Name: K.Nandini
Roll.no: CB.EN.U4AIE22030

Design a Hospital Database Management System Using Python

- Entity-Relationship (ER) diagram representing hospital management system

- complete Python code

created following tables based on the system requirements:

1. Patient Information

- **Patients Table**: Stores patient demographics.
- **Medical History Table**: Tracks the medical history of patients.

2. Appointments

- **Appointments Table**: Manages patient-doctor appointments.

3. Doctors and Medical Staff

- **Doctors Table**: Stores doctor details.
- **Staff Table**: Contains hospital staff details.

4. Medical Records

- **Prescriptions Table**: Records prescriptions given to patients.
- **Lab Tests Table**: Details about lab tests.
- **Vital Signs Table**: Tracks patient vital signs.

5. Billing and Finance

- **Invoices Table**: Handles financial transactions.
- **Payments Table**: Logs payments by patients.

6. Pharmacy

- **Medication Inventory Table**: Tracks medication availability.
- **Medication Dispensation Table**: Logs dispensed medications.

7. Departments

- **Departments Table**: Stores hospital department details.

8. Room and Bed Management

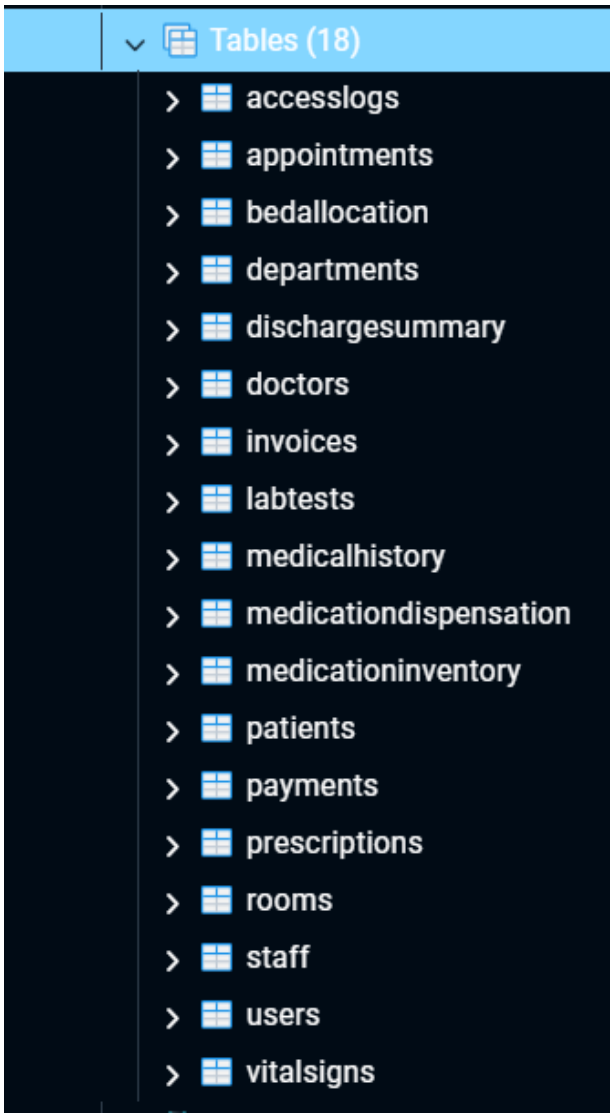
- **Rooms Table**: Manages hospital rooms.
- **Bed Allocation Table**: Tracks the allocation of beds to patients.

9. Reports

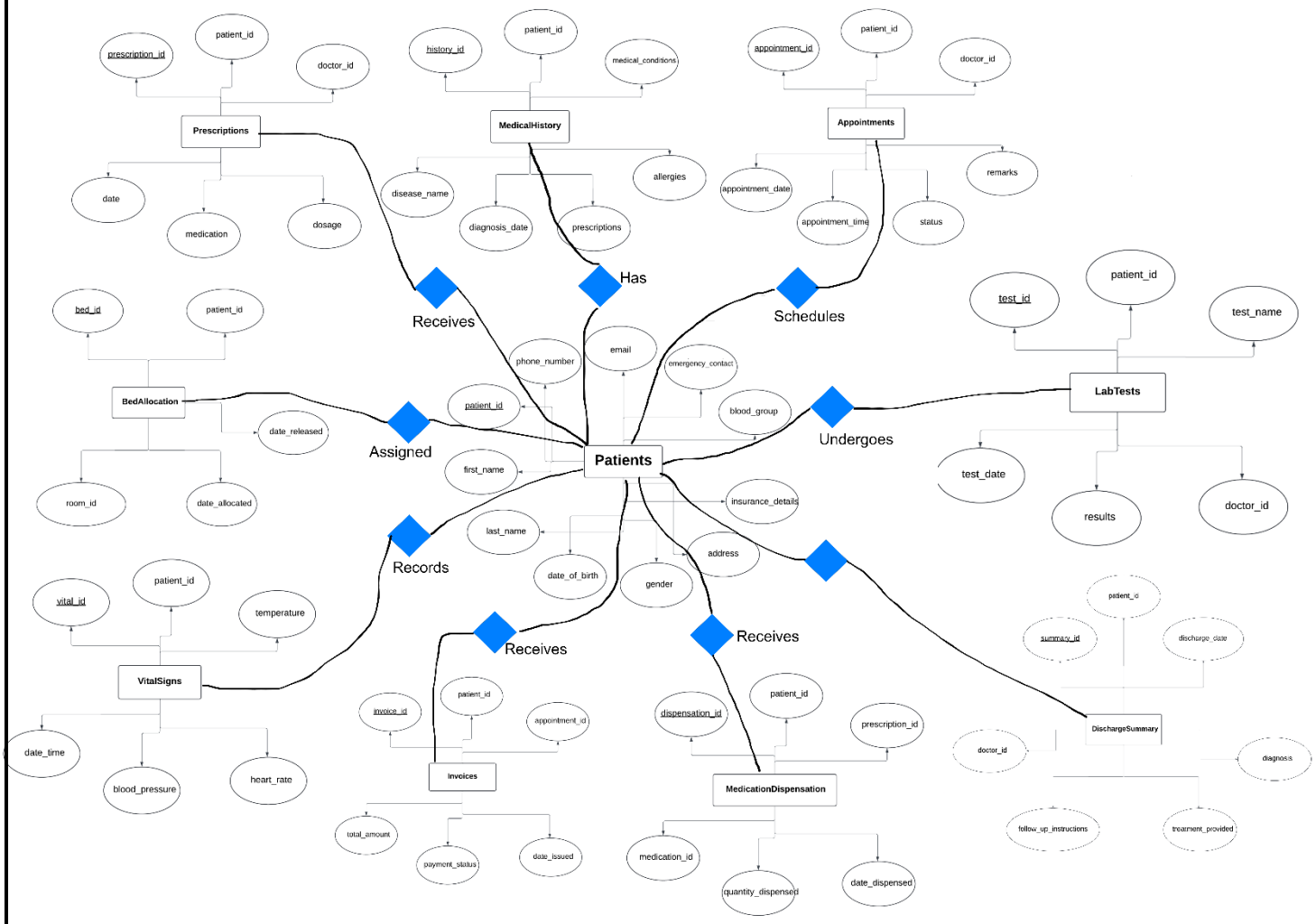
- **Discharge Summary Table**: Records discharge summaries.

