Virtual Doctor Software

📐 Technology Responsibilities

🎨 Frontend (UI & User Experience)

- **HTML** → Basic structure of the web pages (skeleton).
- **CSS** → Custom styling, layout, colors, fonts.
- **Tailwind CSS** → Utility-first CSS framework to speed up design (responsive, clean UI, dark mode, etc.).
- **JavaScript** → Client-side interactivity (form validation, dynamic UI changes).
- **React.js** → Build reusable components (Patient Dashboard, Doctor Profile, Appointment Card).
- **Next.js** →
    - Server-side rendering (SSR) for SEO & performance.
    - Routing system for pages (/patient, /doctor, /admin).
    - API routes (small serverless functions if you don't want full Express).

🔐 Authentication & Notifications

- **Firebase** →
    - Authentication (Google login, Email/Password, OTP).
    - Push Notifications (appointment reminders).
    - Optional: Firestore for chat/messages if you don't want to use Mongo for that.

⚙️ Backend (Business Logic & API Layer)

- **Node.js** → Runtime environment to run JavaScript on the server.
- **Express.js** → Backend framework to build REST APIs (Appointments API, Prescription API, Payment API).
- **Authentication Middleware** → Verify JWT tokens (from Firebase).
- **Payment Integration** → Use Express routes to connect with payment providers.

🗄️ Database (Data Storage)

- **MongoDB** → Stores app data:
  - Patients (name, history, reports, login info).
  - Doctors (profile, specialty, availability).
  - Appointments (time, status, payment).
  - Prescriptions (medicine, dosage, doctor's notes).

🔄 Example Workflow

1. **Frontend (React + Next.js + Tailwind)** → Patient fills appointment form.
2. **API Call (Axios/Fetch)** → Data sent to **Express API**.
3. **Backend (Express + Node.js)** → Validates & stores appointment in **MongoDB**.
4. **Firebase Auth** → Ensures only logged-in users can book.
5. **Doctor Dashboard (React/Next.js)** → Fetches appointments from MongoDB API.
6. **Notifications (Firebase)** → Sends reminders to patient & doctor.

Virtual Doctor Software – Step-by-Step UI & Flow

1️⃣ Landing / Login Screen

Users: Patients, Doctors, Admin

UI Elements:

- Welcome message / app logo
- Login options: Email/password, Google, Phone OTP (Firebase)
- Sign-up link for new patients/doctors
- Minimal design (Tailwind makes it clean & responsive)

Flow:

1. User selects role (Patient / Doctor / Admin)
2. Logs in → redirected to respective dashboard

UI Idea:

- Left: Logo & welcome image
- Right: Login form + "Sign Up" link

2️⃣ Patient Dashboard

UI Elements:

- Sidebar / Top navigation: Home, Appointments, Symptom Checker, Medical Reports, Chat, Profile
- Main panel:
    - Upcoming appointments (list/cards)
    - Quick "Book Appointment" button

    - Symptom checker widget
- Notifications: Reminder alerts

Flow:

1. Patient sees upcoming appointments
2. Can click “Book Appointment” → opens doctor search + calendar
3. Can enter symptoms in AI Symptom Checker → see suggestions
4. Click appointment → join video call at scheduled time

UI Idea:

- Cards for appointments with doctor photo, time, status
- Floating “Check Symptoms” button

3️⃣ Symptom Checker / AI Assistant (Patient)

UI Elements:

- Textarea to enter symptoms
- Submit button → AI returns possible disease(s) + suggested medicine(s)
- Option to send AI suggestion to doctor for approval
- History of symptoms & AI suggestions
- Graph/chart of recurring symptoms over time

Flow:

1. Patient enters symptoms → submits
2. AI processes & returns prediction + medicines
3. Patient can request doctor review → added to prescription draft

UI Idea:

- Card for AI suggestion with clear color-coded confidence

- Button: “Send to Doctor for Review”

4️⃣ Patient Appointments

UI Elements:

- Calendar / list view
- Book new appointment: select doctor, date & time, optionally prepay
- Appointment cards: status (Pending, Confirmed, Completed)

Flow:

1. Search doctor by specialty, language, fee, rating
2. Select slot → confirm booking → payment (optional)
3. Appointment shows on dashboard

UI Idea:

- Cards with doctor photo + name + specialty + fee
- Calendar showing booked/free slots

