Project Report

Project Title : Digital Hospital Management System

Organization: Lovely Professional University

Prepared For: Dr. Siddardha Dasari

Prepared By:

S no.	Name	Registration No.
1	Amanjot Singh	12309766
2	Roshan Pal	12325961
3	Palika Ghai	12310349

Date: 20-Feb-2026

1. Executive Summary

The Hospital Management System (HMS) is a comprehensive web-based healthcare management platform developed to modernize and digitize hospital operations. The system is designed to replace traditional paper-based processes with a centralized, automated, and efficient digital solution that improves workflow coordination between patients, pharmacists, and hospital staff.

In conventional hospital environments, manual record-keeping and fragmented systems often lead to data inconsistencies, prescription errors, delayed billing, and inefficient communication between departments. These challenges reduce operational efficiency and negatively impact patient satisfaction. The proposed Hospital Management System addresses these issues by integrating patient management, prescription handling, pharmacy operations, inventory management, and billing processes into a unified platform.

The primary objective of this project is to develop a secure and scalable web application that simplifies hospital operations while enhancing data accuracy and service delivery speed. By utilizing modern web technologies such as Python and Django for backend development and HTML/CSS for frontend design, the system ensures structured workflow management, secure authentication, and reliable database handling.

The system provides the following core functionalities:

- **Secure patient registration and login system**
- **Digital prescription viewing and uploading**
- **Online medicine ordering with prescription verification**
- **Pharmacy dashboard for order management and inventory control**
- **Automated shopping cart and bill generation system**
- **Role-based access control for different hospital departments**

The application follows a modular architecture, enabling easy scalability and future enhancements such as payment gateway integration, real-time notifications, analytics dashboards, and cloud deployment.

The expected outcome of this project is a user-friendly, secure, and efficient hospital management solution that reduces administrative workload, minimizes errors, improves inventory tracking, and enhances overall patient experience. The system demonstrates how digital transformation can significantly improve healthcare service delivery and operational transparency.

2. Project Overview

Business Problem Statement

Traditional hospital management systems often rely on fragmented software solutions and manual record-keeping processes, which result in operational inefficiencies, data redundancy, billing inaccuracies, delayed prescription processing, and poor coordination between departments such as pharmacy, reception, and patient services. These legacy workflows reduce service speed and compromise data integrity, especially when handling prescription uploads, medicine inventory management, and billing automation.

The technical challenge was to design and implement a secure, database-driven hospital management platform capable of handling patient authentication, prescription uploads, pharmacy inventory control, shopping cart billing workflows, and order fulfillment processes within a centralized, scalable, and modular architecture. The system needed to ensure secure session-based authentication, structured database relationships, controlled access to sensitive modules, and seamless workflow transitions between patient and pharmacy interfaces.

Project Objectives

Objective ID	Description	Key Deliverable
O-01	Secure Role-Based Authentication	Implement a session-based authentication system using Django’s framework, ensuring user credentials are securely managed with form validation, CSRF protection, and controlled dashboard access for Patients, Pharmacists, and Receptionists.
O-02	Prescription & File Management	Develop a secure file upload system using Django’s MEDIA handling to allow patients to upload prescriptions while ensuring structured storage and retrieval by the pharmacy module.
O-03	Pharmacy Inventory & Order Workflow	Design a relational database schema (models.py) to manage medicines, shopping cart items, and billing workflows while maintaining real-time stock updates and accurate order tracking.

Scope:

Category	Description	Status
Authentication	User registration, login, session management, and role-based access control.	In Scope
Prescription System	Secure prescription upload, viewing, and pharmacy access.	In Scope
Pharmacy Dashboard	Medicine CRUD operations, inventory management, cart handling, and bill generation.	In Scope
Billing & Cart Module	Dynamic cart calculation, order fulfillment, and bill completion workflow.	In Scope
Online Payment Gateway	Integration with external payment systems (Razorpay/Stripe).	Out-of-Scope (Phase 1)
AI-driven Analytics	Predictive analytics, automated stock forecasting, or ML-based recommendation systems.	Out-of-Scope (Future Enhancement)

Key Deliverables & Success Criteria

The project success was measured across three primary dimensions:

● Functional Codebase:

A fully operational Django-based Hospital Management System implementing secure authentication, prescription upload handling, pharmacy inventory management, and automated billing workflow.

● Technical Documentation:

Comprehensive documentation including architecture design, database schema explanation, system workflow, and security posture analysis.

● System Performance & Reliability:

Efficient database queries using Django ORM, elimination of redundant queries, structured model relationships, and stable session-based authentication without unauthorized access to restricted modules.

3. Solution Architecture & Design

3.1 System Overview

The foundational architecture of the Hospital Management System (HMS) adheres to a Modular Multi-Tier Architecture Pattern, built upon Django's Model-View-Template (MVT) framework. The system is designed to ensure separation of concerns between presentation logic, application processing, and data persistence layers while maintaining scalability and security.

The system operates primarily through a synchronous server process:
Synchronous WSGI Layer:
Handles structured HTTP request-response cycles including user registration, login authentication, prescription uploads, medicine inventory management, cart operations, and billing workflows. This layer processes user requests through Django views and renders dynamic templates based on role-based access permissions.

Unlike real-time chat systems, the HMS platform is workflow-driven rather than event-stream driven. Therefore, the architecture focuses on transactional consistency, secure session handling, and structured database interaction instead of persistent WebSocket communication.

The application enforces role-based separation across:

- **Patient Interface**
- **Pharmacy Dashboard**
- **Reception Management Module**

Each module communicates with the centralized backend while maintaining controlled access through session validation and middleware enforcement.

The system follows a three-layer logical separation:

1. **Presentation Layer (HTML/CSS Dashboards)**
2. **Application Layer (Django Business Logic)**
3. **Data Layer (Relational Database using ORM)**

3.2 Technology Stack

Component Category		Technology
Backend Framework		Python 3.x, Django
Frontend Interface		HTML5, CSS3
Primary Database		SQLite (Development), PostgreSQL-ready
Authentication Module		Django Session-Based Authentication
File Management		Django MEDIA Handling
Email Integration		Django SMTP (send_mail)
Development Environment		VS Code, Git, Django Admin

3.3 Integration Points

● File Storage (Prescription Upload):
User-uploaded prescription documents are stored securely using Django's MEDIA configuration. Files are saved in a dedicated media directory on the server, and only file paths are stored in the relational database. This ensures structured file handling without bloating database storage.

● Email Notification System:
The system integrates Django's SMTP utilities to send automated email notifications upon user registration and order confirmation. Email dispatch is handled within the application layer to maintain workflow continuity.

● Session-Based Access Control:
All dashboard modules enforce controlled access via session validation. Unauthorized users attempting to access restricted endpoints are redirected to the login interface.

3.4 Security

The system's security posture is based on layered protection mechanisms implemented through Django's framework:

● Session-Based Authentication:
User authentication is managed using Django’s secure session framework. Session identifiers are stored server-side to prevent credential exposure and unauthorized access.
● CSRF Protection:
All form submissions utilize Django’s CSRF middleware, which enforces randomized tokens for state-changing HTTP requests, mitigating Cross-Site Request Forgery attacks.
● Input Validation & ORM Protection:
All database interactions are handled through Django ORM, preventing raw SQL query execution and mitigating SQL injection vulnerabilities.
● Controlled File Upload Handling:
Prescription file uploads are validated through server-side checks to ensure safe file handling and structured storage.
● Role-Based Access Enforcement:
Dashboard routes are protected to ensure that only authorized users (Patients, Pharmacists, Receptionists) can access their respective modules.

3.5 Scalability

The architecture is designed with forward scalability considerations:
● Modular Application Structure:
The system is divided into separate Django applications (patients, pharmacy, accounts), allowing independent scaling and maintainability.
● Database Migration Readiness:
Although SQLite is used during development, the database schema is fully compatible with PostgreSQL or MySQL for production deployment.
● Optimized Database Queries:
The system utilizes efficient ORM querying strategies to minimize redundant database calls and prevent inefficient data retrieval patterns.
● Horizontal Deployment Capability:
The application can be deployed using WSGI servers such as Gunicorn combined with Nginx, enabling horizontal scaling across multiple instances in a production environment.

Architectural Summary
The Hospital Management System demonstrates a structured, secure, and modular backend architecture that efficiently handles authentication, prescription uploads, inventory management, and billing workflows. The design prioritizes transactional integrity, secure access control, and scalable deployment readiness, ensuring that the system can evolve into a production-grade .

4. Implementation Plan (9-Day Timeline)

The Hospital Management System (HMS) was developed within a focused 9-day intensive development cycle. The implementation was divided into four structured phases to ensure systematic progress, risk control, and feature validation.

