PROJECT SYNOPSIS REPORT

PROJECT TITLE:  
Hospital Management System Using MERN Stack

SUBMITTED TO:  
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

FOR:  
MERN FULL STACK DEVELOPMENT

SUBMITTED BY:  
Name(s): V.N.Nishith

University Roll No(s): 12213109  
Semester: 7th  
Session: 2024–2025

# Index

|  |  |  |
| --- | --- | --- |
| Sr. No | Topic | Page No |
| 1 | Problem Statement | 2 |
| 2 | Title of Project | 2 |
| 3 | Objectives & Key Learnings | 3 |
| 4 | Project Features (Backend & Frontend) | 4 |
| 5 | Technologies Used | 6 |
| 6 | Hosting Links | 7 |
| 7 | Booster Functionalities | 8 |
| 8 | References | 9 |

# 1. Problem Statement

Design and develop a full-stack Hospital Management System that allows administrators to create, update, view, and delete hospital details through a user-friendly frontend and robust backend. The system should be capable of filtering hospitals by city, managing hospital data efficiently, and be deployed online for real-time access.

# 2. Title of Project

Hospital Management System Using Node.js, Express.js, MongoDB, and React.js

# 3. Objectives & Key Learnings

- Understand how to build RESTful APIs using Node.js and Express.  
- Learn MongoDB for backend database operations.  
- Practice React.js to build modern frontend UI.  
- Apply full-stack integration and deployment practices.  
- Implement CRUD operations and frontend forms/data visualization.  
- Deploy backend and frontend using Render and Netlify.

# 4. Project Features

## Backend Tasks (API - RESTful Services)

• Create Hospital  
 - Endpoint: POST /api/v1/hospitals/create  
 - Description: Accepts hospital details and creates a new record in MongoDB.

• Get Hospitals by City  
 - Endpoint: GET /api/v1/hospitals?city=delhi  
 - Description: Returns hospitals matching a specified city.

• Delete Hospital  
 - Endpoint: DELETE /api/v1/hospitals/delete?id=<hospitalId>  
 - Description: Deletes a hospital using the provided ID.

• Update Hospital  
 - Endpoint: PUT /api/v1/hospitals/update?id=<hospitalId>  
 - Description: Updates hospital fields such as rating, image, etc.

• Add Hospital Details  
 - Endpoint: POST /api/v1/hospitals/details?id=<hospitalId>  
 - Description: Adds or updates description, number of doctors, images, etc.

## Frontend Tasks (React UI)

• Hospital Creation Form: Form for name, city, image URL, specialities (multi-select), rating.

• Display by City: Page with dropdown/search bar to filter hospitals by city.

• Hospital Details Page: View hospital's full data (name, city, rating, doctors, departments, etc.)

• Edit Hospital Details: Form with pre-filled data to update hospital information.

• Delete Hospital: Button to delete hospital with confirmation popup.

# 5. Technologies Used

|  |  |
| --- | --- |
| Category | Technology |
| Frontend | React.js, HTML, CSS |
| Backend | Node.js, Express.js |
| Database | MongoDB Compass |
| API Client | Postman |
| Hosting | Netlify (Frontend), Render (Backend) |

# 6. Hosting Links

- Frontend: https://your-frontend.netlify.app  
- Backend: https://your-backend.onrender.com  
- GitHub: https://github.com/Nishith4711

# 7. Booster Functionalities

- Authentication & Authorization: JWT-based login system for secure access to create/edit/delete operations.  
- Context API for State Management: Used in React to share state like user data and hospital list globally.  
- RAG-based QNA Bot: Added simple Q&A bot in frontend to answer hospital-related queries using text prompts.

# 8. References

- https://react.dev  
- https://expressjs.com  
- https://mongoosejs.com  
- https://docs.mongodb.com  
- https://jwt.io  
- https://react-select.com