PROJECT REPORT

Hospital Management System

Vinitkumar Dobariya
Jaykumar Sutariya

13.15.2023

Database management system (CS631)

Brief Overview of the Project

Our project is a comprehensive hospital management system designed to streamline hospital operations through a robust web-based platform. Utilizing a blend of front-end technologies like HTML, CSS, and JavaScript, along with a Python and Django framework backend, the system offers a dynamic and user-friendly interface. The database is managed using SQLite, ensuring efficient data handling.

System Capabilities

- **User Interface**: The system provides a user-friendly website where patients can book appointments, submit queries, and access various hospital services online.
- **Admin Panel:** A specialized admin section allows for comprehensive data management, granting full authorization for data access and manipulation. This includes the ability to manually adjust website components using Django model.
- **Data Accessibility:** Admins can delegate data access permissions to other staff members, ensuring controlled and secure data handling.
- **Comprehensive Information Display:** The website showcases all essential hospital information, including doctor availability, staff details (doctors, nurses, and other staff), and room/bed availability.
- **Contact and Support:** A 'Contact Us' feature enables users to raise queries or seek assistance, enhancing patient engagement and support.

Purpose and Scope of the System

- **Data Management and Efficiency:** The system is designed to efficiently manage all hospital data, including appointments, salaries, and patient bed allocations.
- **Streamlining Hospital Operations:** By digitizing key operations, the system significantly reduces administrative burdens, making it easier to handle complex hospital functions.
- **Accessibility and Convenience:** With online appointment bookings and data access, the system enhances the overall patient experience and operational efficiency.

System Requirements Summary

Patient Management

- **Patient Information:** Manage patient details and medical history.
- **Appointment Scheduling:** Book and manage appointments with doctors.
- **Diagnoses History:** Access and review past medical diagnoses.

