Project Title	**Hospital Management System**
Technologies	MERN
Domain	Healthcare
Project Level	Difficult

Table

2 Problem Statement:

It takes a lot of time and is quite error-prone to handle the record manually. The goal of this project is to streamline daily operations like room duties, new patient admission, patient discharge, assigning a doctor, and finally bill computation. My best efforts have been made to simplify the challenging process. Management System for Hospitals. A structured, modular design and a menu-driven UI to make it as easy as possible. I made an attempt to create the software so that users wouldn't encounter any problems using it and that additional growth was doable with little work. Despite the fact that I cannot guarantee that this work is exhaustive, the major goal of my exercise is to carry out each Hospital's activity via computer rather than manually which is time consuming.

3 Introduction:

MERN stack, E-commerce, React, JavaScript, Node, Ex-press, MongoDB, web development Nowadays, technology is growing incredibly fast. The rapid innovation of hardware devices makes software technologies to advance as well, automatically take place of old technologies. Because of the significant expanding in the number of electronic devices that use Internet and real-time feature, performance is key.

1. Goal:

1. Automation of routine task
2. *Improved communication and coordination of staff*
3. Achieving a patient-centered approach

4 Categories:

a. Admin
b. Doctor
c. Staff
d. Patient

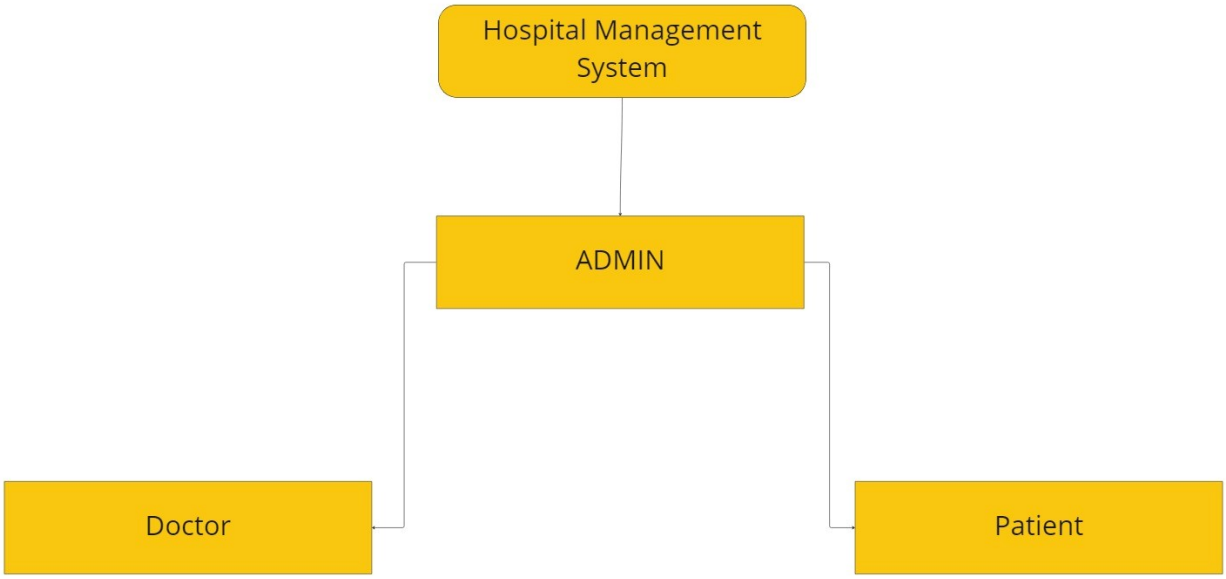
5 Functionality

e. Admin
   i. Patient Registration
   ii. Assigning Doctor and Staff to the Patients
   iii. Patient's checkout
   iv. Treatment
   v. Data Report Generation
   vi. Billing System

f. Doctor
   i. Update patient's daily checkup report on system
   ii. Update medicines for patients

g. Patient

i. First time Registration ii. Able to see Doctors reports, medicines report, bills



miro

6. Project Evaluation metrics:

6.1. Code:

- You are supposed to write code in a modular fashion ● Safe: It can be used without causing harm.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works similarly in every environment (operating system).
- You have to maintain your code on GitHub.
- You must keep your GitHub repo public so anyone can check your code.
- Proper readme file you have to maintain for any project development.

- You should include the basic workflow and execution of the entire project in the readme file on GitHub.
- Follow the coding standards.

6.2. Database:

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

6.3. API Details or User Interface:

You have to expose your complete solution as an API or try to create a user interface for your model testing. Anything will be fine for us.

6.4. Deployment:

Deploy the application on your preferred service.

6.5. Solutions Design:

You have to submit complete solution design strategies in High-level Document (HLD), Lowlevel Document (LLD), and Wireframe documents.

6.6. System Architecture:

You have to submit a system architecture design in your wireframe document and architecture document.

6.7. Optimization of solutions:

Try to optimize your solution on the code level, and architecture level, and mention all of these things in your final submission.

Mention your test cases for your project.

7. Submission requirements:

7.1. High-level Document:

You have to create a high-level document design for your project. You can reference the HLD form below the link.

Sample link: HLD Document Link

7.2. Low-level document:

You have to create a Low-level document design for your project; you can refer to the LLD from the link below.

Sample link: LLD Document Link

7.3. Architecture:

You have to create an Architecture document design for your project; you can refer to the Architecture from the link below.

Sample link: Architecture sample link

7.4. Wireframe:

You have to create a Wireframe document design for your project; refer to the Wireframe from the link below.

Demo link: Wireframe Document Link

7.5. Project code:

You have to submit your code to the GitHub repo in your dashboard when the final submission of your project.

Demo link: Project code sample link

7.6. Detail project report:

You have to create a detailed project report and submit that document as per the given sample.

Demo link: DPR sample link

7.7. Project demo video:

You have to record a project demo video for at least 5 Minutes and submit that link.

The project LinkedIn a post:

You have to post your project details on LinkedIn and submit that post link in your dashboard in your respective field.