Phase Breakdown

Phase / Days	Focus Area	Key Activities	Risk Assessment & Mitigation
Phase 1: Days 1–2	Requirements Analysis, Architecture Setup & Authentication Module	Finalize system architecture (Django MVT structure). Create project structure and apps (patients, pharmacy, accounts). Implement database schema (models.py). Develop secure registration and login functionality with session handling. Configure SQLite and Django Admin panel.	Risk: Database migration conflicts — Mitigation: Strict execution of version-controlled migrations (makemigrations / migrate) after every model change.
Phase 2: Days 3–5	Core Development: Pharmacy Module & Inventory Management	Implement Medicine Model with full CRUD operations. Develop Pharmacy Dashboard UI. Create Shopping Cart and Order Models. Implement cart logic and total price calculation. Integrate role-based dashboard routing.	Risk: Incorrect stock calculation or data inconsistency — Mitigation: Implement validation checks before order confirmation and enforce controlled database updates via ORM.
Phase 3: Days 6–7	Prescription Upload & Order Workflow Integration	Implement Prescription Model and file upload handling using Django MEDIA configuration. Develop patient interface for	Risk: File upload vulnerabilities — Mitigation: Enforce file validation, controlled

		uploading prescriptions. Link prescription orders to pharmacy dashboard. Integrate billing generation workflow. Add email notification functionality for registration and order confirmation.	storage paths, and server-side validation checks.
Phase 4: Days 8–9	UI Refinement, Testing & Quality Assurance	Refine dashboard UI/UX (Landing Page, Login, Pharmacy Dashboard). Conduct unit testing of authentication and cart workflow. Test order-to-bill lifecycle. Validate session-based access control. Finalize project documentation and performance checks.	Risk: Unauthorized dashboard access — Mitigation: Implement session validation checks on all protected routes and verify CSRF protection across forms.

Resource Allocation

The 9-day development sprint required efficient utilization of software tools, development frameworks, and infrastructure resources to ensure timely and high-quality project delivery.

● Software and Development Tools

○ Backend Framework: Python 3.x, Django
○ Frontend Technologies: HTML5, CSS3
○ Database: SQLite (Development Environment)
○ Email Integration: Django SMTP (send_mail)
○ Source Control: Git & GitHub
○ Development Environment: Visual Studio Code

• Infrastructure

○ Development Environment:
Local development server using Django's built-in development server for rapid prototyping and testing.
○ Testing Environment:
Local browser-based testing across multiple modules including authentication, cart operations, billing workflow, and file upload functionality.
○ Production Readiness (Future Scope):
The system architecture supports migration to PostgreSQL and deployment using Gunicorn + Nginx on cloud platforms such as AWS or Heroku.

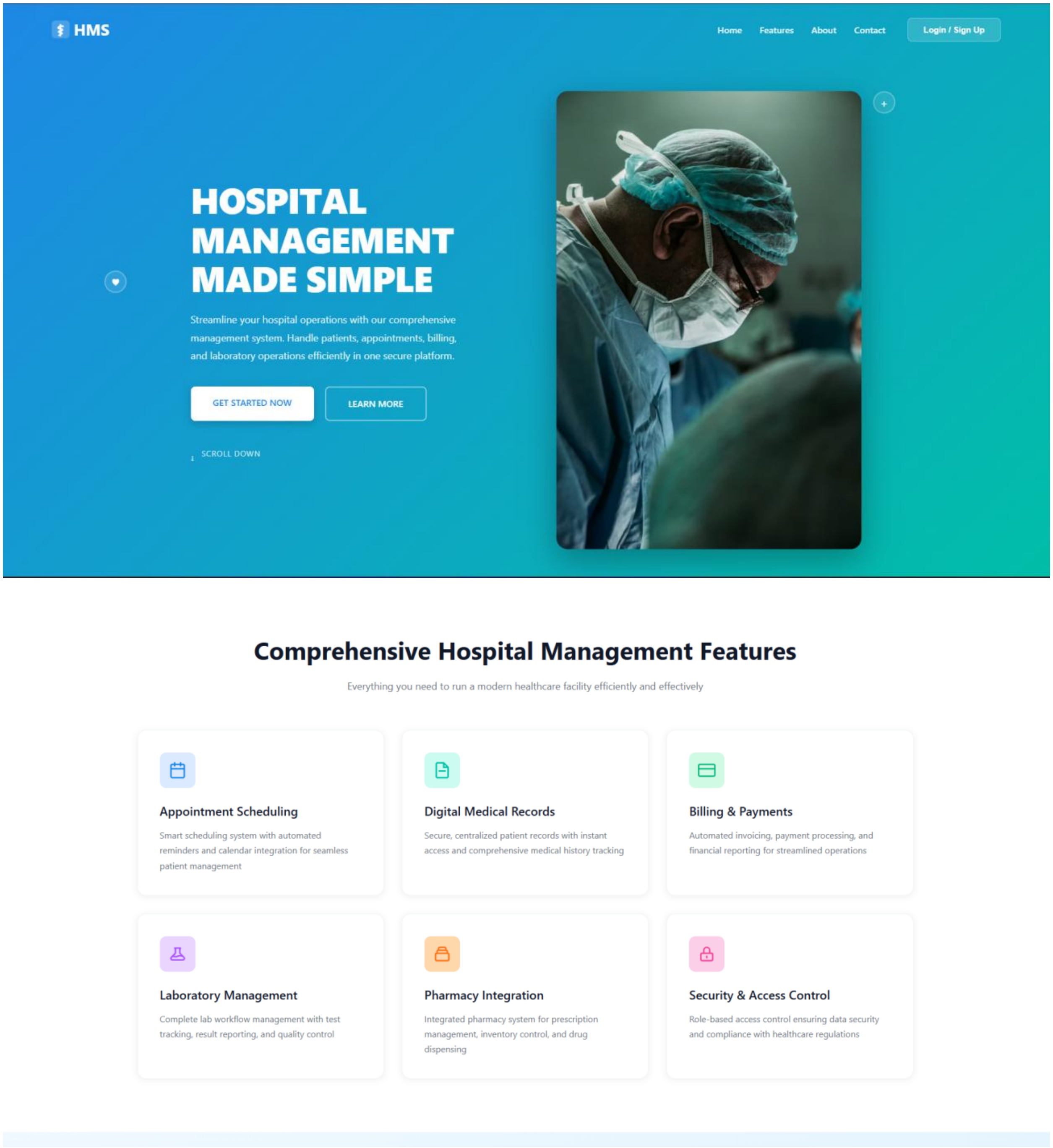
5. Development Team Introduction

The 9-day project was executed by a specialized team of four, with roles designed for maximum focus and efficiency across the key architectural domains:

Role	Name	Experience	Responsibilities
Project Lead & Backend Architect	**Amanjot**	**Student**	**Overall system architecture design, database schema modeling (models.py), authentication system implementation, session management, and core business logic development.**
Isne kuch nhi kiya	**Palika**	**Student**	**Design and implementation of responsive dashboards (Patient, Pharmacy, Landing Page), HTML/CSS styling, user interface optimization, and form validation integration.**
Database & Workflow Integration Developer	**Roshan**	**Student**	**Implementation of pharmacy inventory module, cart & billing workflow logic, prescription upload handling, testing, and integration between patient and pharmacy modules.**

6.ScreenShots of website

1-Home Page



HOSPITAL MANAGEMENT MADE SIMPLE

Streamline your hospital operations with our comprehensive management system. Handle patients, appointments, billing, and laboratory operations efficiently in one secure platform.

