



**CSE471: System Analysis and Design**  
**Assignment on Functional Requirements**  
**Proposed Project Title: Smart Clinic**

<b>Group No: 03, CSE471 Lab Section: 10, Spring 2025</b>	
<b>ID</b>	<b>Name</b>
24241375	Munira Rawshan
24141079	Nahian Al Fiza
24141080	Ahanaf Abrar
21201653	Enamul Haque Shafin

**Submission Date: 03 November, 2025**

# Tech Stack

- **Language:** JavaScript (TypeScript optional)
- **Frontend:** React.js (Vite) + React Router
- **Styling:** TailwindCSS (+ Anime.js for micro-animations)
- **Backend:** Node.js + Express.js
- **Real-time:** Socket.IO (queues + notifications)
- **Database:** MongoDB Atlas (NoSQL)
- **ODM:** Mongoose
- **Auth:** JWT (httpOnly) + Google Sign-In (OAuth)
- **File Storage (reports):** S3-compatible (Cloudflare R2 / AWS S3) via presigned URLs
- **Email:** Resend or SendGrid (booking/reminder & prescription emails)
- **External APIs:** RxNav (drug interactions), HuggingFace (health tips)
- **Deployment:**
  - Frontend: Vercel or Netlify
  - Backend: Render or Railway
  - Database: MongoDB Atlas
  - Object Storage: R2/S3

# Functional Requirements

## Module 1: Profile Management

- ❖ [Munira] Patients and doctors can create accounts by entering name, email, and password; secure login/logout with **Google Sign-In (OAuth)** as an alternative. Successful login routes users to their personalized dashboards.
- ❖ [Nahian] Users can view and update profile information. Doctors can update specialization, clinic details, and weekly availability. Patients can manage preferred doctors.
- ❖ [Shafin] Admin can manage user roles (patient, doctor, admin) and permissions (RBAC), including activate/deactivate and promote/demote.
- ❖ [Ahanaf] Patients and doctors can view a past activity timeline, including previous appointments, prescriptions, and uploaded reports.

## Module 2: Appointments & Queue

- ❖ [Munira] Patients can search doctors, view date-based availability, book appointments, and cancel if needed. Doctors can accept or reject appointment requests.
- ❖ [Nahian] The system shows real-time doctor availability and assigns a serial number for hospital queues (walk-ins or online bookings), updating clients live via [Socket.IO](#).
- ❖ [Shafin] Doctors can open an appointment to see patient context (history, prior prescriptions, and reports) and mark visits as completed or no-show.
- ❖ [Ahanaf] Advanced search and filter APIs let users find doctors or appointments by name, specialty, or date; patient/doctor history endpoints power these views.

## Module 3: Consultation, Dashboards, and Notifications

- ❖ [Munira]
  - Feature 1: A consultation screen where doctors review the current visit and create a **print-ready HTML prescription view** (browser “Save as PDF” supported).

- Feature 2: Two dashboards—Doctor Dashboard (upcoming appointments, live queue, recent prescriptions, reports) and Patient Dashboard (booked appointments, upcoming visits, prescriptions, reports, health tips).

❖ [Nahian]

- Feature 1: Secure report uploads and viewing (PDF/images) using presigned links to object storage.
- Feature 2: Email notifications for booking confirmations/updates and time-based reminders with downloadable .ics files and Google Calendar add links. (External API: Resend/SendGrid)

❖ [Shafin]

- Feature 1: Structured prescription creation and management (medicines, dosage, frequency, duration) with audit logging.
- Feature 2: **Prescription email delivery** to patients containing a link to the HTML prescription page. (External API: Resend/SendGrid)

❖ [Ahanaf]

- Feature 1: **RxNav** integration suggests medicines and warns about potential drug interactions during prescribing.
- Feature 2: Health tips generated via a lightweight **HuggingFace** model and an in-app notifications center for appointment reminders and updates.