

Health Care Connect: A Comprehensive Hospital Management System

¹E. Neelima, ²Chirudeep, ³Supriyo Saha, ⁴Tarrunee

¹Assistant Professor, Dept. of CSE, GITAM (Deemed to be University), Visakhapatnam, Andhra Pradesh, India.

^{2,3,4}Student, GITAM (Deemed to be University), Visakhapatnam, Andhra Pradesh, India.

Abstract - Modern healthcare institutions need the Hospital Management System (HMS) to improve patient care, optimize resource allocation, and simplify operations. In this work, an advanced HMS that leverages the MERN (MongoDB, Express.js, React.js, Node.js) stack—a popular technological stack for creating scalable web applications—is designed and developed. Patient registration, appointment scheduling, inventory control, billing, reporting, and electronic health records (EHR) administration are just a few of the components that make up the proposed system. Express.js and Node.js offer a solid backend framework for handling server-side functionality and API endpoints, while MongoDB enables for flexible data modelling and scalability. The frontend uses React.js to produce a user interface that is easy to use and offers dynamic user experiences. Rapid development, scalability, and adaptability are just a few benefits that make the MERN stack a great option for developing an all-inclusive and flexible HMS. Through the use of contemporary online technologies, this system seeks to improve patient outcomes and satisfaction by increasing healthcare services' accessibility, accuracy, and efficiency

Keywords: ReactJs, MongoDB, ExpressJs, Nodejs, Hospital Management system, HMS, Patient

1. INTRODUCTION

In the field of healthcare, providing excellent patient care and making the best use of available resources depend heavily on the effective administration of hospital operations. To increase productivity overall, increase accuracy, and streamline operations, digital solutions are gradually replacing traditional paper-based methods. The Hospital Management System (HMS), a complete software platform created to automate and integrate several clinical and administrative operations inside healthcare institutions, is one such creative approach. There are several advantages to switching from old systems to digital HMS. Our solution claims to eliminate

paperwork, minimize mistakes, and simplify operations with capabilities including inventory control, appointment scheduling, and electronic health records (EHR) administration. Furthermore, the adaptability of the MERN stack makes it simple to integrate and customize with the hospital's current infrastructure, guaranteeing a seamless transition with little interference with day-to-day operations. This study will examine our MERN-based HMS's design, development, and implementation, stressing its salient characteristics, important technological issues, and possible drawbacks. Our goal is to enable healthcare businesses to maximize operational efficiency and provide better patient care by utilizing the power of contemporary online technology. Our ultimate objectives are to enhance healthcare management systems and raise the standard of care that is ultimately given to patients throughout the globe.

2. Literature Survey

[1] The creation and implementation of a complete system to oversee hospital operations is described in the article "Design and Implementation of a Hospital Management System". In order to expedite hospital procedures, it probably covers a number of issues including patient data, scheduling, invoicing, and inventory management. By offering a customized way to improve productivity and patient care in healthcare facilities, this research advances the area of healthcare informatics.

[2] A document titled "Design and Implementation of Hospital Emergency Nursing Information Management System" provides details about how an emergency nursing information system made especially for hospitals was developed. It probably outlines the capabilities, features, and structure of the system, all of which are intended to facilitate effective administration of emergency nurse responsibilities and patient care. This research uses information technology to improve nursing staff coordination and decision-making, which helps to improve healthcare service delivery during crises. Finding nearby medical services is one of the features that the article "Healthcare

Management System and Domain Search of Nearest Medical Services" investigates in relation to the creation of a healthcare management system. The design, functionality, and deployment of the system are probably covered, with an emphasis on enhancing patient access to medical resources. The goal of this research is to improve patient happiness and healthcare outcomes by leveraging technology to make medical services easily accessible to user.

[3] In-depth analysis of hospital management systems is provided in "A Comprehensive Review of the Design and Implementation of Hospital Management Systems: Challenges and Solutions," which was published in IEEE Access in 2020. This study carefully examines the intricacies, difficulties, and potential fixes related to the planning and execution of such systems. It provides insightful analysis of current frameworks, technical developments, and real-world obstacles, making it a useful resource for legislators, system developers, and healthcare professionals who aim to improve patient care and hospital operations.

[4] In their paper titled "Intelligent Hospital Management System (IHMS)," Baki and Hakan Koyuncu present a novel concept for managing hospitals. This concept involves creating an integrated healthcare management system that combines cognitive technologies like machine learning and artificial intelligence to enhance both patient care and hospital operations. The IHMS aims to improve decision-making processes within healthcare organizations, optimize resource allocation, and reduce manual effort by leveraging these advanced technologies. The paper explores the design, implementation, and potential benefits of the IHMS, providing valuable insights for researchers and healthcare practitioners.