Comprehensive Hospital Management Features

Everything you need to run a modern healthcare facility efficiently and effectively

Appointment Scheduling

Smart scheduling system with automated reminders and calendar integration for seamless patient management

Digital Medical Records

Secure, centralized patient records with instant access and comprehensive medical history tracking

Billing & Payments

Automated invoicing, payment processing, and financial reporting for streamlined operations

Laboratory Management

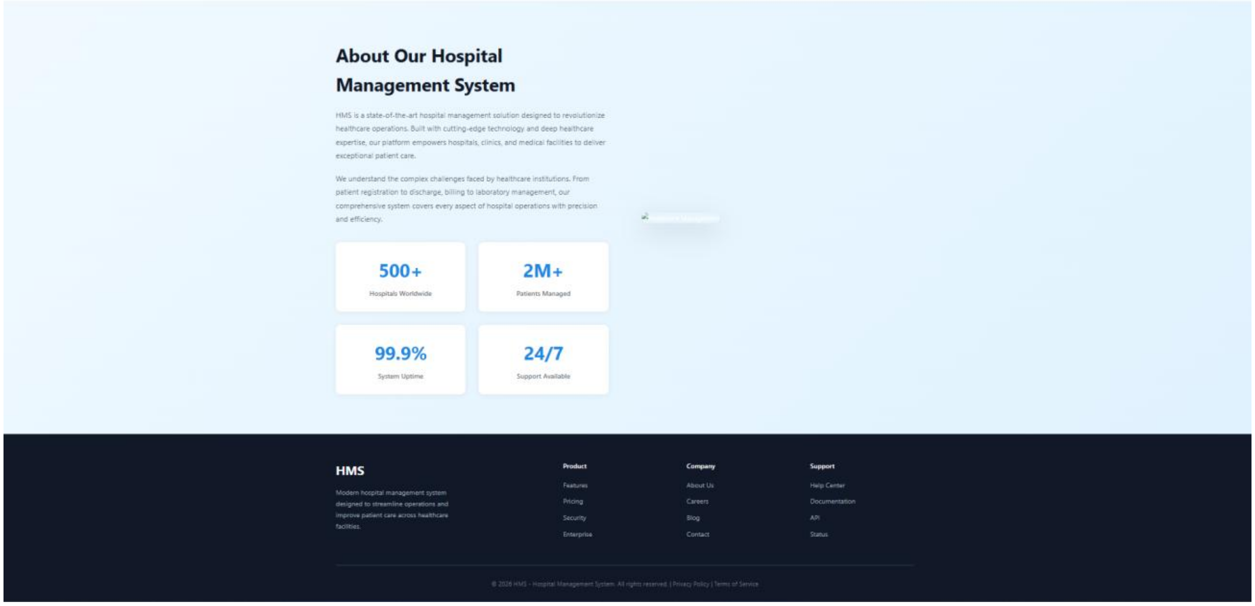
Complete lab workflow management with test tracking, result reporting, and quality control

Pharmacy Integration

Integrated pharmacy system for prescription management, inventory control, and drug dispensing

Security & Access Control

Role-based access control ensuring data security and compliance with healthcare regulations



About Our Hospital Management System

HMS is a state-of-the-art hospital management solution designed to revolutionize healthcare operations. Built with cutting-edge technology and deep healthcare expertise, our platform empowers hospitals, clinics, and medical facilities to deliver exceptional patient care.

We understand the complex challenges faced by healthcare institutions. From patient registration to discharge, billing to laboratory management, our comprehensive system covers every aspect of hospital operations with precision and efficiency.

500+
Hospitals Worldwide

2M+
Patients Managed

99.9%
System Uptime

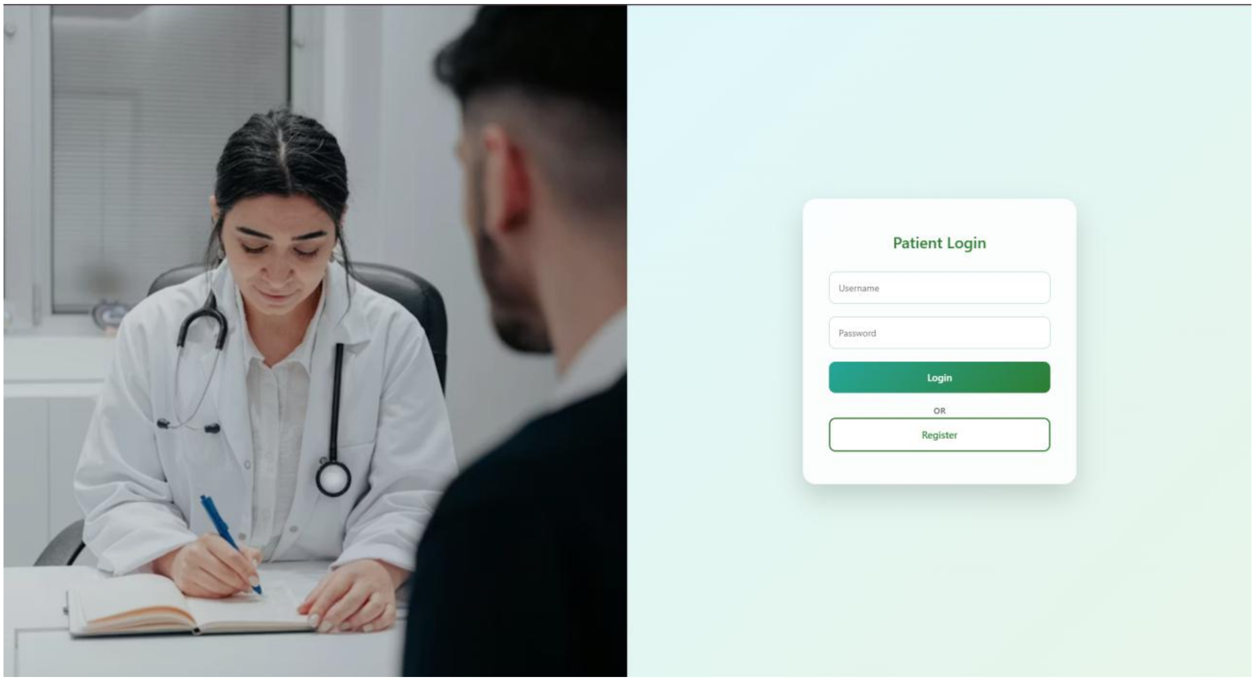
24/7
Support Available

HMS

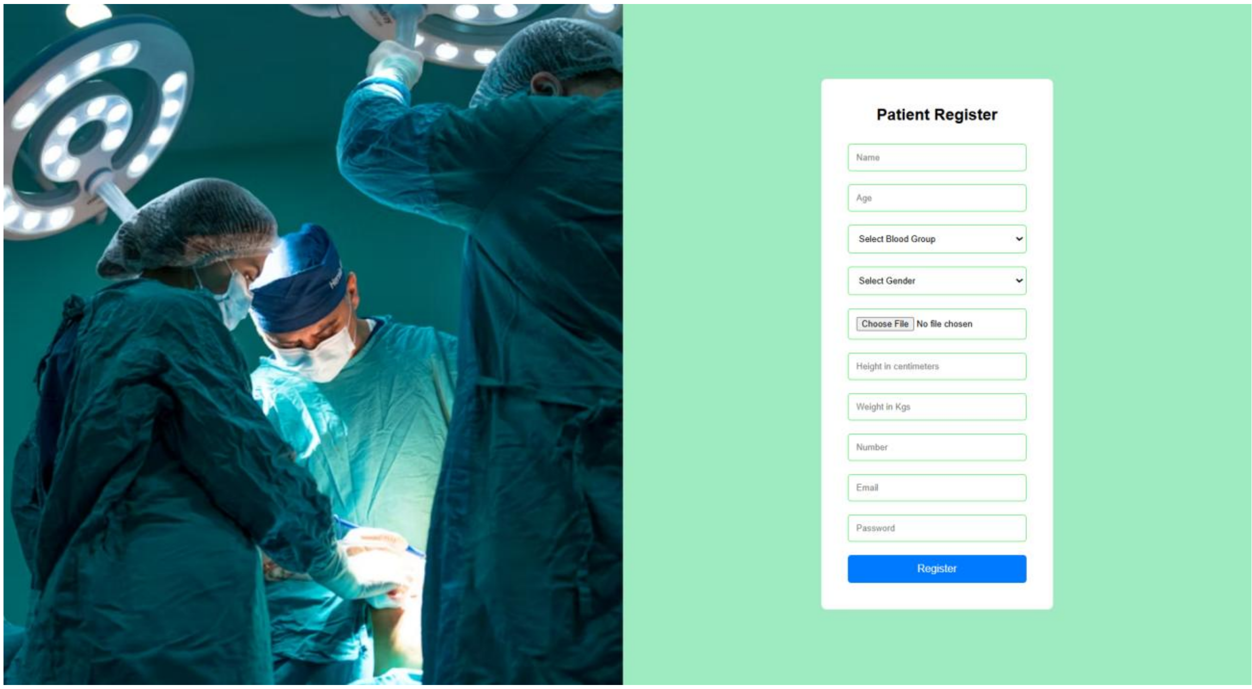
Modern hospital management system designed to streamline operations and improve patient care across healthcare facilities.

