VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi-590 018

A Mini - Project Report

On

“AI-DRIVEN HOSPITAL QUEUE AND BED AVAILABILITY TRACKING SYSTEM”

Submitted in partial fulfillment of the requirements for the **MINI PROJECT (BCD586)** course of the 5th semester

Bachelor of Engineering

In

Computer Science & Engineering (DATA SCIENCE)

Submitted by

Mr. Ganesh H

(4AI23CD018)

Ms. Manasa YK

(4AI23CD027)

Ms. Megha GK

(4AI23CD028)

Mr. Nilesh jha

(4AI23CD032)

Under the guidance of

Ms. GAGANA DEEPA J.

Assistant Professor

Department of CS&E (DATA SCIENCE)

Adichunchanagiri Institute of Technology

CHIKKAMAGALURU - 577102

2025-26

ADHICHUNCHANAGIRI INSTITUTE OF TECHNOLOGY

Jyothinagar, Chikkamagaluru-577102

DEPARTMENT OF CS&E (DATA SCIENCE)

CERTIFICATE

This is to certify that the Mini project work entitled **“AI-DRIVEN HOSPITAL QUEUE AND BED AVALIABILITY TRACKING SYSTEM”** is a bonafied work carried out by **Mr. Ganesh H (4AI23CD018), Ms. Manasa Y K (4AI23CD027), Ms. Megha G K (4AI23CD028), Mr. Nilesh Kumar Jha (4AI23CD032)** in partial fulfillment for the **Mini Project (BCS586)** course of 5th semester Bachelor of Engineering in **Computer Science and Engineering (Data Science)** of the Visvesvaraya Technological University, Belagavi during the academic year **2025-2026**. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The Mini project report has been approved as it satisfies the academic requirements in respect of Project Work prescribed for the said Degree.

Signature of the Guide
Ms. Gagana Deepa J B.E., M.Tech
Assistant Professor

Signature of Coordinator
Mrs. Shilpa K V. B.E., M.Tech
Assistant Professor

Signature of the HOD
Dr. Adarsh M J B.E., M.Tech., Ph.D
Associate Professor and Head

ABSTRACT

Hospital resource management is a critical aspect of healthcare operations, directly influencing patient satisfaction, treatment efficiency, and overall service quality. Traditional methods such as manual queue handling, verbal bed allocation updates, and paper-based appointment scheduling are time-consuming, error-prone, and lack real-time visibility. The AI-Driven Hospital Queue and bed availability tracking system is designed to address these challenges by automating hospital workflow processes, improving accuracy, and enhancing data management across departments.

This project implements a robust, user-friendly hospital management platform using web-based interfaces that enable seamless interaction for patients, staff, and administrators. Its key components include user authentication, real-time queue tracking, bed availability management, doctor scheduling, appointment booking, and AI-powered chatbot assistance. The system is built using Django for the backend, HTML/CSS for the frontend, and SQLite for reliable database management, with Google Gemini API integrated for intelligent chatbot responses by offering an efficient, automated, and real-time solution, this project contributes significantly to improving hospital operations, reducing manual workload, minimizing errors, and enhancing patient experience through intelligent digital healthcare management.

ACKNOWLEDGEMENTS

We express our humble pranamas to his holiness Divine Soul Parama Poojya Jagadguru Padmabushana Sri Sri Sri Dr.Balagangadharanatha Maha Swamiji and Parama Poojya Jagadguru Sri Sri Sri Dr. Nirmalanandanatha Maha Swamiji Pontiff, Sri Adichunchanagiri Maha Samsthana Matt and Sri Sri Gunanatha Swamiji, Chikkamagaluru branch, Sringeri who have showered their blessings on us.

The completion of any project involves the efforts of many people. We have been lucky enough to have received a lot of help and support from all quarters during the making of this project, so with gratitude, we take this opportunity to acknowledge all those whose guidance and encouragement helped us emerge successful.

We express our gratitude to Dr. C K Subbaraya, Director, Adichunchanagiri Institute of Technology.

We express our sincere thanks to our beloved principal, Dr. C T Jayadeva for having supported us in our academic endeavors.

We are also indebted to Dr. Adarsh M J, HOD of CS&E (DATA SCIENCE) Department, for the facilities and support extended towards us.

We thank our project coordinator Mrs. Shilpa K V Asst. Professor, Department of CS&E (DATA SCIENCE), for her lively correspondence and assistance in carrying on with this project

We are thankful to the resourceful guidance, timely assistance and graceful gesture of our guide Ms. Gagana Deepa J, Asst. Professor, Department of CS&E (DATA SCIENCE), who has helped us in every aspect of our project work.

We would be very pleased to express our heart full thanks to all the teaching and non- teaching staff of CS&E (DATA SCIENCE) Department and our friends who have rendered their help, motivation and support.

Ganesh H
Manasa Y K
Megha G K
Nilesh Jha

CONTENTS

Chapter 1

INTRODUCTION

1.1 Background

- **Context**: Hospitals today manage hundreds of patients daily, requiring efficient handling of queues, bed allocation, and appointments. Traditionally, these tasks are performed manually using registers, verbal communication, or basic systems that lack real-time updates
- **Problem**: Manual queue and bed management systems are slow, error-prone, and lack real-time visibility. Patients often wait for long hours with no clarity on their position in the queue or bed availability. Staff members struggle to coordinate between departments, frequently relying on phone calls or manual records that are outdated within minutes.
- **Opportunity**: The rise of digital healthcare technologies creates a major opportunity to modernize hospital operations. By implementing an AI-driven system, hospitals can automate queue tracking, monitor bed availability in real time, offer online appointment scheduling, and assist patients through an intelligent chatbot.

1.2 Problem Statement

- **Overview of the Problem**: Hospitals struggle with long queues, overcrowded waiting areas, and poor patient flow due to absence of a unified and real-time system. The lack of instant updates on bed and appointment availability leads to delays, miscommunication, and operational inefficiency. An AI-driven platform is needed to streamline patient movement, improve transparency, and optimize hospital resource management.

