Project Report: E-Healthcare Management System

# 1. Title of the Project

E-Healthcare Management System

# 2. Introduction

This project aims to create an E-Healthcare Management System that allows patients to schedule appointments online, receive consultations, and manage their health records digitally.

# 3. Objective

• To provide a user-friendly platform for booking medical appointments online.  
• To facilitate doctors in managing appointments and patient data.  
• To ensure secure and real-time communication.  
• To reduce paperwork and streamline tasks.

# 4. Scope of the Project

• Patients: Register, login, view doctors, and book appointments.  
• Doctors/Admins: Approve or reject appointments and manage slots.  
• Future: Telemedicine, prescription uploads, notifications.

# 5. System Requirements

Hardware:  
• Intel i3+, 4 GB RAM, 100 MB storage  
Software:  
• Windows/Linux/macOS, Node.js, MongoDB, Chrome

# 6. Technology Stack

Frontend: HTML, CSS, JavaScript  
Backend: Node.js, Express.js  
Database: MongoDB  
API: RESTful APIs  
Tools: VS Code, Postman, Git

# 7. System Design

Architecture:  
Frontend ↔ API ↔ Backend ↔ MongoDB

# 8. Module Description

User Module:  
• Register/Login  
• Book & view appointments  
Admin/Doctor Module:  
• View/manage bookings  
• Approve/Reject

# 9. Folder Structure

project-root/  
├── backend/  
│ ├── models/  
│ ├── routes/  
│ └── server.js  
├── frontend/  
│ ├── index.html  
│ └── script.js  
└── README.md

# 10. API Endpoints

Users:  
• POST /api/users/register  
• POST /api/users/login  
Appointments:  
• POST /api/appointments/book  
• GET /api/appointments/:userId  
• PUT /api/appointments/:id

# 11. User Interface

Screens:  
• Home Page  
• Dashboard  
• Admin Panel

# 12. Testing

Manual Testing:  
• Registration, login, booking  
Tools:  
• Postman, DevTools

# 13. Challenges Faced

• API security and validation  
• Backend-frontend communication  
• MongoDB setup

# 14. Security Considerations

• Input validation  
• (Planned) Password encryption with bcrypt  
• (Planned) JWT session management

# 15. Future Enhancements

• Video consultation (WebRTC/Zoom)  
• Notifications  
• Analytics Dashboard  
• Role-based access

# 16. Conclusion

A scalable digital healthcare platform for remote consultations and appointment management.

# 17. Screenshots

(Attach UI screenshots here)

# 18. References

• Node.js: https://nodejs.org/  
• MongoDB: https://www.mongodb.com/  
• Express.js: https://expressjs.com/  
• MDN Docs: https://developer.mozilla.org/