© 2026 HMS - Hospital Management System. All rights reserved. | Privacy Policy | Terms of Service

2-Patient(login page & home page)



Patient Login



Patient Register

Name

Age

Select Blood Group

Select Gender

Choose File No file chosen

Height in centimeters

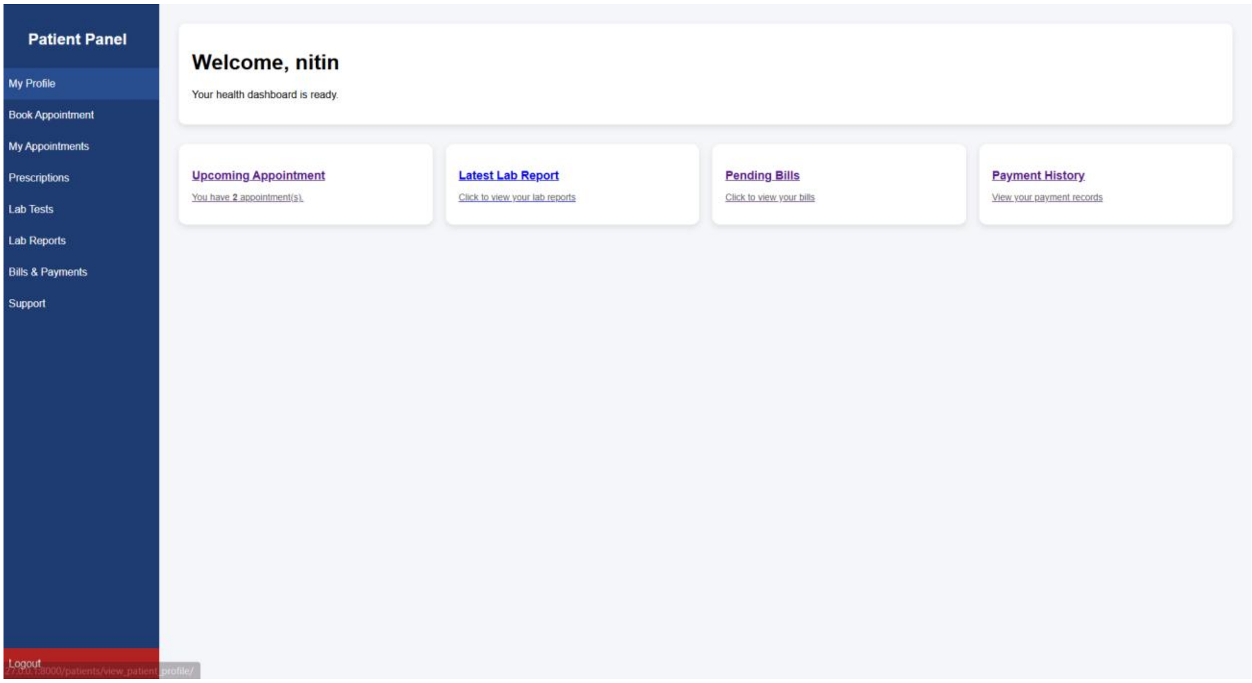
Weight in Kgs

Number

Email

Password

Register



Patient Panel

- My Profile
- Book Appointment
- My Appointments
- Prescriptions
- Lab Tests
- Lab Reports
- Bills & Payments
- Support

Welcome, nitin

Your health dashboard is ready.

Upcoming Appointment
You have **2** appointment(s).

Latest Lab Report
Click to view your lab reports

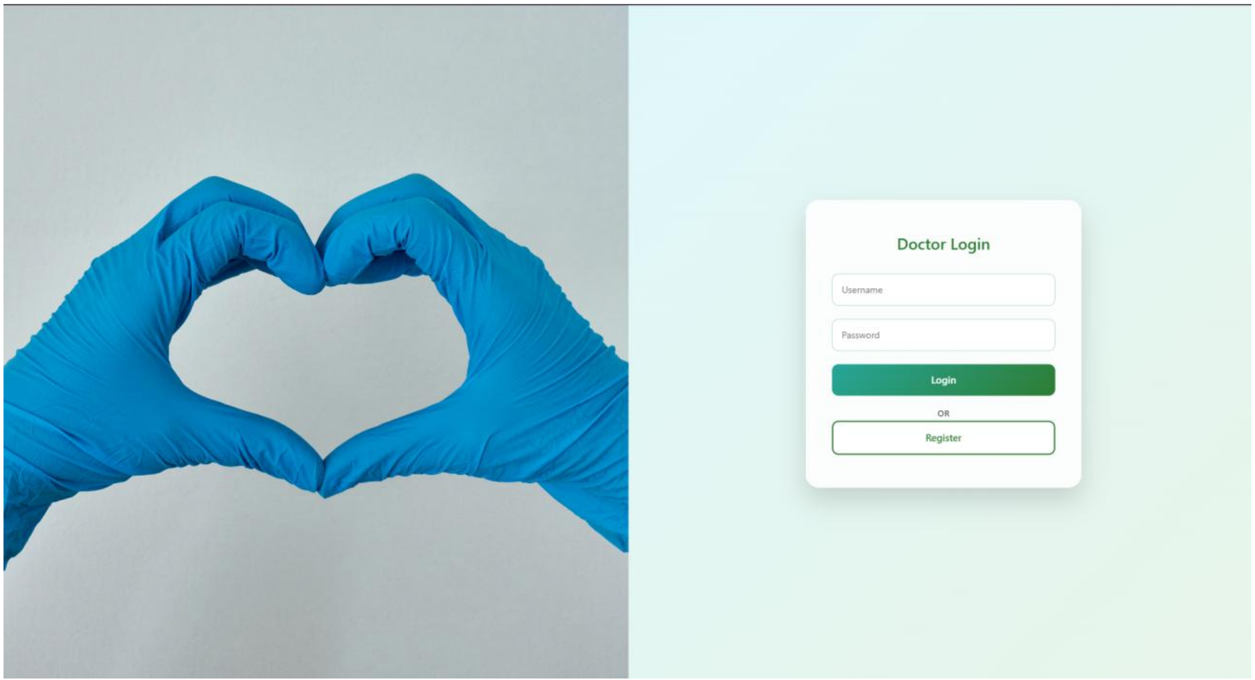
Pending Bills
Click to view your bills

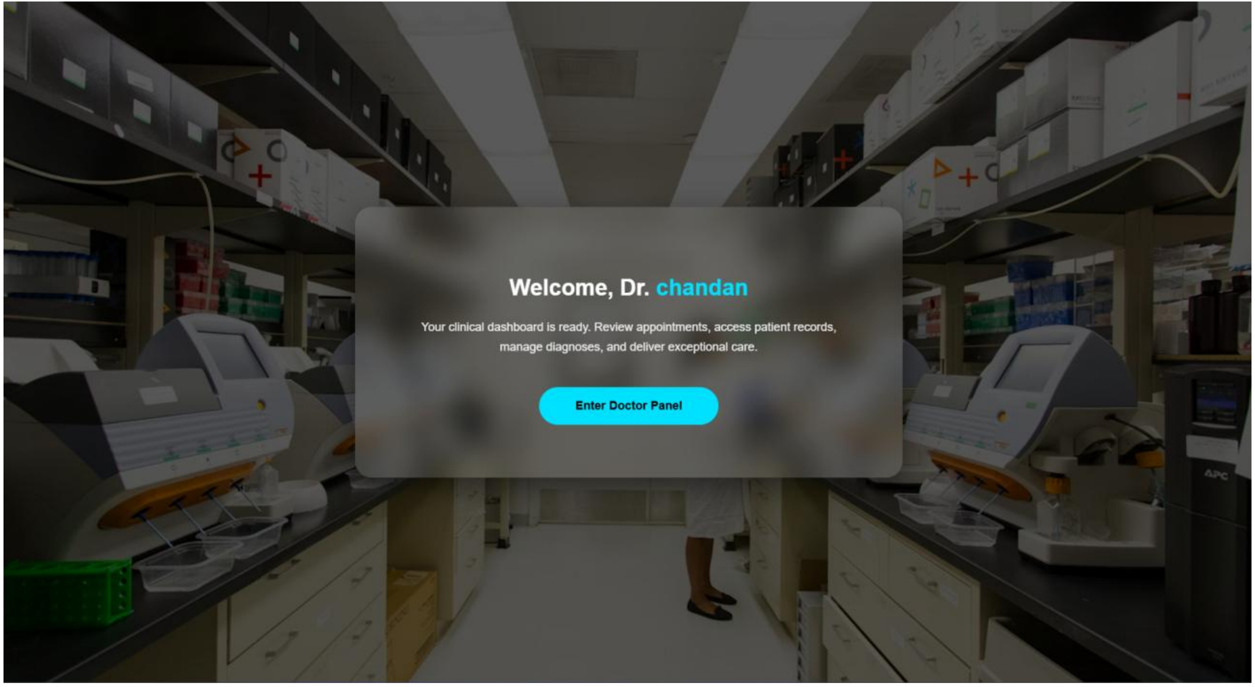
Payment History
View your payment records