- **Specific Issues**:
    - Lack of Real-Time Updates.
    - Manual Errors in Record-Keeping.
    - Long Waiting Times for Patient.
    - Difficulty Handling High Patient Volume.
    - Poor Communication Across Departments.

1.3 Objective of the System

- The main objective of the **AI-Driven Hospital Queue & Bed Availability Tracking System** is to automate and streamline patient flow in hospitals using a digital, intelligent, and real-time system.

Core Objectives

- Provide **real-time queue tracking** with automated token generation
- Display **live bed availability** across all wards
- Offer **doctor scheduling & appointment booking**
- Provide **AI-powered chatbot support** using Google Gemini

- Enable **role-based authentication** (admin/patient)
- Streamline patient flow and reduce manual processes

Key Goals

- Improve hospital transparency
- Reduce waiting times
- Digitize core patient-facing operations
- Provide an intelligent support system
- Optimize resource allocation.

1.4 Significance of the System

• **Efficiency:** Automates queue, bed, doctor, appointment management, reducing manual work and saving time.

• **Accuracy:** Minimizes human errors by maintaining real-time and reliable hospital data.
• **Transparency:** Patients can easily view live queue status and bed availability.

• **Better Decision Making:** Doctors and administrators can make faster and more informed decisions using updated data.

• **AI-Driven Support:** The integrated Gemini chatbot answers common patient queries, reducing staff workload.

• **Real-Time Monitoring:** Hospital staff can monitor patient flow, bed usage, appointments instantly, improving coordination and hospital operations.

1.5 Scope of the Project

- **In Scope**:
  - Token generation and patient flow management
  - Live bed availability dashboard
  - Real-time queue tracking
  - Appointment booking module
  - AI Chatbot for hospital-related questions
  - Secure user authentication
  - Admin panel for hospital staff

- **Out of Scope**:
  - Full hospital billing system
  - Detailed electronic medical records (EHR)

1.6 Methodology

- **Approach**: The system will be developed using a modern, modular, and scalable approach to ensure smooth performance, maintainability, and real-time responsiveness. It uses a combination of web technologies, backend services, database management, and AI integration to deliver an efficient hospital workflow solution.

1.7 Technologies Used

- **Frontend:** HTML, CSS , JavaScript
- **Backend:** Django (Python)
- **Database:** SQLite
- **AI Integration:** Gemini API , Tensor flow
- **Auth:** Django Authentication
- **Version Control:** GitHub
- **Testing Tools:** Django Test Framework

This project follows an **Agile development methodology**, focusing on iterative progress, continuous testing, and rapid feedback cycles.

1.7.1 Iterative Development: The system is built in small modules (Queue, Beds, Appointments, Chatbot), developed and improved in iterations.

1.7.2 User Feedback Integration: Each iteration is tested and reviewed, allowing changes based on real hospital workflow needs.

1.7.3 Continuous Integration & Updates: Features are added gradually with frequent updates and refinements.

1.7.4 Improved **Flexibility:** Agile allows easy modification of requirements as the project evolves.

- **Better Quality Assurance:** Errors and issues are identified early during each sprint.

1.8 Target Audience

- **Patients:** To track queue , book digital token, appointments, and ask queries.
- **Hospital admins:** To manage live queue, beds, and appointments.
- **Doctors:** To view schedules and appointments.

1.9 Overview of the Report

This report is organized as follows:

- **Chapter 2:** System Design
- **Chapter 3:** Implementation
- **Chapter 4:** Testing
- **Chapter 5:** Results and Discussion
- **Chapter 6:** Conclusion and Future Enhancement

Chapter 2

SYSTEM DESIGN

This chapter describes the technical design of the system which follows a client–server architecture where users interact through Django-rendered web pages. The backend handles queue logic, bed updates, appointment scheduling, doctor management, and AI chatbot operations. The database stores all patient flow and resource-related records.

2.1 System Architecture

High-Level Overview: The system supports real-time communication where patient queues, bed availability, doctor schedules, and appointments are updated instantly across all connected users. With the integration of an AI-powered chatbot using the Gemini API, patients receive quick assistance for their queries. The layered architecture of the system ensures modularity, scalability, and smooth functioning of all hospital operations through a structured flow of data between the UI, backend, and database.

Architecture Diagram: Include a diagram illustrating all key components of the system: the frontend user interface, the Django backend server, and the database layer.

- **Components:**
  o **Frontend:** A web-based interface built using Django templates where patients, staff, and administrators interact with the system.
  o **Backend Server:** The Django backend that processes user requests , manages queue logic, bed availability updates, appointments, and handles AI chatbot communication.
  o **Database:** Stores all structured data , including user accounts, patient queues, bed records, appointment details, doctor information, and chatbot interaction logs.

2.2 Module Design

- The system is divided into functional modules, each handling a specific task.

2.2.1 Queue Management Module

- Generates tokens
- Tracks real-time queue position
- Displays live patient flow

2.2.2 Bed Availability Module

- Shows real-time available/occupied beds
- Bed hold and view held beds.

2.2.3 Appointment Scheduling Module

- Book appointments/View doctor availability
- Cancel/reschedule appointments

2.2.4 User Authentication Module

- Login / Sign-up for patient and admin
- Secure access to modules

2.2.5 AI Chatbot Module

Powered by Gemini API, Answers patient queries.

