

MEDPREP

INTERDISCIPLINARY PROJECT

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology in Information Technology

by

ABHILASH B (41120002)



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY-1 UNIVERSITY BY UGC

Accredited "A++" by NAAC

12B Status by UGC |Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

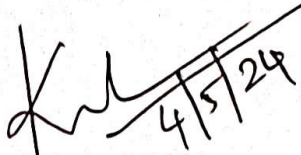
Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of
ABHILASH B (41120002) who carried out the project titled
"MEDPREP (PERN)" under my supervision from **January 2024** to
May 2024.


Internal Guide

Dr. KAMATCHI K S, M.E., Ph.D.,



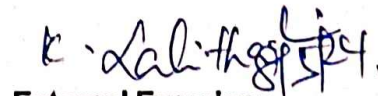
Head of the Department

Dr. R. SUBHASHINI, M.E., Ph.D.

Submitted for viva voce Examination held on 08/05/2024.



Internal Examiner


External Examiner

DECLARATION

I, ABHILASH B (41120002) hereby declare that the Project Report entitled "MEDPREP (PERN)" done by me under the guidance of Dr. Kamatchi K S, M.E.,Ph.d., Department of Information Technology, is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 08/05/2024

PLACE: Chennai

Abhilash.B

SIGNATURE OF CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean, School of Computing, Dr.R. Subhashini, M.E.,Ph.D. Head of the Department of Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. Kamatchi K S M.E.,Ph.D.**, for her valuable guidance, suggestions and constant encouragement that paved the way for the successful completion of my project work.

I wish to express my thanks to all teaching and non-teaching staff members of the **Department of Information Technology** who were helpful in many ways throughout the completion of the project.

OFFER LETTER



+91-9551125126



versalgo@versalgo.in



D-102, 10th floor, Lavender Block, Embassy Residency, Cheran Nagar, Perumbakkam, Chennai, Tamil Nadu, India

19 December 2023

Dear Mr. Abhilash B

Subject: Internship Appointment Letter with Probation Period Details

We are thrilled to offer you the Software Developer Intern position at VersalGo Pvt. Ltd. based on your exceptional qualifications, enthusiasm, and passion for software development demonstrated during the probationary period.

Your internship is scheduled to commence on 20th December 2023, and the probationary period will span the first four months of your internship. This probationary period serves as an evaluation phase to assess your performance and suitability for a continued internship with VersalGo Pvt. Ltd.

During the probationary period, your main responsibilities will include:

- Assisting in the development, testing, and debugging of software applications.
- Collaborating with team members to contribute to project requirements and specifications.
- Participating in code reviews and providing constructive feedback.
- Conducting research and staying updated on the latest trends and technologies in software development.
- Assisting in the documentation and maintenance of software projects.

At the end of the probationary period, your performance will be thoroughly evaluated, and, subject to a successful assessment, your internship will continue as planned. You will be expected to work 18 hours per week, and your specific working hours and schedule will be communicated to you by your supervisor.

As an intern at VersalGo Pvt. Ltd., adherence to company policies, including confidentiality, code of conduct, and professional ethics, is mandatory. We expect you to maintain a high level of professionalism and actively contribute to the success of the team and the organization.

To confirm your acceptance of this internship position, please sign and return a copy of this appointment letter and the Non-Disclosure Agreement no later than 2 days from the date of this letter. If you have any questions or require further information, please do not hesitate to contact Mr. Kishore Kumar K at cmo@versalgo.in or 98849 48682.

Congratulations once again on your appointment as a Software Developer Intern at VersalGo Pvt. Ltd. We believe this internship will provide you with an excellent learning experience, and we look forward to your contributions to our organization.

Warm Regards,


Alina Mushtaq
CEO of VersalGo Pvt. Ltd.



ABSTRACT

MedPrep Hospital Management System (HMS) is a comprehensive software solution revolutionizing healthcare operations through its utilization of the PERN stack (PostgreSQL, Express.js, React.js, and Node.js). Tailored for patients, doctors, nurses, and administrators, MedPrep boasts intuitive features like patient management, appointment scheduling, and electronic health records (EHR) management. With React.js, the system ensures a seamless user experience across web and mobile platforms, promoting accessibility and user-friendliness. Its standout feature is multilingual capability, facilitating interaction in users' preferred languages and enhancing communication and engagement for a patient-centric approach. Patients can effortlessly register, schedule appointments, and access medical records via the MedPrep website, while healthcare professionals benefit from real-time updates and secure communication channels. Administrators leverage comprehensive analytics and reporting tools to drive data-driven decisions and optimize hospital operations. Prioritizing data security and regulatory compliance, MedPrep utilizes PostgreSQL for secure data storage and encryption protocols to safeguard sensitive patient information. Built on Node.js, the system ensures scalability and seamless integration with existing hospital infrastructure, laying a solid foundation for efficient healthcare management despite lacking billing and inventory management functionalities.

