

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	Implement Medicine Model with full CRUD operations. Develop Pharmacy Dashboard UI.	Risk: Incorrect stock calculation or data inconsistency — Mitigation: Implement validation checks

	Management	Create Shopping Cart and Order Models. Implement cart logic and total price calculation. Integrate role-based dashboard routing.	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 uploading prescriptions. Link prescription orders to pharmacy dashboard. Integrate billing generation workflow. Add email notification functionality for registration and order confirmation.	Risk: File upload vulnerabilities — Mitigation: Enforce file validation, controlled 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.

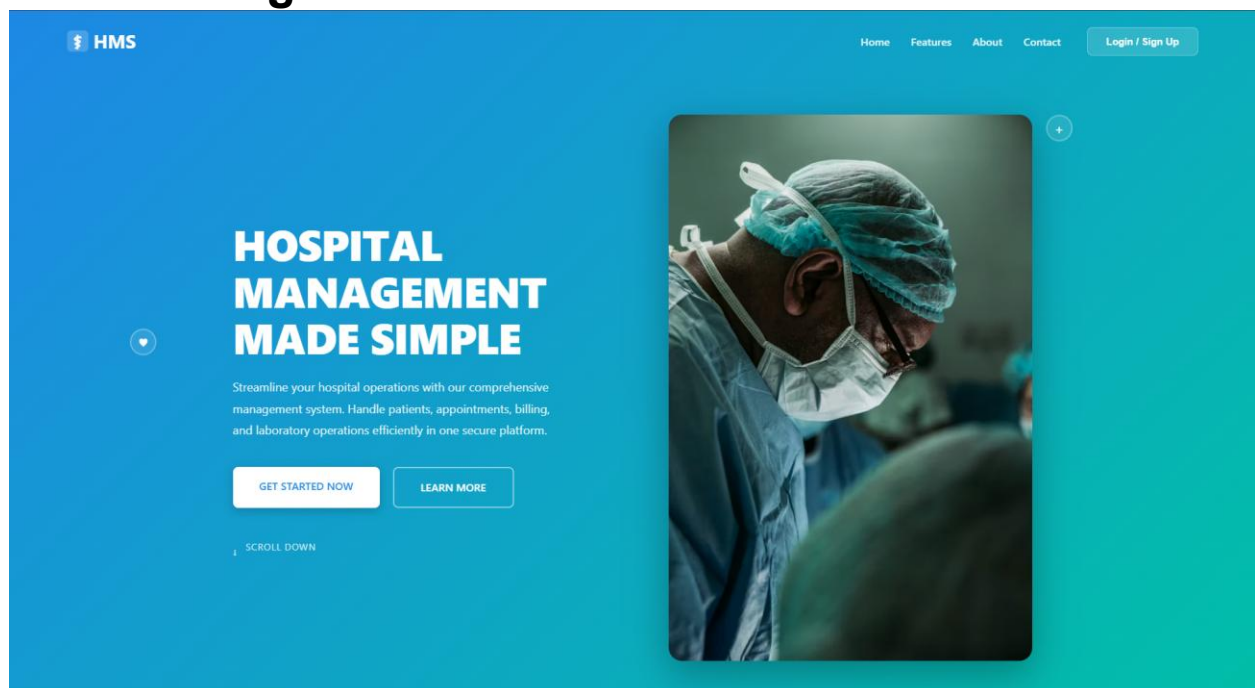
5. Development Team Introduction

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.
Frontend & UI Developer	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



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.