2.3 Database Design

Key Tables

2.3.1 User – id, username, password
2.3.2 Hospital – id, name
2.3.3 Doctor – id, name, specialization
2.3.4 Queue – id, token_no, patient_name, status
2.3.5 Bed – status
2.3.5 5Appointment – id, doctor, patient, date, time
2.3.6 Chatlogs – question, response

2.4 User Interface (UI) Design

Main Screens:

- Login Page
- Patient Dashboard
- Queue Status Page
- Bed Availability Display
- Appointment Booking Screen
- Admin Panel
- AI Chatbot Window

2.5 Technology Stack

- **Frontend:** Built using HTML, CSS , and Django Templates to create a clean, responsive, and user-friendly interface.
- **Backend:** Implemented using Python Django, which handles business logic, routing, authentication, AI chatbot integration, and all hospital operations. It processes user requests, updates the database, and communicates with the Gemini API for chatbot responses.
- **Database:** Uses SQLite to store all structured data such as users, queue tokens, bed status, doctor details, and appointments.

Chapter 3

Implementation

This chapter explains how the AI-Driven Hospital Queue & Bed Availability Tracking System was developed. It covers the backend, frontend, database, APIs, integration process, and the overall workflow of the system

3.1 Backend Implementation

Backend logic includes:

3.1 Queue token generation
3.2 Bed status changes
3.3 Appointment CRUD operations
3.3 AI chatbot integration using Gemini API
3.5 Doctor and user management

API Endpoints

3.1.1 Authentication

- POST /login
  Authenticates users (admin/patient) and starts a secure session.
- POST /logout
  Logs out the current user and ends the session.

3.1.2 Queue Management

- POST /queue/generate
  Generates a new queue token for a patient.
- GET /queue/status
  Returns the live queue list and current token status.
- POST /queue/update/<id>
  Updates the status of a token (waiting, completed, skipped).

3.1.3 Bed Management

- GET /beds/
  Displays real-time bed availability across different wards.
- POST /beds/update/<id>
  Admin updates bed status as available, occupied, or cleaning.

3.1.4 Appointment System

- POST /appointment/book
  Books an appointment with doctor, date, and time.
- GET /appointment/list
  Returns the list of all upcoming appointments for the user.
- POST /appointment/cancel/<id>
  Cancels an existing appointment.

3.2 Frontend Implementation

The frontend of this system is developed using Django Templates, HTML, CSS and JavaScript, ensuring a clean, responsive, and user-friendly interface. The UI allows patients, doctors, and hospital staff to easily navigate through all modules such as queues, beds, appointments, and chatbot support without any complexity.

Main UI Components

3.2.1 Login Page

- Provides secure login access for patients and hospital staff.
- Authenticates user credentials before redirecting to the appropriate dashboard.

3.2.2 Dashboard

- Displays quick access links to Queue, Beds, Doctors, Appointments, and Chatbot modules.
- Admin dashboard includes additional management controls such as updating beds, doctors, and viewing all appointments.

3.2.3 Queue Status Screen

- Shows the generated token number, current queue position, and live status updates.
- Uses periodic refresh (auto-reload) to display real-time queue progress.

3.2.4 Bed Availability Screen

- Displays a table of available, occupied, and holding beds.

3.2.5 Appointment Booking Page

- Allows patients to select a doctor, choose a date and time, and confirm appointments.
- Users can view their upcoming bookings and cancel appointments if required.

3.2.6 AI Chatbot Window

- Provides an interactive chat interface connected to the Gemini AI API.
- Gives instant responses to hospital-related questions, reducing staff workload.

3.3 Database Implementation

This system uses SQLite as its relational database to ensure structured data storage, reliability, and efficient management of hospital records. Django’s ORM (Object Relational Mapping) is used to interact with the database, handle queries, and maintain data consistency across all modules such as queues, beds, doctors, appointments, and chatbot logs.

3.3 Key Tables

3.3.1 users

- user_id
- username
- email
- role (patient/admin/staff)
- password (hashed using Django authentication)

3.3.2 queue

- id
- token_number
- patient_name
- status (waiting, completed)
- timestamp

3.3.3 bed

- id
- status (available, occupied)

3.3.4 appointments

- id
- patient_id
- doctor_id
- date
- time

3.3.5 chat_logs

- user_id
- question
- response
- timestamp

Chapter 4

Testing

This chapter describes the testing strategies, test cases, and validation processes used to ensure that the **AI-Driven hospital queue and bed availability tracking system** functions accurately, reliably, and securely. Testing was conducted for all major modules—including queue management, bed tracking, doctor management, appointment handling, authentication, and AI chatbot integration—to verify that the system performs as expected under different conditions and user scenarios

4.1 Testing Objectives

4.1.1 Verify that all system modules operate according to functional requirements.
4.1.2 Validate user interactions for patients, administrators, and hospital staff.
4.1.3 Ensure real-time accuracy of queue positions, bed status, and appointment updates.
4.1.4 Confirm security of user authentication and protection of sensitive data.
4.1.5 Assess the system's responsiveness, stability, and behavior under various loads.
4.1.6 Detect and resolve bugs, inconsistencies, and performance issues prior to deployment.

4.2 Testing Environment

•**Hardware**: Laptop/PC with minimum 8GB RAM and multi-core processor.

• **Software:** Python 3.x, Django Framework, JavaScript.

• **Testing Tools:** Django Test Framework (backend unit testing)

• **Database:** SQLite

• **Operating System:** Windows 11

• **Browsers:** Google Chrome and Microsoft Edge for UI and cross-browser.

4.3 Types of Testing

4.3.1 Unit Testing

Unit tests were performed on individual backend functions and Django views to ensure correct behaviour in isolation.

- Queue token generation logic
- Bed status update function
- Appointment booking validation
- Login and authentication logic
- Chatbot request-processing and response handling
- Form validation and field-level error handling

4.3.2 Integration Testing

Integration testing ensured that the different modules of the system worked together smoothly.

4.3.2.1 Django frontend templates correctly receiving data from backend views
4.3.2.2 Proper interaction between Django views and database models using ORM
4.3.2.3 Queue and bed updates reflecting instantly in the corresponding UI pages
4.3.2.3 Appointment bookings successfully stored and retrieved from the database
4.3.2.4 Chatbot module correctly sending queries to the Gemini API and displaying responses
4.3.2.5 Role-based access (admin/patient) working across integrated modules

4.3.3 Functional Testing

This verifies that the system meets all functional requirements from the user's perspective.