TABLE OF CONTENT

Chapter No.	TITLE	Page No.
	ABSTRACT	VI
	LIST OF ABBREVIATIONS	IX
	LIST OF FIGURES	X
1	INTRODUCTION	1
	1.1 React.js	2
	1.2 PostgreSQL	3
2	METHODOLOGY	4
	2.1 Architecture	4
	2.2 User interface	6
	2.2.1 Integration with Material UI	7
	2.2.2 Integration with backend	8
	2.3 USER ACCESSIBILITY	9
	2.3.1 User registration	9
	2.3.2 User dashboard	10
	2.3.3 Appointment management	12
	2.3.4 Access control	13
	2.3.5 Reports and prescription management	14
	2.4 Multi language support	16
	2.5 Administrative functions	17
	2.5.1 Adding doctor and nurse	18
	2.5.2 Managing doctors and nurses	18

	2.6 Backend development	19
	2.6.1 Building API endpoints	20
	2.6.2 Designing database	21
	2.6.3 User Authentication	22
	2.7 Deployment and Hosting	23
3	RESULTS AND DISCUSSION	24
4	CONCLUSION AND FUTURE WORKS	28
	4.1 Conclusion	28
	4.2 Future Works	29
	REFERENCES	31
	APPENDICES	32
	A Sample Code	32
	B Screen Shots	37

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
PERN	- PostgreSQL, Express.js, React.js, Node.js
API	- Application Programming Interface
HTML	- Hypertext Markup Language
CSS	- Cascading Style Sheets
UI	- User Interface
UX	- User Experience
GUI	- Graphical User Interface
SQL	- Structured Query Language
JSON	- JavaScript Object Notation
CRUD	- Create, Read, Update and Delete
HTTPS	- Hypertext Transfer Protocol Secure
SSL	- Secure Sockets Layer
URL	- Uniform Resource Locator
IDE	-Integrated Development Environment
DNS	- Domain Name System
SSH	- Secure Shell
JWT	- JSON Web Token
CI/CD	-Continuous Integration/Continuous Deployment

LIST OF FIGURES

Figure No.	Name of the Figure	Page No.
2.1	Architecture Diagram	5
2.2	User Registration	10
2.3	User Dashboard	12
2.4	Appointment Management	13
2.5	Prescription and Reports	15
2.6	Multi Language	17
3.1	Multi User Login Page	24
3.2	Patient Signup	24
3.3	Home Page	25
3.4	Book Appointment	26
3.5	Prescription and Reports	26
3.6	Multi-Lingual Support	27

CHAPTER 1

INTRODUCTION

In the rapidly evolving landscape of healthcare management, the integration of advanced technological solutions has become imperative to streamline operations, enhance patient care, and improve overall efficiency within hospital settings. Recognizing this need, the development of MedPrep Hospital Management System (HMS) emerges as a pivotal step towards modernizing healthcare administration. MedPrep is a comprehensive software solution meticulously designed and implemented using the PERN stack, aimed at revolutionizing the way hospitals manage their operations and interact with patients, doctors, nurses, and administrators.

The complexity of managing a hospital encompasses numerous facets, ranging from patient registration and appointment scheduling to electronic health records (EHR) management and secure communication channels between healthcare providers. Historically, these processes have been fragmented, leading to inefficiencies, communication gaps, and compromised patient care. MedPrep addresses these challenges by offering an integrated platform that centralizes hospital operations, fosters collaboration among stakeholders, and promotes a patient-centric approach to healthcare delivery.

At the heart of MedPrep lies its commitment to accessibility and inclusivity. Recognizing the diverse linguistic backgrounds of patients and healthcare professionals, MedPrep incorporates a multilingual feature, enabling users to interact with the system in their preferred language. This feature not only enhances communication but also empowers users by allowing them to navigate the system with ease and confidence.

