e)
```

```
    try:
        print("=" * 30)
        print("Hospital Management System")
        print("=" * 30)
        print("Welcome to Hospital Management System!")

        hospital_management_system = HospitalServiceImpl()

        while True:
            print("1.Patient 2.Doctor 3.Appointment 0.EXIT")
            ch = int(input("Enter choice: "))
            if ch == 1:
                p = PatientDao()
                p.perform_patient_actions()
            elif ch == 2:
                d = DoctorDao()
                d.perform_doctor_actions()
            elif ch == 3:
                a = AppointmentDao()
                a.perform_appointment_actions()
            elif ch == 0:
                break
            else:
                print("Invalid choice")

        hospital_management_system=HospitalServiceImpl()

        while True:
            print("=" * 10)
            print("---MENU---")
            print("=" * 10)
            print("1.Get Appointment By ID\n2.Get Appointment For
Patient\n3.Get Appointment For Doctor\n4.Cancel Appointment\n0.EXIT")
            ch = int(input("Enter choice: "))
            if ch == 1:

print(hospital_management_system.getAppointmentById(int(input('Enter
Appointment ID to see the Appointment'))))
            elif ch == 2:

print(hospital_management_system.getAppointmentsForPatient(int(input('Enter
Patient ID of the Patient to see the Appointment: '))))
            elif ch == 3:

print(hospital_management_system.getAppointmentsForDoctor(int(input('Enter
Doctor ID of the Doctor to see the Appointment: '))))
            elif ch == 4:

print(hospital_management_system.cancelAppointment(int(input('Enter
Appointment ID to to cancel the Appointment: '))))
            elif ch == 0:
                break
            else:
                print("Invalid choice")

    except PatientNumberNotFound as e:
        print(e)

    finally:
        dbconnection.close()
        print("Thankyou for visiting Hospital Management System!")
```

```
        print("--Connection Is Closed:--")


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

util Package

DBConnUtil.py

```
class DBConnection:
    def open(self):
        try:
            connection_properties=PropertyUtil.getConnectionString()
            self.conn=sql.connect(**connection_properties)
            self.stmt=self.conn.cursor()
        except Exception as e:
            print(str(e) + '--Database Is Not Connected:--')
            sys.exit(1)

    def close(self):
        self.conn.close()
```

DBPropertyUtil.py

```
class PropertyUtil:
    connection_properties= None

    @staticmethod
    def getConnectionString():
        if PropertyUtil.connection_properties is None:
            host='localhost'
            database='hospital_db'
            user='root'
            password='Muskan20'

PropertyUtil.connection_properties={'host':host,'database':database,'user':
user,'password':password}
        return  PropertyUtil.connection_properties
```

Output Snippets

```
Run    Main    patientdao
(Patient) 1.CREATE 2.INSERT 3.UPDATE 4.DELETE 5.SELECT 0.EXIT
Enter choice: 2
Enter Patient ID: 1
Enter First Name: Tanmay
Enter Last Name: Shrivastava
Enter DOB: 2000-01-01
Enter Gender: Male
Enter Number: 6263582103
Enter address: Gwalior
True
```

```
Enter address: Gwalior
True
(Patient) 1.CREATE 2.INSERT 3.UPDATE 4.DELETE 5.SELE
Enter choice: 2
Enter Patient ID: 2
Enter First Name: Muskan
Enter Last Name: Saxena
Enter DOB: 2001-08-20
Enter Gender: Female
Enter Number: 7987666716
Enter address: Bhopal
True
```

```
Run    Main    patientdao
==========
1.Get Appointment By ID
2.Get Appointment For Patient
3.Get Appointment For Doctor
4.Cancel Appointment
0.EXIT
Enter choice: 1
Enter Appointment ID to see the Appointment1
Enter Appointment ID to get Information:
[(1, 1, 1, datetime.date(2024, 3, 20), 'Stomach Ache')]
Hospital Management System > main > Main.py
17°C
Mostly cloudy
Search
```

Run Main patientdao

```
Enter choice: 2
Enter Patient ID of the Patient to see the Appointment: 2
Enter Patient ID to get Information:
[(2, 2, 2, datetime.date(2023, 3, 2), 'Head Ache')]
==========
---MENU---
==========
1.Get Appointment By ID
2.Get Appointment For Patient
3.Get Appointment For Doctor
```

Hospital Management System > main > Main.py

17°C
Mostly cloudy

Search

Run Main patientdao

```
==========
1.Get Appointment By ID
2.Get Appointment For Patient
3.Get Appointment For Doctor
4.Cancel Appointment
0.EXIT
Enter choice: 2
Enter Patient ID of the Patient to see the Appointment: 2
Enter Patient ID to get Information:
[(2, 2, 2, datetime.date(2023, 3, 2), 'Head Ache')]
```