Product

Features
Pricing
Security
Enterprise

Company

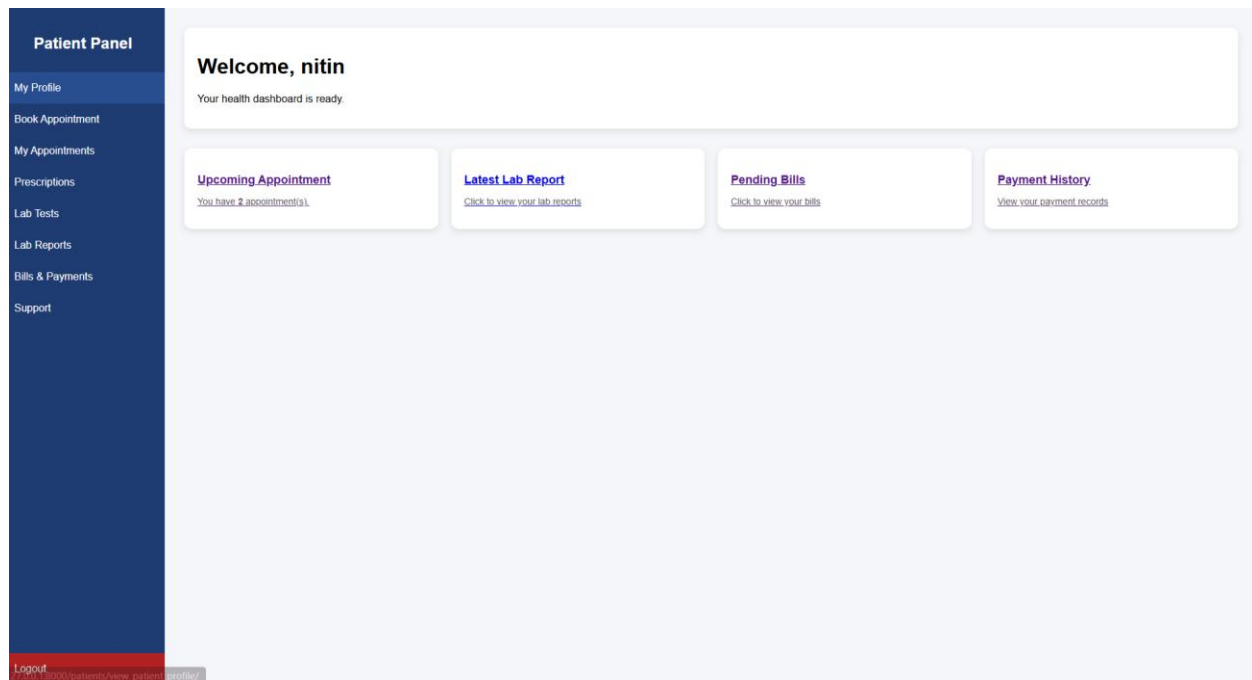
About Us
Careers
Blog
Contact

Support

Help Center
Documentation
API
Status

© 2024 HMS - Hospital Management System. All rights reserved. | Privacy Policy | Terms of Service

2-Patient(login page & home page)



3-Doctor(login page & home page)



