

Abstract / Synopsis

Mcare (Integrated Patient & Doctor Healthcare Management System).

Introduction

Mcare is a web application designed to improve healthcare by simplifying appointment scheduling, treatment documentation, and medication adherence. It offers a user-friendly interface for patients and providers, addressing key challenges such as managing appointments, maintaining records, and ensuring medication compliance. By Tying to Integrate leveraging cloud technologies, Mcare enhances the efficiency, accessibility, and security of healthcare services, resulting in improved patient satisfaction and optimized provider workflows.

Objective

Mcare improves healthcare by simplifying appointment management, providing secure prescriptions, and automated medication reminders. It supports telemedicine with secure video consultations and centralizes patient records, enhancing efficiency and security for both patients and providers. New features can be added as per requirements.

Scope

- Facilitate appointment scheduling, treatment documentation, and medication reminders for patients.
- Enable doctors to manage schedules, treatments, and ensure patient compliance.
- Support telemedicine for secure remote consultations.
- Ensure secure access to health records and patient data.

- Focus on improving healthcare efficiency, accessibility, and patient satisfaction through a user-friendly interface.

Key Features

- **User Registration and Authentication:** Secure login with role-based access.
- **Appointment Management:** Search doctors, flexible booking, and schedule management.
- **Treatment and Prescription Management:** Electronic prescriptions and access to treatment plans.
- **Medication Reminders:** Automated reminders via SMS, email, or push notifications.
- **Health Records Management:** Centralized records with upload and access for patients and doctors.
- **Telemedicine:** Secure video consultations and integrated messaging.
- **Dashboard Features:**
 - Patient: Manage appointments, records, prescriptions, and consultations.
 - Doctor: Manage schedules, records, prescriptions, and communication.
 - Admin: Oversee usage, roles, security, reports, and feedback

Software Requirements

- **Frontend Technologies:** React.js, Angular.js, Tailwind CSS.
- **Backend Technologies:** Node.js, Express.js
- **Cloud Services(optional):** AWS, GCP, Azure
- **Visual Studio Code, Git**

- **Database:** MongoDB, RDS, Cloud SQL

Hardware Requirements

- Minimum 8 GB RAM or higher.
- Intel core i5 processor.
- Android device for testing and with internet

Advantages

- Centralized management of appointments and records.
- Better communication and coordination between patients and doctors.
- Automated reminders for improved medication adherence.
- Secure and scalable cloud infrastructure.
- Support for in-person and telemedicine consultations.

Disadvantages

- Requires internet connectivity for access.
- Login is necessary to get services from this application.
- Dependency on cloud service providers for uptime and security

ACKNOWLEDGEMENT

I would like to express my thanks to the people who have helped me most throughout my project. I am grateful to my **Prof. Randeep Singh Ghai** for nonstop support for the project. I can't say thank you enough for him tremendous support and help.

I owe my deep gratitude to our HOD of Information Technology Department **Mrs. Pinky Panda** who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

At last but not the least I want to thank all of my friends who helped/treasured me out in completing the project, where they all exchanged their own interesting ideas, thoughts and made this possible to complete my project with all accurate information. I wish to thank my parents for their personal support or attention who inspired/encouraged me to go my own way.

DECLARATION

I hereby declare that the project entitled, “**Mcare Integrated Patient and Doctor Healthcare Management System**” done at **Guru Nanak Khalsa College**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfilment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as semester 5 project as part of our curriculum.

Rajan Vinod Bhagat

Table of Contents

Chapter 1: Introduction	8
1.1 Background.....	8
1.2 Objective.....	9
1.3 Purpose, Scope and Applicability.....	9
1.3.1 Purpose	9
1.3.2 Scope	10
1.3.3 Applicability	10
1.4 Achievements:	11
 Chapter 2: Survey of Technologies.....	 12
2.1 Introduction.....	12
2.2 Existing System	12
2.3 Market Survey	14
2.4 Proposed System.....	16
 Chapter 3: Requirements and Analysis.....	 18
3.1 Problem Definition	18
3.2 Requirements Specification	19
3.3 Planning and Scheduling	21
3.3.1 Gantt Chart	21
3.3.2 WBS Chart	22
3.4 Software and Hardware Requirements	23
3.4.1 Software Requirements:	23
3.4.2 Hardware Requirements:	23
3.5 Preliminary Product Description	23
3.6 Conceptual Model.....	25
3.6.1 Use Case Diagram	25

Chapter 4: System Design	27
4.1 Basic Module	27
4.2 Procedural Design.....	30
4.2.1 Sequence Diagram.....	30
4.2.2 Activity Diagram	31
4.2.3 ER Diagram	33
4.3 User Interface Design	35
4.3.1 Login Page Design	35
4.3.2 Sign In Design	35
4.3.3 Doctor Dashboard Design	36
4.3.4 Patient Dashboard.....	36
4.3.5 Home Page Design	37
4.3.6 Telemedicine Feature Design.....	37

Chapter 1: Introduction

1.1 Background

In recent years, healthcare systems worldwide have encountered numerous inefficiencies, particularly in managing appointments, treatment records, and medication adherence. Traditional systems often lack the technological infrastructure needed to support seamless communication between patients and healthcare providers, leading to suboptimal patient outcomes and administrative burdens on medical staff. Additionally, the growing demand for healthcare services has intensified the need for scalable solutions that can accommodate an increasing volume of patients without sacrificing quality of care.

The **Mcare** project is designed as a comprehensive solution to address these gaps. **Mcare** simplifies the process of appointment booking, rescheduling, and cancellation, allowing patients to interact with healthcare providers with minimal friction. For healthcare professionals, **Mcare** offers an organized system to manage patient interactions, document treatments, and ensure adherence to prescribed medication regimens. A central feature of **Mcare** is its dashboard system, which offers tailored interfaces for patients, doctors, and administrators, ensuring an optimized and efficient experience for all users.

Through features such as seamless appointment management, secure treatment documentation, and telemedicine support, **Mcare** aims to improve the quality and accessibility of healthcare services. Additionally, it ensures the secure and centralized storage of patient health records, enhancing coordination between healthcare stakeholders.

1.2 Objective

The objective of the Mcare project is to create a healthcare management platform. This platform will streamline the management of healthcare services, including the booking, rescheduling, and cancellation of appointments, and support appointment scheduling for doctors. It will also provide secure treatment documentation, medication reminders, and telemedicine support. It Provide a secure platform for doctors to prescribe treatments and medications. The primary aim of the project is to provide patients and healthcare providers with an efficient and user-friendly solution that facilitates easy appointment management, secure medical reports, effective treatment documentation, and seamless telemedicine interactions, ensuring improved healthcare experiences for all users.

1.3 Purpose, Scope and Applicability

1.3.1 Purpose

The purpose of the Mcare project is to create a healthcare management platform that addresses inefficiencies in traditional systems. This platform is designed to enhance patient engagement by providing a user-friendly interface for managing appointments, accessing medical records, and receiving medication reminders. It aims to streamline clinical workflows by offering a centralized system for healthcare providers to manage patient interactions, document treatments, and ensure adherence to medication regimens. Additionally, Mcare seeks to improve telemedicine capabilities with secure video consultations and integrated messaging. The platform will ensure secure and scalable data management to maintain the confidentiality, integrity, and availability of patient health records. Ultimately, Mcare strives to improve the quality, accessibility, and efficiency of healthcare services for patient & doctor.

1.3.2 Scope

The Mcare project aims to improve inefficient manual healthcare systems by developing a comprehensive clinic management platform, replacing manual data storage and paper forms with a more efficient solution.

The scope includes developing a user-friendly platform for patients to manage appointments, access records, and receive medication reminders, while providing healthcare providers with a centralized system for managing interactions, documenting treatments, and monitoring medication adherence. It will also feature a telemedicine module for secure video consultations and messaging.

The project will create a secure, scalable cloud infrastructure for managing patient health records, ensuring data confidentiality, integrity, and availability. It will feature patient registration and authentication, appointment and treatment management, medication reminders, health records management, and a dashboard for all users.

1.3.3 Applicability

The Mcare project has a wide range of applicability in the healthcare industry, particularly in clinics and hospitals in India. The project's automated administration and management system can be applied in various settings, including:

- **Clinics:** The project can be implemented in clinics of all sizes, from small private clinics to large multi-specialty clinics.
- **Hospitals:** The project can be applied in hospitals, including government and private hospitals, to improve the efficiency of their administration and management systems.

- **Rural Healthcare:** The project can be particularly useful in rural areas where access to healthcare is limited and manual systems are still prevalent.
- **Telemedicine:** The project's telemedicine feature can be applied in various settings, including remote healthcare services, online consultations, and telehealth services.
- **Healthcare Chains:** The project can be implemented in healthcare chains, including multi-specialty hospitals and clinics, to standardize their administration and management systems.

1.4 Achievements:

- Successful integration of a secure telemedicine platform allowing patients and doctors to conduct video consultations.
- Implementation of an intuitive scheduling system that enables easy booking, rescheduling, and cancellation of appointments.
- Development of a user-friendly dashboard for patients, doctors, and administrators, enhancing user experience and interaction.
- Integration of secure payment and appointment systems for seamless and confidential transactions.
- Establishing secure medical record access with encryption to protect sensitive patient data, ensuring compliance with healthcare standards.
- Implementation of automated medication reminders to improve patient adherence to treatment plans and promote better health outcomes.

