**HealthConnect**

**1. Introduction**

HealthConnect is a web-based application designed to facilitate seamless appointment booking between doctors and patients. The system aims to provide users with an efficient way to schedule appointments based on their preferences, whether by choosing a specific doctor or selecting a hospital based on the medical condition.

**2. System Architecture**

The system follows a client-server architecture, with a web-based front-end for users (doctors and patients) and a back-end server handling data processing. The application utilizes a RESTful API to fetch hospital information and manage appointment bookings.

**3. User Roles**

**a. Patient**

- Register/Login: Patients can create accounts and log in to the system.

- Browse Doctors: View a list of registered doctors along with their specialties.

- Book by Doctor: Send appointment requests to specific doctors.

- Book by Hospital: Search hospitals based on location and medical condition.

- View Appointment Status: Check the status of appointment requests and confirmations.

**b. Doctor**

- Register/Login: Doctors can create accounts and log in to the system.

- Approve/Reject Appointments: Doctors can manage appointment requests from patients.

- View Appointment Schedule: Access a schedule of approved appointments.

- Update Profile: Maintain personal and professional information.

**4. Appointment Booking**

**a. Book by Doctor**

1. User selects the doctor from the list of registered doctors.

2. User sends an appointment request, including preferred date and time.

3. Doctor receives and manages appointment requests.

4. Doctor approves or rejects the request, and the user is notified.

**b. Book by Hospital**

1. User selects the medical condition and location (state).

2. System fetches a list of hospitals with available departments.

3. User selects a hospital and chooses a specific department.

4. System displays available appointments, ensuring a maximum of 5 per day for each department.

5. User selects an appointment slot, and the appointment is confirmed.

**5. Latest Health News**

The system integrates an API to fetch and display the latest health news. Users can access relevant health information, articles, and updates from reputable sources.

**6. Technologies Used**

- Front-end: Tailwind, JavaScript, React.js

- Back-end: Node.js, Express.js

- Database: MongoDB

- Authentication: JWT (JSON Web Tokens)

- External APIs: Fetch hospital information and health news.

**7. API Design**

* **Authentication APIs:**
  + /hc/signup: Register a new user.
  + /hc/login: Authenticate user credentials and generate a JWT token and log the user.
* **Booking APIs:**
  + /hc/book-doctor: To book an appointment for doctor.
  + /hc/book-hospital: To book an appointment for hospital.
* **Search APIs:**
  + /hc/search-doctor: To search for the doctor.
  + /hc/search-hospital: To search for hospital.
* **Appointment APIs:**
  + /hc/user-appointments: To list the appointments for the user logged in.
  + /hc/doctor-appointments: Get the list or schedule of appointments for the doctor logged in.
  + /hc/approve-appointments/:appointmentId : To approve the appointment by doctor/hospital.
  + /hc/reject-appointments/:appointmentId : To reject the appointment by doctor/hospital.
* **External APIs:** 
  + newsapi.org/v2/topheadlines/ : Top health headlines from India.
  + /resource/7d208ae4-5d65-47ec-8cb8-2a7a7ac89f8c : To Get data of Hospital and Doctors with Geo Code.

**8. Database Design for HealthConnect:**

1. Users Table:

* Fields:
  + **UserID** (Primary Key)
  + **Name**
  + **Contact details**
  + **Password**
  + **Role** (Patient or Doctor)
  + **Email**

2. Doctors Table:

* Fields:
  + **Doctor ID** (Primary Key)
  + **User ID** (Foreign Key referencing Users)
  + **Specialty**
  + **Location**

3. Hospitals Table:

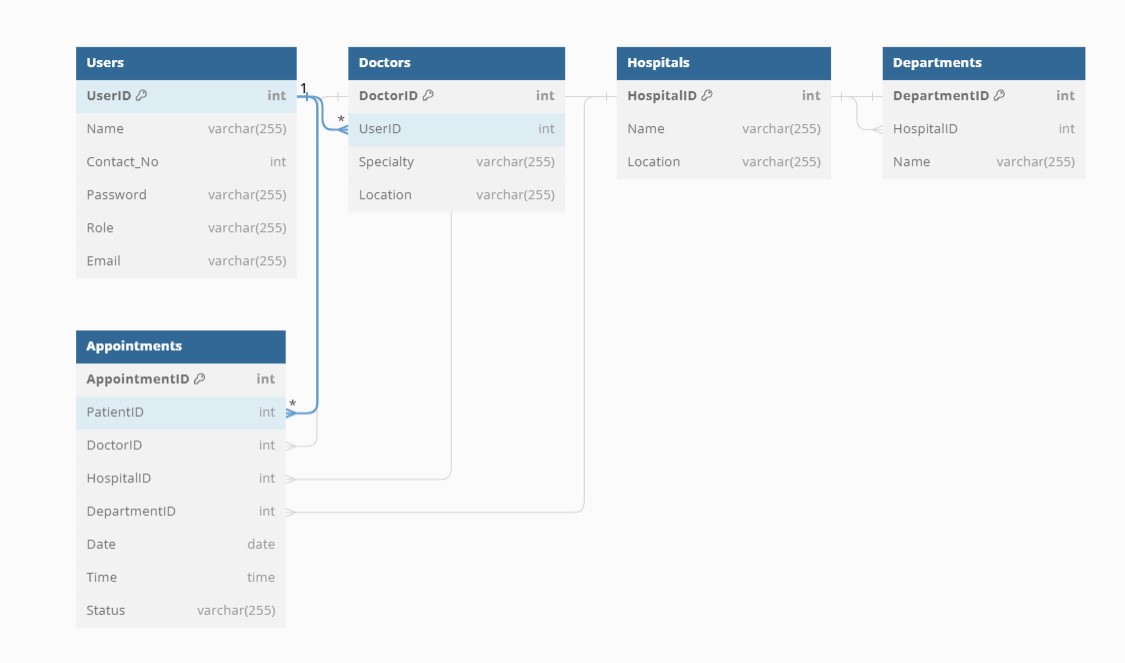
* Fields:
  + **Hospital ID** (Primary Key)
  + **Name**
  + **Location**

4. Departments Table:

* Fields:
  + **DepartmentID** (Primary Key)
  + **HospitalID** (Foreign Key referencing Hospitals)
  + **Name**

5. Appointments Table:

* Fields:
  + **AppointmentID** (Primary Key)
  + **PatientID** (Foreign Key referencing Users)
  + **DoctorID** (Foreign Key referencing Doctors)
  + **HospitalID** (Foreign Key referencing Hospitals)
  + **DepartmentID** (Foreign Key referencing Departments)
  + **Date**
  + **Time**
  + **Status** (Pending, Approved, Rejected)



**9. Security**

- User data is securely stored and transmitted using encryption.

- Authentication tokens are used to ensure secure user access.

- Regular security audits and updates to address vulnerabilities.

**10. Future Enhancements**

- Integration with telemedicine for virtual appointments.

- Enhanced user profiles with medical history and records.

- Notifications via email or SMS for appointment reminders.

**11. Deployment**

HealthConnect can be deployed on cloud platforms such as AWS, Google Cloud Platform, or Microsoft Azure. The Frontend can be deployed by Amazon S3, Netlify, or Vercel , while the backend can be deployed on platforms supporting Node.js applications like AWS Elastic Beanstalk or Heroku. MongoDB Atlas can be used for database hosting.

**12. Conclusion**

HealthConnect is designed to streamline the appointment scheduling process for both doctors and patients, providing a user-friendly interface and ensuring efficient communication and coordination in the healthcare domain.