Key Functional Scenarios Tested

- Patient logs in successfully
- Token is generated and displayed
- Queue updates reflect instantly
- Available/occupied beds are shown accurately
- Appointments are booked and stored correctly
- AI chatbot responds properly

4.4 Test Cases

Below are sample test cases for various components:

Table 4.1: Test Cases

Test Case ID	Description	Test Steps	Expected Result	Status
TC-001	Login with valid credentials	Enter valid username and password → Click Login	User is redirected to the dashboard	Pass
TC-002	Login with invalid credentials	Enter invalid username/password → Click Login	Error message displayed, login denied	Pass
TC-003	Generate patient queue token	Navigate to Queue → Click Generate Token	Token is created and queue list updated	Pass
TC-004	View real-time queue	Navigate to Queue page	Queue list shows updated positions accurately	Pass
TC-005	View bed availability	Open Bed Status page	Correct and real-time bed availability displayed	Pass
TC-006	Update bed status (Admin)	Admin marks a bed as Occupied/Available → Save changes	Frontend updates instantly with new status	Pass

Test Case ID	Description	Test Steps	Expected Result	Status
TC-007	Book appointment	Select doctor → Choose available time slot → Confirm appointment	Appointment saved and displayed in Appointments List	Pass
TC-008	Chatbot response	Ask a query in AI chatbot	Accurate response generated by Gemini-based AI	Pass
TC-009	Unauthorized access attempt	Patient tries to access Admin page	Access denied with proper error message	Pass
TC-010	API response validation	Send incorrect or malformed API request	System returns proper error code with message	Pass

Chapter 5

Results and Discussion

This chapter presents the results obtained from the implementation of the **AI-Driven Hospital Queue & Bed Availability Tracking System**, followed by an analysis of its performance, effectiveness, and challenges encountered during development.

Snapshots of the Project and description

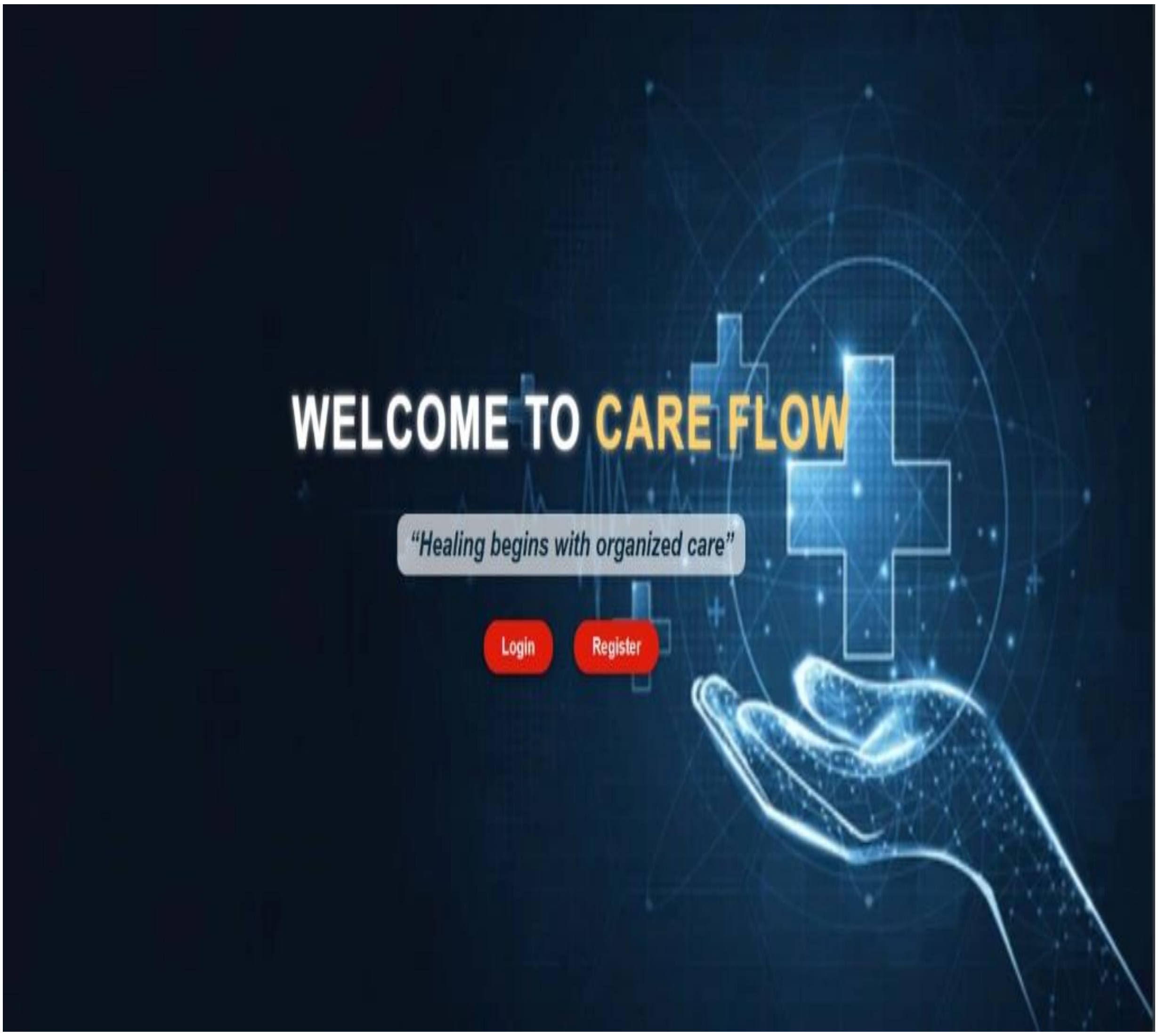


Figure 5.1: Welcome page

The following Figure 5.1 shows that the welcome page provides the main entry point to the system and introduces users to all available modules. It serves as the starting interface where patients and staff begin navigating the application.