Chapter 2: Survey of Technologies

2.1 Introduction

In day to day life, we will need to visit hospitals or clinics for various medical purposes. It may be for doctor's appointments, medical check-ups, or to purchase medicines. Nowadays, it is really hard to get some time to visit hospitals or clinics due to busy lifestyles or long waiting queues. In order to solve this, Clinic Management Systems like Mcare have been developed. Using these systems, patients can manage their medical records, book appointments online, access medical services digitally, and even consult with doctors remotely through telemedicine. Moreover, doctors can also use the system to schedule appointments, provide digital prescriptions, and share medical reports with patients, making healthcare more accessible, convenient, and efficient.

2.2 Existing System

Currently, many healthcare clinic or hospital centers use manual systems to manage and store essential data. This approach involves numerous paper forms and scattered databases throughout the facility. Data often becomes fragmented and fails to adhere to management standards. In many healthcare centers, hospital or clinic traditional manual systems are still used to manage and store patient data. These systems rely heavily on physical documents and manual processes, which can lead to various inefficiencies and challenges. Forms are frequently misplaced during transfers between departments, necessitating thorough reviews to ensure no information is lost. Multiple copies of the same data lead to inconsistencies across various records.

Patients and healthcare providers face difficulties due to fragmented data, lost or misplaced documents, and inconsistent records across different

departments. Manual systems often lead to inefficiencies, such as longer wait times, higher chances of errors, and delays in accessing critical information. Additionally, these systems struggle with data security and require extensive administrative effort to manage and process information.

Disadvantage of current existing manual system of healthcare:

- **Time-Consuming** The existing system is tedious, with a significant amount of time wasted in searching and organizing patient data.
- **Poor Security and Data Protection** The system lacks adequate security measures, putting patient data at risk of unauthorized access and compromising confidentiality.
- **File Mismanagement** There is a high degree of scattering of patient files, making it difficult to locate and retrieve information when needed.
- **Long Wait Times** Patients are forced to wait for longer periods due to the inefficiencies of the manual system.
- **Data Inconsistencies** The manual system is prone to errors, miscopying, and inconsistencies in data entry, which can lead to inaccurate diagnoses and treatment plans.
- **Limited Data Sharing and Patient Services** The existing system hinders effective data sharing and patient services, leading to poor healthcare outcomes.
- **Lack of Security** The system lacks adequate security measures, putting patient data at risk of unauthorized access and compromising confidentiality.
- **Data Duplication** The manual system leads to redundant data entry, resulting in data inconsistencies and errors.

2.3 Market Survey

The healthcare industry is witnessing a significant shift towards digitalization, with various clinic management systems emerging to cater to the needs of patients and healthcare providers. However, many of these systems have limitations and drawbacks. The market for healthcare management systems is diverse, with various systems available across different price ranges and from numerous providers, including Practo, Zocdoc, Healthgrades, and others. Many alternative systems lack a comprehensive dashboard, making it difficult for patients, doctors, and administrators to manage appointments, records, and communication. Additionally, existing systems often fail to provide secure prescription management, putting patient data at risk of unauthorized access. Furthermore, automated medication reminders are often absent, leading to poor medication adherence. Finally, few systems integrate telemedicine capabilities, limiting the scope of remote consultations.

The following key points were observed during the market survey:

Alternative Systems and their Problems

- **Manual Systems:** Manual systems are still prevalent in many healthcare centers, relying on physical documents and manual processes. These systems are time-consuming, prone to errors, and lack data security.
- **Existing Digital Systems:** Many existing digital systems are fragmented, with limited features and functionalities. They often lack user-friendly interfaces, making it difficult for patients and healthcare providers to navigate.
- **Telemedicine Platforms:** Telemedicine platforms are limited in their scope, focusing primarily on video consultations and lacking comprehensive features for clinic management.

Alternative Websites in market

1. Practo:

- Disadvantages: Limited features for clinic management, no integrated telemedicine, and no secure prescription management.
- Features Missing: Comprehensive dashboard, automated medication reminders, and secure prescription management.

2. Zocdoc:

- Disadvantages: Primarily focused on appointment scheduling, lacks comprehensive clinic management features, and no telemedicine integration.
- Features Missing: Secure prescription management, automated medication reminders, and integrated telemedicine.

3. Healthgrades:

- Disadvantages: Limited features for clinic management, no telemedicine integration, and no secure prescription management.
- Features Missing: Comprehensive dashboard, automated medication reminders, and integrated telemedicine.

The existing healthcare management systems have several limitations, including the lack of comprehensive dashboards, secure prescription management, automated medication reminders, and integrated telemedicine capabilities, which can lead to inefficiencies, errors, and poor patient outcomes.

2.4 Proposed System

To remove all the disadvantages of conventional methods, **Mcare** is a system is proposed which is a comprehensive cloud-based healthcare management platform. The purpose of **Mcare** system is to improve the efficiency and accessibility of healthcare services, reduce wait times, and enhance patient engagement. One can access medical records, schedule appointments, and receive medication reminders online. This system can save time and improve healthcare outcomes because it provides a centralized and automated management system for healthcare providers and patients.

This platform offers seamless access to health records, appointment scheduling, and secure online communication for both patients and healthcare professionals. The project also ensures data security and confidentiality, providing a user-friendly interface for administrators to manage clinic operations.

To further enhance the **Mcare** system, we are considering the integration of advanced cloud technology. This would improve scalability, data accessibility, and overall system efficiency, aiming to provide a more seamless and reliable experience for users. However, the implementation of this feature is still under exploration, and its feasibility will depend on future assessments of the project's needs and available resources.

Mcare application provides various features such as

- **Appointment Management:** Schedule, reschedule, and manage appointments.
- **Treatment and Prescription Management:** Document treatments and issue electronic prescriptions.
- **Medication Reminders:** Automated reminders for medication adherence.

- **Health Records Management:** Access and manage medical records securely.
- **Telemedicine:** Secure video consultations and integrated messaging.
- **Comprehensive Dashboard:** Centralized interface for patients, doctors, and administrators to manage their activities.
- **Optional Feature:** We are exploring the integration of **cloud technology** into the **Mcare** project to enhance the system's scalability, data accessibility, and performance, allowing for more flexibility in future expansions.

Advantages of Proposed System

- Enhanced patient care through online appointments and medication reminders.
- Secure access to patient health records for both doctors and patients.
- Telemedicine feature allows for remote consultations, improving healthcare accessibility.
- Reduces paperwork and administrative delays.
- Streamlines clinic operations, improving efficiency and reducing human errors.
- Real-time access to patient information, ensuring better decision-making.
- Ensures data security, privacy, and compliance with healthcare standards.
- Supports multi-user roles (patients, doctors, and administrators) with personalized dashboards.

Chapter 3: Requirements and Analysis

3.1 Problem Definition

The healthcare system faces numerous challenges related to inefficiencies in appointment scheduling, manual record-keeping, and the lack of timely communication between patients and healthcare providers. Traditional systems often rely on outdated methods, such as paper-based forms and manual processes, which hinder the ability to provide timely and personalized healthcare services. This leads to long wait times, missed appointments, and inefficient use of healthcare resources. For patients, accessing medical services is often cumbersome due to the lack of streamlined communication and ease in booking appointments or receiving updates on treatment plans.

On the healthcare provider side, managing schedules, patient records, and treatment histories remains a time-consuming task, compounded by the absence of a secure and centralized platform for data management. The reliance on manual records not only increases the risk of errors but also limits the ability to provide holistic and continuous care. With the growing demand for personalized healthcare services, healthcare professionals need a more efficient system to manage their interactions with patients, especially when dealing with follow-up treatments, medication adherence, and telemedicine consultations.

The Mcare project seeks to address these problems by offering a comprehensive, web-based solution designed to streamline appointment scheduling, enhance patient-provider communication, and secure management of medical records. By transitioning away from paper-based systems, Mcare will improve efficiency, reduce errors, and support automated medication reminders to ensure better adherence. Additionally,

it will enable secure remote consultations via telemedicine, helping patients receive care from the comfort of their homes. Through a centralized and user-friendly interface, Mcare will not only enhance healthcare accessibility for patients but also improve operational efficiency for healthcare providers, ensuring a higher standard of care for all stakeholders.

3.2 Requirements Specification

Following Requirements will be implemented in the Mcare web application:

3.2.1 Functional Requirements:

1. **User Roles and Authentication:** Support for Patient, Doctor, and Admin roles with secure login. Admins manage users and system performance.
2. **Appointment Management:** Patients can search doctors, book, reschedule, and cancel appointments. Doctors manage schedules, and notifications are sent automatically.
3. **Medical Records and Treatment:** Doctors update patient records; patients view medical history and receive electronic prescriptions.
4. **Medication Reminders:** Automated reminders for patients to take medications.
5. **Telemedicine Support:** Video consultations and secure messaging for follow-up.
6. **Admin Features:** Admins manage accounts, monitor performance, ensure security, and resolve user complaints.