Doctor Login

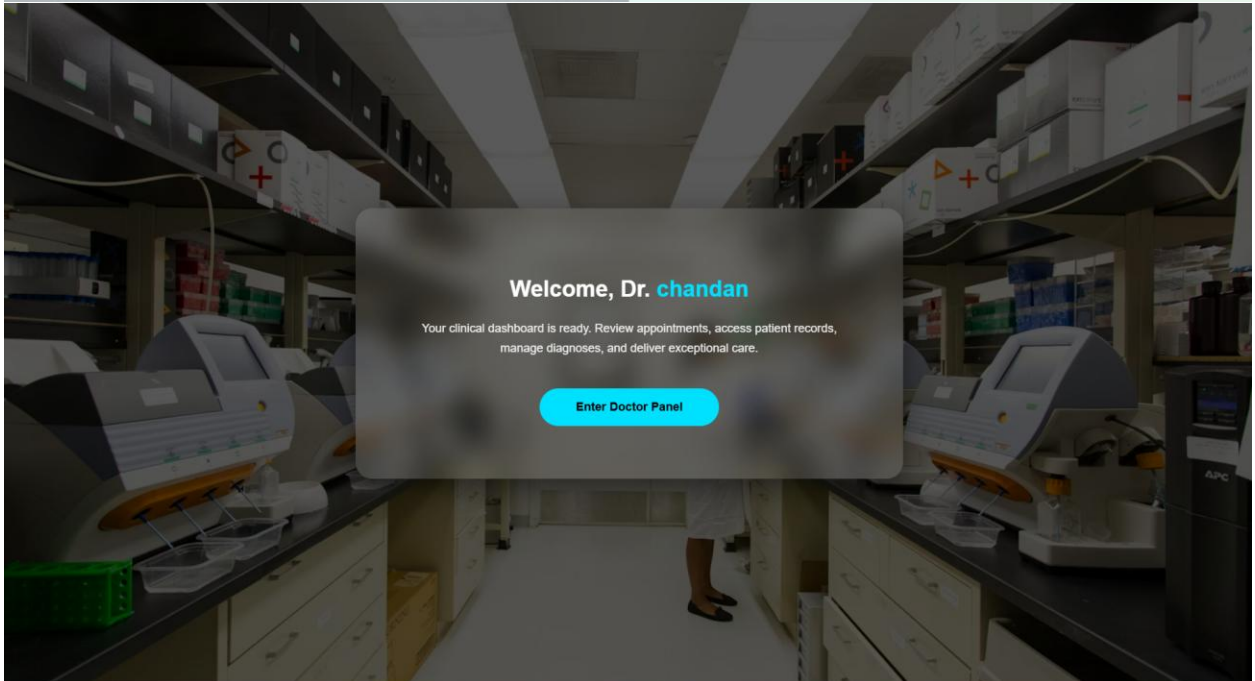
Username

Password

[Login](#)

OR

[Register](#)



Dr. Dashboard

Dashboard

My Appointments

Patient Records

Prescriptions

Profile

Logout

Welcome, Dr.