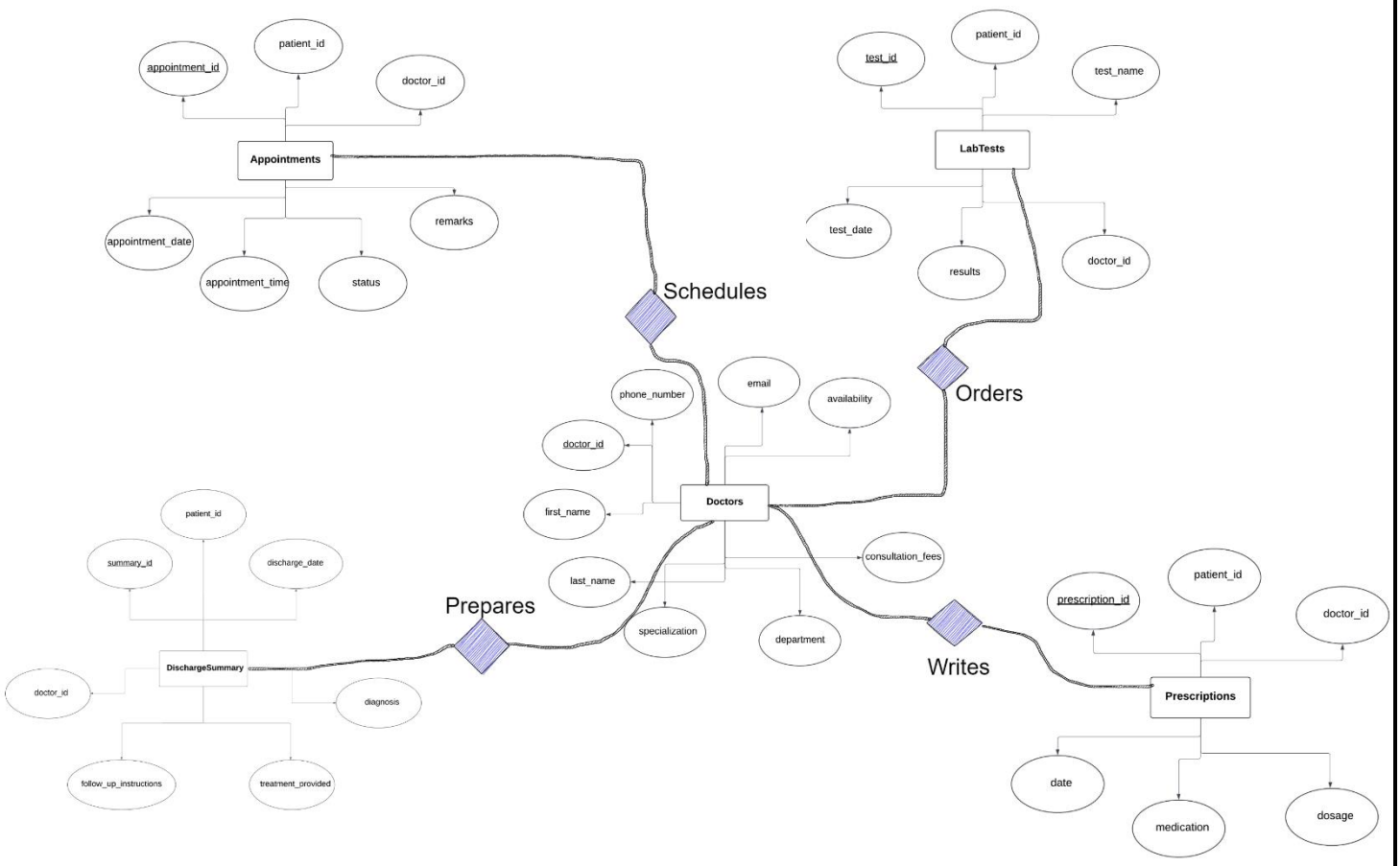
10. Administration

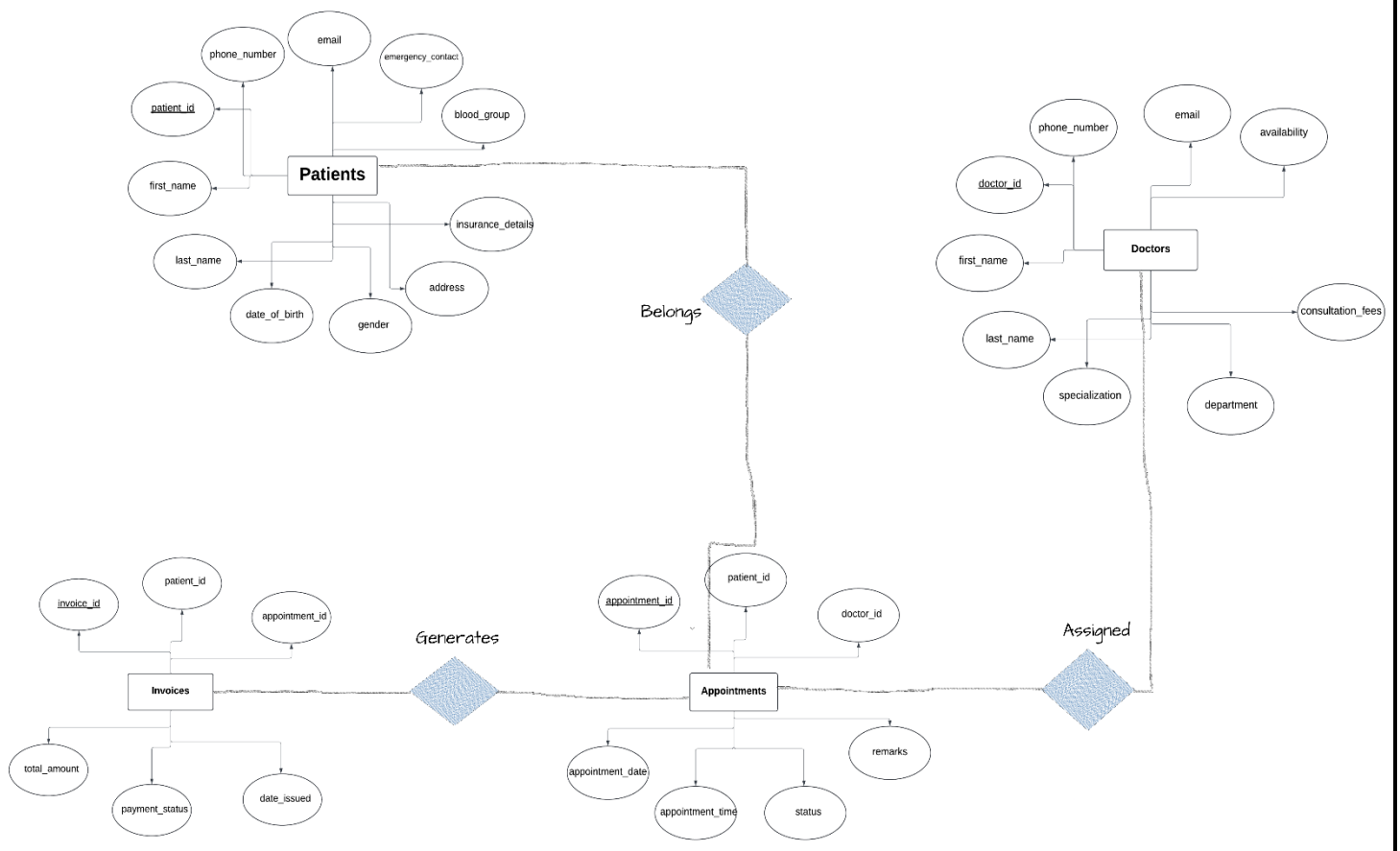
- **Users Table**: Stores user login details.
- **Access Logs Table**: Tracks user access.