3.2.2 Non-Functional Requirements:

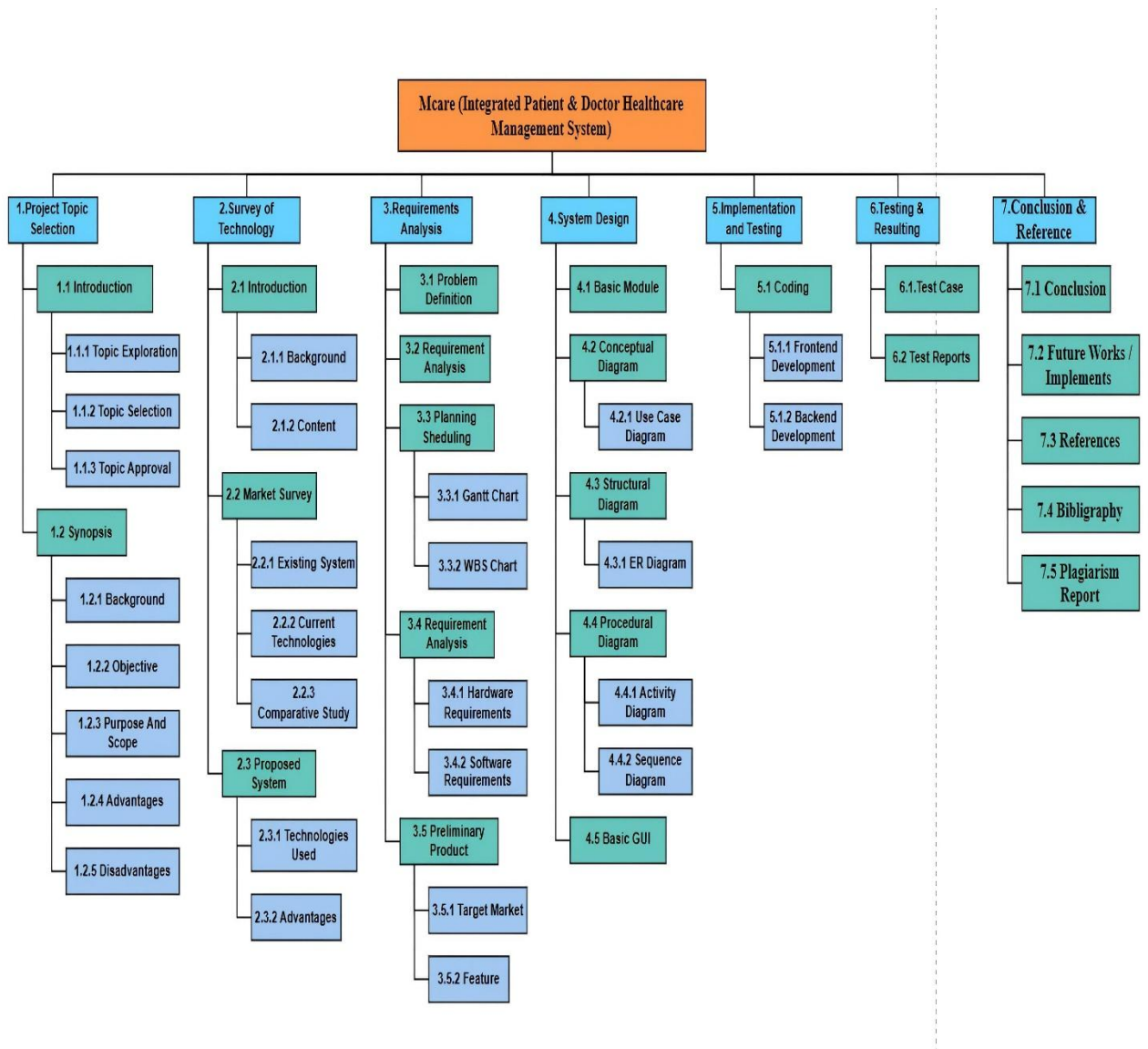
1. **Security:** Data encryption, role-based access control, and compliance with privacy laws.
2. **Performance:** Handle multiple users with real-time processing.
3. **Usability:** Intuitive, responsive interface for all devices.
4. **Scalability:** Expandable to accommodate more users and clinics.
5. **Reliability:** High availability with regular backups.
6. **Data Storage:** Secure centralized storage with daily backups.

3.3 Planning and Scheduling

3.3.1 Gantt Chart



3.3.2 WBS Chart



3.4 Software and Hardware Requirements

3.4.1 Software Requirements:

- **Frontend:** HTML, JavaScript, React.js or Angular.js for UI, and Tailwind CSS for styling.
- **Backend:** Node.js and Express.js for server-side operations; MongoDB for data management.
- **Cloud Services (Optional):** AWS, GCP, or Azure for scalable deployment.
- **Development Tools:** Visual Studio Code for coding; Git for version control.
- **Database:** MongoDB for user data, Cloud SQL for relational databases (optional).

3.4.2 Hardware Requirements:

- **Development & Testing:** Minimum 8 GB RAM, Intel Core i5 processor, 100 GB storage.
- **Mobile Testing:** Android device with internet for telemedicine and mobile responsiveness.
- **Network:** Stable internet connection and high-speed broadband for real-time operations.

3.5 Preliminary Product Description

The Mcare system is designed to be implemented in healthcare institutions ranging from small clinics to large multi-specialty hospitals. This system provides a seamless platform where patients can book appointments with doctors through a mobile application or web portal. It includes a user-

friendly interface to manage doctor availability, schedule patient appointments, and track medical history.

The system supports 24/7 appointment scheduling, and real-time notifications help patients stay updated about upcoming visits. With integrated telemedicine capabilities, patients can consult doctors remotely, enhancing healthcare accessibility. The system's secure platform ensures that sensitive medical records are safely stored and shared.

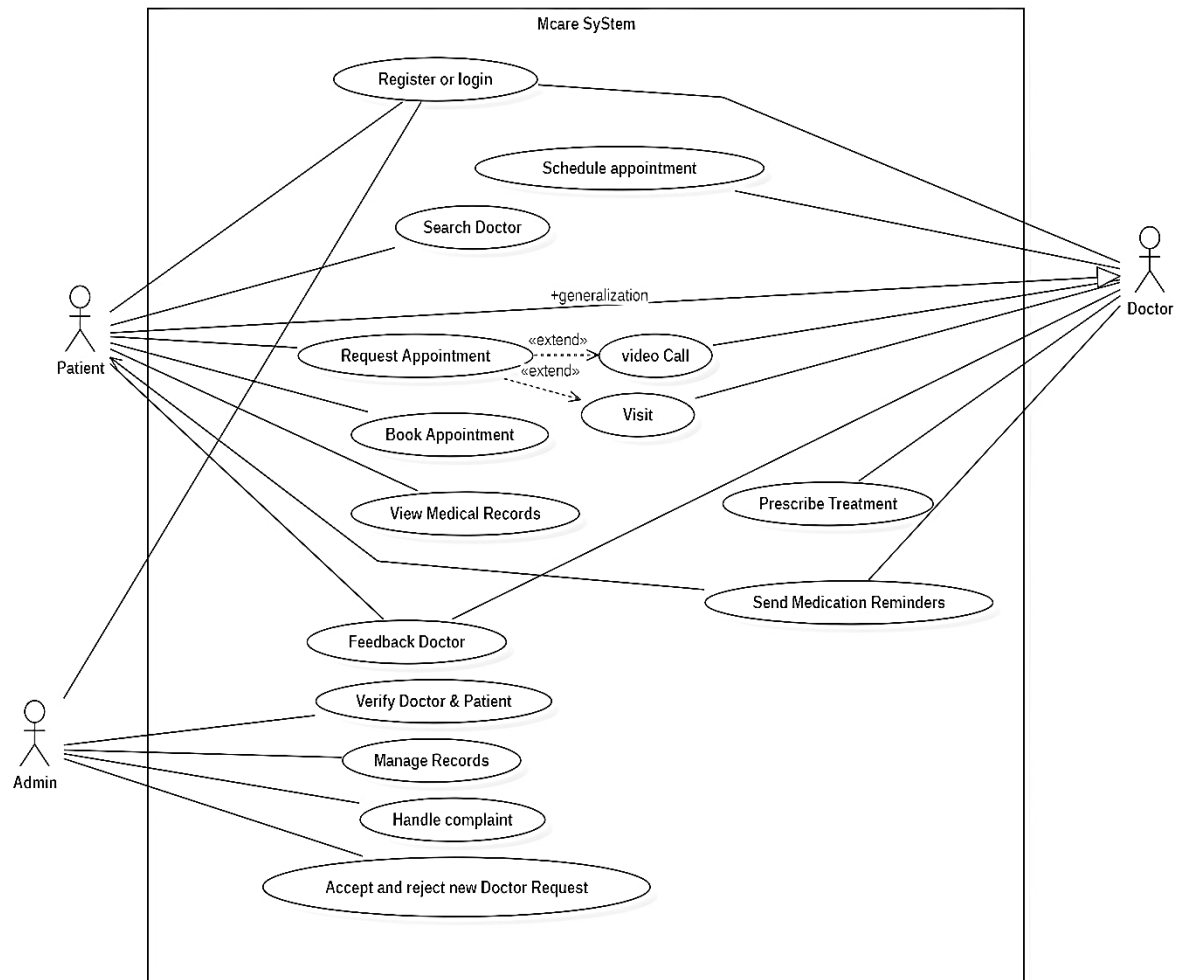
Since the application is available on smartphones, it allows easy access from anywhere, enabling both doctors and patients to interact conveniently. The platform reduces manual administrative tasks, allowing medical staff to focus more on patient care.