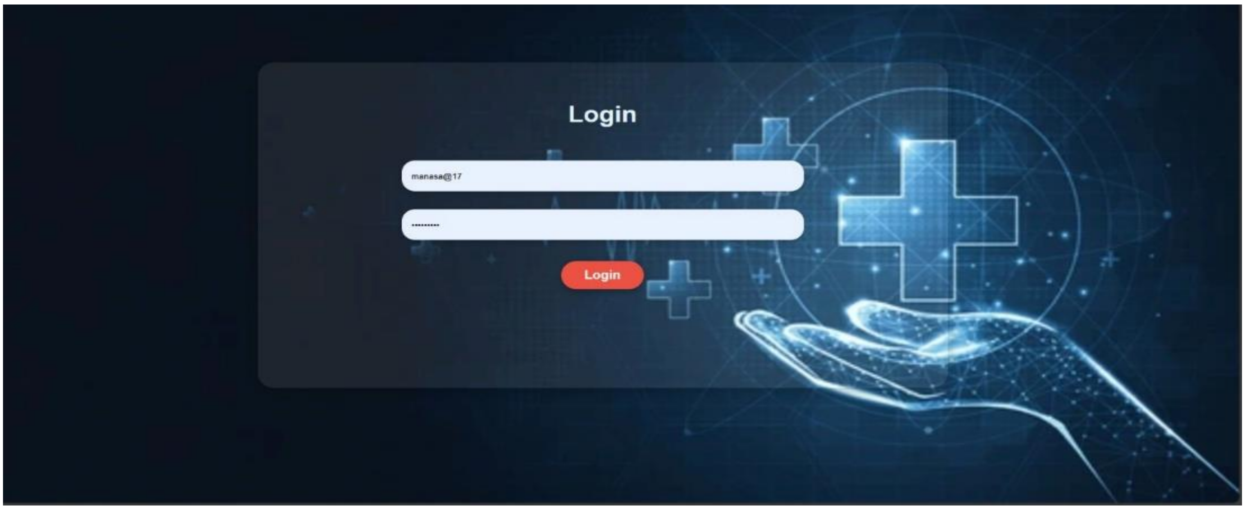


Figure 5.2: Login page

The following Figure 5.2 shows that the login page ensures secure access by allowing users to enter valid credentials before proceeding.It verifies user identity and redirects them to the appropriate dashboard based on their role.

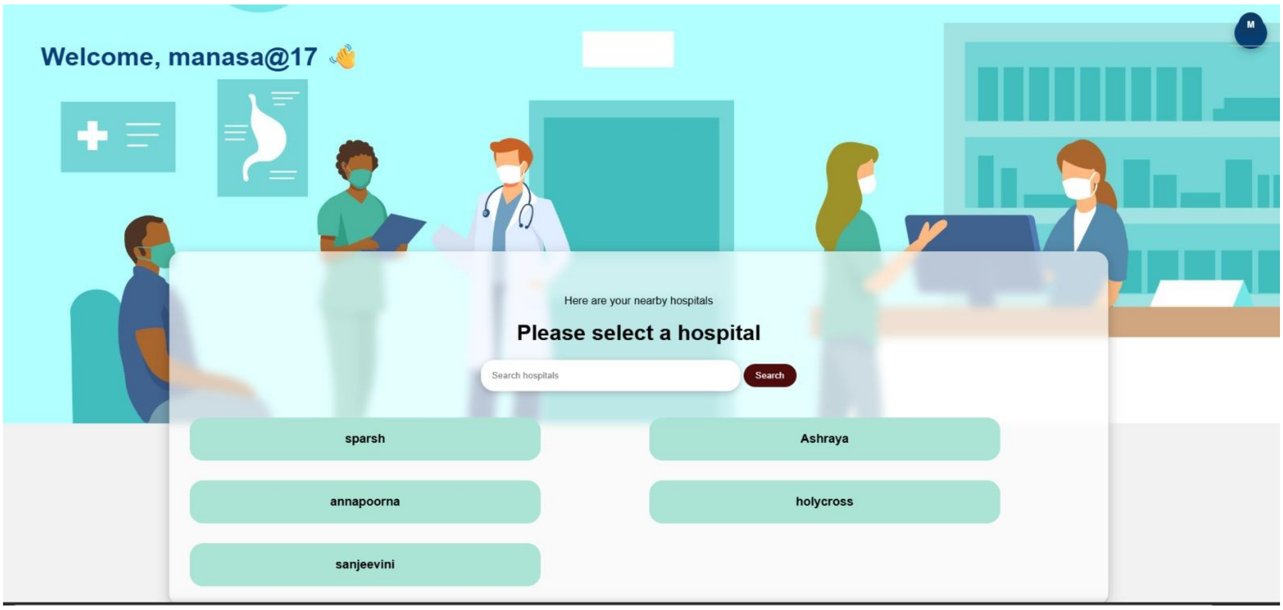


Figure 5.3 : Home page

The following Figure 5.3 shows that the home page offers an organized layout with quick links to queue tracking, bed status, appointments, and chatbot.
It acts as the central navigation hub for both patients and admins.