Logout

27.0.0.1:8000/patients/view_patient_profile/

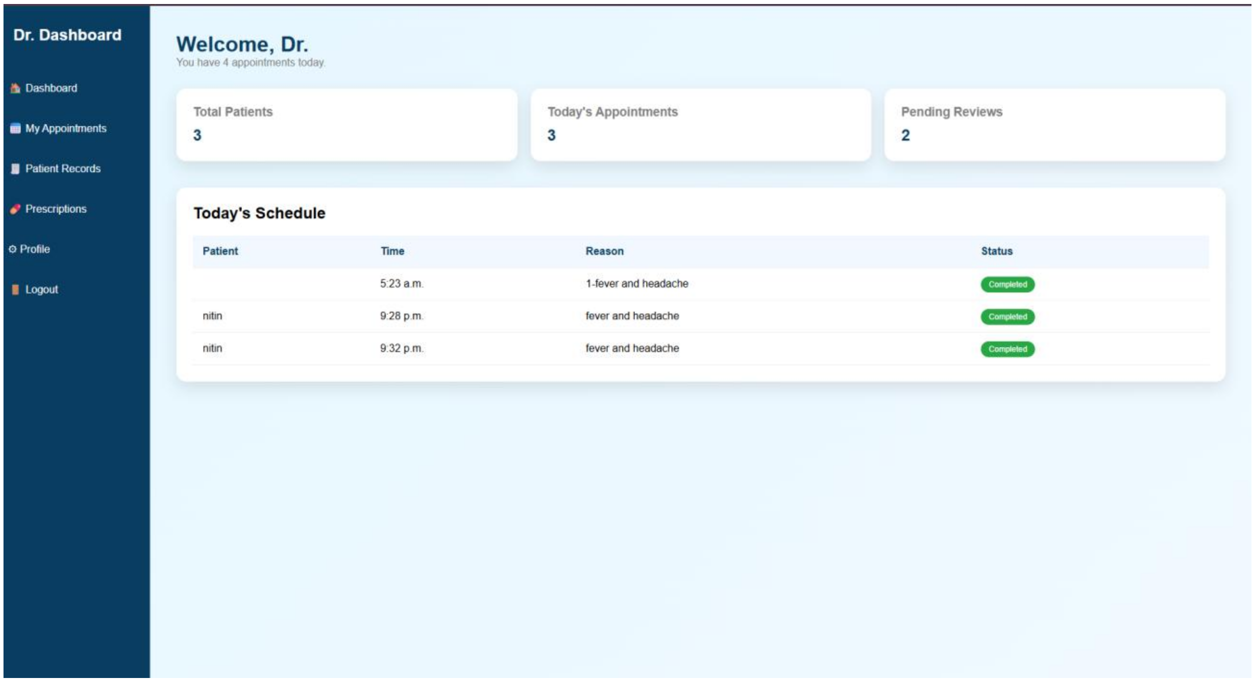
3-Doctor(login page & home page)





Welcome, Dr. chandan

Your clinical dashboard is ready. Review appointments, access patient records, manage diagnoses, and deliver exceptional care.



Welcome, Dr.

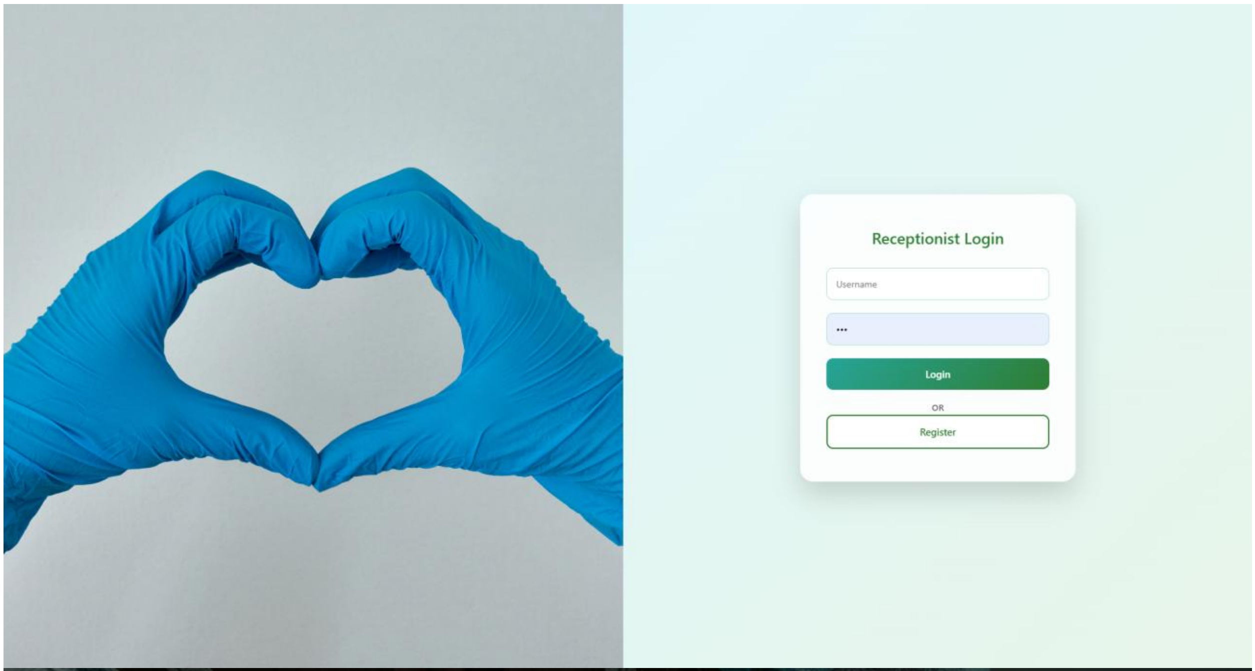
You have 4 appointments today.

Total Patients	Today's Appointments	Pending Reviews
3	3	2

Today's Schedule

Patient	Time	Reason	Status
	5:23 a.m.	1-fever and headache	Completed
nitin	9:28 p.m.	fever and headache	Completed
nitin	9:32 p.m.	fever and headache	Completed

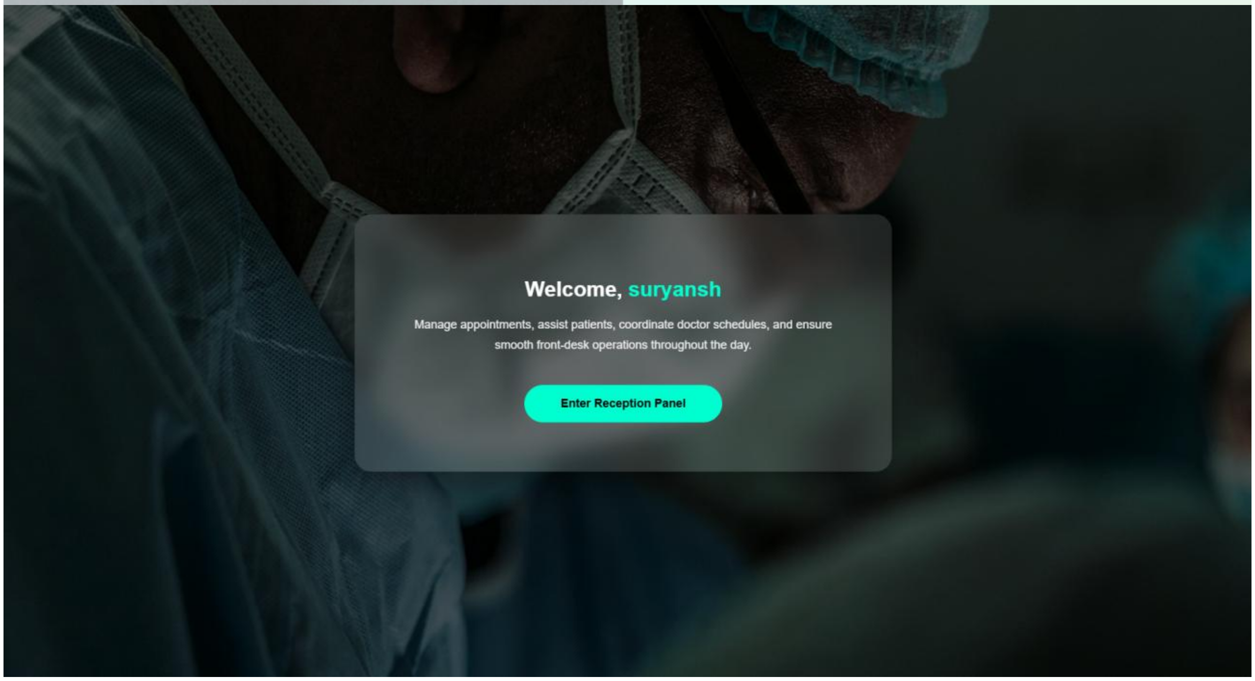
4-Recptionist(login page & home page)