3.Objectives

The principal aim of this project is to establish, execute, and construct a system that provides assistance to the hospital administration, which can only be achieved by complete fulfilment of the ancillary goals that will be discussed shortly. Improving usage efficiency, as determined by the expressivity and consistency of the graphical user interface, is one of the project's goals. When a task takes less time to complete each time they use the system, the user is said to be efficient. Building a

system that enables future enhancements and extensions of the existing capability is another goal. The system ought to include the ability to manage patient and doctor information, make appointments, check prescriptions, and place orders purchasing medicines online and paying online.

4.Proposed Methodology

The project's main goal has been to cut down on the time and amount of paperwork associated with these procedures. Three modules make up the design of this project. A module with a doctor, pharmacy and a patient. A web interface was created for the patient module that enables users to book appointments, pay bills, register on the hospital website, and access, when a patient makes a request, the system assigns the patient to the closest available doctor. Another web interface is the doctor module, which allows a doctor to view the patients that are assigned to him, as well as their histories. include his remarks regarding the patient's visit. The pharmacy has access to two additional sub modules where they can communicate results and bills. Additionally, there is an admin section that allows you to add new staff members and amend current staff members in addition to viewing all patient and doctor records.

5.Methodology

The "Hospital Administration using REACT" project methodology employs a methodical approach for creating a dependable and efficient hospital management system. The project can be divided into numerous sections, each of which concentrates on a certain healthcare area. oversight. Here is a detailed explanation of the procedure, organized by module.

Database Design: Construct a MongoDB structure for databases that is suitable for storing medical records. Indicate which collections are made for specific people, like patients, doctors, appointments, drugs, etc. Make sure your indexing is accurate for efficient searches.

Backend Programming (Express and Node.js): Set up a Node.js environment. Create RESTful APIs with Express.js to handle CRUD operations for various entities. Install authentication and permission systems to provide secure resource access. Mongoose is used to do database operations and integrate with MongoDB. When doing operations like scheduling appointments and creating invoices, use business logic. Make a project using React.js for front-end development. Provide user interfaces (UIs) for several user roles, including administrator, doctor, nurse, and patient. Install the components for inventory management,

appointment scheduling, and viewing patient records. Use state management libraries, like Redux, to manage an application's state. Verify that the user interface is responsive and easily accessed. Combining and Examining: Link the back-end and front- end APIs. Unit tests are used to test front-end and back- end components. Conduct integration tests to ensure smooth communication between the frontend and backend. Check the program for issues related to performance and security. Deployment: Set up the environments for development, staging, and production. Configure pipelines for CI/CD deployment remotely. Surveillance and upkeep: Send out alerts for significant issues. Regularly update dependencies and fix security vulnerabilities. Obtain input from stakeholders and users to aid in future development. Security: When working with sensitive data, encrypt it. Use HTTPS for secure communication. Implement validation and input sanitization to prevent injection attacks. Regularly check the application for security vulnerabilities.

- After logging in, the user can view earlier visits, medications taken, and search for the appointment he needs.
- After this stage, if any patient makes an appointment, a notification will be sent to the doctor's end detailing the appointment and the time it is reserved for him.
- After talking with the doctor, the patient will be requested to take any necessary medications and undergo any necessary tests.
- It will be displayed in the Patient's account after payment for drugs or laboratory testing.

6.Results and discussion

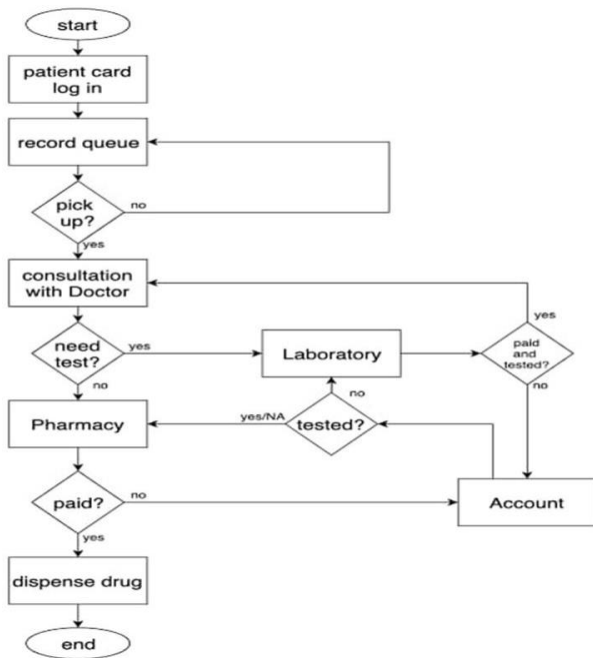


Figure 1: This flowchart describes how the proposed methodology works

- As we can see, the patient must first sign up for the page by entering information such as their phone number, name, password, and email address. If the patient already has an account, he must log in using his credentials.