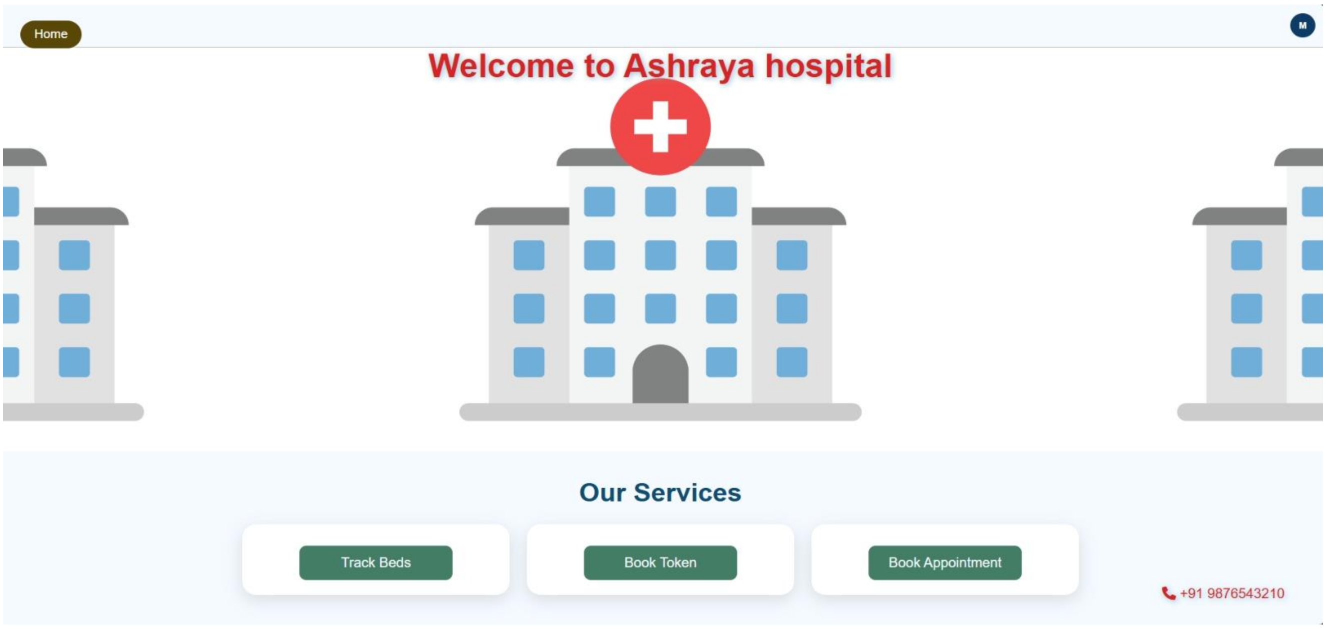
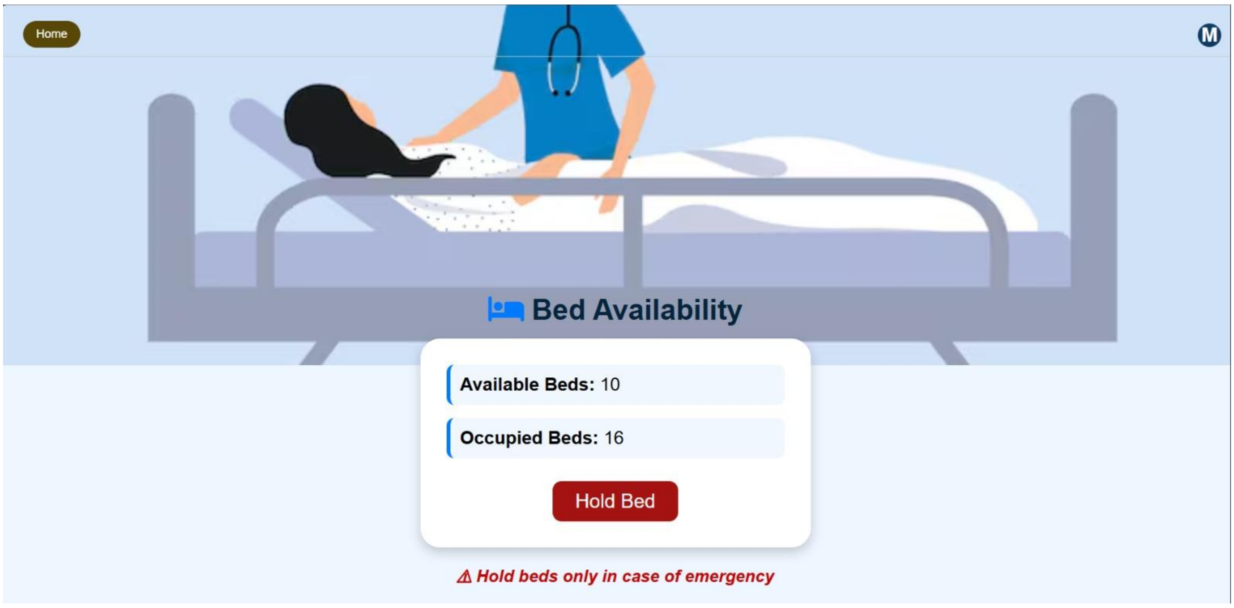


Figure 5.4: Hospital dashboard

The following Figure 5.4 shows that the hospital dashboard provides administrators with real-time insights into queue status, bed usage, and appointments.
It helps staff monitor hospital operations efficiently from a single interface.