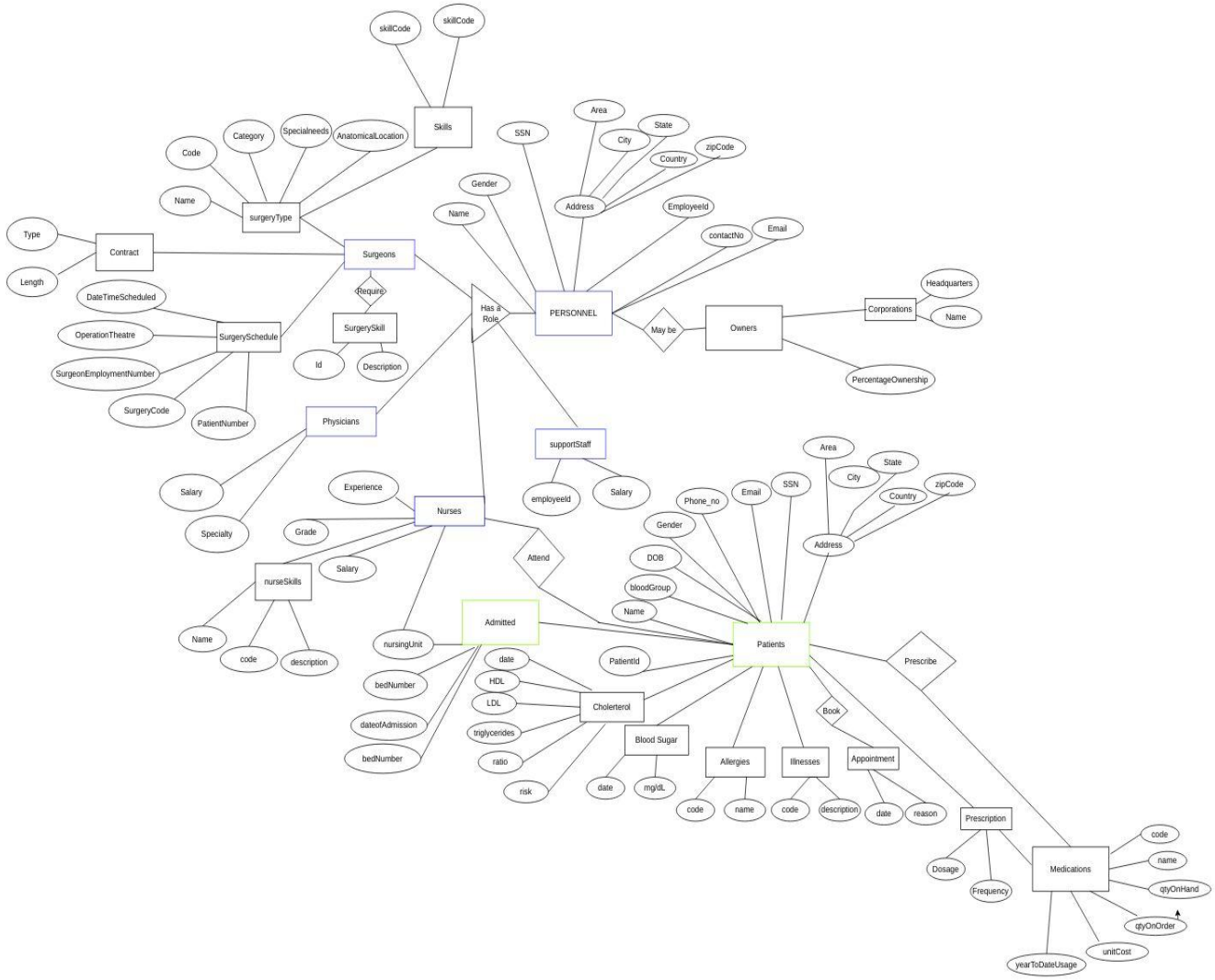
In-Patient Management

- **Room/Beds:** Monitor and assign hospital rooms and beds.
- **Staff Assignments:** Allocate doctors and nurses to patients.
- **Surgery Scheduling:** Organize and manage surgical procedures.

Medical Staff Management

- **Staff Administration:** Add or remove hospital staff members.
- **Staff Overview:** View staff details based on job types.
- **Shift Scheduling:** Arrange and manage work shifts for staff.

Diagram of the Entity-Relationship (ER) Model



Description of Entities and Relationships

- **Patients:** Central to the system, holding information like ID, name, blood group, cholesterol levels, blood sugar, allergies, and illnesses.
- **Medical Staff (Surgeons, Nurses, Support Staff):** Detailed with attributes like employment number, skills, specialties, experience, and roles.

- **Appointments and Prescriptions:** Manage patient appointments and prescriptions, including details like dosage, frequency, and medication.
- **Surgery Schedule:** Outlines surgeries, including type, date, and assigned surgeon.
- **Medications:** Inventory management for medications, including quantity on hand, usage, and cost.

ER Design Choices

- **Comprehensive Patient Management:** The design focuses on detailed patient records, enabling effective tracking of medical history, current treatments, and prescriptions.
- **Integrated Staff Management:** Reflects the complex roles and responsibilities of different medical staff members, facilitating efficient allocation and scheduling.
- **Surgery and Medication Management:** Ensures precise scheduling and inventory management, critical for hospital operations.