Applications:

- The system can allow patients to create accounts, store medical history, and track appointments.
- It can send health alerts for follow-up appointments, medication schedules, and test results.
- Integration with digital payment platforms can streamline payments and billing.
- Video consultations can be added for remote check-ups.
- Patients can choose specialized services like consultations, labs, or diagnostic tests when booking appointments.

3.6 Conceptual Model

3.6.1 Use Case Diagram



Description

This use case diagram represents the interaction between three key actors—**Patient**, **Doctor**, and **Admin**—within the **Mcare system**. Each actor has specific roles and interactions with the system:

1. **Patient**: The patient can register or log in, search for a doctor, request and book appointments, and view their medical records. They can also provide feedback on doctors and use video call options for telemedicine consultations.

2. **Doctor:** The doctor manages appointments, prescribes treatments, and sends medication reminders. They can also consult patients via video calls or visits.
3. **Admin:** The admin handles verification of both doctors and patients, manages medical records, handles complaints, and can accept or reject new doctor requests.

Overall, the system promotes efficient communication, seamless appointment management, and secure handling of medical data, improving the overall healthcare experience for all users.

Chapter 4: System Design

4.1 Basic Module

The **Mcare Project** is structured into key functional modules that streamline user registration, appointment scheduling, medical record management, and telemedicine support. These modules ensure an efficient, scalable, and user-friendly platform, optimizing specific tasks to enhance the overall healthcare experience for patients, doctors, and administrators.

Here are some Module:

1. User Management Module:

Features:

- Registration and login for Patients, Doctors, and Admins.
- Role-based access control with permissions for each user type.
- Profile management allowing users to view, edit, and update personal details.

2. Appointment Management Module:

Features:

- Patients can search for doctors by specialization, location, and availability.
- Appointment booking, rescheduling, and cancellation for patients.
- Doctors can manage, approve, or reject appointments.
- Notification system for appointment reminders via email/SMS.

3. Medical Records Management Module:

Features:

- Doctors can update patient medical records and treatment details.
- Patients can view their medical history, diagnoses, and prescriptions.
- Secure storage of medical records in a centralized database

4. Telemedicine Module:

Features:

- Video consultations between patients and doctors.
- Messaging system for follow-up consultations.
- Telemedicine appointment scheduling and management.

5. Admin Management Module:

Features:

- Manage patient and doctor accounts (add, update, remove users).
- Monitor system performance and user activity.
- Review and resolve complaints from patients or doctors.
- Generate system reports and logs for audits.

6. Notification & Reminder Module:

Features:

- Automatic reminders for upcoming appointments.
- Medication reminders for patients.
- Notification delivery via email or SMS for both patients and doctors

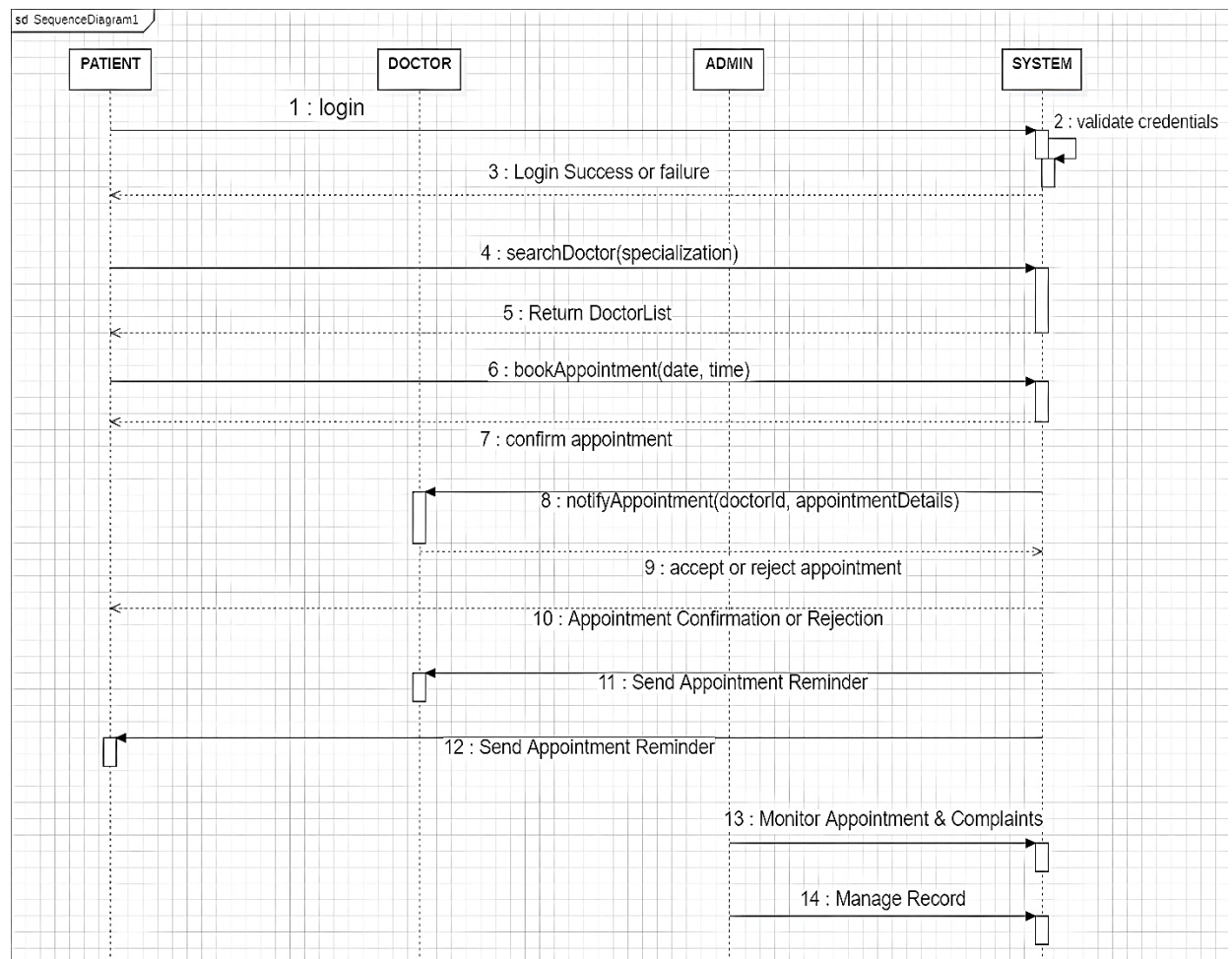
7. Payment and Billing Module (Optional):

Features:

- Patients can make payments for consultations via a secure gateway eg..GPay, Paytm,Paypal etc
- Billing and payment tracking for telemedicine or in-person consultations.