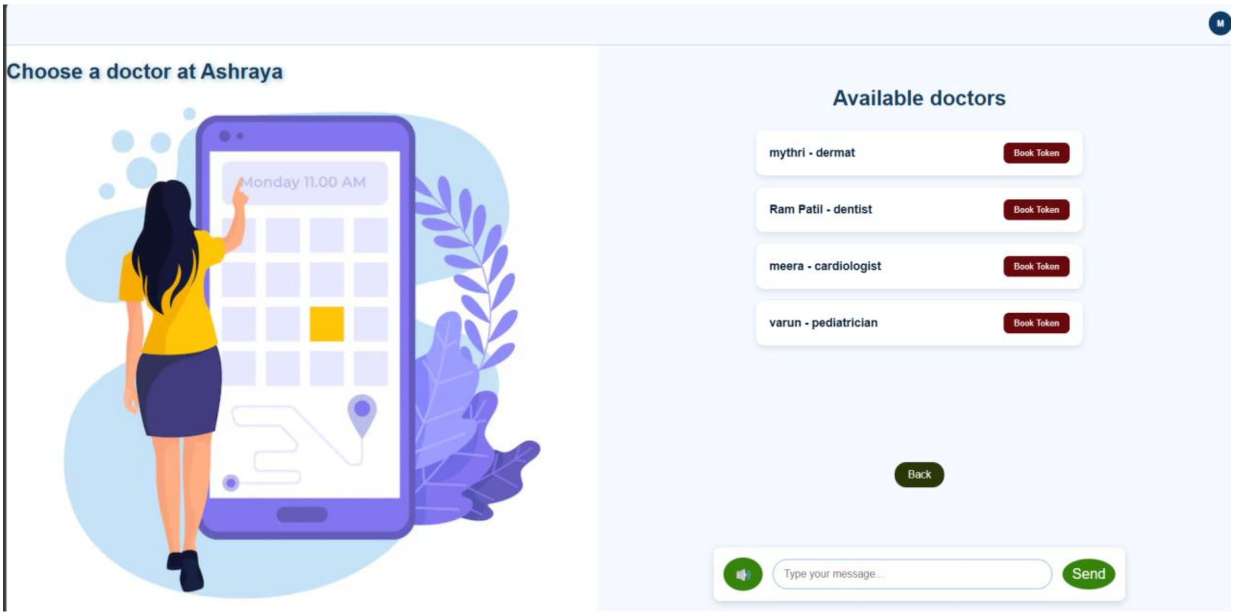
Available Beds: 10

Occupied Beds: 16

⚠ *Hold beds only in case of emergency*

Figure 5.5: Bed availability

The following Figure 5.5 shows that this screen displays up-to-date information on available, occupied, and reserved hospital beds.



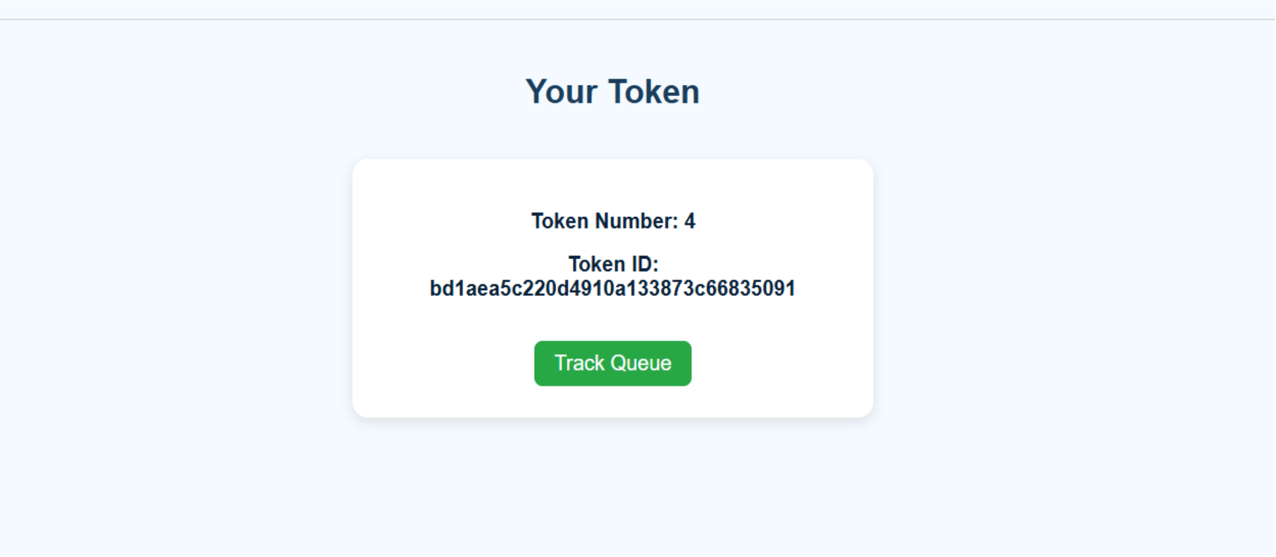
Choose a doctor at Ashraya

Available doctors

mythri - dermat	Book Token
Ram Patil - dentist	Book Token
meera - cardiologist	Book Token
varun - pediatrician	Book Token

Figure 5.6: Digital token booking

The following Figure 5.6 shows that the digital token booking page enables patients to generate a queue token for consultation. It automates the queueing process and reduces manual interruptions.