Representation of the Relational Database Schema

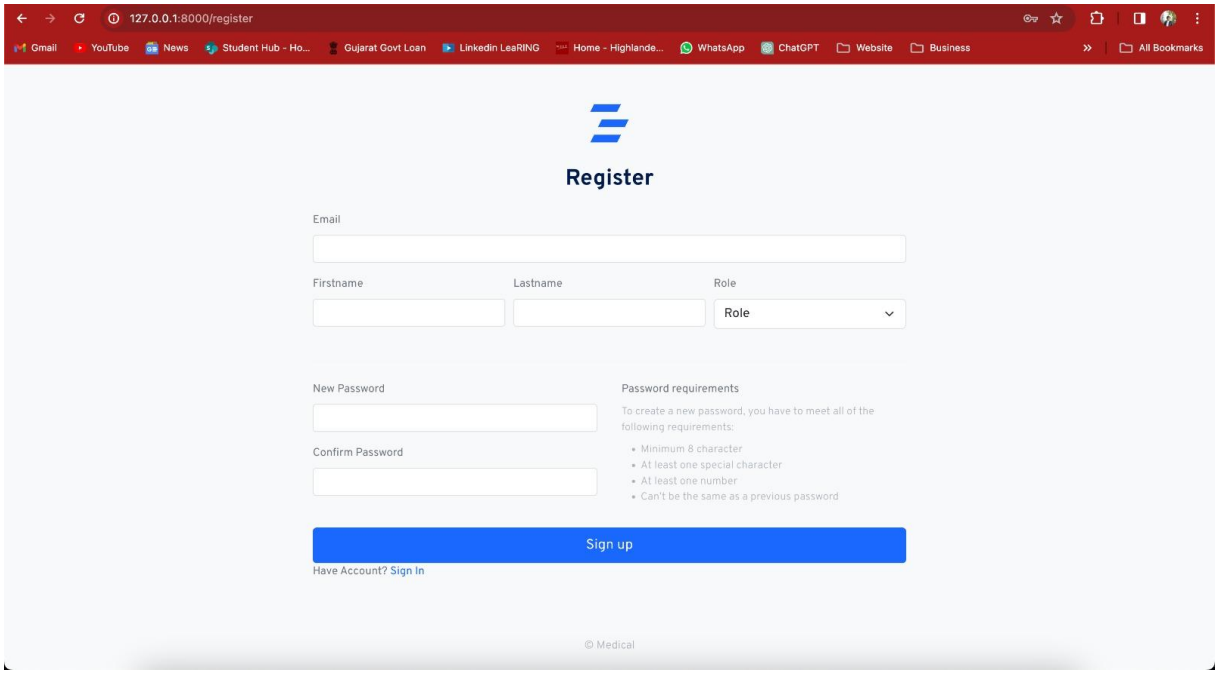
- **Schema Overview:** The database schema consists of multiple interrelated tables, each representing entities like Patients, Medical Staff, Appointments, Surgeries, and Medications.
- **Key Tables and Fields:**
    - **Patients:** Fields include Patient ID, Name, Blood Group, Contact Details, Medical History.
    - **Medical Staff:** Tables for Doctors, Nurses, each with fields for ID, Specialization, Contact Information, Schedule.
    - **Appointments:** Contains Appointment ID, Patient ID, Doctor ID, Date, Time, Purpose.
    - **Medications:** Details about medication including Medication ID, Name, Dosage, Quantity, Cost.
- **Relationships:** The relationships between tables are defined through foreign keys, ensuring data consistency and integrity.

Design Decisions

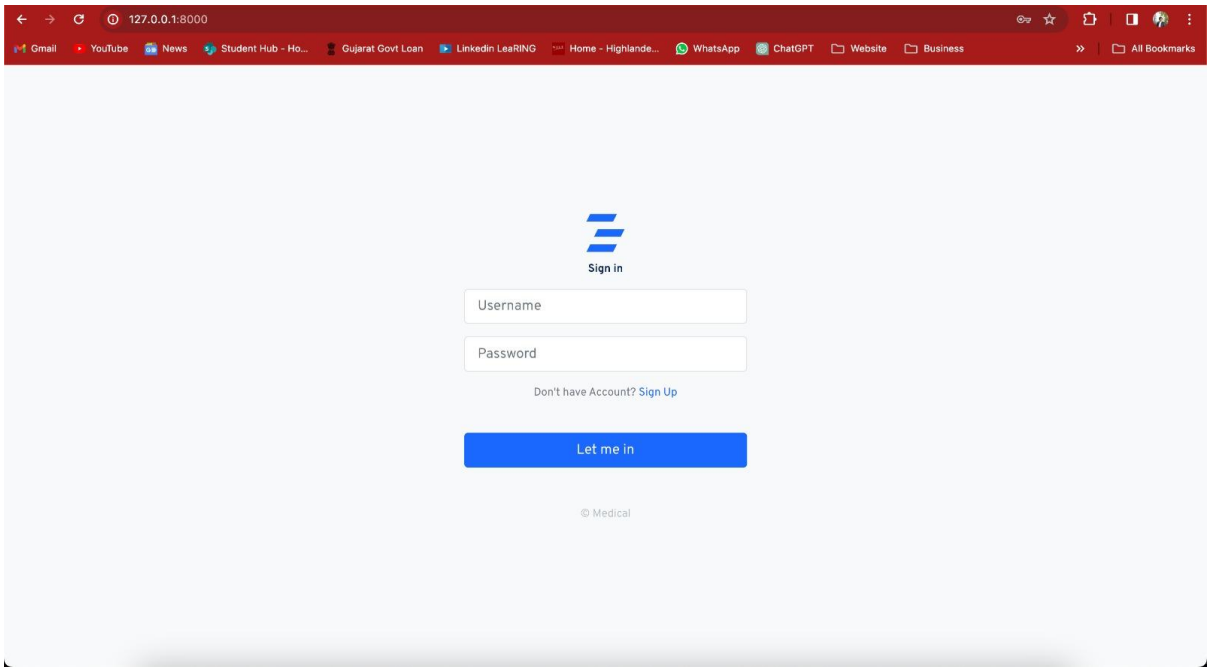
- **Choice of Keys:** Primary keys are carefully chosen for each table to uniquely identify records, ensuring data integrity. Foreign keys establish relationships between tables, allowing for efficient data retrieval and updates.
- **Normalization:** The database is normalized to reduce redundancy and improve data integrity. This minimizes the possibility of data anomalies and ensures efficient storage.
- **Scalability and Flexibility:** The schema is designed to be scalable and flexible, accommodating future expansions and changes in hospital operations.
- **Security and Accessibility:** The design takes into account data security and accessibility, ensuring sensitive information is protected while still being accessible to authorized personnel.