4.2 Procedural Design

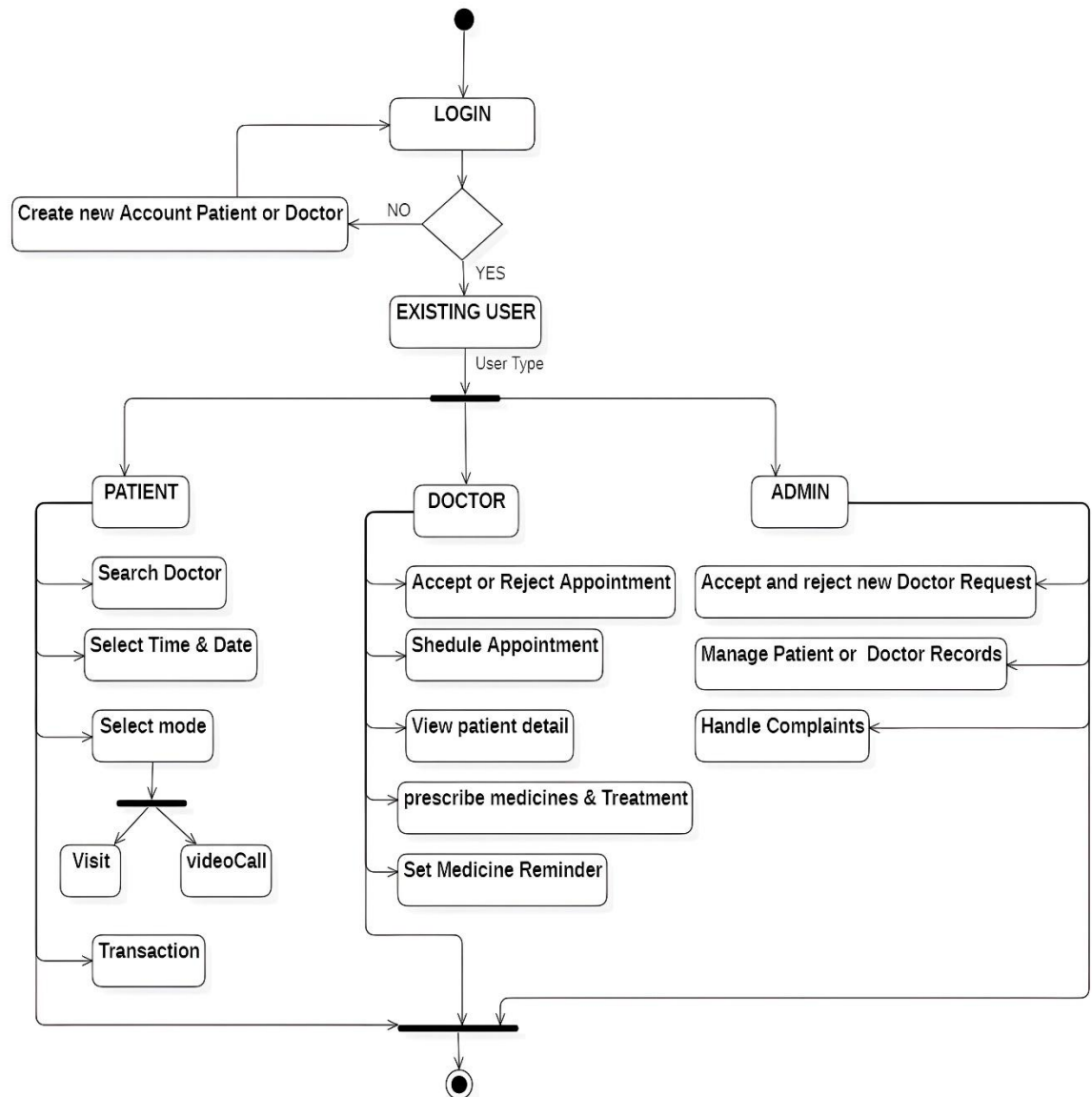
4.2.1 Sequence Diagram



Description

The sequence diagram outlines the Mcare system's process for managing patient-doctor appointments. It starts with the patient logging in, after which they search for doctors and book an appointment by selecting a time and date. The system confirms the booking and notifies both the doctor and admin. The doctor can then accept or reject the appointment, and the patient is informed accordingly. Automated reminders are sent to both parties before the appointment, while the admin monitors the process and manages complaints. The system also handles the secure management of patient records.

4.2.2 Activity Diagram



Description

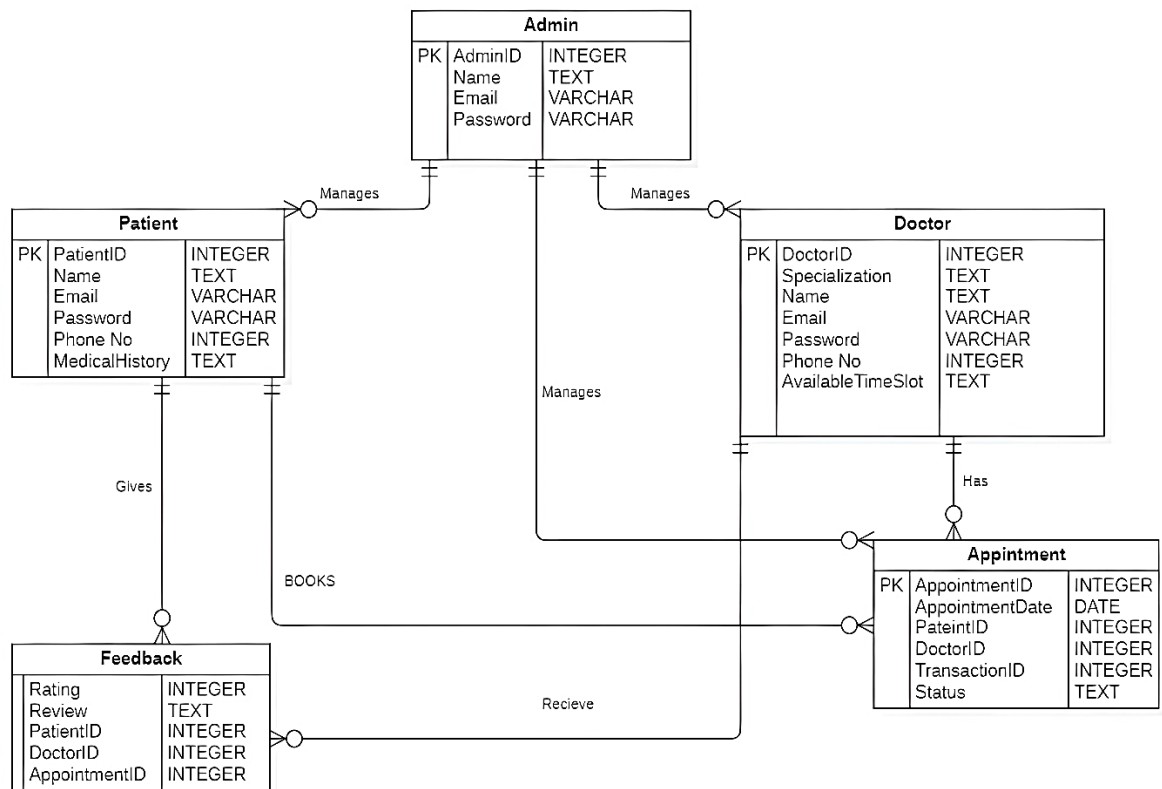
It starts with a Login node, followed by a decision point for "Create new Account Patient or Doctor." If a user is existing, the system checks their type: Patient, Doctor, or Admin.

Patient Flow: The patient can Search Doctor, Select Time & Date, and choose their preferred appointment mode Select mode either visit or video Call. The flow then leads to a Transaction node, implying payment or confirmation.

Doctor Flow: Doctors can Accept or Reject Appointment, Schedule Appointment, View patient detail, prescribe medicines & Treatment, and Set Medicine Reminder.

Admin Flow: Admin users can Accept and reject new Doctor Request, Manage Patient or Doctor Records, and Handle Complaints.

4.2.3 ER Diagram



Description

This is an Entity-Relationship Diagram (ERD) for a doctor-patient appointment management system. Here's a brief explanation of the entities and their relationships:

1. **Admin:** Manages both doctors and patients. Contains attributes like AdminID, Name, Email, and Password.
2. **Doctor:** Manages appointments and has a relationship with the **Appointment** entity. Contains attributes like DoctorID, Specialization, Name, Email, Password, Phone Number, and Available Time Slot.


3. **Patient:** Can book appointments and give feedback. Contains attributes like PatientID, Name, Email, Password, Phone Number, and Medical History.
4. **Appointment:** Contains information about appointments, with relationships to both patients and doctors. Attributes include AppointmentID, AppointmentDate, PatientID, DoctorID, TransactionID, and Status.
5. **Feedback:** Patients provide feedback on appointments, linking the PatientID, DoctorID, and AppointmentID, along with rating and review.

The relationships between the entities are clear, such as:


- Admin manages doctors and patients.
- Patients book appointments with doctors.
- Patients give feedback after appointments, which doctors receive.


4.3 User Interface Design


4.3.1 Login Page Design



LOGIN


 USERNAME


 PASSWORD


 USER TYPE ▼ DOCTOR

LOGIN NOW

OR LOGIN WITH

 LOGIN WITH GOOGLE





 LOGIN WITH FACEBOOK




McCare


Empowering Health, Anytime, Anywhere

- ▶ Connecting Patients and Doctors with Ease.
- ▶ Your Health, Our Priority
- ▶ 24/7 Access to Doctors
- ▶ Easy Appointment Scheduling




CONTACT US


 +1012 3456 789

 demo@gmail.com

4.3.2 Sign In Design

SIGN IN


 EMAIL ID


 PASSWORD


SIGN UP

ALREADY HAVE ACCOUNT [LOGIN](#)

OR SIGN IN WITH

 LOGIN WITH GOOGLE





 LOGIN WITH FACEBOOK




McCare


Empowering Health, Anytime, Anywhere

- ▶ Connecting Patients and Doctors with Ease.
- ▶ Your Health, Our Priority
- ▶ 24/7 Access to Doctors
- ▶ Easy Appointment Scheduling