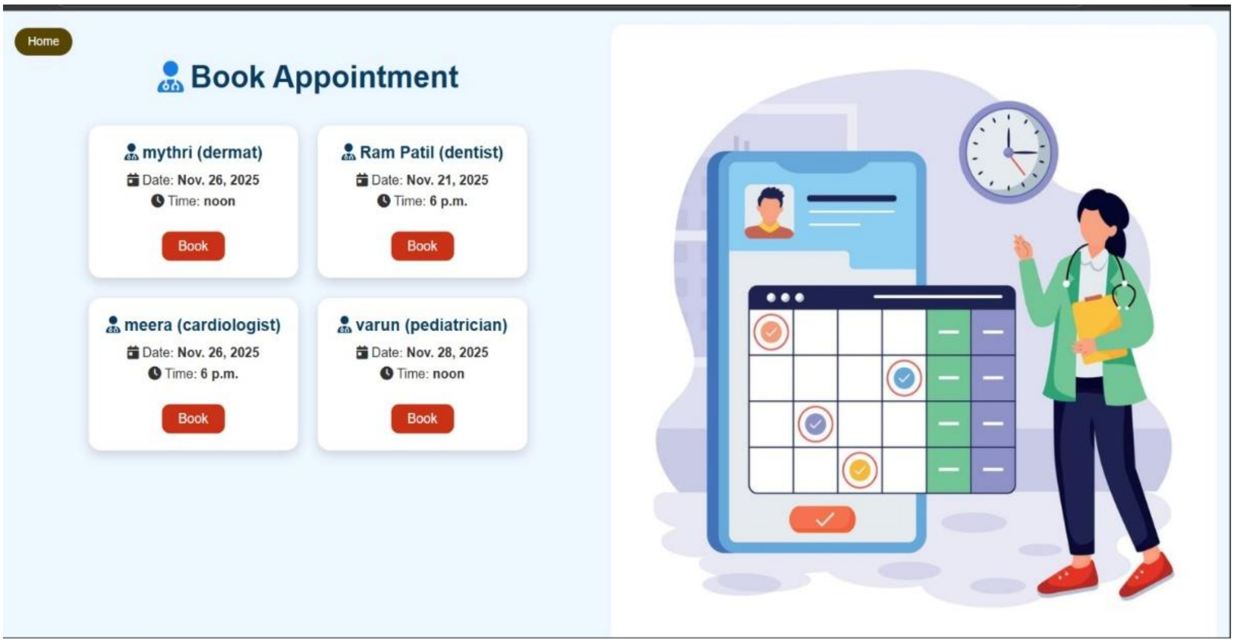
Your Token

Token Number: 4

Token ID: bd1aea5c220d4910a133873c66835091

Figure 5.7: Token

The following Figure 5.7 shows that the token screen shows the generated token number along with the patient's current position in the queue. It provides clarity and transparency by updating the queue status in real time



Book Appointment

mythri (dermat)
Date: Nov. 26, 2025
Time: noon

Ram Patil (dentist)
Date: Nov. 21, 2025
Time: 6 p.m.

meera (cardiologist)
Date: Nov. 26, 2025
Time: 6 p.m.

varun (pediatrician)
Date: Nov. 28, 2025
Time: noon

Figure 5.8: Appointment booking

The following Figure 5.8 shows that the appointment booking page lets users choose a doctor, date, and time slot for consultation. It simplifies appointment scheduling and ensures smooth patient flow.

5.2Discussion

Effectiveness of the System

5.2.1 Improved Transparency:

Patients can easily track their queue position, check bed availability, and receive real-time updates without repeatedly asking hospital staff.

5.2.2 Reduced Waiting Time:

Automated queue management helps minimize unnecessary delays , improving overall patient throughput.

5.2.3 Operational Efficiency:

Staff can allocate beds faster and manage queues more effectively due to centralized, digital information.

5.2.4 Enhanced Patient Experience:

Patients experience less confusion and have clearer insights into their queue status and room availability.

5.2.5 AI Assistance:

The integrated chatbot handles repetitive queries, freeing staff to focus on critical tasks and improving service efficiency.

5.3 Challenges Encountered

During development and testing, several challenges were identified:

5.3.1 Real-time Synchronization

Ensuring instantaneous updates to queue and bed status for all users required optimized API

5.3.2 Handling Concurrent Requests

Simultaneous token generation or bed updates necessitated robust backend mechanisms to prevent data conflicts.

5.3.3 Chatbot Limitations

AI responses depend on prompt context, requiring fine-tuning to provide accurate and relevant guidance.

5.3.4 Database Coordination

Managing multiple modules queue, beds, and appointments within a single database demanded careful relational design and indexing.

5.3.5 User Interface Complexity

Designing an intuitive interface for both patients and staff required multiple iterations to balance simplicity with functionality.

Limitations of the Current System

- Absence of a mobile app version
- Lack of notification system for patients and staff
- Limited analytics and reporting capabilities
- No integration with external hospital systems
- Fully dependent on internet connectivity

Chapter 6

Conclusion and Future Enhancements

6.1 Conclusion

The AI-Driven Hospital Queue & Bed Availability Tracking System effectively addresses critical challenges in hospitals, including patient flow management, bed allocation, and appointment scheduling. By replacing manual processes with an intelligent digital system, the project enhances transparency, reduces waiting times, and improves the overall patient experience.

Key achievements of the system include:

- **Real-time Queue Tracking:** Automated token generation and status updates keep patients informed.
- **Accurate Bed Availability:** Color-coded visual representation provides clear insights into bed occupancy.
- **Simplified Appointment Management:** Easy booking and tracking of patient appointments.
- **Interactive AI Chatbot:** Guides patients and answers common queries efficiently.
- **Role-Based Access Control:** Ensures data security and privacy for patients and staff.

Testing and evaluation demonstrated that the system is efficient, reliable, and user-friendly across all modules. It reduces staff workload, minimizes errors, and streamlines hospital operations, making it a valuable tool for modern healthcare facilities.

6.2 Future Enhancements

To further improve the effectiveness and scalability of the system the following enhancements are recommended in future versions:

- Mobile Application Development
- Push Notifications & Alerts
- Predictive Analytics for Bed Demand
- Integration With EHR (Electronic Health Records)
- Multi-Language Chatbot Support
- Voice-Enabled Assistance
- Advanced Admin Dashboard

REFERENCES

Citation Format:

Books:

- Pressman, R. S. (2014). *Software Engineering: A Practitioner's Approach* (8th ed.). McGraw-Hill.
- Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson.

Online Resources:

- Google Gemini API. (2024). *Google AI Developer Documentation*. Retrieved from: https://ai.google.dev
- W3Schools. (2024). *HTML, CSS & JavaScript Tutorials*. Retrieved from: https://www.w3schools.com

Software Tools:

- Python Software Foundation. (2024). *Python 3.10 Documentation*. Retrieved from https://www.python.org
- GitHub, Inc. (2024). *Version Control Platform*. Retrieved from https://github.com

Example

Database Tool

[1] SQLite Documentation. (2024). *SQLite: Lightweight Relational Database.* Retrieved from https://www.sqlite.org

[2] Reference for database design and management of patient records and queue tokens.