Implementation

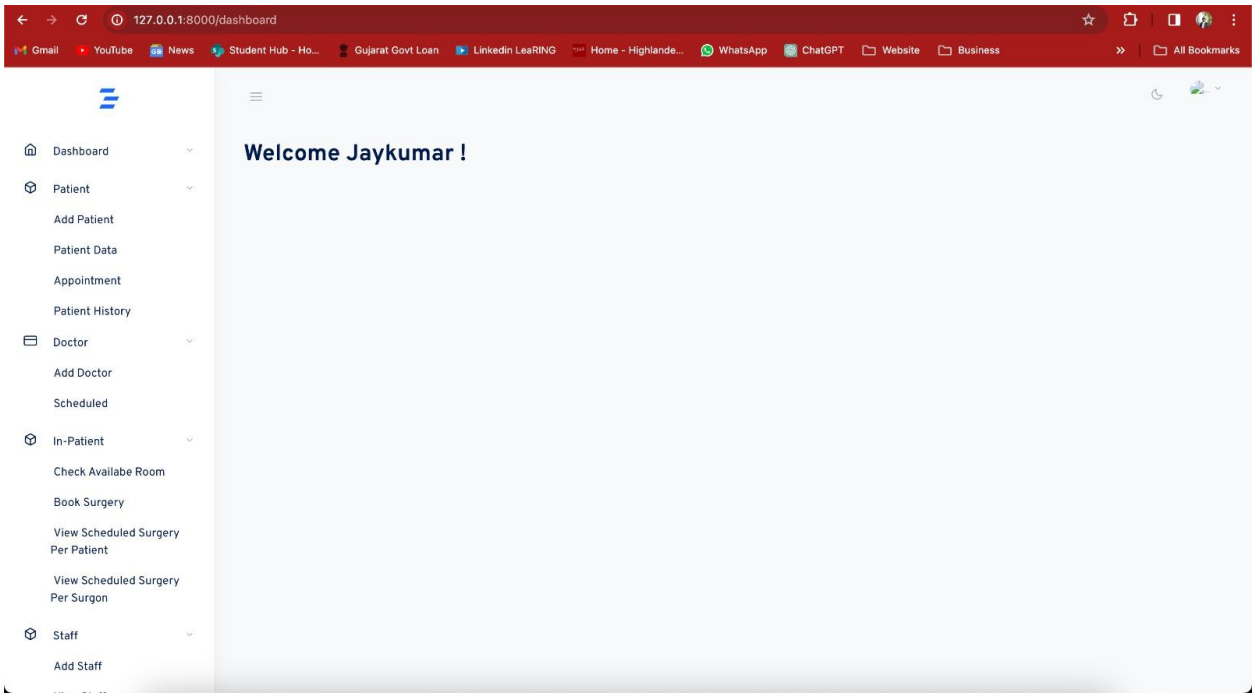
User Registration Page



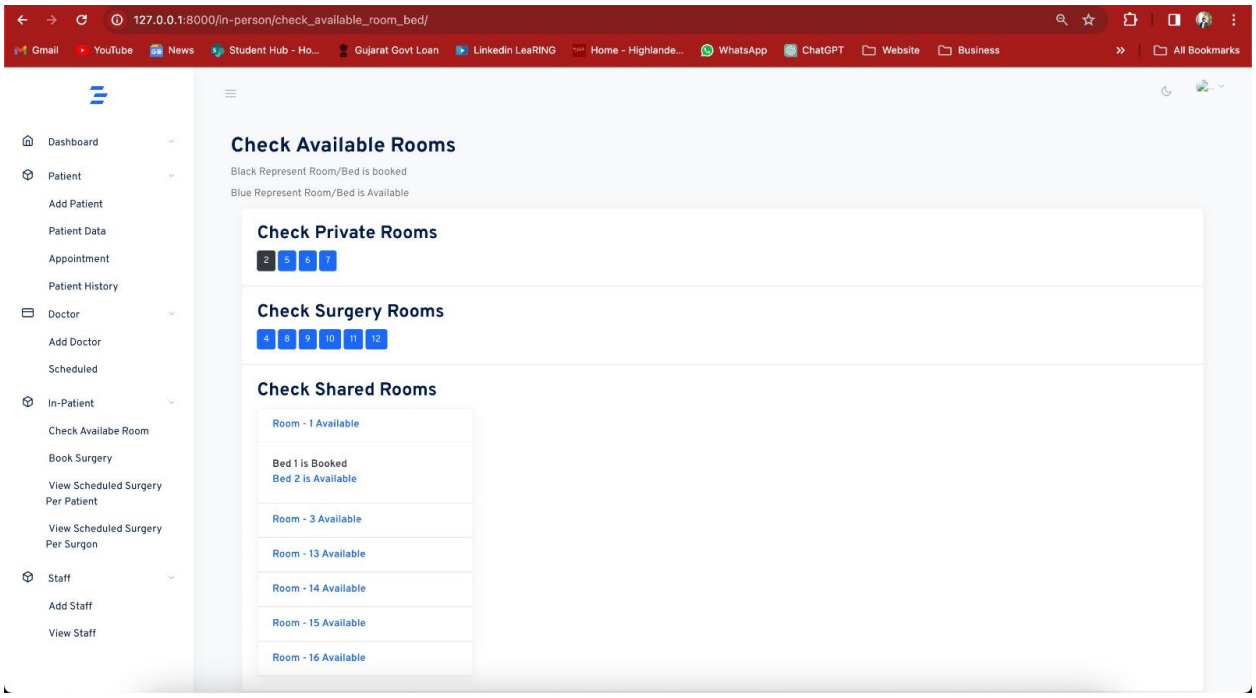
User LoginPage



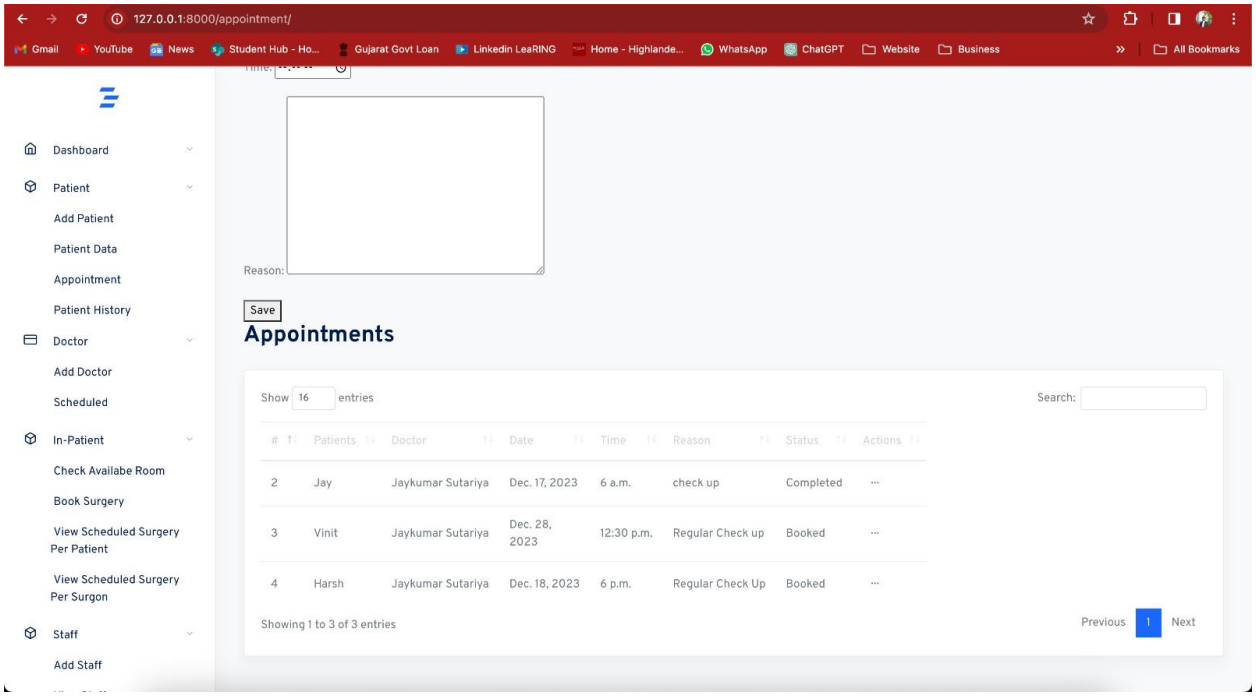
Welcome Page



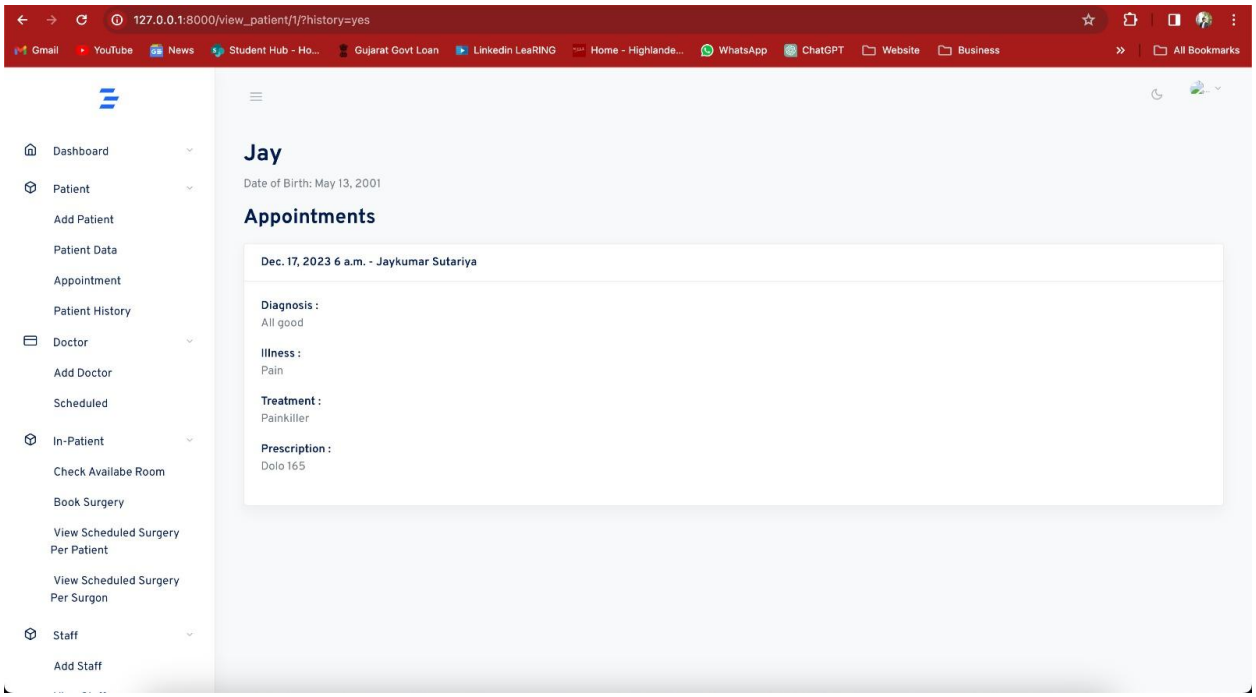
Reserving Room



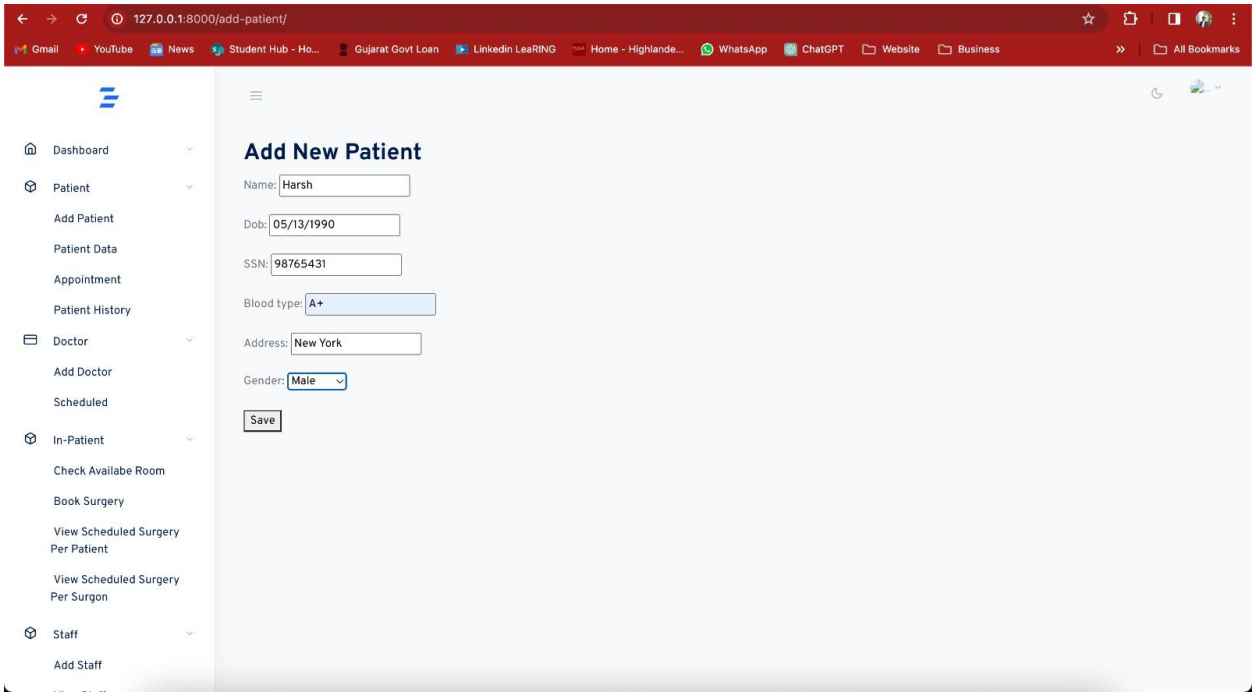