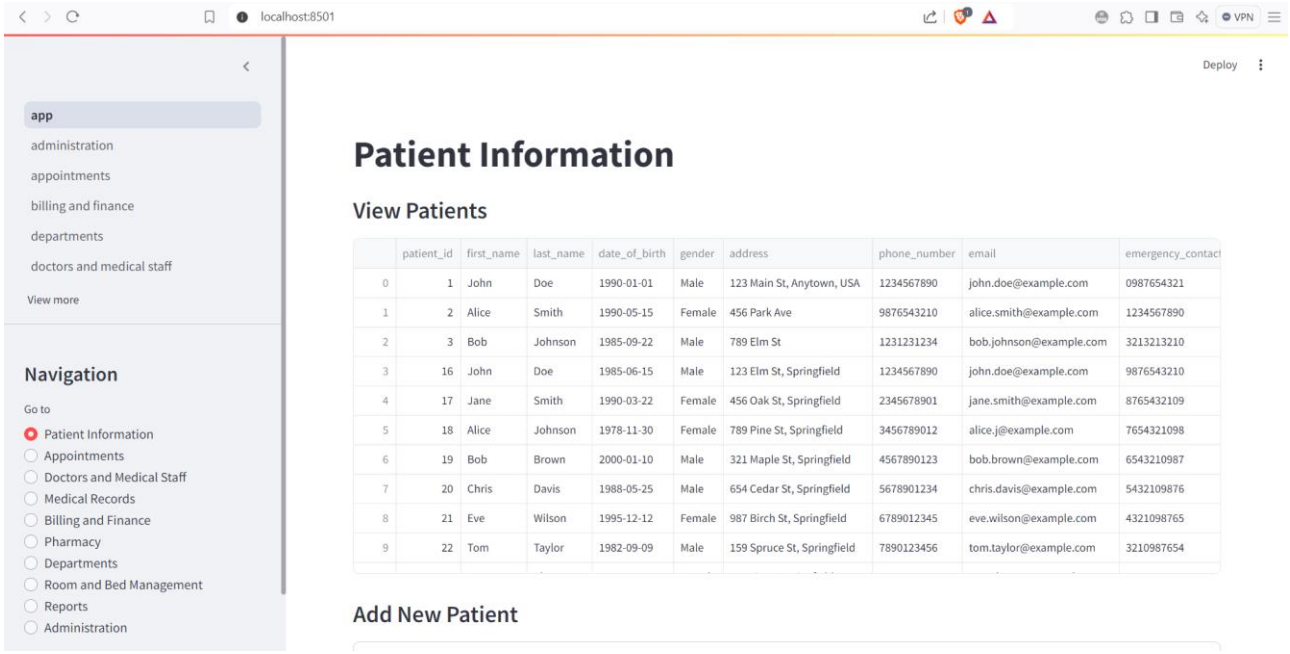
ER diagrams:

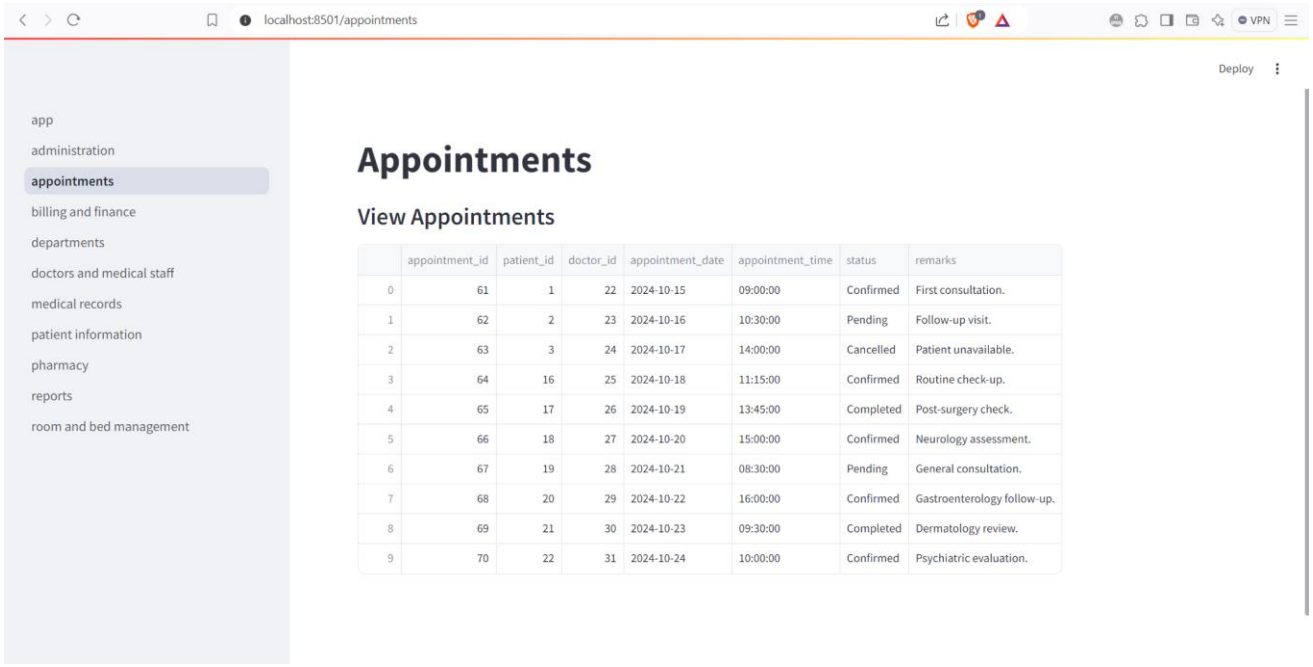

Patients
Prescriptions
prescription_id
patient_id
doctor_id
date
medication
dosage
MedicalHistory
history_id
patient_id
medical_conditions
disease_name
diagnosis_date
prescriptions
allergies
Appointments
appointment_id
patient_id
doctor_id
appointment_date
appointment_time
status
remarks
Has
Receives
Schedules
BedAllocation
bed_id
patient_id
date_released
room_id
date_allocated
Assigned
LabTests
test_id
patient_id
test_name
test_date
results
doctor_id
Undergoes
phone_number
email
emergency_contact
blood_group
patient_id
first_name
last_name
date_of_birth
gender
address
insurance_details
Records
VitalSigns
vital_id
patient_id
temperature
date_time
blood_pressure
heart_rate
Receives
Invoices
invoice_id
patient_id
appointment_id
total_amount
payment_status
date_issued
Receives
MedicationDispensation
dispensation_id
patient_id
prescription_id
medication_id
quantity_dispensed
date_dispensed
DischargeSummary
patient_id
summary_id
discharge_date
doctor_id
diagnosis
follow_up_instructions
treatment_provided

