**1. Patient Referral Management System**

**Objective**: A system that facilitates the referral process between general practitioners (GPs) and specialists.

**Features**:

* **Referral Generation**:
  + GPs can create and send electronic referrals to specialists, including patient medical history, symptoms, and test results.
* **Specialist Dashboard**:
  + Specialists can manage their incoming referrals, view patient history, and schedule consultations.
* **Appointment Integration**:
  + Once the specialist receives a referral, the system can automatically schedule the consultation and send reminders to patients.
* **Referral Tracking**:
  + Track the status of a referral (sent, accepted, scheduled, completed).
* **Inter-Doctor Communication**:
  + Doctors can communicate directly with each other through a secure messaging system within the platform.(optional)

**Technology Stack**:

* **Backend**: Node.js with Express for API handling referrals, managing appointments, and doctor interactions.
* **Frontend**: React.js for easy navigation between patient history, referrals, and scheduling.
* **Database**: MongoDB for storing patient referrals, doctor data, and scheduling info.
* **Authentication**: for doctor and patient authentication, ensuring secure communication.

**2. Clinical Trial Management System**

**Objective**: An application to manage and track clinical trials, from recruiting patients to monitoring their progress.

**Features**:

* **Patient Enrollment**:
  + Patients can apply to participate in clinical trials and submit necessary medical documentation.
* **Trial Scheduling**:
  + Researchers can schedule trial-related appointments and track patient progress.
* **Data Collection**:
  + Securely collect patient health data throughout the trial, including vitals, symptoms, and test results.
* **Trial Monitoring**:
  + Researchers can monitor trial progress, including participant compliance and adverse effects.
* **Compliance & Reporting**:
  + Generate compliance reports and monitor the trial's adherence to legal and ethical standards.
* **Patient Notifications**:
  + Patients receive reminders about upcoming trial appointments, data submission, and follow-ups.

**Technology Stack**:

* **Backend**: Node.js with Express to handle trial data, patient enrollments, and scheduling.
* **Frontend**: React.js for researchers to manage trials and for patients to track their trial progress.
* **Database**: MongoDB for storing trial data, patient info
* **Authentication**: for doctor and patient authentication, ensuring secure communication.

**3. Doctor Appointment and Waitlist Management System**

**Objective**: A system that helps doctors manage patient appointments and waitlists, ensuring a smooth workflow in a clinic or hospital setting.

**Features**:

* **Appointment Scheduling**:
  + Patients can schedule appointments online, choosing time slots based on doctor availability.
* **Waitlist Management**:
  + If a patient cancels an appointment, the system automatically places another patient from the waitlist in that slot.
* **Doctor Dashboard**:
  + Doctors can view upcoming appointments, check patient details, and manage their schedule.
* **Real-Time Availability**:
  + Integrates with a real-time calendar to show live availability and prevent double-booking.
* **Patient Notifications**:
  + Automated reminders for patients regarding upcoming appointments or cancellations.

**Technology Stack**:

* **Backend**: Node.js with Express to handle scheduling, patient data, and waitlist management.
* **Frontend**: React.js for a user-friendly interface for patients and doctors to manage appointments.
* **Database**: PostgreSQL for structured appointment and patient data.
* **Authentication**: for doctor and patient authentication, ensuring secure communication