Project Title:

Smart Health Tracker and Online Appointment System

Project Description:

This project aims to create a **web-based platform** that helps users monitor their health metrics, maintain medical records, and schedule appointments with doctors efficiently. The system will allow users to log their daily health stats (e.g., weight, blood pressure, sugar levels) and track trends over time using interactive charts. Additionally, it will provide a robust appointment booking module where users can search for doctors, view available slots, and book appointments.

The platform will consist of two primary modules:

1. **User Module:** For patients to track health, upload reports, and book appointments.
2. **Admin/Doctor Module:** For doctors to manage schedules, view patient history, and approve/reject appointments.

Key Features:

1. **User Registration and Login (PHP + MySQL):**
   - Secure user authentication (including password hashing).
   - Role-based access (Patient/Doctor).
2. **Health Tracking Dashboard (HTML, CSS, JS, PHP):**
   - Users can log daily health metrics.
   - Interactive visualization using **JavaScript charts (Chart.js/D3.js)**.
3. **Medical Record Upload (HTML, PHP, MySQL):**
   - Patients can upload prescriptions, test reports, and other medical documents.
   - Secure file storage and retrieval.
4. **Search and Book Appointments (PHP + MySQL):**
   - Search doctors by specialty, location, or availability.
   - Book, reschedule, or cancel appointments.
5. **Doctor's Dashboard (HTML, CSS, PHP, JS):**
   - View and manage appointment requests.

    - Access patient health records and history.

6. **Notification System (JS, PHP):**
    - Email or SMS notifications for appointment confirmations and reminders.
    - In-app notifications for upcoming appointments or health tracking alerts.
7. **Admin Module (HTML, CSS, PHP, MySQL):**
    - Manage users (patients and doctors).
    - View platform analytics (e.g., total appointments, active users).
8. **Responsive Design (HTML, CSS):**
    - Mobile-first approach using **CSS frameworks** like Bootstrap or Tailwind CSS.

Technology Stack:

- **Frontend:** HTML, CSS, JavaScript (with libraries like Chart.js or D3.js for data visualization).
- **Backend:** PHP.
- **Database:** MySQL for storing user, health, and appointment data.
- **Additional Tools:** AJAX for dynamic content loading, jQuery for simplified DOM manipulation.

Potential Innovations:

1. **AI Integration:** Use basic algorithms (optional) to provide health recommendations based on user logs (e.g., high blood pressure trends trigger lifestyle tips).
2. **Voice Commands:** Allow users to log health data or search for doctors using voice commands (leveraging JS APIs like Web Speech API).
3. **Geolocation-Based Search:** Enable users to find doctors nearest to their location using Google Maps API.

Deliverables:

1. Fully functional web application with intuitive UI.
2. User and Doctor dashboards tailored to specific needs.

3. Interactive and responsive health data visualizations.
4. Admin module for system management.
5. Comprehensive documentation for deployment and maintenance.

This project not only demonstrates technical skills across HTML, CSS, JavaScript, PHP, and MySQL but also addresses a real-world problem, making it impactful and innovative for final-year BTech students

Database Structure

Tables:

1. **Users**
   - user_id (Primary Key, Auto Increment)
   - name
   - email (Unique)
   - password (Hashed)
   - role (Patient or Doctor)
   - created_at
   - updated_at
2. **Health_Logs**
   - log_id (Primary Key, Auto Increment)
   - user_id (Foreign Key, references Users.user_id)
   - date
   - weight
   - blood_pressure
   - sugar_level
   - notes
3. **Appointments**
   - appointment_id (Primary Key, Auto Increment)

- patient_id (Foreign Key, references Users.user_id)
- doctor_id (Foreign Key, references Users.user_id)
- appointment_date
- time_slot
- status (Pending, Approved, Rejected)

4. **Medical_Records**
    - record_id (Primary Key, Auto Increment)
    - user_id (Foreign Key, references Users.user_id)
    - file_name
    - upload_date
5. **Notifications**
    - notification_id (Primary Key, Auto Increment)
    - user_id (Foreign Key, references Users.user_id)
    - message
    - type (Email, In-App)
    - created_at
6. **Analytics**
    - metric_id (Primary Key, Auto Increment)
    - total_users
    - total_doctors
    - appointments_booked
    - health_logs_count

Modules and Roles

1. User Module (Patient)

Role: Enable patients to log health data, view trends, upload medical records, and book appointments.

Pages:

- **Registration/Login** (Sign up, log in securely)
- **Dashboard** (View health stats and trends using charts)
- **Log Health Metrics** (Form to enter daily stats)
- **Upload Medical Records** (Upload and view prescriptions/reports)
- **Search Doctors** (Search and book appointments)
- **Manage Appointments** (View, reschedule, or cancel appointments)

---

2. Doctor Module

Role: Allow doctors to manage their schedules, view patient histories, and approve/reject appointments.

Pages:

- **Login** (Doctor-specific access)
- **Dashboard** (View appointment requests and patient health logs)
- **Appointment Management** (Approve, reject, or reschedule)
- **Patient Health History** (View uploaded records and health trends)

---

3. Admin Module

Role: Manage users, oversee platform analytics, and handle reports.

Pages:

- **Admin Login** (Secure admin access)
- **User Management** (Add/edit/delete patients or doctors)
- **View Analytics** (Charts and tables showing platform activity)
- **Notifications** (Send global updates or alerts)

---

4. Notification System

Role: Deliver email, SMS, and in-app notifications.

Functionality:

- Automated notifications for appointment confirmations, health tracking reminders, and reminders for upcoming appointments.

---

5. Responsive Frontend

Role: Create an intuitive and responsive interface for all user roles.

Pages:

- Mobile-friendly layouts for all dashboards.
- Interactive health tracking charts (using Chart.js or D3.js).

---

6. AI & Geolocation Features (Optional)

Role: Provide advanced functionalities like health tips, voice commands, and doctor search based on location.