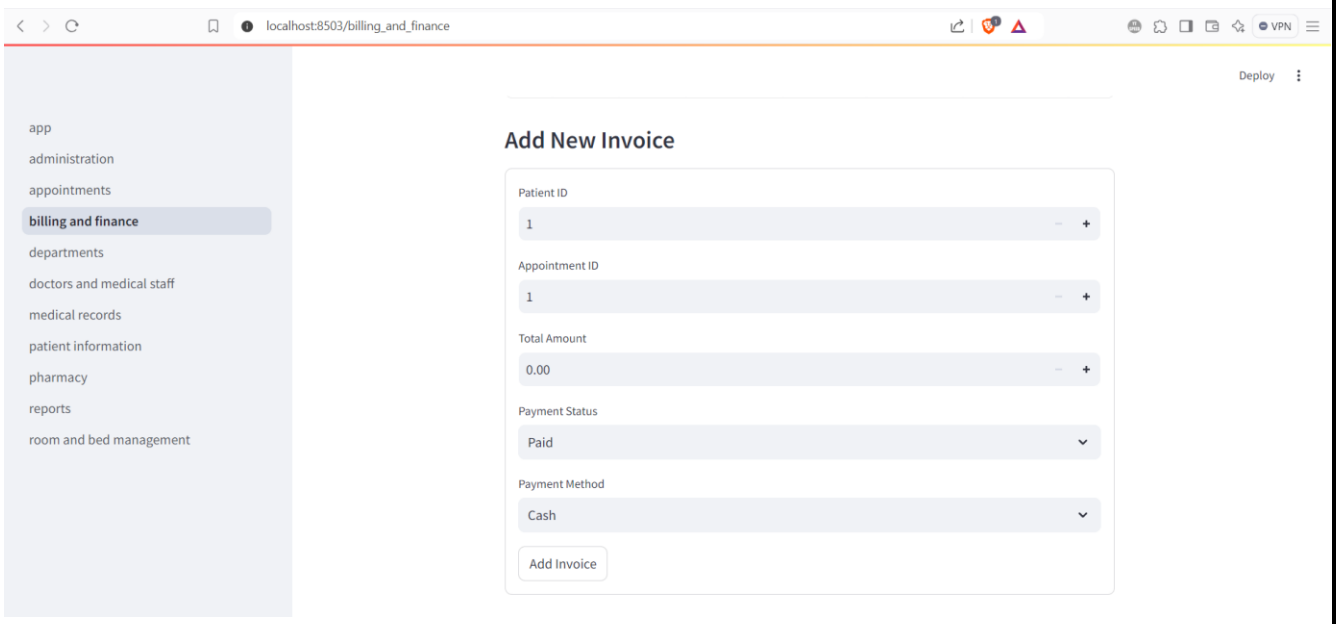

appointment_id
patient_id
doctor_id
Appointments
appointment_date
appointment_time
status
remarks
Schedules
test_id
patient_id
test_name
LabTests
test_date
results
doctor_id
Orders
phone_number
email
availability
doctor_id
first_name
Doctors
last_name
specialization
department
consultation_fees
Prepares
Writes
patient_id
summary_id
discharge_date
DischargeSummary
doctor_id
diagnosis
follow_up_instructions
treatment_provided
prescription_id
patient_id
doctor_id
Prescriptions
date
medication
dosage

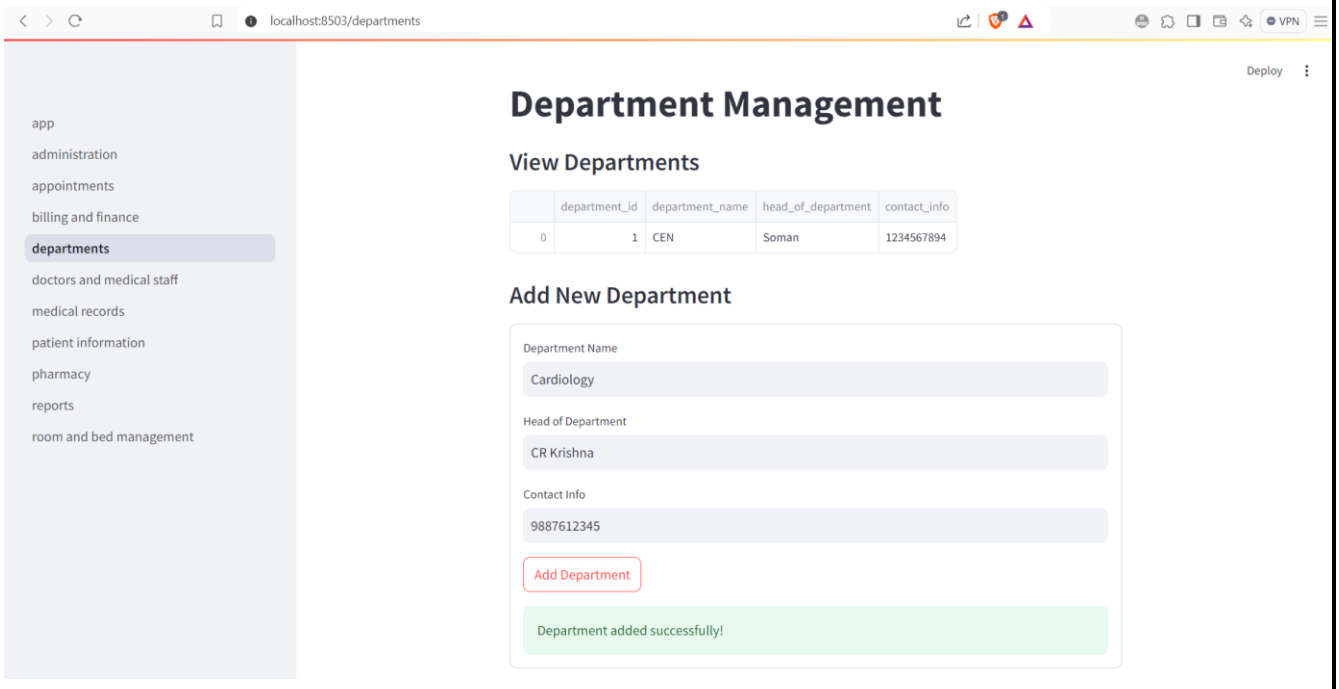

phone_number
email
emergency_contact
patient_id
blood_group
first_name
Patients
insurance_details
last_name
address
date_of_birth
gender
Belongs
phone_number
email
availability
doctor_id
first_name
Doctors
consultation_fees
last_name
specialization
department
invoice_id
patient_id
appointment_id
Generates
Invoices
total_amount
payment_status
date_issued
appointment_id
patient_id
doctor_id
Appointments
appointment_date
appointment_time
status
remarks
Assigned

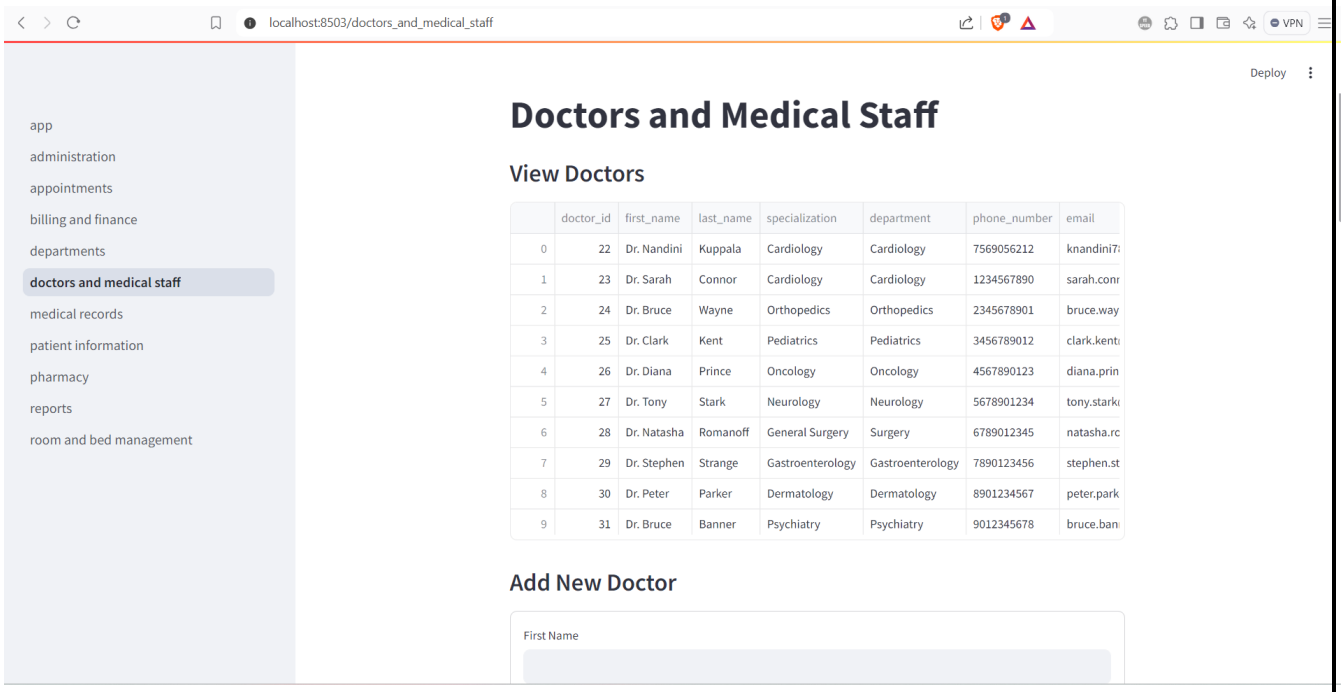
APP:

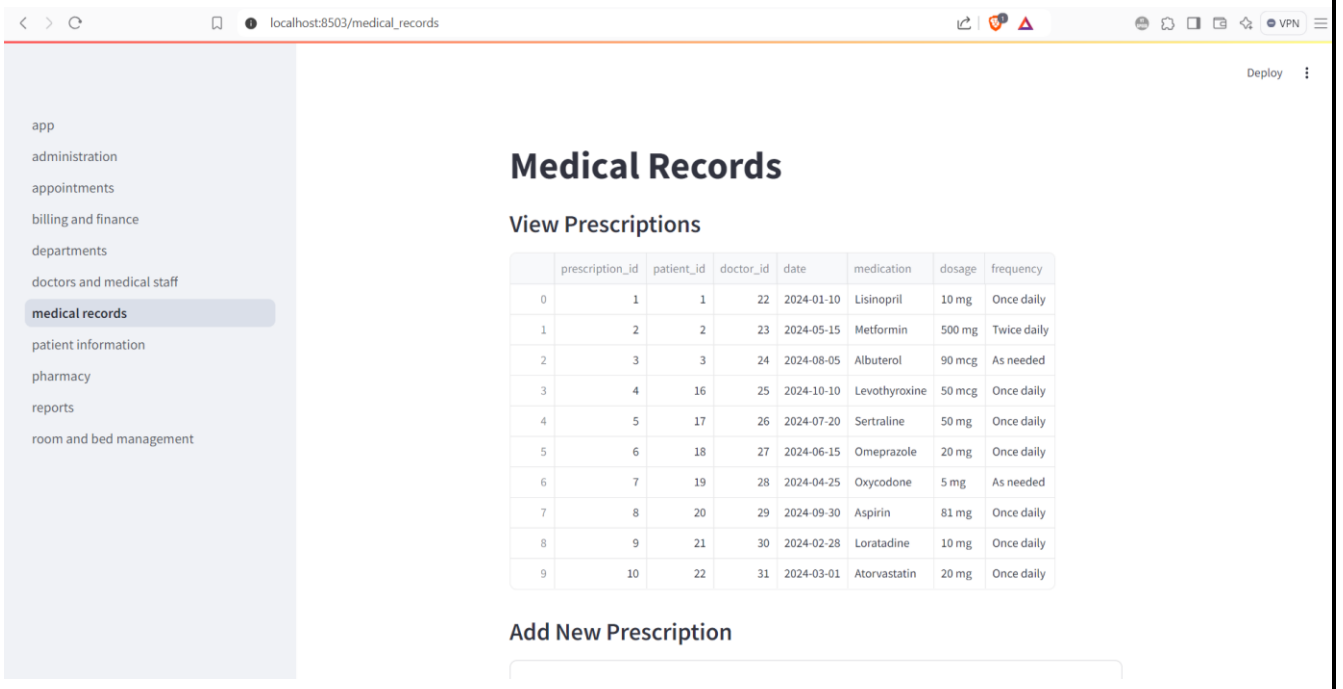

localhost:8501
Deploy
app
administration
appointments
billing and finance
departments
doctors and medical staff
View more
Navigation
Go to
Patient Information
Appointments
Doctors and Medical Staff
Medical Records
Billing and Finance
Pharmacy
Departments
Room and Bed Management
Reports
Administration
Patient Information
View Patients
patient_id first_name last_name date_of_birth gender address phone_number email emergency_contact
0 1 John Doe 1990-01-01 Male 123 Main St, Anytown, USA 1234567890 john.doe@example.com 0987654321
1 2 Alice Smith 1990-05-15 Female 456 Park Ave 9876543210 alice.smith@example.com 1234567890
2 3 Bob Johnson 1985-09-22 Male 789 Elm St 1231231234 bob.johnson@example.com 3213213210
3 16 John Doe 1985-06-15 Male 123 Elm St, Springfield 1234567890 john.doe@example.com 9876543210
4 17 Jane Smith 1990-03-22 Female 456 Oak St, Springfield 2345678901 jane.smith@example.com 8765432109
5 18 Alice Johnson 1978-11-30 Female 789 Pine St, Springfield 3456789012 alice.j@example.com 7654321098
6 19 Bob Brown 2000-01-10 Male 321 Maple St, Springfield 4567890123 bob.brown@example.com 6543210987
7 20 Chris Davis 1988-05-25 Male 654 Cedar St, Springfield 5678901234 chris.davis@example.com 5432109876
8 21 Eve Wilson 1995-12-12 Female 987 Birch St, Springfield 6789012345 eve.wilson@example.com 4321098765
9 22 Tom Taylor 1982-09-09 Male 159 Spruce St, Springfield 7890123456 tom.taylor@example.com 3210987654
Add New Patient

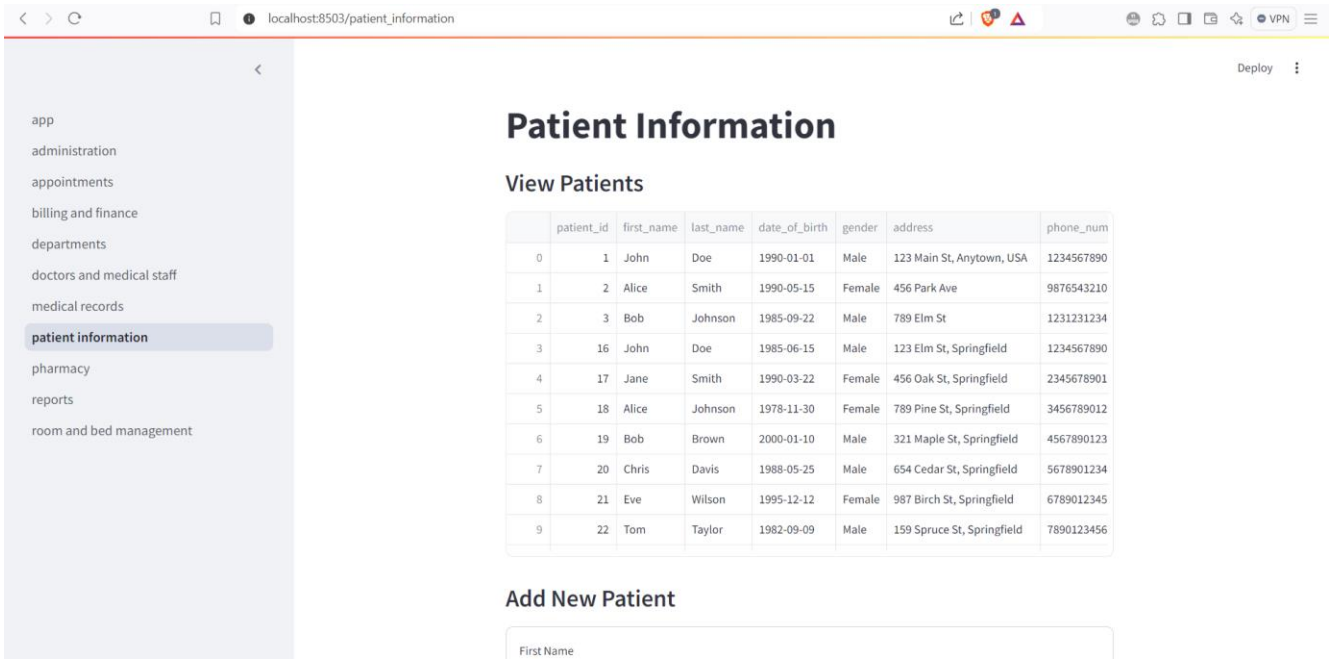

localhost:8501/appointments
Deploy
app
administration
appointments
billing and finance
departments
doctors and medical staff
medical records
patient information
pharmacy
reports
room and bed management
Appointments
View Appointments
appointment_id patient_id doctor_id appointment_date appointment_time status remarks
0 61 1 22 2024-10-15 09:00:00 Confirmed First consultation.
1 62 2 23 2024-10-16 10:30:00 Pending Follow-up visit.
2 63 3 24 2024-10-17 14:00:00 Cancelled Patient unavailable.
3 64 16 25 2024-10-18 11:15:00 Confirmed Routine check-up.
4 65 17 26 2024-10-19 13:45:00 Completed Post-surgery check.
5 66 18 27 2024-10-20 15:00:00 Confirmed Neurology assessment.
6 67 19 28 2024-10-21 08:30:00 Pending General consultation.
7 68 20 29 2024-10-22 16:00:00 Confirmed Gastroenterology follow-up.
8 69 21 30 2024-10-23 09:30:00 Completed Dermatology review.
9 70 22 31 2024-10-24 10:00:00 Confirmed Psychiatric evaluation.