Run Main patientdao

```
3.Get Appointment For Doctor
4.Cancel Appointment
0.EXIT
Enter choice: 3
Enter Doctor ID of the Doctor to see the Appointment: 1
Enter Appointment ID to get Information:
[(1, 1, 1, datetime.date(2024, 3, 20), 'Stomach Ache')]
==========
---MENU---
==========
```

```
3.Get Appointment For Doctor
4.Cancel Appointment
0.EXIT
Enter choice: 4
Enter Appointment ID to to cancel the Appointment: 4
Enter Appointment ID to cancel appointment:
True
==========
---MENU---
==========
```

```
(Doctor) 1.CREATE 2.INSERT 3.UPDATE 4.DELETE 5.SELECT 0.EXIT
Enter choice: 2
Enter Doctor ID: 5
Enter First Name: Sumit
Enter Last Name: Singh
Specialization: ENT
Enter Number: 6789934455
True
(Doctor) 1.CREATE 2.INSERT 3.UPDATE 4.DELETE 5.SELECT 0.EXIT
Enter choice:
```

```
True
(Doctor) 1.CREATE 2.INSERT 3.UPDATE 4.DELETE 5.SELECT 0.EXIT
Enter choice: 5
Records In Doctor Table:
(1, 'Ravi', 'Shukla', 'MD', '9087654321')
(2, 'Rahul', 'Gandhi', 'Surgeon', '6789054312')
(3, 'Priya', 'priya', 'Gynaec', '5643217890')
(5, 'Sumit', 'Singh', 'ENT', '6789934455')
(Doctor) 1.CREATE 2.INSERT 3.UPDATE 4.DELETE 5.SELECT 0.EXIT
Enter choice:
```

Run Main patientdao

```
True
(Doctor) 1.CREATE 2.INSERT 3.UPDATE 4.DELETE 5.SELECT 0.EXIT
Enter choice: 5
Records In Doctor Table:
(1, 'Ravi', 'Shukla', 'MD', '9087654321')
(2, 'Rahul', 'Gandhi', 'Surgeon', '6789054312')
(3, 'Priya', 'priya', 'Gynaec', '5643217890')
(5, 'Sumit', 'Singh', 'ENT', '6789934455')
(Doctor) 1.CREATE 2.INSERT 3.UPDATE 4.DELETE 5.SELECT 0.EXIT
Enter choice:
```

Hospital Management System > main > Main.py

```
(Doctor) 1.CREATE 2.INSERT 3.UPDATE 4.DELETE 5.SELECT 0.EXIT
Enter choice: 5
Records In Doctor Table:
(1, 'Ravi', 'Shukla', 'MD', '9087654321')
(2, 'Rahul', 'Gandhi', 'Surgeon', '6789054312')
(3, 'Priya', 'priya', 'Gynaec', '5643217890')
(5, 'Sumit', 'Singh', 'ENT', '6789934455')
(Doctor) 1.CREATE 2.INSERT 3.UPDATE 4.DELETE 5.SELECT 0.EXIT
Enter choice: 0
1.Patient 2.Doctor 3.Appointment 0.EXIT
Enter choice: 1
(Patient) 1.CREATE 2.INSERT 3.UPDATE 4.DELETE 5.SELECT 0.EXIT
Enter choice: 5
Records In Patient Table:
(1, 'Tanmay', 'Shrivastava', datetime.date(2000, 1, 1), 'Male', '6263582103', 'Gwalior')
(2, 'Muskan', 'Saxena', datetime.date(2001, 8, 20), 'Female', '7987666716', 'Bhopal')
(3, 'Rockey', 'Randhawa', datetime.date(1993, 10, 10), 'Male', '0987654321', 'Mumbai')
(Patient) 1.CREATE 2.INSERT 3.UPDATE 4.DELETE 5.SELECT 0.EXIT
Enter choice:
```

```
1.Patient 2.Doctor 3.Appointment 0.EXIT
Enter choice: 1
(Patient) 1.CREATE 2.INSERT 3.UPDATE 4.DELETE 5.SELECT 0.EXIT
Enter choice: 5
Records In Patient Table:
(1, 'Tanmay', 'Shrivastava', datetime.date(2000, 1, 1), 'Male', '6263582103', 'Gwalior')
(2, 'Muskan', 'Saxena', datetime.date(2001, 8, 20), 'Female', '7987666716', 'Bhopal')
(3, 'Rockey', 'Randhawa', datetime.date(1993, 10, 10), 'Male', '0987654321', 'Mumbai')
(Patient) 1.CREATE 2.INSERT 3.UPDATE 4.DELETE 5.SELECT 0.EXIT
Enter choice: 0
1.Patient 2.Doctor 3.Appointment 0.EXIT
Enter choice: 2
(Doctor) 1.CREATE 2.INSERT 3.UPDATE 4.DELETE 5.SELECT 0.EXIT
Enter choice: 5
Records In Doctor Table:
(1, 'Ravi', 'Shukla', 'MD', '9087654321')
(2, 'Rahul', 'Gandhi', 'Surgeon', '6789054312')
(3, 'Priya', 'priya', 'Gynaec', '5643217890')
(5, 'Sumit', 'Singh', 'ENT', '6789934455')
(Doctor) 1.CREATE 2.INSERT 3.UPDATE 4.DELETE 5.SELECT 0.EXIT
Enter choice:
```