1.1 REACT.JS

React.js, often simply referred to as React, is an open-source JavaScript library primarily used for building user interfaces (UIs) for web applications. It was developed by Facebook and maintained by both Facebook and a community of individual developers and companies. React was first released in 2013, and since then, it has gained immense popularity within the web development community due to its efficiency, flexibility, and performance. At its core, React follows a component-based architecture, which means that UIs are built using reusable and modular components. These components encapsulate both the structure and behavior of a part of the user interface, making it easier to manage and maintain complex applications. React components can be composed together to form larger, more complex UIs.

One of the key features of React is its use of a virtual DOM (Document Object Model), which is a lightweight representation of the actual DOM. React efficiently updates the virtual DOM based on changes in the application's state, and then selectively updates the real DOM, minimizing the number of DOM manipulations and resulting in better performance. React also promotes a declarative programming style, where developers describe the desired UI state, and React takes care of updating the DOM to match that state. This approach makes code more predictable and easier to debug. React is often used in conjunction with other libraries and frameworks, such as Redux for managing application state, React Router for handling routing within single-page applications, and Axios for making HTTP requests. Additionally, React has a thriving ecosystem with a wide range of tools, libraries, and community resources to aid developers in building robust and efficient web applications.

1.2 POSTGRESQL

PostgreSQL, often referred to as Postgres, stands as a stalwart within the realm of relational database management systems (RDBMS), renowned for its robustness, reliability, and advanced feature set. Originating from the University of California, Berkeley, in the 1980s, PostgreSQL has since matured into a formidable database system favored by developers and organizations globally. Its adherence to the ACID (Atomicity, Consistency, Isolation, Durability) principles ensures data integrity and reliability, even amidst system failures, fostering trust and confidence among users. PostgreSQL's extensibility is a hallmark feature, offering a rich array of data types, support for user-defined functions and extensions, and scalability options like sharding and replication. Moreover, PostgreSQL boasts comprehensive SQL compliance, empowering developers with advanced query capabilities and procedural languages for stored procedures and triggers. With native support for JSON data and robust full-text search capabilities, PostgreSQL caters to applications with flexible schema requirements and diverse data structures. Its concurrency control mechanisms, including multiversion concurrency control (MVCC), facilitate concurrent access to data while maintaining consistency and isolation. Security is paramount, with features such as authentication methods, SSL/TLS encryption, and role-based access control (RBAC), ensuring stringent data protection. PostgreSQL's support for replication methods and high availability features, coupled with its vibrant community and rich ecosystem, solidify its position as a premier choice for applications spanning small-scale projects to enterprise-level systems, delivering unparalleled flexibility, performance, and reliability.

CHAPTER 2

METHODOLOGY

In this chapter, we provide an overarching view of our web application project, emphasizing its architectural components, user authentication mechanisms, and key features. This comprehensive overview lays the foundation for a deeper exploration of the project's various facets.

2.1 ARCHITECTURE

The architecture of MedPrep is meticulously designed to meet the complex demands of hospital management and healthcare delivery. Embracing a modern and flexible approach, MedPrep follows a microservices architecture, breaking down the application into smaller, autonomous services. This modular design facilitates better scalability, fault isolation, and maintenance, ensuring that each component can be developed, deployed, and updated independently.

At the frontend, MedPrep leverages React.js, a widely-used JavaScript library renowned for building dynamic and responsive user interfaces. This choice enables patients, doctors, nurses, and administrators to interact seamlessly with the system through web browsers or dedicated mobile applications. Meanwhile, the backend relies on Node.js and Express.js, providing a robust foundation for handling HTTP requests, routing, and middleware integration. Data management is handled by PostgreSQL, a powerful relational database known for its reliability and performance, ensuring efficient storage and retrieval of critical healthcare data.

Authentication and authorization are paramount in MedPrep's architecture. JSON Web Tokens (JWT) are employed for authentication, while role-based access control (RBAC) dictates user permissions, safeguarding sensitive data and functionalities from unauthorized access. Integrations with external systems further enhance MedPrep's capabilities.

Scalability and high availability are fundamental considerations in MedPrep's architecture. Horizontal scalability allows for the addition of more resources to handle increased loads or traffic, while load balancers distribute incoming requests across multiple instances of the application for optimal performance. Redundant components and failover mechanisms minimize downtime, ensuring uninterrupted service even in the event of component failures.

Monitoring and logging play a crucial role in maintaining the health and performance of MedPrep's architecture. Tools such as Prometheus and Grafana monitor system performance in real-time, while centralized logging solutions like the ELK stack aggregate and analyze log data for proactive maintenance and optimization.