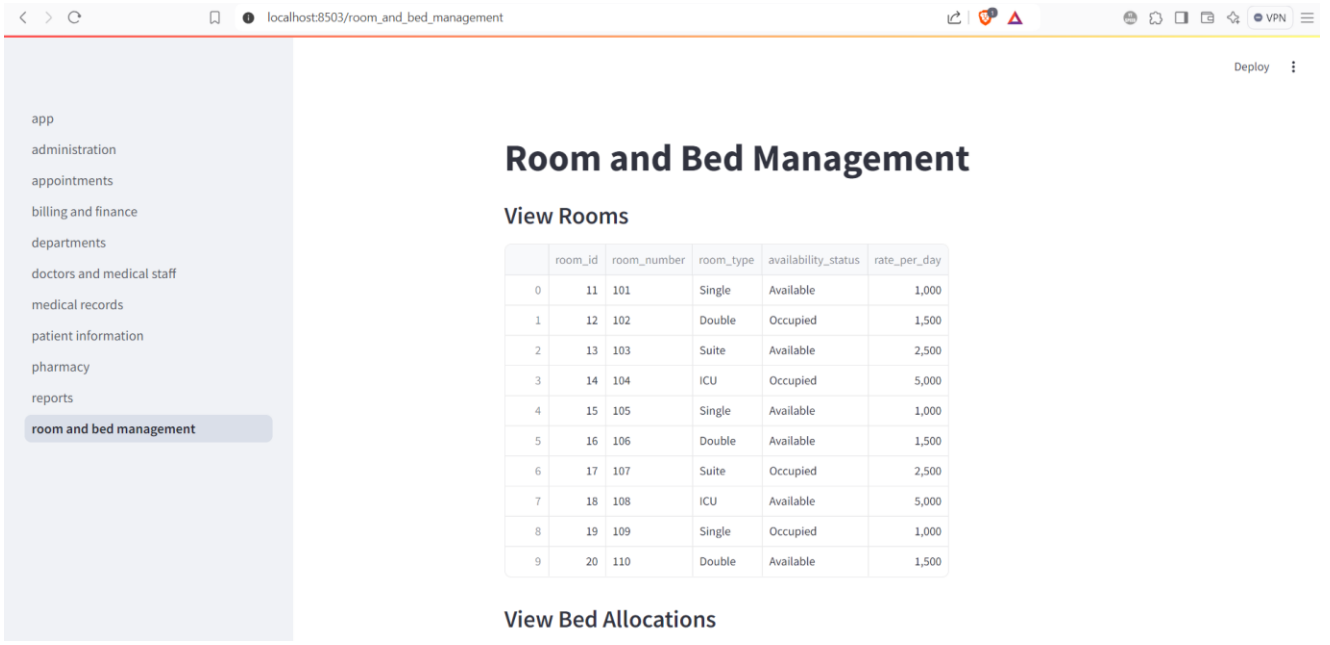

localhost:8503/billing_and_finance
VPN
Deploy
app
administration
appointments
billing and finance
departments
doctors and medical staff
medical records
patient information
pharmacy
reports
room and bed management
Add New Invoice
Patient ID
1
Appointment ID
1
Total Amount
0.00
Payment Status
Paid
Payment Method
Cash
Add Invoice


localhost:8503/departments
VPN
Deploy
app
administration
appointments
billing and finance
departments
doctors and medical staff
medical records
patient information
pharmacy
reports
room and bed management
Department Management
View Departments
department_id
department_name
head_of_department
contact_info
0
1
CEN
Soman
1234567894
Add New Department
Department Name
Cardiology
Head of Department
CR Krishna
Contact Info
9887612345
Add Department
Department added successfully!


localhost:8503/doctors_and_medical_staff
VPN
Deploy
app
administration
appointments
billing and finance
departments
doctors and medical staff
medical records
patient information
pharmacy
reports
room and bed management
Doctors and Medical Staff
View Doctors
doctor_id first_name last_name specialization department phone_number email
0 22 Dr. Nandini Kuppala Cardiology Cardiology 7569056212 knandini7
1 23 Dr. Sarah Connor Cardiology Cardiology 1234567890 sarah.conr
2 24 Dr. Bruce Wayne Orthopedics Orthopedics 2345678901 bruce.way
3 25 Dr. Clark Kent Pediatrics Pediatrics 3456789012 clark.kent
4 26 Dr. Diana Prince Oncology Oncology 4567890123 diana.prin
5 27 Dr. Tony Stark Neurology Neurology 5678901234 tony.stark
6 28 Dr. Natasha Romanoff General Surgery Surgery 6789012345 natasha.rc
7 29 Dr. Stephen Strange Gastroenterology Gastroenterology 7890123456 stephen.st
8 30 Dr. Peter Parker Dermatology Dermatology 8901234567 peter.park
9 31 Dr. Bruce Banner Psychiatry Psychiatry 9012345678 bruce.ban
Add New Doctor
First Name


localhost:8503/medical_records
VPN
Deploy
app
administration
appointments
billing and finance
departments
doctors and medical staff
medical records
patient information
pharmacy
reports
room and bed management
Medical Records
View Prescriptions
prescription_id patient_id doctor_id date medication dosage frequency
0 1 1 22 2024-01-10 Lisinopril 10 mg Once daily
1 2 2 23 2024-05-15 Metformin 500 mg Twice daily
2 3 3 24 2024-08-05 Albuterol 90 mcg As needed
3 4 16 25 2024-10-10 Levothyroxine 50 mcg Once daily
4 5 17 26 2024-07-20 Sertraline 50 mg Once daily
5 6 18 27 2024-06-15 Omeprazole 20 mg Once daily
6 7 19 28 2024-04-25 Oxycodone 5 mg As needed
7 8 20 29 2024-09-30 Aspirin 81 mg Once daily
8 9 21 30 2024-02-28 Loratadine 10 mg Once daily
9 10 22 31 2024-03-01 Atorvastatin 20 mg Once daily
Add New Prescription


localhost:8503/patient_information
VPN
Deploy
app
administration
appointments
billing and finance
departments
doctors and medical staff
medical records
patient information
pharmacy
reports
room and bed management
Patient Information
View Patients
patient_id first_name last_name date_of_birth gender address phone_num
0 1 John Doe 1990-01-01 Male 123 Main St, Anytown, USA 1234567890
1 2 Alice Smith 1990-05-15 Female 456 Park Ave 9876543210
2 3 Bob Johnson 1985-09-22 Male 789 Elm St 1231231234
3 16 John Doe 1985-06-15 Male 123 Elm St, Springfield 1234567890
4 17 Jane Smith 1990-03-22 Female 456 Oak St, Springfield 2345678901
5 18 Alice Johnson 1978-11-30 Female 789 Pine St, Springfield 3456789012
6 19 Bob Brown 2000-01-10 Male 321 Maple St, Springfield 4567890123
7 20 Chris Davis 1988-05-25 Male 654 Cedar St, Springfield 5678901234
8 21 Eve Wilson 1995-12-12 Female 987 Birch St, Springfield 6789012345
9 22 Tom Taylor 1982-09-09 Male 159 Spruce St, Springfield 7890123456
Add New Patient
First Name


localhost:8503/room_and_bed_management
VPN
Deploy
app
administration
appointments
billing and finance
departments
doctors and medical staff
medical records
patient information
pharmacy
reports
room and bed management
Room and Bed Management
View Rooms
room_id room_number room_type availability_status rate_per_day
0 11 101 Single Available 1,000
1 12 102 Double Occupied 1,500
2 13 103 Suite Available 2,500
3 14 104 ICU Occupied 5,000
4 15 105 Single Available 1,000
5 16 106 Double Available 1,500
6 17 107 Suite Occupied 2,500
7 18 108 ICU Available 5,000
8 19 109 Single Occupied 1,000
9 20 110 Double Available 1,500
View Bed Allocations