Receptionist Login

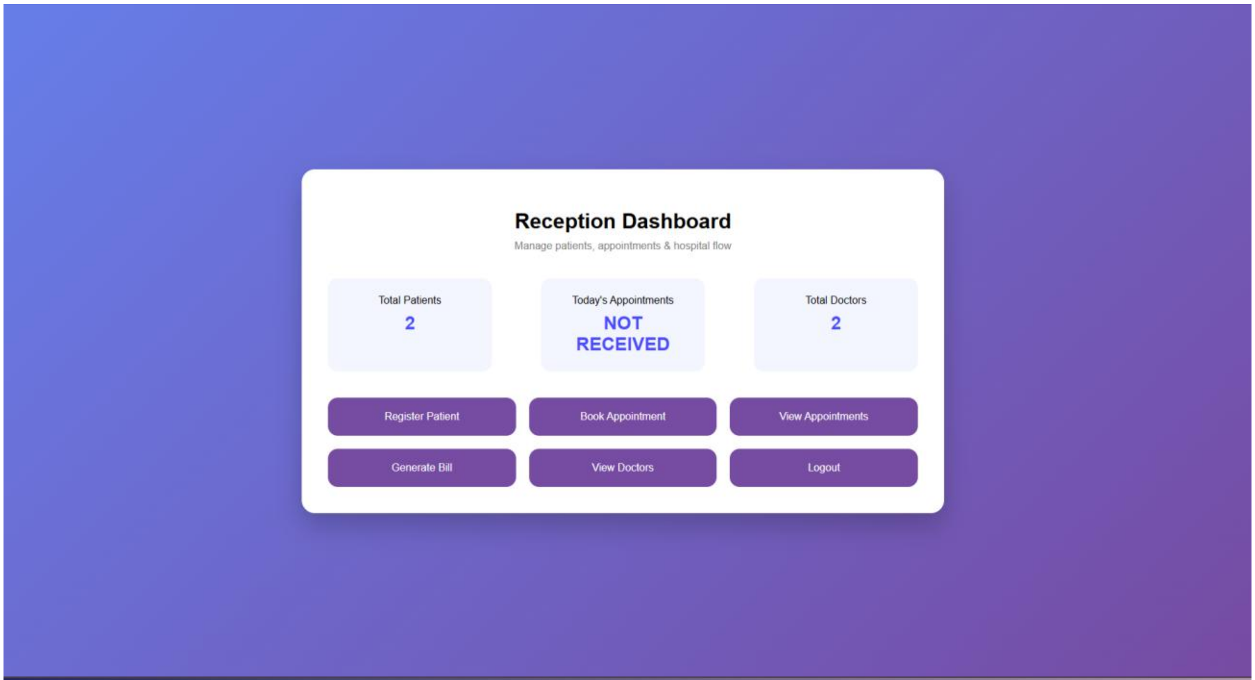
Username

•••



Welcome, suryansh

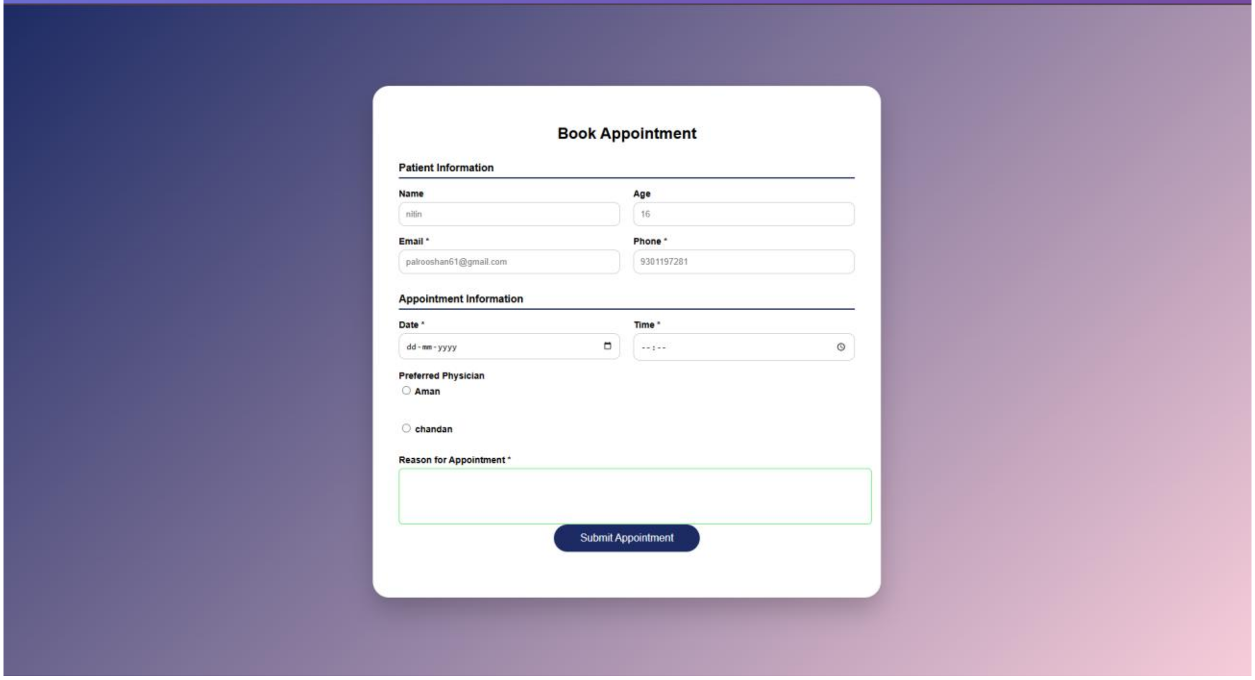
Manage appointments, assist patients, coordinate doctor schedules, and ensure smooth front-desk operations throughout the day.



Reception Dashboard

Manage patients, appointments & hospital flow

Total Patients	Today's Appointments	Total Doctors
2	NOT RECEIVED	2



Book Appointment

Patient Information

Name: nitin

Age: 16

Email *: palrooshan61@gmail.com

Phone *: 9301197281

Appointment Information

Date *: dd - mm - yyyy

Time *: - - : - -

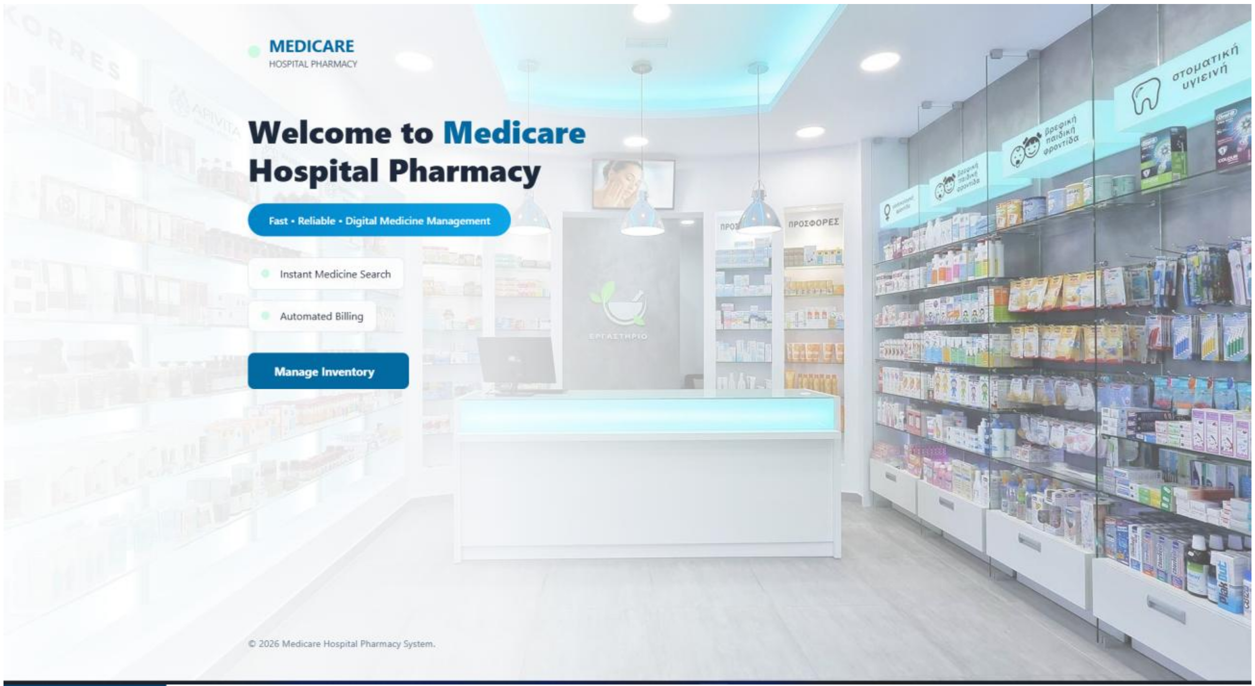
Preferred Physician

○ Aman

○ chandan

Reason for Appointment *

5-Pharmacist (login page & home page)