You have 4 appointments today

Total Patients

3

Today's Appointments

3

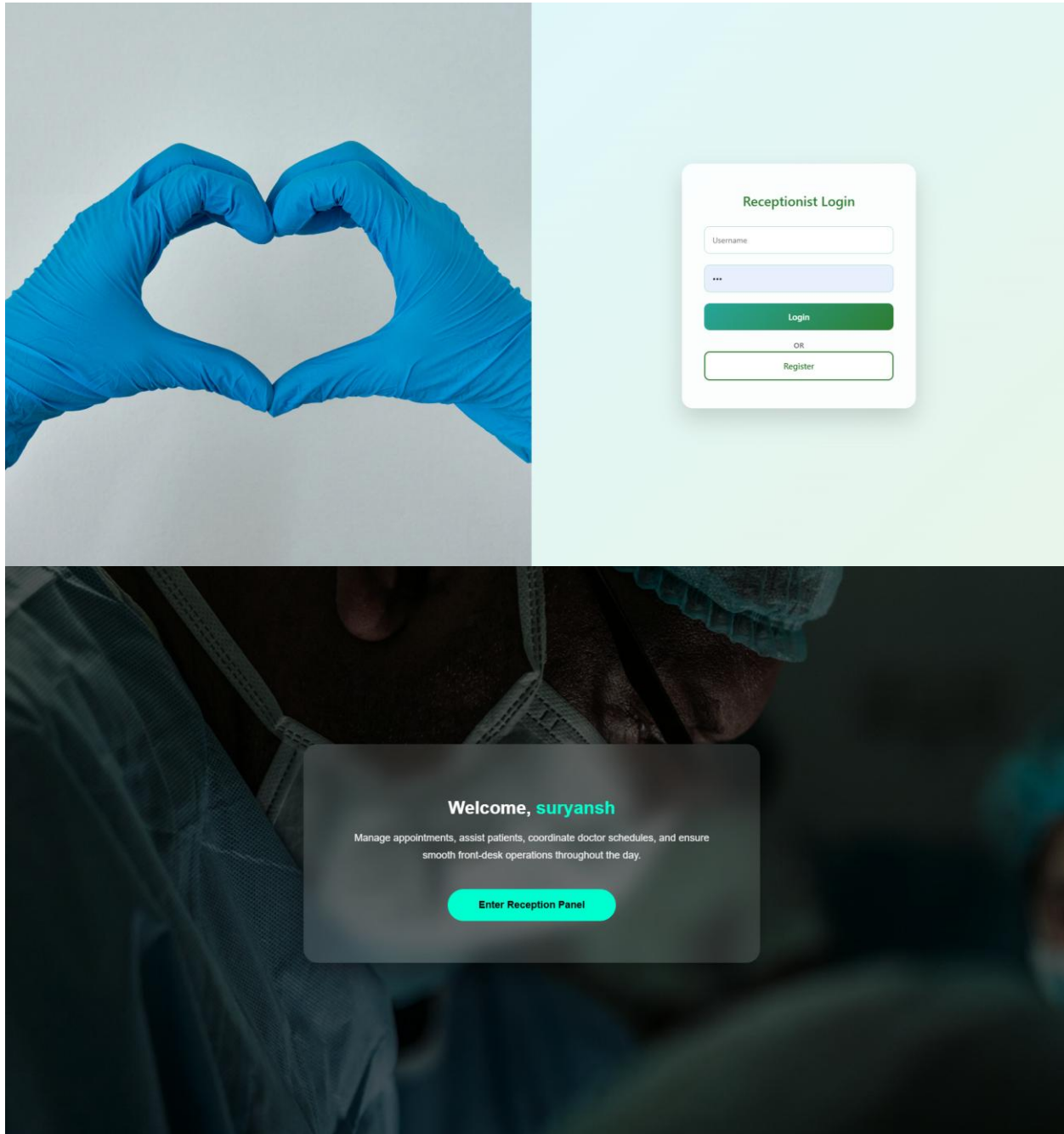
Pending Reviews

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)



Reception Dashboard

Manage patients, appointments & hospital flow

Total Patients

2

Today's Appointments

NOT
RECEIVED

Total Doctors

2

Register Patient

Book Appointment

View Appointments

Generate Bill

View Doctors

Logout

Book Appointment

Patient Information

Name

nitin

Age

16

Email *

patrooshan61@gmail.com

Phone *

9301197281

Appointment Information

Date *

dd-mm-yyyy

Time *

--:--

Preferred Physician

☐ Aman

☐ chandan

Reason for Appointment *

Submit Appointment

5-Pharmacist (login page & home page)

The image displays the Medicare Hospital Pharmacy login page and the Pharmacist Panel dashboard. The login page features a background image of a pharmacy interior with shelves stocked with medications. The Medicare logo is at the top left, and the text "Welcome to Medicare Hospital Pharmacy" is prominently displayed. Below this, there are three buttons: "Instant Medicine Search", "Automated Billing", and "Manage Inventory". A copyright notice "© 2026 Medicare Hospital Pharmacy System." is at the bottom.

The Pharmacist Panel dashboard is shown below the login page. It includes a search bar at the top right labeled "Search medications, patients...". The dashboard is divided into three main sections: "Pending Orders" (0), "Today's Revenue" (\$0), and "Low Stock Items" (1). Below these are two tables: "Recent Prescriptions" and "Inventory Alerts".

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

[View All Alerts](#)

The dashboard also includes a sidebar with navigation links: "Dashboard", "Inventory", and "Logout". At the bottom left, there is a user profile for "Dr. nitin Pharmacist".

Medicine Inventory

Search medicines...

Search

View Cart

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
citradyme	11-11-2025	60	\$14.0	Nov. 12, 2027	Add to Cart

6-Admin page

HMS Admin

Dashboard

Patients

Doctors

Appointments

Billing

Reports

Add Medicine

Search...

Admin

Hospital Management Dashboard

Patients

0

Doctors

0

Staff

0

Departments

0

Inventory Items

0

Total Revenue

\$0

Recent Appointments

No appointments found.

7. 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.