Python code:

to connect:

```
import psycopg2
from psycopg2 import sql

# Define your database connection parameters
DB_HOST = "localhost"
DB_NAME = "hospital_management"
DB_USER = "postgres"
DB_PASSWORD = "nandini@108"  # replace with your actual password

# Establish the connection
try:
    connection = psycopg2.connect(
        host=DB_HOST,
        database=DB_NAME,
        user=DB_USER,
        password=DB_PASSWORD
    )
    cursor = connection.cursor()
    print("Connected to the database.")

except Exception as error:
    print(f"Error connecting to database: {error}")
```

Created tables In Pg Admin:

```
CREATE TABLE Patients (
    patient_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    date_of_birth DATE,
    gender VARCHAR(10),
    address TEXT,
    phone_number VARCHAR(15),
    email VARCHAR(50),
    emergency_contact VARCHAR(15),
    insurance_details TEXT,
    blood_group VARCHAR(5)
);

CREATE TABLE MedicalHistory (
    history_id SERIAL PRIMARY KEY,
    patient_id INT REFERENCES Patients(patient_id),
    disease_name VARCHAR(100),
    diagnosis_date DATE,
    prescriptions TEXT,
    allergies TEXT,
    medical_conditions TEXT
);

CREATE TABLE Doctors (
```

```
    doctor_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    specialization VARCHAR(100),
    department VARCHAR(50),
    phone_number VARCHAR(15),
    email VARCHAR(50),
    availability TEXT,
    consultation_fees DECIMAL(10, 2)
);

CREATE TABLE Appointments (
    appointment_id SERIAL PRIMARY KEY,
    patient_id INT REFERENCES Patients(patient_id),
    doctor_id INT REFERENCES Doctors(doctor_id),
    appointment_date DATE,
    appointment_time TIME,
    status VARCHAR(20),
    remarks TEXT
);

CREATE TABLE Staff (
    staff_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    role VARCHAR(50),
```

```
    department VARCHAR(50),
    phone_number VARCHAR(15),
    email VARCHAR(50),
    shift_schedule TEXT
);

CREATE TABLE Prescriptions (
    prescription_id SERIAL PRIMARY KEY,
    patient_id INT REFERENCES Patients(patient_id),
    doctor_id INT REFERENCES Doctors(doctor_id),
    date DATE,
    medication TEXT,
    dosage TEXT,
    frequency TEXT
);

CREATE TABLE LabTests (
    test_id SERIAL PRIMARY KEY,
    patient_id INT REFERENCES Patients(patient_id),
    test_name VARCHAR(100),
    test_date DATE,
    results TEXT,
    doctor_id INT REFERENCES Doctors(doctor_id)
);

CREATE TABLE VitalSigns (
```

```
    vital_id SERIAL PRIMARY KEY,
    patient_id INT REFERENCES Patients(patient_id),
    date_time TIMESTAMP,
    blood_pressure VARCHAR(20),
    heart_rate INT,
    temperature DECIMAL(4, 2)
);

CREATE TABLE Invoices (
    invoice_id SERIAL PRIMARY KEY,
    patient_id INT REFERENCES Patients(patient_id),
    appointment_id INT REFERENCES Appointments(appointment_id),
    total_amount DECIMAL(10, 2),
    payment_status VARCHAR(20),
    payment_method VARCHAR(20),
    date_issued DATE
);

CREATE TABLE Payments (
    payment_id SERIAL PRIMARY KEY,
    invoice_id INT REFERENCES Invoices(invoice_id),
    payment_date DATE,
    amount_paid DECIMAL(10, 2),
    payment_method VARCHAR(20)
);
```

```
CREATE TABLE MedicationInventory (
    medication_id SERIAL PRIMARY KEY,
    name VARCHAR(100),
    quantity_in_stock INT,
    price DECIMAL(10, 2),
    expiration_date DATE
);

CREATE TABLE MedicationDispensation (
    dispensation_id SERIAL PRIMARY KEY,
    patient_id INT REFERENCES Patients(patient_id),
    prescription_id INT REFERENCES Prescriptions(prescription_id),
    medication_id INT REFERENCES MedicationInventory(medication_id),
    quantity_dispensed INT,
    date_dispensed DATE
);

CREATE TABLE Departments (
    department_id SERIAL PRIMARY KEY,
    department_name VARCHAR(100),
    head_of_department VARCHAR(100),
    contact_info VARCHAR(100)
);

CREATE TABLE Rooms (
    room_id SERIAL PRIMARY KEY,
```

```
    room_number VARCHAR(10),
    room_type VARCHAR(50),
    availability_status VARCHAR(20),
    rate_per_day DECIMAL(10, 2)
);

CREATE TABLE BedAllocation (
    bed_id SERIAL PRIMARY KEY,
    patient_id INT REFERENCES Patients(patient_id),
    room_id INT REFERENCES Rooms(room_id),
    date_allocated DATE,
    date_released DATE
);

CREATE TABLE DischargeSummary (
    summary_id SERIAL PRIMARY KEY,
    patient_id INT REFERENCES Patients(patient_id),
    discharge_date DATE,
    diagnosis TEXT,
    treatment_provided TEXT,
    doctor_id INT REFERENCES Doctors(doctor_id),
    follow_up_instructions TEXT
);

CREATE TABLE Users (
    user_id SERIAL PRIMARY KEY,
```

```
    username VARCHAR(50),
    password VARCHAR(255),
    role VARCHAR(50),
    staff_id INT REFERENCES Staff(staff_id),
    last_login TIMESTAMP
);

CREATE TABLE AccessLogs (
    log_id SERIAL PRIMARY KEY,
    user_id INT REFERENCES Users(user_id),
    action TEXT,
    timestamp TIMESTAMP
);
```

Inserted data from python code:

```
INSERT INTO Patients (first_name, last_name, date_of_birth, gender, address, phone_number, email, emergency_contact, insurance_details, blood_group) VALUES
('John', 'Doe', '1985-05-15', 'Male', '123 Main St, Springfield', '9876543210', 'john.doe@example.com', '911', 'Health Insurance A', 'O+'),
('Jane', 'Smith', '1990-03-25', 'Female', '456 Oak St, Springfield', '8765432109', 'jane.smith@example.com', '912', 'Health Insurance B', 'A+'),
('Alice', 'Johnson', '1988-08-30', 'Female', '789 Pine St, Springfield', '7654321098', 'alice.johnson@example.com', '913', 'Health Insurance C', 'B+'),
('Bob', 'Brown', '1980-12-20', 'Male', '321 Maple St, Springfield', '6543210987', 'bob.brown@example.com', '914', 'Health Insurance D', 'AB+'),
('Charlie', 'Davis', '1995-01-10', 'Male', '654 Elm St, Springfield', '5432109876', 'charlie.davis@example.com', '915', 'Health Insurance E', 'O-'),
```

('Emily', 'Wilson', '1992-06-15', 'Female', '987 Cedar St, Springfield', '4321098765', 'emily.wilson@example.com', '916', 'Health Insurance F', 'A-'),

('Frank', 'Martinez', '1987-09-05', 'Male', '159 Birch St, Springfield', '3210987654', 'frank.martinez@example.com', '917', 'Health Insurance G', 'B-'),