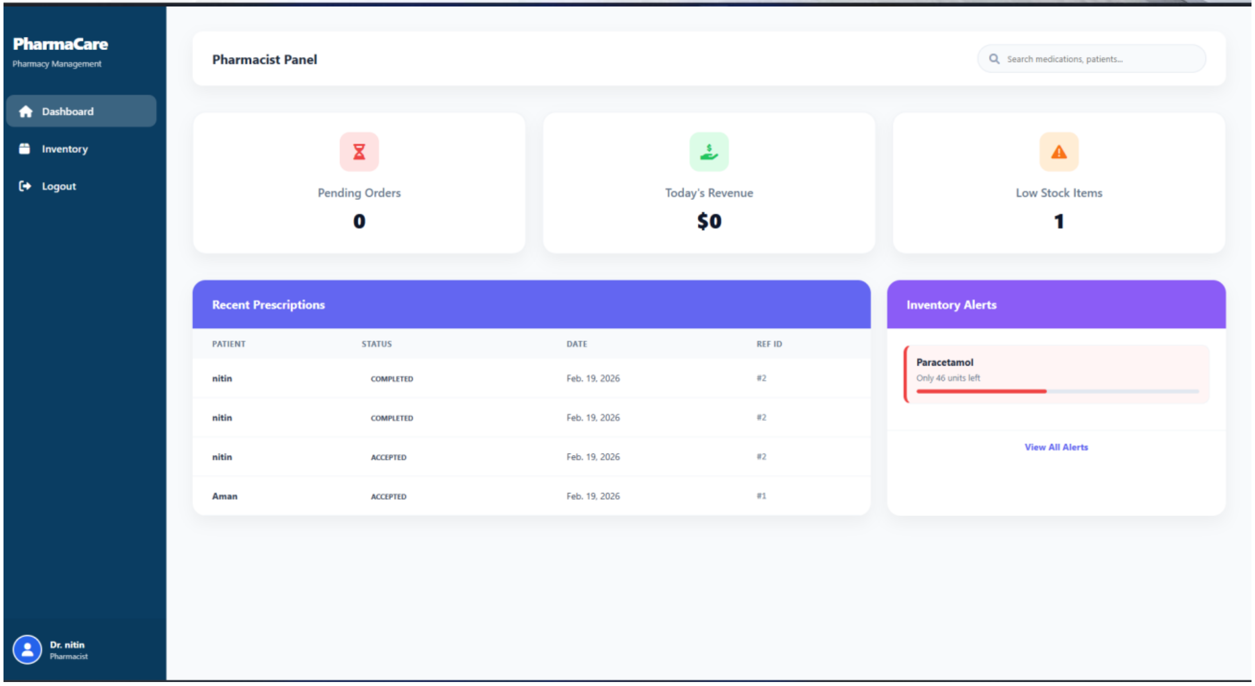
MEDICARE
HOSPITAL PHARMACY

Welcome to Medicare Hospital Pharmacy

Fast • Reliable • Digital Medicine Management

- Instant Medicine Search
- Automated Billing

© 2026 Medicare Hospital Pharmacy System.



Pharmacist Panel

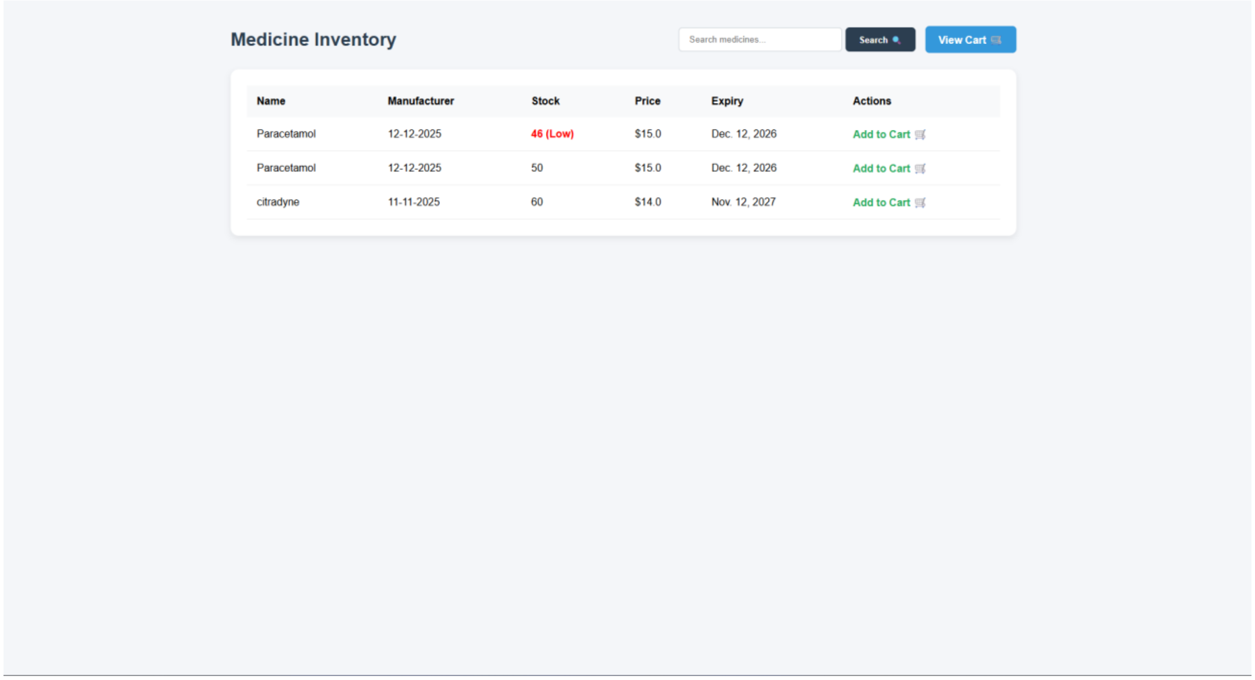
Pending Orders	Today's Revenue	Low Stock Items
0	$0	1

Recent Prescriptions

PATIENT	STATUS	DATE	REF ID
nitin	COMPLETED	Feb. 19, 2026	#2
nitin	COMPLETED	Feb. 19, 2026	#2
nitin	ACCEPTED	Feb. 19, 2026	#2
Aman	ACCEPTED	Feb. 19, 2026	#1