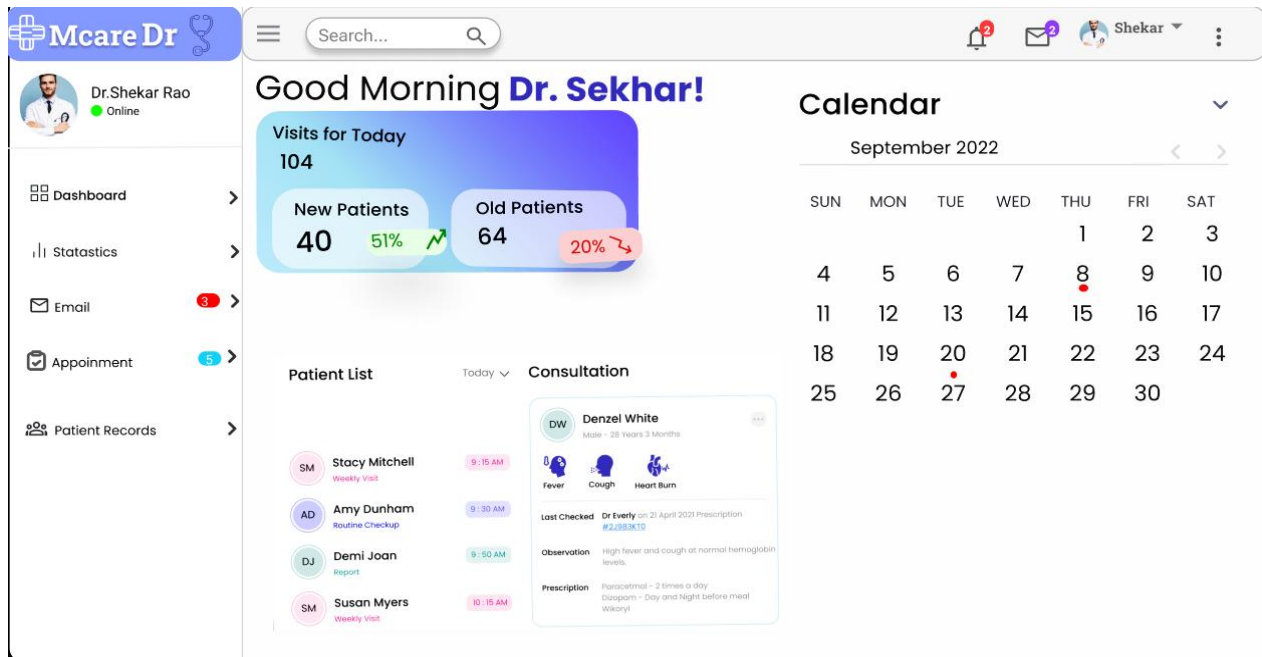


CONTACT US

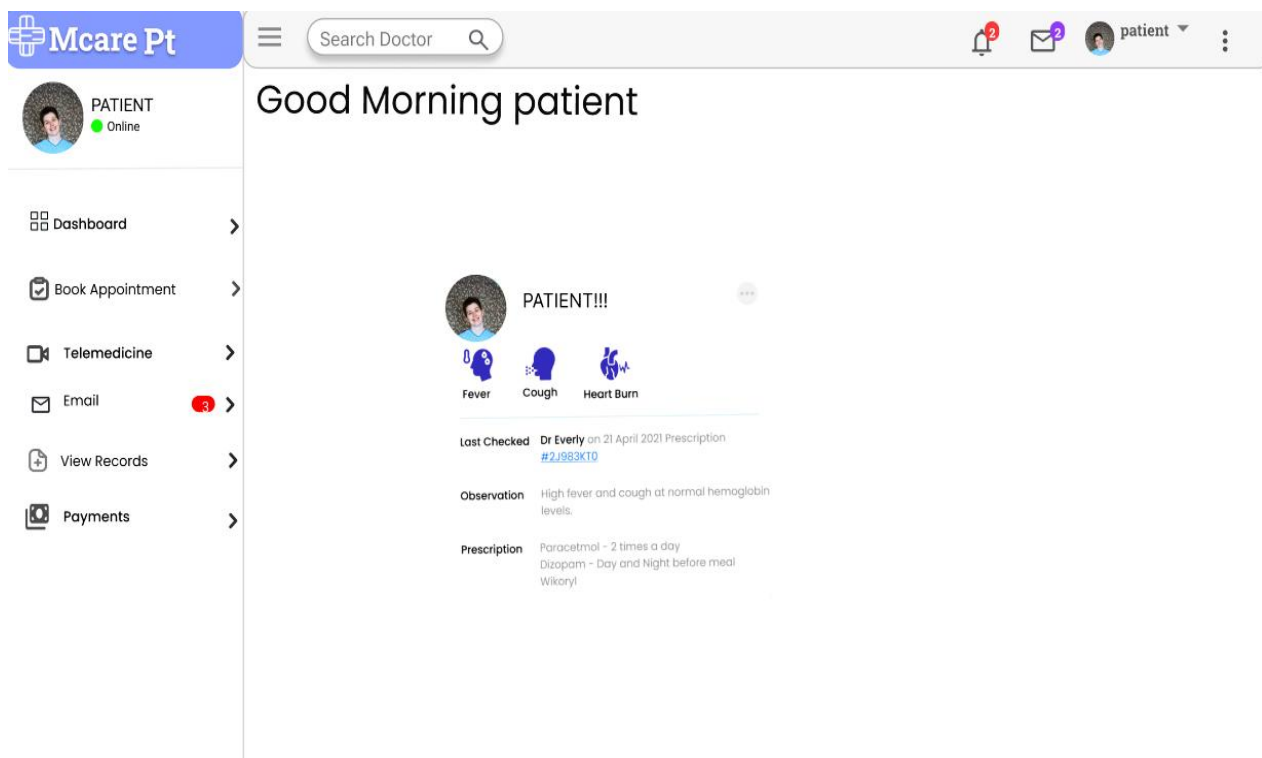
 +1012 3456 789

 demo@gmail.com

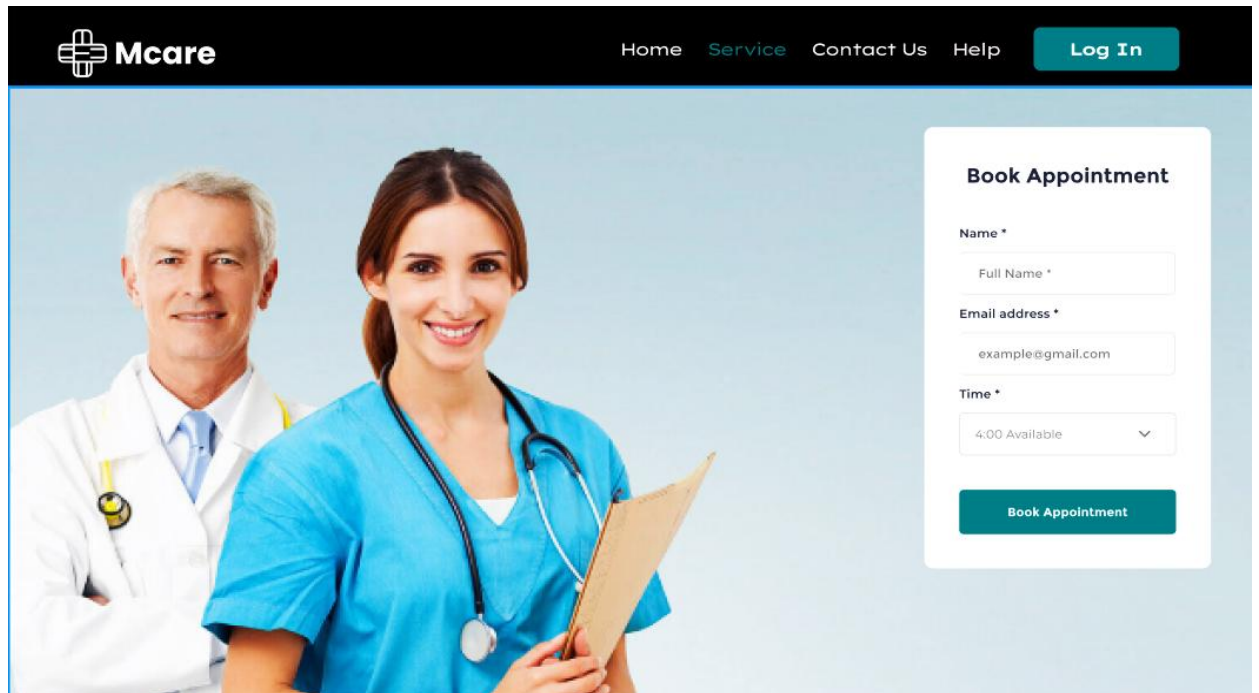
4.3.3 Doctor Dashboard Design



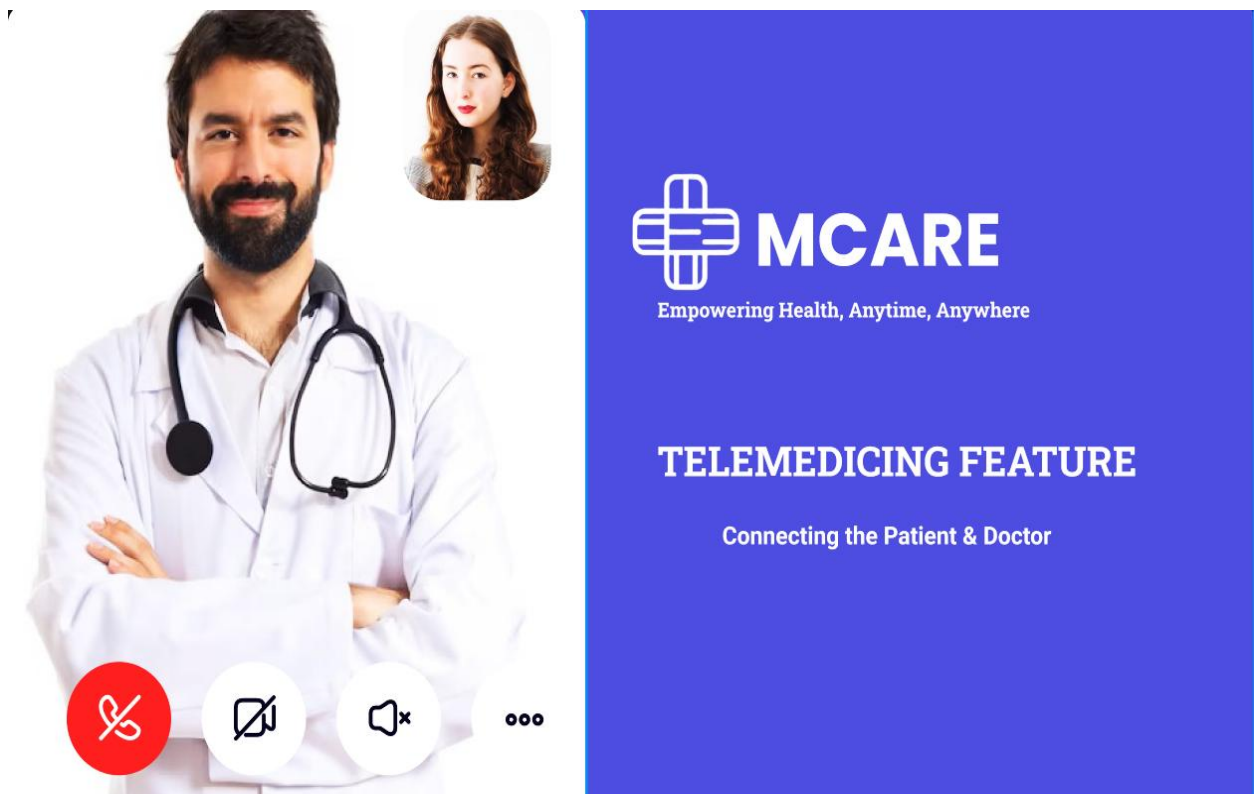
4.3.4 Patient Dashboard



4.3.5 Home Page Design



4.3.6 Telemedicine Feature Design



Chapter 5: Implementation

5.1 Coding

5.1.1 Appointment Requests Component (Frontend)

```
export function AppointmentRequests() {

  const [requests, setRequests] = useState<AppointmentRequest[]>([])

  const [isLoading, setIsLoading] = useState(true)

  const { toast } = useToast()

  const user = JSON.parse(localStorage.getItem("user") || "{}")

  const socket = useSocket(user.id, "doctor")

  const fetchRequests = useCallback(async () => {

    try {

      const token = localStorage.getItem("token")

      const response = await fetch(
`http://localhost:4000/api/appointments?userId=${user.id}&userType=doctor&status
=pending`,

      {

        headers: {

          Authorization: `Bearer ${token}`,

        },

      },

    )

    if (!response.ok) {

      throw new Error("Failed to fetch appointment requests")

    }

  })

}
```

```

}

const data = await response.json()

if (data.success) {

  // Process the appointments to ensure dates are properly formatted

  const formattedAppointments = data.appointments.map((appointment) => {

    // Format date properly

    let formattedDate = "Date not available"

    let formattedTime = "Time not available"

    try {

      if (appointment.date) {

        const dateObj = new Date(appointment.date)

        if (!isNaN(dateObj.getTime())) {

          formattedDate = dateObj.toLocaleDateString()

        }

      }

      if (appointment.timeSlot && appointment.timeSlot.startTime) {

        formattedTime = appointment.timeSlot.startTime

      }

    } catch (error) {

      console.error("Error formatting date/time:", error)

    }

    return {

      id: appointment._id || appointment.id,

      patientName: appointment.patientName || "Unknown Patient",

```

```

        requestedDate: formattedDate,

        requestedTime: formattedTime,

        note: appointment.symptoms || "",

        type: appointment.consultationType || "Regular",

    }

}))

setRequests(formattedAppointments)

console.log("Formatted appointments:", formattedAppointments)

}

} catch (error) {

    console.error("Error fetching requests:", error)

    toast({

        title: "Error",

        description: "Failed to load appointment requests",

        variant: "destructive",

    })

} finally {

    setIsLoading(false)

}

}, [user.id, toast])

// Rest of the component...

}

```


Chapter 6: Testing and Result

6.1 Test Cases

- User Login Test Cases

Test Case No	Module	Input	Expected Output	Actual Result	Pass/fail
TC001	Login	Valid email and password for a patient	User is authenticated and redirected to patient dashboard	Same as Expected Result	Pass
TC002	Login	Valid email and password for a doctor	User is authenticated and redirected to doctor dashboard	Same as Expected Result	Pass
TC003	Login	Valid email and password for an admin	User is authenticated and redirected to admin dashboard	Same as Expected Result	Pass
TC004	Login	Valid email but incorrect password	Error message: "Invalid credentials"	Same as Expected Result	Pass

TC005	Login	Email not registered in the system	Error message: "User not found"	Same as Expected Result	Pass
TC006	Login	Empty email field and Empty password field	Form validation error for email field and Password field	Same as Expected Result	Pass
TC007	Login	Invalid email format (missing @ symbol)	Form validation error for email format	Same as Expected Result	Pass

- **User Sign Up Test Cases**

Test Case No	Module	Input	Expected Output	Actual Result	Pass/fail
TC001	Sign up	Valid new email, password, name, and selecting "Patient" role	Account created successfully, verification email sent, redirect to login page	Same as Expected Result	Pass
TC002	Sign up	Valid new email, password, name, and selecting "Doctor" role	Account created successfully, verification email sent, redirect to login page	Same as Expected Result	Pass

TC003	Sign up	Email already registered in the system	Error message: "Email already in use"	Same as Expected Result	Pass
TC004	Sign up	Password without required complexity (missing uppercase/number/special char)	Form validation error for password complexity	Same as Expected Result	Pass
