1. User Roles:

- Patient:
    - Can sign up with basic information.
    - Search for doctors by location or chamber.
    - Book appointments online.
    - Receive SMS notifications for appointment confirmation.
    - Access appointment history.
    - Can log in for future appointments.
- Doctor:
    - Must sign up with a detailed profile including available slots.
    - Choose pre or post payment options for appointments.
    - Define pre-payment options (full or partial) for each appointment.
    - Manage multiple chambers within a single profile.
    - View and manage appointment lists.
    - Receive notifications via email or dashboard.
    - Can log in to the system for management.
- Doctor Staff:
    - Can log in via the doctor portal.
    - View and manage appointment lists for specific chambers.
    - Access limited to a specific chamber or location.
- Super Admin:
    - Manages the overall system.
    - User management for patients, doctors, and staff.
    - Access to appointment data and statistics.
    - Subscription management for doctors.
    - System configuration settings.

2. Functionalities:

2.1 Patient Interface:

- Homepage:
    - Search bar for finding doctors by location or chamber.
    - Featured doctors or specialties.
    - Quick access to appointment booking.
- Doctor Search Results:
    - List of doctors with details.

        - Filter options for refining search results.
        - Book Appointment button for each doctor.
- Appointment Booking:
    - Calendar for selecting appointment date and time.
    - Patient information form (including password for account creation).
    - Payment options based on doctor-wise pre-payment setup.
    - Confirmation page with appointment details.
    - SMS notification for appointment confirmation.
- User Dashboard:
    - Upcoming appointments.
    - Appointment history.
    - Profile settings.

2.2 Doctor Interface:

- Dashboard:
    - Overview of upcoming appointments.
    - Quick access to appointment management.
    - Profile settings.
- Appointment Management:
    - Calendar view with appointment slots.
    - List of upcoming appointments with patient details.
    - Option to confirm, reschedule, or cancel appointments.
- Chamber Management:
    - List of chambers with details.
    - Ability to add/edit chamber details.
- Profile Settings:
    - Personal information.
    - Availability settings.
    - Subscription status and payment.

2.3 Doctor Staff Interface:

- Login:
    - Access to the Doctor Portal with unique credentials.
- Appointment List:
    - List of upcoming appointments for a specific chamber/location.
    - Option to mark appointments as attended or update status.
- Chamber Overview:

        - Details of the assigned chamber.
        - Quick access to chamber-related information.
- Profile Settings:
    - View and edit personal information.
    - Change password.

2.4 Super Admin Interface:

- Admin Dashboard:
    - Overview of the entire system.
    - User statistics, subscription status.
- User Management:
    - List of all users with roles and subscription statuses.
    - Option to add/remove users and update their details.
- Subscription Management:
    - Overview of subscription status for doctors.
    - Option to track and manage subscription payments.
- System Settings:
    - Configuration options for the overall system.
    - System health and performance.

3. Database Schema:

4. Future Scope:

- Telemedicine Integration:
    - Allow doctors to conduct virtual appointments.
- Prescription Management:
    - Enable doctors to electronically prescribe medications.
- Integration with Electronic Health Records (EHR):
    - Streamline patient information management.
- Enhanced Analytics:
    - Provide detailed statistics and insights for doctors and administrators.
- Advanced Notifications:
    - Implement more customizable and automated notification systems.
- Mobile App Enhancements:

        - Include additional features for a more seamless mobile experience.
- Billing and Payment Integration:
    - Facilitate online payment for appointment fees.

5. Subscription Model:

- Subscription Fee Management:
    - Admin can restrict doctor or staff accounts if subscription fee is not paid.
- Automated Subscription Renewal:
    - Implement an automated system for subscription fee renewal.
- Notifications for Subscription Expiry:
    - Provide alerts to doctors before subscription expiration.
- Billing History:
    - Enable doctors to view and download their billing history.
- Tiered Subscription Plans:
    - Offer different subscription tiers with varying features.
- Promotional Subscription Plans:
    - Implement promotional subscription plans for attracting new doctors.

API

1. Authentication:

- `POST /api/auth/login`: User login with email and password.
- `POST /api/auth/logout`: User logout.

2. Patient Operations:

- `POST /api/patients/signup`: Patients sign up with basic information.
- `GET /api/patients/{patientId}`: Retrieve patient details.
- `POST /api/patients/appointments/book`: Book an appointment.
- `GET /api/patients/appointments/{patientId}`: Retrieve patient's appointments.
- `POST /api/patients/login`: Patient login for future appointments.

3. Doctor Operations:

- `POST /api/doctors/signup`: Doctor sign up with detailed profile.
- `GET /api/doctors/{doctorId}`: Retrieve doctor details.
- `POST /api/doctors/appointments/create`: Create a new appointment slot.
- `GET /api/doctors/appointments/{doctorId}`: Retrieve doctor's appointments.
- `PUT /api/doctors/appointments/{appointmentId}`: Confirm, reschedule, or cancel an appointment.

4. Doctor Staff Operations:

- `POST /api/doctor-staff/login`: Doctor staff login via doctor portal.
- `GET /api/doctor-staff/appointments/{staffId}`: Retrieve staff's appointments.
- `PUT /api/doctor-staff/appointments/{appointmentId}`: Update appointment status.

5. Super Admin Operations:

- `POST /api/super-admin/login`: Super admin login.
- `GET /api/super-admin/users`: Retrieve a list of all users.
- `GET /api/super-admin/doctors`: Retrieve a list of all doctors.
- `GET /api/super-admin/subscriptions`: Retrieve subscription status.
- `PUT /api/super-admin/doctor-status/{doctorId}`: Update doctor's subscription status.