Inventory Alerts

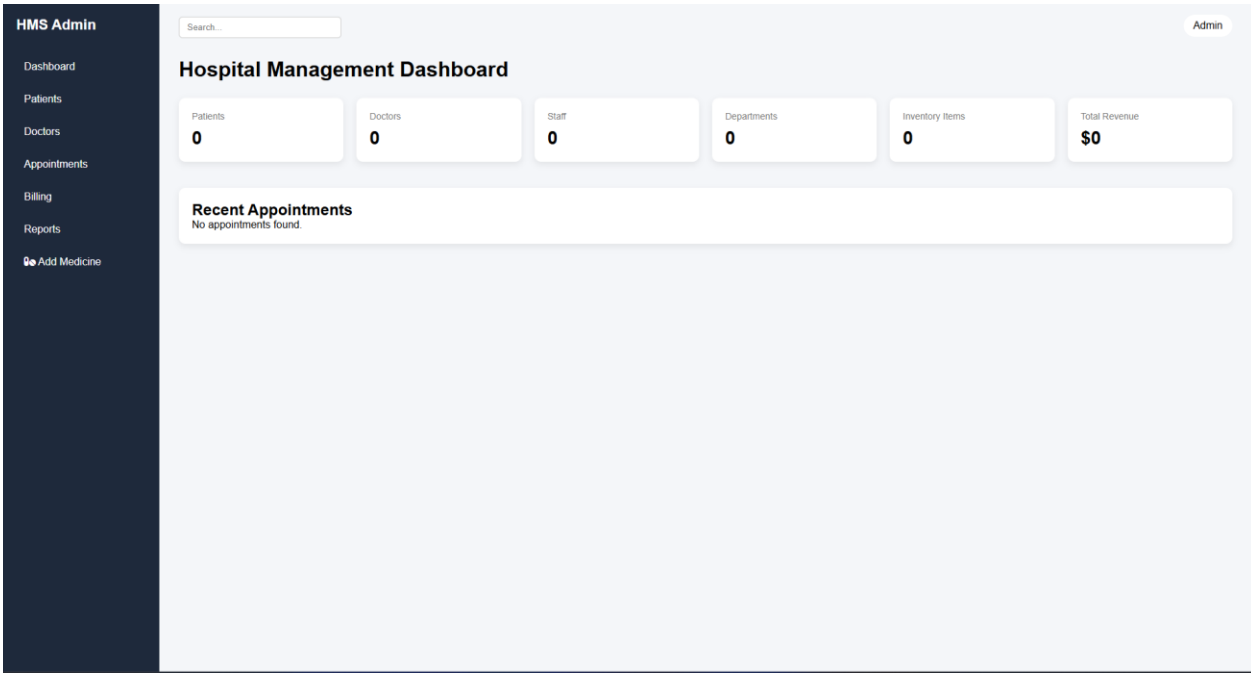
Paracetamol
Only 46 units left



Medicine Inventory

Name	Manufacturer	Stock	Price	Expiry	Actions
Paracetamol	12-12-2025	46 (Low)	$15.0	Dec. 12, 2026	Add to Cart 🛒
Paracetamol	12-12-2025	50	$15.0	Dec. 12, 2026	Add to Cart 🛒
citradyne	11-11-2025	60	$14.0	Nov. 12, 2027	Add to Cart 🛒

6-Admin page



Hospital Management Dashboard

Patients	Doctors	Staff	Departments	Inventory Items	Total Revenue
0	0	0	0	0	$0

Recent Appointments
No appointments found.

7. User Manual

6.1 Overview

The Hospital Management System (HMS) is a web-based application that digitizes hospital workflows such as patient registration, appointment booking, prescription management, pharmacy operations, and billing within a single, role-based platform. It supports four primary user roles: Patient, Physician, Pharmacist, and Receptionist, each with dedicated dashboards and access-controlled functionality.

6.2 Accessing the System

- **Open a supported browser (Chrome, Edge, Firefox).**
- **Navigate to the deployed HMS URL provided by the administrator.**
- **Use the “Login” or “Register” options on the homepage depending on your role and access status.**

6.3 Account Registration and Login

6.3.1 Registration (Patient)

- **Click on “Register” on the homepage.**
- **Select the “Patient” role.**
- **Fill in required details (name, email, password, contact information, etc.) and submit the form.**
- **Verify your email if verification is enabled.**
- **After approval/verification, you can log in using your registered credentials.**

6.3.2 Login (All Roles)

- **Click “Login”.**
- **Enter your registered email/username and password.**
- **On successful authentication, you will be redirected to the role-specific dashboard:**
    - **Patient Dashboard**
    - **Physician Dashboard**
    - **Pharmacist Dashboard**
    - **Receptionist Dashboard.**

If login fails, check your credentials, ensure caps lock is off, and, if necessary, use the “Forgot Password” flow (if configured).

6.4 Patient Module – How to Use

After login, patients see a dashboard with navigation links for profile, prescriptions, appointments, and orders.

1. **Update Profile**
    - **Go to “Profile”.**
    - **Review and update contact details if needed.**
    - **Save changes.**

2. **Upload Prescription**
    - **Navigate to “Prescriptions” → “Upload Prescription”.**
    - **Choose a valid file (e.g., PDF/image as per allowed types).**
    - **Add optional notes for the pharmacist or physician.**
    - **Click “Upload” and confirm that the prescription appears in your list.**
3. **View Prescription Status**
    - **In “Prescriptions”, check the status (e.g., Uploaded, Under Review, Approved, Processed).**
    - **Use filters/search to find specific prescriptions if available.**
4. **Place Medicine Order**
    - **Go to “Pharmacy” or “Shop Medicines”.**
    - **Search or browse medicines.**
    - **Add medicines to cart and verify quantities.**
    - **Proceed to “Cart”, review items, and confirm the order.**
    - **View the generated bill and order status in “My Orders”.**
5. **Manage Appointments**
    - **Navigate to “Appointments”.**
    - **Choose a physician and available time slot.**
    - **Submit the appointment request.**
    - **Track confirmation status in the same view.**

6.5 Physician Module – How to Use

Physicians get a dashboard focused on appointments and patient clinical records.

1. **View Today’s Appointments**
    - **On dashboard, see the list of scheduled appointments.**
    - **Use filters for date or patient name if available.**
2. **Access Patient Records**
    - **Click an appointment to open the patient record.**
    - **Review past prescriptions, visit history, and relevant details.**
3. **Update Clinical Notes / Prescriptions**
    - **Within the patient record, add consultation notes.**
    - **Upload or generate a digital prescription if required by workflow.**
    - **Save the record so that it is available to the pharmacy and patient.**
4. **Manage Appointment Status**
    - **Mark appointments as “Completed”, “Cancelled”, or “No Show” after the consultation.**
    - **This updates the patient-facing status and helps reception manage flow.**

6.6 Pharmacist Module – How to Use

Pharmacists manage inventory, process orders, and generate bills via their dedicated

dashboard.

1. Manage Medicine Inventory
    - Navigate to “Medicines”.
    - Add new medicine with name, description, price, and stock quantity.
    - Edit existing medicines to update price or stock.
    - Delete obsolete entries if allowed.
2. Process Patient Prescriptions
    - Open “Prescriptions” or “Orders from Prescriptions”.
    - View uploaded prescriptions and verify their validity.
    - Map prescribed items to available medicines in inventory.
    - Approve, partially fulfill, or reject prescriptions as per policy.
3. Handle Cart and Billing
    - Access “Orders” or “Cart Management”.
    - For each order, verify medicines and quantities.
    - Confirm stock availability; update quantities if needed.
    - Generate the bill; the system will calculate totals automatically based on medicine price and quantity.
    - Mark orders as “Billed” or “Completed” once dispensed.

6.7 Receptionist Module – How to Use

Receptionists coordinate appointments and overall patient flow.

1. Register Walk-in Patients
    - Use “Register Patient” option if self-registration is not done.
    - Enter basic details and assign login credentials as per policy.
2. Manage Appointments
    - Navigate to “Appointments”.
    - Create new appointments for patients by selecting physician, date, and time slot.
    - Reschedule or cancel appointments based on patient or physician request.
    - Monitor queue and appointment order to manage patient flow.
3. Coordinate with Physicians and Pharmacy
    - Use dashboard to check which patients are completed, in-progress, or waiting.
    - Communicate changes in schedule or urgent cases to physicians and pharmacists as needed.

6.8 Security and Best Practices (For Users)

- Never share your password; change it periodically.
- Always log out from the dashboard using the “Logout” option, especially on shared systems.
- Upload only valid prescription files; avoid executable or suspicious files.

- **Verify bill details before making payment (if payment gateway is integrated in future versions).**

6.9 Common Errors and Troubleshooting

- **Login Failed**
  - **Check email/username and password.**
  - **Use “Forgot Password” if available.**
- **File Upload Error**
  - **Ensure file format and size comply with allowed limits.**
  - **Retry with a stable internet connection.**
- **No Medicines in Cart**
  - **Add at least one medicine from the pharmacy catalog before proceeding to billing.**
- **Permission Denied / Unauthorized Access**
  - **You may be trying to access a page outside your role.**
  - **Log in with the correct role or contact the administrator.**

8. Conclusion

The architectural design of the Hospital Management System (HMS) represents a structured and scalable implementation of a modular, database-driven healthcare management platform built using the Django framework. The system successfully integrates authentication management, prescription upload handling, pharmacy inventory control, shopping cart workflow, and automated billing processes within a unified multi-tier architecture.

The relational database schema enforces Referential Integrity across core entities such as Patients, Pharmacists, Medicines, Prescriptions, and Orders, ensuring consistent and reliable transactional workflows. The use of Django ORM provides optimized database interaction while preventing common security vulnerabilities such as SQL injection. Session-based authentication combined with CSRF middleware ensures secure user access and controlled dashboard routing.

The system demonstrates a balanced integration of backend logic and frontend dashboard interfaces, ensuring both operational efficiency and user-friendly interaction. By leveraging Django's structured application separation, the architecture remains modular, maintainable, and extendable for future enhancements.

Overall, the project achieves its objective of digitizing hospital and pharmacy operations, reducing manual workload, improving inventory tracking accuracy, and enhancing patient service efficiency through secure and structured automation.