Scalable Multi-Clinic Management System - Product & Technical Specification (Django + HTML)

1. System Overview (Non-Technical Explanation)

This system is designed as a platform where multiple hospitals or clinics can register and manage their daily operations digitally. Each hospital gets its own secure workspace. Receptionists register patients, doctors provide consultation and prescriptions, and administrators monitor overall operations. The platform owner (Super Admin) manages multiple hospitals from a central dashboard.

2. High-Level Workflow

1. Hospital registers on platform
2. Hospital Admin configures hospital details, logo and doctors
3. Receptionist registers patients
4. Appointments and tokens created
5. Doctor consultation and prescription
6. Test recommendation and report uploads
7. Admission and treatment tracking (if required)
8. Discharge process
9. Follow-up scheduling
10. Dashboard analytics and reports

3. User Roles

1. Super Admin – Platform owner managing all hospitals and subscriptions.
2. Clinic Admin – Manages hospital settings, staff, reports.
3. Doctor – Consultation, treatment, prescriptions.
4. Receptionist – Patient registration, appointment management.

4. User Stories (Simple Language)

1. As a Hospital Owner, I want to register my hospital so I can manage patients digitally.
2. As a Receptionist, I want to register new patients quickly.
3. As a Doctor, I want to see patient history before consultation.
4. As a Doctor, I want to create prescriptions and suggest tests.
5. As an Admin, I want to see how many patients were treated or admitted.
6. As a Super Admin, I want to manage multiple hospitals from one dashboard.

5. Database Schema (Simple Explanation)

Each hospital is treated as a tenant using clinic_id. All tables include clinic_id to separate hospital data securely.

Table Name	Purpose
clinics	Stores hospital details, logo, branding information
users	All login users with roles (doctor/admin/receptionist)
patients	Patient profile and history
appointments	Doctor schedule and patient tokens
consultations	Doctor notes, symptoms and diagnosis
prescriptions	Medicine details and dosage
test_requests	Recommended medical tests
test_reports	Uploaded lab reports
admissions	Patient admission records
treatment_logs	Daily treatment tracking
discharges	Discharge summaries
followups	Routine checkup scheduling

6. Screens & Dashboards

1. Hospital Registration Screen – Hospital enters name, logo, address, contact details.
2. Doctor Dashboard – Today's appointments, patient history, prescription creation.
3. Reception Dashboard – Register patient, manage queue and tokens.
4. Clinic Admin Dashboard – Treated patients count, admitted patients, revenue and reports.
5. Super Admin Dashboard – All clinics overview, subscriptions, system analytics.
6. Patient Registration Screen – Add demographics and medical history.
7. Prescription Screen – Medicine list, dosage, downloadable PDF.
8. Admission Screen – Admit patient, assign bed or ward.
9. Treatment Monitoring – Daily updates and treatment notes.

7. Analytics Dashboard (Hospital Insights)

1. Total patients treated
2. Currently admitted patients
3. Daily/Monthly patient visits
4. Doctor performance summary
5. Follow-up pending patients
6. Revenue overview (optional)

8. Scalable Architecture Approach

The system should be designed as a multi-tenant SaaS platform using Django backend. Each hospital has isolated data using clinic_id. Role-based permissions ensure security. The architecture should support adding new modules like AI recommendations, WhatsApp notifications, and white-label branding without redesigning the system.