TC005	Sign up	Invalid email format	Form validation error for email format	Same as Expected Result	Pass
TC006	Sign Up	Empty Fields	Form Validation Error for Empty field	Same as Expected Result	Pass

- **Forgot Password Test Cases**

Test Case No	Module	Input	Expected Output	Actual Result	Pass/fail
TC001	Forgot Password	Registered email address	Success message: "Password reset link sent to email"	Same as Expected Result	Pass
TC002	Forgot Password	Email not registered in the system	Error message: "Email not found"	Same as Expected Result	Pass
TC003	Forgot Password	Empty email field	Form validation error for email field	Same as Expected Result	Pass
TC004	Forgot Password	Invalid email format	Form validation error for email format	Same as Expected Result	Pass
TC005	Forgot Password	Expired reset token	Error message about expired token, option to request new link	Same as Expected Result	Pass

TC006	Forgot Password	New password same as old password	Error or warning message about using a different password	Same as Expected Result	Pass
-------	-----------------	-----------------------------------	---	-------------------------	------

- **Appointment Module Test Cases**

Test Case No	Module	Input	Expected Output	Actual Result	Pass/fail
TC001	Doctor Dashboard Appointments Module	Doctor login and navigates to appointments page	All pending and upcoming appointments are displayed correctly	Same as Expected Result	Pass
TC002	Doctor Dashboard Appointments Module	Doctor clicks "Accept" on a pending appointment	Appointment status changes to "Confirmed", notification sent to patient	Same as Expected Result	Pass
TC003	Doctor Dashboard Appointments Module	Doctor clicks "Reject" on a pending appointment	Appointment status changes to "Rejected", notification sent to patient	Same as Expected Result	Pass

TC004	Doctor Dashboard Appointments Module	Doctor clicks "Complete" on an in-progress appointment	Appointment status changes to "Completed", patient record updated	Same as Expected Result	Pass
TC005	Doctor Dashboard Appointments Module	Doctor adds notes to an appointment	Notes are saved and visible on appointment details	Same as Expected Result	Pass
TC006	Doctor Dashboard Appointments Module	Doctor reschedules an appointment	Appointment time is updated, notification sent to patient	Same as Expected Result	Pass
TC007	Doctor Dashboard Appointments Module	Doctor views appointment history	Past appointments are displayed with correct status and details	Same as Expected Result	Pass
TC008	Doctor Dashboard Appointments Module	Doctor has no appointments	Empty state is displayed with appropriate message	Same as Expected Result	Pass
TC009	Patient Appointment Booking	Patient selects a doctor from the list	Doctor's profile and available time slots are displayed	Same as Expected Result	Pass

TC010	Patient Appointment Booking	Patient selects an available time slot	Time slot is temporarily reserved during booking process	Same as Expected Result	Pass
TC011	Patient Appointment Booking	Patient completes booking form with valid details	Appointment is created with "Pending" status, notification sent to doctor	Same as Expected Result	Pass
TC012	Patient Appointment Booking	Patient attempts to book an already booked time slot	Error message indicating time slot is no longer available	Same as Expected Result	Pass
TC013	Patient Appointment Booking	Patient cancels an upcoming appointment	Appointment status changes to "Cancelled", slot becomes available again	Same as Expected Result	Pass
TC014	Patient Appointment Booking	Patient reschedules an appointment	Old slot is freed, new slot is booked, notification sent to doctor	Same as Expected Result	Pass

TC015	Patient Appointment Booking	Patient views appointment history	All past and upcoming appointments are displayed with correct status	Same as Expected Result	Pass
-------	-----------------------------	-----------------------------------	--	-------------------------	------

- Doctor Availability Management**

Test Case No	Module	Input	Expected Output	Actual Result	Pass/Fail
TC001	Availability Management	Doctor sets regular weekly schedule	Schedule is saved and reflected in booking system	Same as Expected Result	Pass
TC002	Availability Management	Doctor blocks specific dates (vacation/leave)	Dates are marked as unavailable in booking system	Same as Expected Result	Pass
TC003	Availability Management	Doctor modifies existing availability	Changes are saved and reflected in booking system	Same as Expected Result	Pass