5️⃣ Video Consultation (Patient & Doctor)

UI Elements:

- Remote video (large) + local video (small)
- Chat sidebar for messages & file sharing
- Prescription button for doctor
- Controls: mute/unmute, camera toggle, end call

Flow:

1. Patient joins video call at scheduled time
2. Doctor sees patient info + AI suggestion

3. Doctor fills prescription → generates PDF → stores in patient's history

UI Idea:

- Left: large remote video
- Right: chat + patient details + prescription button

6️⃣ Prescriptions & Medical Reports (Patient)

UI Elements:

- List of prescriptions (PDF download)
- Upload lab reports, imaging files
- Medication tracker with reminders

Flow:

1. Doctor submits prescription → patient downloads
2. Patient uploads reports → linked to medical history
3. Track medication intake with notifications

UI Idea:

- Grid view for reports/prescriptions
- Download / Upload buttons with clear icons

7️⃣ Doctor Dashboard

UI Elements:

- Sidebar: Dashboard, Appointments, Patients, Prescriptions, Profile
- Main panel:
    - Today's schedule (timeline/cards)
    - Patient list for video calls

- Pending AI suggestions

Flow:

1. Doctor sees today's appointments & AI symptom suggestions
2. Starts video consultation → writes / approves prescription
3. Views patient history and uploaded reports

UI Idea:

- Timeline view for daily schedule
- Patient detail panel on right
- Modal for prescription form

8️⃣ AI-Assisted Case Notes & Prescription (Doctor)

UI Elements:

- AI suggestion panel: disease + medicine + confidence
- Case notes editor
- Approve / Edit / Save buttons

Flow:

1. AI suggests diagnosis based on patient symptoms
2. Doctor reviews → edits or approves → generates prescription PDF
3. Prescription stored in patient's history

UI Idea:

- Side panel with AI suggestions
- Editable prescription modal with structured fields

9️⃣ Admin Dashboard

UI Elements:

- Sidebar: Dashboard, Doctors, Patients, Appointments, Reports, Payments, Analytics
- Main panel:
  - KPIs: revenue, appointments, top specialties
  - Pending doctor verification → approve/reject
  - System-wide analytics

Flow:

1. Admin logs in → sees system KPIs
2. Approves pending doctor registrations
3. Monitors revenue, appointments, and patient safety metrics

UI Idea:

- Charts for revenue and appointments (Recharts / Chart.js)
- Tables for pending doctor approvals
- Buttons for approve/reject actions

1 0 Notifications & Messaging

UI Elements:

- Real-time chat (patient-doctor, family-doctor)
- Push notifications (appointment reminders, AI suggestions, lab results)
- Inbox panel with unread indicators

Flow:

1. Send/receive messages during video or outside consultation
2. Receive reminders for appointments, medications, follow-ups

UI Idea:

- Sidebar for chats
- Popup notifications for important alerts

1️⃣1️⃣ Advanced Modules (Optional / Future-Proof)

Features:

- IoT vitals integration (BP, sugar, temperature)
- AI 24/7 Chatbot for triage
- Multi-language support (English / Bengali)
- Offline Mode / PWA
- Feature flags to enable/disable modules
- Recording & transcription (optional)
- Multi-clinic support

UI Idea:

- Small dashboard widgets for vitals
- Toggle buttons for offline mode or language
- Chatbot panel as floating button

1️⃣2️⃣ Full Software Flow Summary

Patient:

Login → Dashboard → Symptom Checker → Book Appointment → Video Call → Prescription → Upload Reports → Analytics

Doctor:

Login → Dashboard → AI Suggestion Review → Video Call → Prescription → Patient History → Analytics

Admin:

Login → Dashboard → Doctor Verification → Patients → Appointments → Payments → Reports / Analytics

- **Tech Stack & Roles in Software**

Stack / Tool	Role in Software
React + Next.js + Tailwind	Patient/Doctor/Admin UI – behaves like desktop software
Node.js + Express	Backend logic, AI integration, appointment/payment handling
MongoDB	Store users, appointments, prescriptions, reports, chat logs
Firebase	Auth, Storage, Notifications
WebRTC / Twilio / PeerJS	Video consultations
OpenAI / AI Engine	Symptom analysis, disease prediction, prescription suggestions
Chart.js / Recharts	Dashboard graphs & analytics

Optional IoT	Capture vitals, integrate with patient dashboard