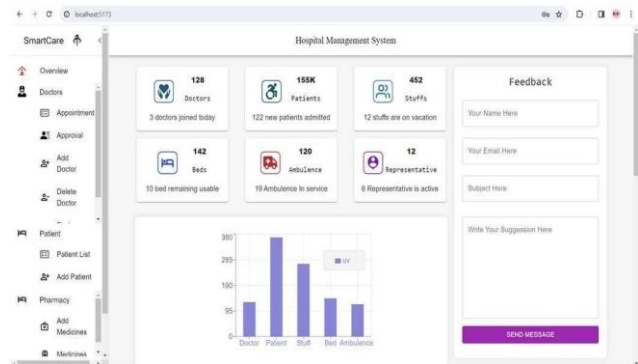


Figure 2: Overview

Main page shows the number of doctors available, number of patients admitted and a dashboard which has modules like doctor, patient, pharmacy.

In addition, the dashboard offers access to functions for scheduling appointments, monitoring medications, and adding and removing doctors and patients.

Overall, the dashboard provides a quick overview of the hospital current state and allows for easy access to commonly used features.

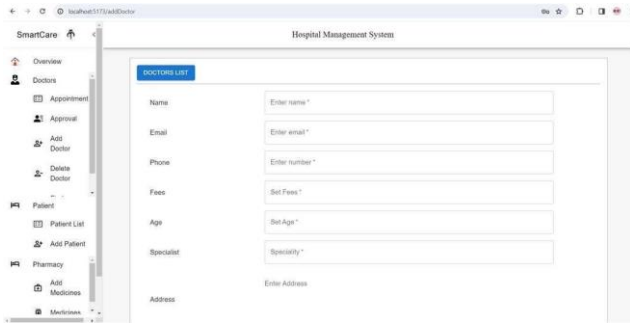


Figure 3: Adding Doctors

- This page allows users to enter information for adding a new doctor to the system. Here are the specific details of the form:
- Name: Enter name - This field allows the user to enter the doctor's name.
- Email: Enter email - This field allows the user to enter the doctor's email address.
- Phone: Enter number - This field allows the user to enter the doctor's phone number.
- Fees: Set Fees - This field allows the user to enter the doctor's fees.
- Set Fees: The user may input the doctor's fees in this section.

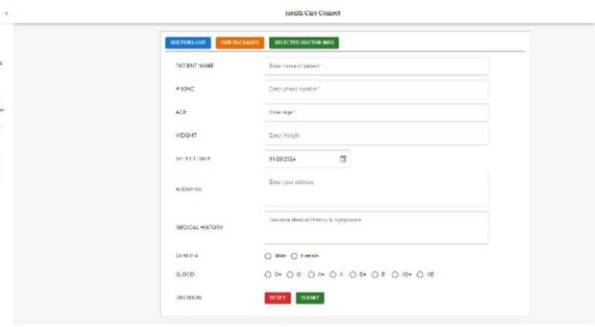


Figure 5: Patient Information

From the figure first we need to select the doctor that we need to make appointment with, after selecting the doctor the patient's basic information is collected like patient's name, phone number, height, weight, appointment date, their address, medical history, gender, blood group. By clicking submit the appointment for that patient is created.

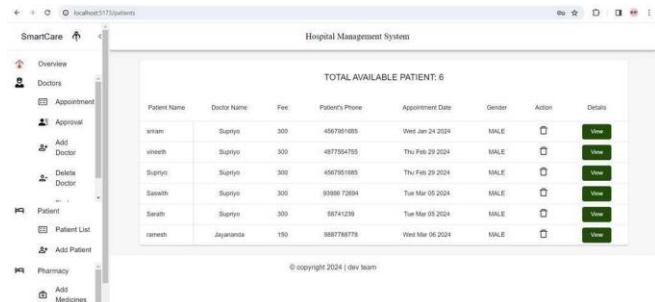


Figure 6: List of Patients

- This page shows the patients list which has some fields like:
- Approval: The appointment's status is displayed in this column. It may be blank or read "Approval," which indicates that it is probably awaiting approval. Patient Name: Name of the patient.
- Doctor: The physician who has been assigned to the case.
- charge: This appointment's consultation charge with the doctor.
- Patient's Phone: The number that the patient is using. Date of Appointment: The time and date of the scheduled meeting.

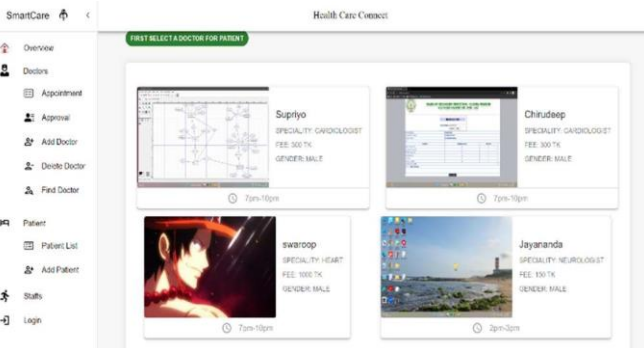


Figure 4: Dashboard