TC004	Availability Management	Doctor sets availability with invalid time range (end before start)	Validation error shown for invalid time range	Same as Expected Result	Pass
TC005	Availability Management	Doctor views calendar with existing appointments	Calendar shows booked and available slots correctly	Same as Expected Result	Pass

- **Online Prescription & Medicine Reminder Module**

Test Case No	Module	Input	Expected Output	Actual Result	Pass/Fail
TC001	Online Prescription	Doctor creates a new prescription with valid medication details, dosage, and duration	Prescription is successfully created and saved to the database	Same as Expected Result	
TC002	Online Prescription	Doctor creates a prescription with missing medication name	System shows validation error for required field	Same as Expected Result	

TC003	Online Prescription	Doctor adds special instructions to prescription	Special instructions are saved and displayed correctly	Same as Expected Result	
TC004	Online Prescription	Doctor creates a prescription with invalid dosage format	System shows validation error for dosage format	Same as Expected Result	
TC005	Online Prescription	Doctor creates a prescription with past date Doctor creates a prescription with past date	System shows validation error for invalid date	Same as Expected Result	
TC006	Online Prescription	Doctor sends prescription to patient	Patient receives notification about new prescription	Same as Expected Result	
TC007	Online Prescription	Doctor updates an existing prescription	Changes are saved and patient is notified of updates	Same as Expected Result	

TC008	Online Prescription	Patient views their prescription list	All prescriptions are displayed in chronological order	Same as Expected Result	
TC009	Online Prescription	Patient downloads prescription as PDF	PDF is generated with correct formatting and all prescription details	Same as Expected Result	
TC010	Medicine Reminder	Doctor sets up medication reminder with valid medication name, dosage, frequency, and duration	Reminder is successfully created and scheduled	Same as Expected Result	
TC011	Medicine Reminder	Doctor sets reminder with missing medication name	System shows validation error for required field	Same as Expected Result	

TC012	Medicine Reminder	Doctor sets multiple reminders for different medications	All reminders are correctly scheduled	Same as Expected Result	
TC013	Medicine Reminder	Patient receives reminder notification at scheduled time	Notification is delivered with correct medication details	Same as Expected Result	
TC014	Medicine Reminder	Patient marks reminder as "taken"	Reminder status is updated and logged	Same as Expected Result	
TC015	Medicine Reminder	Patient snoozes reminder	Reminder is rescheduled for the specified snooze duration	Same as Expected Result	
TC016	Medicine Reminder	Patient misses reminder (no action taken)	System logs missed reminder and sends follow- up notification	Same as Expected Result	

TC017	Medicine Reminder	Patient views all active medication reminders	All active reminders are displayed with correct details	Same as Expected Result	
TC018	Medicine Reminder	Doctor views patient's medication adherence report	Report shows accurate statistics of taken/missed medications	Same as Expected Result	
TC019	Medicine Reminder	System handles reminder when patient is offline	Reminder is delivered when patient comes online	Same as Expected Result	
TC020	Medicine Reminder	Multiple reminders scheduled for same time	All reminders are delivered correctly without conflicts	Same as Expected Result	

- **Notification System**

Test Case No	Module	Input	Expected Result	Actual Result	Pass/ fail
TC001	Notifications	New appointment is booked	Doctor receives notification about pending appointment	Same as Expected Result	

TC002	Notifications	Doctor accepts appointment	Patient receives notification about confirmed appointment	Same as Expected Result	
TC003	Notifications	Doctor rejects appointment	Patient receives notification about rejected appointment	Same as Expected Result	
TC004	Notifications	Appointment reminder	Both doctor and patient receive reminder notifications	Same as Expected Result	
TC005	Notifications	Doctor reschedules appointment	Patient receives notification about rescheduled appointment	Same as Expected Result	
TC006	Notifications	Patient cancels appointment	Doctor receives notification about cancelled appointment	Same as Expected Result	
TC007	Notifications	Doctor uploads new document/prescription	Patient receives notification about new document	Same as Expected Result	

6.2 Test Reports

Test Reports	
Project Name	Mcare (Integrated Patient & Doctor Healthcare Management System)
Test Type	BlackBox Testing
Pass	61
Fail	5
Total	71

Chapter 7: Conclusion

7.1 Conclusion

The Mcare project represents a significant advancement in healthcare management technology, successfully addressing critical inefficiencies in traditional healthcare systems. Through the development of this integrated patient and doctor healthcare management system, we have created a comprehensive solution that bridges the gap between healthcare providers and patients, enhancing the overall quality of healthcare delivery.

Impact and Benefits

The Mcare system delivers substantial benefits to all stakeholders in the healthcare ecosystem:

For Patients: Enhanced accessibility to healthcare services, reduced wait times, improved medication adherence, and greater engagement in their own healthcare journey.

For Healthcare Providers: Streamlined workflows, reduced administrative burden, improved patient communication, and more efficient practice management.

For Healthcare System: Better resource utilization, reduced paperwork, enhanced data security, and improved coordination between different healthcare entities.

7.2 Future Works / Improvements

While the current MCare system efficiently streamlines healthcare management, several advanced features are planned for future implementation to further enhance accessibility, security, and efficiency in patient care:

- **Telemedicine Integration:** Implementing a secure telemedicine module to enable remote consultations between doctors and patients, ensuring accessibility to healthcare regardless of location.
- **AI-Assisted Diagnosis & Treatment Recommendations:** Utilizing machine learning algorithms to analyze patient history and symptoms, providing doctors with AI-driven suggestions for potential diagnoses and treatments.
- **Multi-Language & Multi-Currency Support:** Expanding the platform to support multiple languages for improved accessibility and enabling transactions in various currencies for seamless global usability.
- **Online Medicine Purchase & Delivery:** Implementing an integrated e-pharmacy system that enables patients to order prescribed medications directly through the platform, ensuring doorstep delivery and medication availability.

7.3 References

YouTube Video Playlist

- YouTube Playlist:

https://www.youtube.com/playlist?list=PL4cUxeGkcC9iJ_KkrkBZWZRHVwnzLIoUE/

<https://youtu.be/FT234TaUyRQ?si=1jpit2yVKaUIMXHy/>

https://youtu.be/oY24fxdTKi8?si=1RtqYPBV50_dsLdl/

GitHub Sample Project

<https://github.com/HashenUdara/edoc-doctor-appointment-system/>

Official Documentation and Websites

1. React.js Official Documentation

<https://reactjs.org/docs/getting-started.html/>

2. Node.js Official Documentation

<https://nodejs.org/en/docs/>

3. Express.js Official Documentation

<https://expressjs.com/>

Healthcare Standards and Regulations

HIPAA Compliance Resources

<https://www.hhs.gov/hipaa/for-professionals/index.html/>

7.4 Bibliography

YouTube Video Playlist

Traversy Media. (n.d.). *MERN Stack Course – MongoDB, Express, React, Node.js Full Tutorial*. [YouTube Playlist](#). Retrieved from YouTube.

- This video series provides an in-depth guide to building full-stack applications using the MERN (MongoDB, Express, React, Node.js) stack. It covers various aspects of frontend and backend development, including authentication, database interactions, and state management.

Traversy Media. (2023, July 15). *MERN Stack Authentication Tutorial – JWT, MongoDB, Express, React, Node.js*. [Video](#). Retrieved from YouTube.

- This tutorial explains user authentication in a MERN stack application using JSON Web Tokens (JWT) and MongoDB. It provides insights into handling user sessions securely and efficiently.

Academind. (2023, August 5). *Building a Full-Stack Doctor Appointment System with React and Node.js*. [Video](#). Retrieved from YouTube.

- This tutorial demonstrates how to build a doctor appointment system using React and Node.js. It includes features such as user authentication, appointment scheduling, and database management.

GitHub Sample Project

Udara, H. (n.d.). *E-Doc Doctor Appointment System*. GitHub Repository. Retrieved from <https://github.com/HashenUdara/edoc-doctor-appointment-system>.

- This open-source project serves as a reference for developing an online doctor appointment booking system using the MERN stack. It includes functionalities such as doctor registration, patient appointment scheduling, and prescription management.

Official Documentation and Websites

React.js Official Documentation. (n.d.). *React – A JavaScript Library for Building User Interfaces*. Retrieved from <https://reactjs.org/docs/getting-started.html>.

- The official React documentation provides guidelines on setting up and using React for building interactive user interfaces. It includes concepts such as components, state management, and lifecycle methods.

Node.js Official Documentation. (n.d.). *Node.js Documentation*. Retrieved from <https://nodejs.org/en/docs/>.

- This resource provides official documentation for Node.js, covering topics such as event-driven programming, modules, and backend development using JavaScript.

Express.js Official Documentation. (n.d.). *Express – Fast, Unopinionated, Minimalist Web Framework for Node.js*. Retrieved from <https://expressjs.com/>.

- Express.js is a minimal and flexible Node.js web application framework. The documentation covers routing, middleware, and API handling.

Healthcare Standards and Regulations

U.S. Department of Health and Human Services. (n.d.). *HIPAA Compliance Resources – Health Insurance Portability and Accountability Act (HIPAA)*. Retrieved from <https://www.hhs.gov/hipaa/for-professionals/index.html>.

- This resource provides comprehensive information about HIPAA regulations, which govern the protection of patient health information in healthcare applications. It outlines compliance requirements for data security, privacy, and breach notifications.

7.5 Plagiarism Report