('Grace', 'Lee', '1993-11-30', 'Female', '753 Spruce St, Springfield', '2109876543', 'grace.lee@example.com', '918', 'Health Insurance H', 'AB-'),

('Henry', 'Garcia', '1986-04-04', 'Male', '369 Fir St, Springfield', '1098765432', 'henry.garcia@example.com', '919', 'Health Insurance I', 'O+'),

('Isabella', 'Hernandez', '1989-07-28', 'Female', '147 Cherry St, Springfield', '0987654321', 'isabella.hernandez@example.com', '920', 'Health Insurance J', 'A+');

INSERT INTO Doctors (first_name, last_name, specialization, department, phone_number, email, availability, consultation_fees) VALUES

('Dr. Nandini', 'Kuppala', 'Cardiology', 'Cardiology', '7569056212', 'knandini7816@gmail.com', 'Yes', 500.00),

('Dr. Sarah', 'Connor', 'Cardiology', 'Cardiology', '1234567890', 'sarah.connor@example.com', 'Mon-Fri', 200.00),

('Dr. Bruce', 'Wayne', 'Orthopedics', 'Orthopedics', '2345678901', 'bruce.wayne@example.com', 'Mon-Sat', 150.00),

('Dr. Clark', 'Kent', 'Pediatrics', 'Pediatrics', '3456789012', 'clark.kent@example.com', 'Tue-Sun', 100.00),

('Dr. Diana', 'Prince', 'Oncology', 'Oncology', '4567890123', 'diana.prince@example.com', 'Mon-Fri', 250.00),

('Dr. Tony', 'Stark', 'Neurology', 'Neurology', '5678901234', 'tony.stark@example.com', 'Mon-Thu', 300.00),

('Dr. Natasha', 'Romanoff', 'General Surgery', 'Surgery', '6789012345', 'natasha.romanoff@example.com', 'Mon-Sun', 180.00),

('Dr. Stephen', 'Strange', 'Gastroenterology', 'Gastroenterology', '7890123456', 'stephen.strange@example.com', 'Tue-Sat', 220.00),

('Dr. Peter', 'Parker', 'Dermatology', 'Dermatology', '8901234567', 'peter.parker@example.com', 'Mon-Fri', 130.00),

('Dr. Bruce', 'Banner', 'Psychiatry', 'Psychiatry', '9012345678', 'bruce.banner@example.com', 'Mon-Thu', 160.00);

INSERT INTO Appointments (patient_id, doctor_id, appointment_date, appointment_time, status, remarks) VALUES

(1, 22, '2024-10-15', '09:00:00', 'Confirmed', 'First consultation.'),

(2, 23, '2024-10-16', '10:30:00', 'Pending', 'Follow-up visit.'),

(3, 24, '2024-10-17', '14:00:00', 'Cancelled', 'Patient unavailable.'),

(16, 25, '2024-10-18', '11:15:00', 'Confirmed', 'Routine check-up.'),

(17, 26, '2024-10-19', '13:45:00', 'Completed', 'Post-surgery check.'),

(18, 27, '2024-10-20', '15:00:00', 'Confirmed', 'Neurology assessment.'),

(19, 28, '2024-10-21', '08:30:00', 'Pending', 'General consultation.'),

(20, 29, '2024-10-22', '16:00:00', 'Confirmed', 'Gastroenterology follow-up.'),

(21, 30, '2024-10-23', '09:30:00', 'Completed', 'Dermatology review.'),

(22, 31, '2024-10-24', '10:00:00', 'Confirmed', 'Psychiatric evaluation.');

INSERT INTO Staff (first_name, last_name, role, department, phone_number, email, shift_schedule) VALUES

('Alice', 'Smith', 'Nurse', 'Nursing', '1112223333', 'alice.smith@example.com', 'Mon-Fri 9am-5pm'),

('Bob', 'Johnson', 'Receptionist', 'Administration', '2223334444', 'bob.johnson@example.com', 'Mon-Sun 8am-4pm'),

('Charlie', 'Brown', 'Pharmacist', 'Pharmacy', '3334445555', 'charlie.brown@example.com', 'Mon-Fri 10am-6pm'),

('Diana', 'Prince', 'Surgeon', 'Surgery', '4445556666',
'diana.prince@example.com', 'Mon-Fri 8am-4pm'),

('Edward', 'Norton', 'Radiologist', 'Radiology', '5556667777',
'edward.norton@example.com', 'Mon-Fri 9am-5pm'),

('Fiona', 'Green', 'Lab Technician', 'Laboratory', '6667778888',
'fiona.green@example.com', 'Mon-Fri 9am-5pm'),

('George', 'White', 'Physiotherapist', 'Rehabilitation', '7778889999',
'george.white@example.com', 'Mon-Sat 9am-5pm'),

('Hannah', 'Williams', 'Admin Assistant', 'Administration', '8889990000',
'hannah.williams@example.com', 'Mon-Fri 8am-4pm'),

('Ian', 'Black', 'IT Support', 'IT', '9990001111', 'ian.black@example.com', 'Mon-Fri 9am-5pm'),

('Julia', 'Gold', 'Dietitian', 'Nutrition', '0001112222', 'julia.gold@example.com',
'Mon-Fri 9am-5pm');

INSERT INTO LabTests (patient_id, test_name, test_date, results, doctor_id)
VALUES

(1, 'Blood Test', '2024-01-12', 'Normal', 22),

(2, 'Glucose Test', '2024-05-17', 'High', 23),

(3, 'Chest X-Ray', '2024-08-07', 'Clear', 24),

(16, 'Thyroid Function Test', '2024-10-12', 'Normal', 25),

(17, 'Anxiety Screening', '2024-07-22', 'Mild', 26),

(18, 'Endoscopy', '2024-06-18', 'Normal', 27),

(19, 'MRI', '2024-04-30', 'Mild issues detected', 28),

(20, 'Lipid Profile', '2024-09-20', 'High Cholesterol', 29),

(21, 'Allergy Test', '2024-02-15', 'Positive for pollen', 30),

(22, 'Cardiac Stress Test', '2024-03-05', 'Normal', 31);

Code for APP

App structure:

```
dbms as3/
|
├── app.py
└── pages/
    ├── patient_information.py
    ├── appointments.py
    ├── doctors_and_medical_staff.py
    ├── medical_records.py
    ├── billing_and_finance.py
    ├── pharmacy.py
    ├── departments.py
    ├── room_and_bed_management.py
    ├── reports.py
    └── administration.py
```

app.py:

```
import streamlit as st

# Set the page title
st.set_page_config(page_title="Hospital Management System", layout="wide")

# Sidebar for navigation
st.sidebar.title("Navigation")
```

```
selection = st.sidebar.radio("Go to", [
    "Patient Information",
    "Appointments",
    "Doctors and Medical Staff",
    "Medical Records",
    "Billing and Finance",
    "Pharmacy",
    "Departments",
    "Room and Bed Management",
    "Reports",
    "Administration"
])

# Dynamic import of pages based on selection
if selection == "Patient Information":
    import pages.patient_information as page
elif selection == "Appointments":
    import pages.appointments as page
elif selection == "Doctors and Medical Staff":
    import pages.doctors_and_medical_staff as page
elif selection == "Medical Records":
    import pages.medical_records as page
elif selection == "Billing and Finance":
    import pages.billing_and_finance as page
elif selection == "Pharmacy":
    import pages.pharmacy as page
```

```
elif selection == "Departments":
    import pages.departments as page
elif selection == "Room and Bed Management":
    import pages.room_and_bed_management as page
elif selection == "Reports":
    import pages.reports as page
elif selection == "Administration":
    import pages.administration as page

# Check if the page has a main function and call it
if hasattr(page, 'main'):
    page.main()
else:
    st.error("The selected page does not have a main function.")
```

codes of other pages are little big mam

my git hub link for full code - https://github.com/nandini-queen-of-my-world/DBMS--AS3-Nandini-30-A

Than you