2025

MEDICARE PRO

TEAM NAME : TEAM V7

TEAM MEMBERS :

Sairam S
Rubesh S
Sam Jeya Ruban S
Rakesh J
Ranjith T

PROBLEM STATEMENT

•Hospitals relying on manual processes face:
•Time-consuming patient registration and appointment scheduling.
•Frequent data entry errors and mismanagement of records.
•Limited transparency in tracking doctor availability and appointment status.
•Security vulnerabilities in handling sensitive patient data.
•A digital, role-based platform is needed to ensure efficiency, security, and accuracy.

SOLUTION

- A web-based Hospital Management System developed with the latest MERN-inspired technologies.
- Enables:
- Patients to self-register, manage profiles, and book appointments.
- Doctors to view and respond to appointment requests.
- Admins to perform full CRUD operations for patients, doctors, and appointments.
- Features secure JWT authentication, real-time updates, and a seamless user experience.

INTRODUCTION

•The Hospital Management System (HMS) is a web-based platform designed to digitize and simplify hospital operations.
•It provides a secure, role-based environment for patients, doctors, and administrators to manage healthcare interactions efficiently.
•The system aims to reduce manual processes, minimize errors, and improve the overall patient experience.

KEY FEATURES

Patients

•Self-register securely and create personal profiles.
•Book, reschedule, or cancel appointments in real time.
•Track appointment status with live updates.

Doctors

•Admin-created accounts ensure secure onboarding.
•View scheduled appointments and patient details.
•Accept or reject appointment requests.

Administrators

•Full CRUD operations for managing patients and doctors.
•Monitor and delete appointments when necessary.
•Maintain system security and data integrity.

SYSTEM ARCHITECTURE

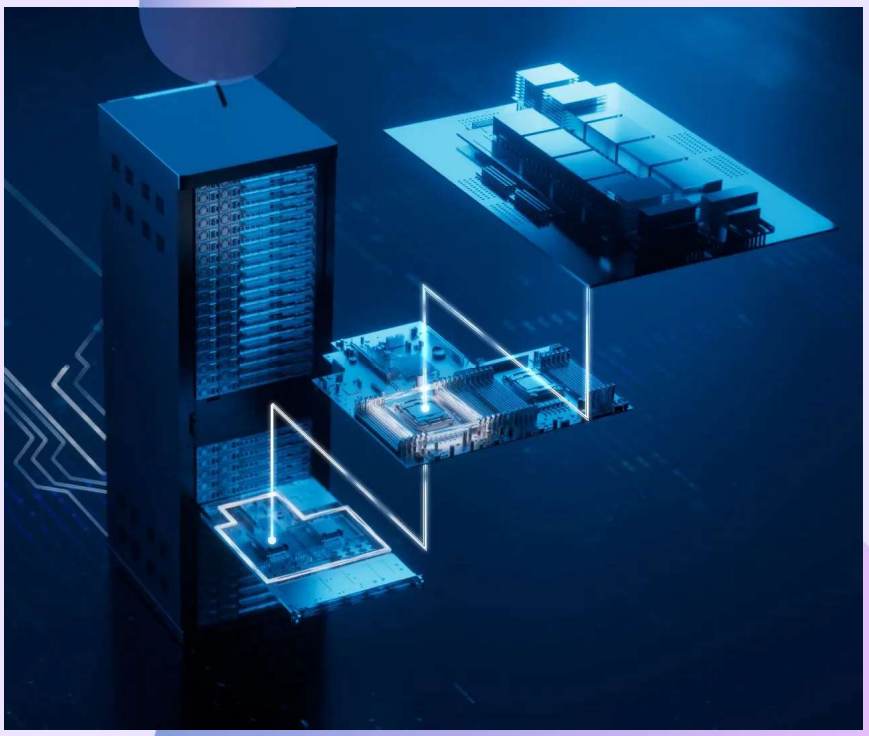
Frontend (Client)
- Tech: React + Vite + Bootstrap
- Role: User interface for Admin, Doctor, Patient.
- Handles login/register, role-based dashboards, and API calls via Axios.

Backend (Server)
- Tech: Node.js + Express
- Role: REST API with JWT authentication & role-based access.
- Handles CRUD for users, profiles, and appointments.

Database
- Tech: MySQL with Prisma ORM
- Role: Stores users, profiles, and appointment data with migrations.

Security
- JWT for sessions, bcrypt for password hashing, strict role-based permissions.

Deployment
- Dev: Local Node & Vite servers.



TECHNOLOGY STACK

Frontend

- React (with Vite) → Fast, modern frontend framework.
- Bootstrap 5 → UI styling and responsive design.
- Axios → For making API requests to the backend.
- NPM → Dependency management.

TECHNOLOGY STACK

Backend

- Node.js → JavaScript runtime environment.
- Express.js → Backend web framework for APIs.
- JWT (JSON Web Token) → Authentication & role-based access control.
- bcrypt.js → Password hashing and security.
- Prisma ORM → Database ORM (Object Relational Mapper).
- MySQL → Relational Database for storing patients, doctors, users, appointments.

Other Tools

- ESLint → Code linting and quality check.
- Vite → Fast development build tool for frontend.
- GitHub → Version control and project hosting.

DATABASE DESIGN

- Users Table: Stores patient and doctor credentials with roles.
- Appointments Table: Maintains appointment details (patient, doctor, status, time).
- Relationships:
- One-to-many mapping between Doctors and Appointments.
- Foreign keys ensure data consistency and easy queries.

CHALLENGES & SOLUTIONS

- **Challenge 1: Role-Based Access Challenge: Restricting Admin, Doctor, and Patient permissions.**
  - Solution: Used JWT + middleware to verify roles before route access.
- **Challenge 2: Frontend–Backend Separation Challenge: Deployment confusion from separate folders.**
  - ·Solution: Managed with two package.json files and clear .env setup.
- **Challenge 3: Password Security Challenge: Secure password storage.**
  - Solution: bcrypt.js hashing with secure login verification.

FUTURE ENHANCEMENTS

- Payment Gateway Integration: Online payments for consultations.
- Automated Notifications: SMS/email alerts for appointment reminders.
- Analytics Dashboard: Hospital performance and patient trend reporting.
- Telemedicine Features: Virtual consultations and video calls.

CONCLUSION

- Developing this Hospital Management System gave hands-on experience in building a full-stack web app with RBAC, JWT+bcrypt authentication, and clear frontend–backend separation.
- Overcoming challenges like role permissions, database relations, and integration improved our problem-solving and structuring skills.
- Future scope includes real-time notifications, medical history tracking, and cloud deployment, making it a strong foundation for healthcare solutions.

Thank You.