6. Subscription Management:

- `POST /api/subscriptions/pay`: Make a subscription payment.
- `GET /api/subscriptions/status/{userId}`: Check subscription status.
- `PUT /api/subscriptions/cancel/{userId}`: Cancel a subscription.

7. Future Scope (Example):

- `POST /api/telemedicine/consultation`: Initiate a telemedicine consultation.
- `POST /api/prescriptions/create`: Create an electronic prescription.
- `GET /api/ehr/patient/{patientId}`: Retrieve electronic health records for a patient.
- `GET /api/analytics/doctors/{doctorId}`: Retrieve analytics for a specific doctor.
- `POST /api/notifications/send`: Send custom notifications to users.

- `POST /api/mobile-app/feedback`: Collect feedback from mobile app users.
- `POST /api/billing/pay`: Initiate an online payment for appointment fees.

8. API Design Considerations:

- Follow RESTful principles for clear and consistent API design.
- Implement secure authentication mechanisms (e.g., OAuth, JWT).
- Include versioning in the API to manage changes over time (e.g., `/v1/`).
- Provide comprehensive documentation for each endpoint using tools like Swagger or OpenAPI.
- Ensure proper error handling and use appropriate HTTP status codes.
- Implement rate limiting to prevent abuse or misuse of the API.

Menu Concept for front end

Patient Frontend Screens:

Homepage:
Doctor Search Results:
Appointment Booking:
User Dashboard:

Doctor Frontend Screens:

Dashboard:
Appointment Management:
Chamber Management:
Profile Settings:

Doctor Staff Frontend Screens:

Login:
Appointment List:
Chamber Overview:
Profile Settings:

Super Admin Frontend Screens:

Admin Dashboard:
User Management:
Subscription Management:
System Settings:

Database Schema

```
-- Users Table: Common fields for Patient, Doctor, DoctorStaff, and SuperAdmin

CREATE TABLE Users (

    UserID INT PRIMARY KEY,

    UserType VARCHAR(20) NOT NULL, -- Patient, Doctor, DoctorStaff, SuperAdmin

    FirstName VARCHAR(50) NOT NULL,

    LastName VARCHAR(50) NOT NULL,

    Email VARCHAR(100) UNIQUE NOT NULL,

    Password VARCHAR(255) NOT NULL,

    PhoneNumber VARCHAR(15),

    SubscriptionStatus BOOLEAN DEFAULT TRUE, -- For Doctors (Subscription Status)

);



-- Patients Table: Additional fields for Patient

CREATE TABLE Patients (
```

```
    PatientID INT PRIMARY KEY,
    UserID INT UNIQUE,
    CONSTRAINT FK_Patient_User FOREIGN KEY (UserID) REFERENCES Users(UserID)
);

-- Doctors Table: Additional fields for Doctor
CREATE TABLE Doctors (
    DoctorID INT PRIMARY KEY,
    UserID INT UNIQUE,
    Specialization VARCHAR(50),
    -- Other Doctor-specific fields
    CONSTRAINT FK_Doctor_User FOREIGN KEY (UserID) REFERENCES Users(UserID)
);

-- DoctorStaff Table: Additional fields for Doctor Staff
CREATE TABLE DoctorStaff (
    StaffID INT PRIMARY KEY,
    UserID INT UNIQUE,
    DoctorID INT,
    -- Other Doctor Staff-specific fields
```

```
    CONSTRAINT FK_DoctorStaff_User FOREIGN KEY (UserID) REFERENCES
Users(UserID),

    CONSTRAINT FK_DoctorStaff_Doctor FOREIGN KEY (DoctorID) REFERENCES
Doctors(DoctorID)

);



-- Appointments Table

CREATE TABLE Appointments (

    AppointmentID INT PRIMARY KEY,

    DoctorID INT,

    PatientID INT,

    AppointmentDateTime DATETIME NOT NULL,

    Status VARCHAR(20) DEFAULT 'Scheduled', -- Scheduled, Confirmed, Canceled,
Completed

    -- Other Appointment-related fields

    CONSTRAINT FK_Appointment_Doctor FOREIGN KEY (DoctorID) REFERENCES
Doctors(DoctorID),

    CONSTRAINT FK_Appointment_Patient FOREIGN KEY (PatientID) REFERENCES
Patients(PatientID)

);



-- SuperAdmins Table: Additional fields for Super Admin

CREATE TABLE SuperAdmins (
```

```
    SuperAdminID INT PRIMARY KEY,

    UserID INT UNIQUE,

    CONSTRAINT FK_SuperAdmin_User FOREIGN KEY (UserID) REFERENCES
Users(UserID)

);



-- Chambers Table: Additional fields for Doctor's Chambers

CREATE TABLE Chambers (

    ChamberID INT PRIMARY KEY,

    DoctorID INT,

    Location VARCHAR(100) NOT NULL,

    -- Other Chamber-related fields

    CONSTRAINT FK_Chamber_Doctor FOREIGN KEY (DoctorID) REFERENCES
Doctors(DoctorID)

);



-- SubscriptionPayments Table: For tracking subscription payments

CREATE TABLE SubscriptionPayments (

    PaymentID INT PRIMARY KEY,

    UserID INT,

    Amount DECIMAL(10, 2) NOT NULL,

    PaymentDate DATETIME NOT NULL,
```

```
    CONSTRAINT FK_SubscriptionPayments_User FOREIGN KEY (UserID) REFERENCES Users(UserID)

);



-- Notifications Table: For storing notifications

CREATE TABLE Notifications (

    NotificationID INT PRIMARY KEY,

    UserID INT,

    Message TEXT NOT NULL,

    DateTime DATETIME NOT NULL,

    IsRead BOOLEAN DEFAULT FALSE,

    CONSTRAINT FK_Notifications_User FOREIGN KEY (UserID) REFERENCES Users(UserID)

);
```