Book an Appoinment



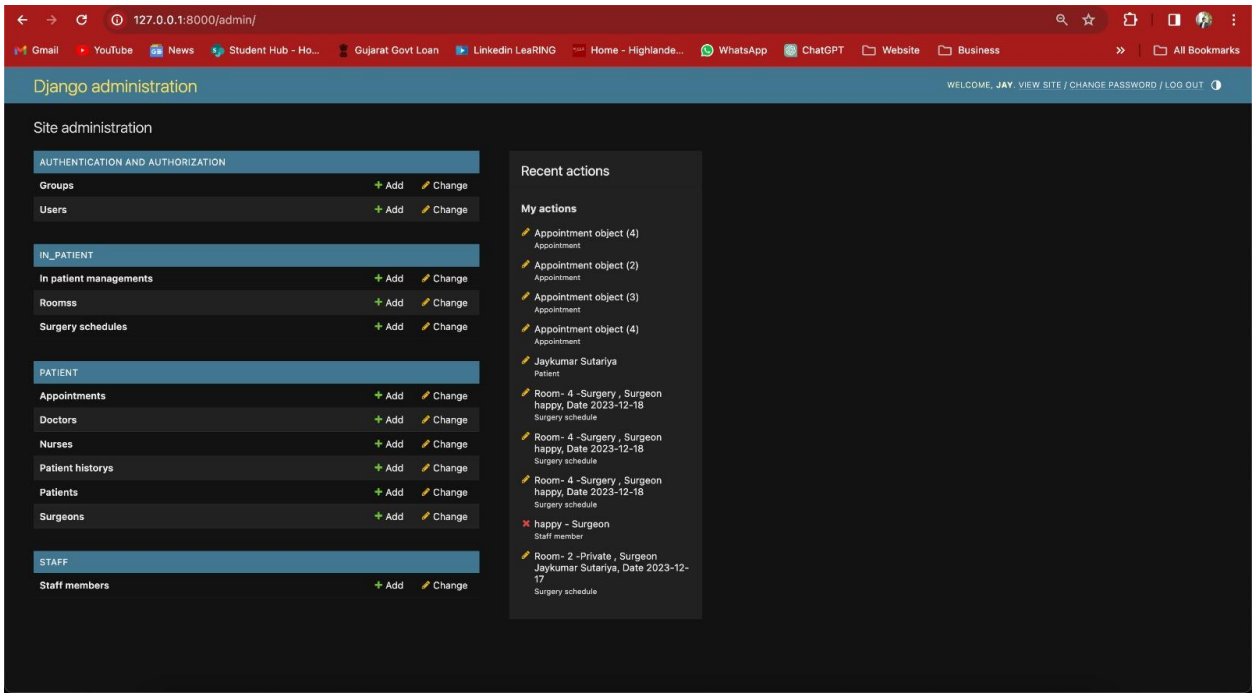
Appointment Booked



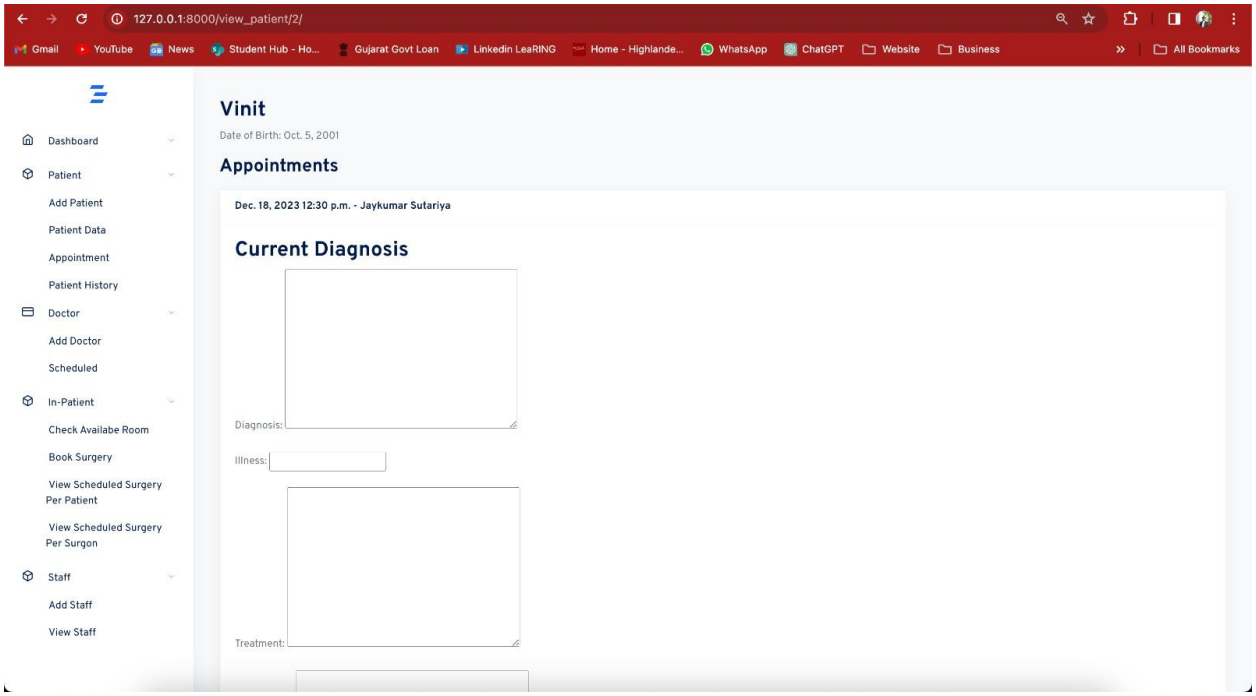
Add a patient



Admin side panel



User can add data



Challenges Faced

- **Data Security: Implementing stringent security measures to protect sensitive patient data.**
- **User Interface Compatibility: Ensuring the system is user-friendly across various devices.**
- **Interdepartmental Data Flow: Achieving seamless data sharing among different hospital departments.**
- **Regulatory Compliance: Aligning system operations with healthcare regulations.**

Conclusion

- **Project Impact: Successfully digitized hospital management, improving efficiency and patient care.**
- **Technological Integration: Showcased the effective use of Python, Django, HTML, and CSS in healthcare.**
- **Learning Experience: Gained valuable insights into the complexities of database systems in medical environments.**
- **Looking Ahead: Confident in the system's foundation and potential for future advancements.**

References

- https://docs.djangoproject.com/en/5.0/
- https://www.w3schools.com/
- https://www.webmd.com/cold-and-flu/what-are-epidemics-pandemics-outbreaks