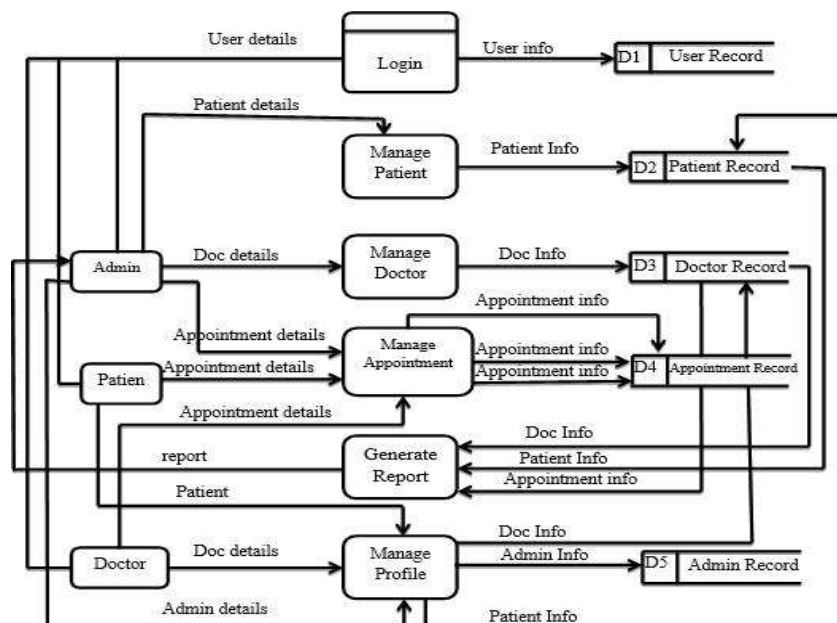


Fig 2.1 Architecture Diagram

2.2 USER INTERFACE

The frontend of our web application is meticulously designed and developed using React.js, a renowned JavaScript library celebrated for its flexibility and reusability. Our focus lies in crafting an intuitive and visually appealing user interface that prioritizes user experience above all else. With an emphasis on user-friendly navigation, responsive design, and clear data visualization, our UI design philosophy aims to enhance usability and engagement for every user interaction.

Central to our approach is the implementation of React components, following a component-based architecture that capitalizes on React.js's modularity. This methodology enables us to create reusable UI elements, fostering consistency across the application while streamlining development efforts. By breaking down complex UI structures into smaller, manageable components, we not only improve code maintainability but also enhance scalability, allowing for seamless integration of new features and functionalities.

Our frontend team's meticulous attention to detail ensures that every aspect of the user interface is thoughtfully crafted to meet the diverse needs and preferences of our users. From intuitive navigation paths to responsive layouts that adapt seamlessly to different devices and screen sizes, we strive to provide a cohesive and immersive user experience. Moreover, clear data visualization techniques are employed to present information in a digestible format, empowering users to comprehend and interact with data effortlessly.

Through thoughtful design decisions, rigorous testing, and continuous iteration, we aim to surpass user expectations and set new standards for usability and engagement in web application interfaces.

2.2.1 INTEGRATION WITH MATERIAL UI

MedPrep's frontend architecture is carefully crafted using React.js, a powerful JavaScript library renowned for its flexibility and performance in building dynamic user interfaces. Complementing React.js is Material-UI, a popular React component library that adheres to Google's Material Design guidelines, offering a plethora of pre-designed and customizable UI components.

The integration of React with Material-UI in MedPrep ensures a seamless and visually appealing user experience throughout the application. Material-UI provides a comprehensive collection of components, ranging from buttons and forms to navigation bars, all of which can be easily tailored to align with MedPrep's branding and design specifications. Furthermore, Material-UI's responsive design and accessibility features enhance the usability of the application, catering to the diverse needs of users across different devices and screen sizes.

One of the key strengths of Material-UI lies in its modular and component-based architecture, which perfectly complements React's own component-based approach. This synergy between React and Material-UI facilitates code reusability and maintainability, enabling developers working on the MedPrep project to expedite development tasks while ensuring consistency in UI design and implementation. With Material-UI's extensive documentation and vibrant community support, developers have access to a wealth of resources to streamline their workflow and address any challenges encountered during development.

In summary, the integration of React.js with Material-UI in MedPrep's frontend architecture not only ensures a visually appealing and consistent user experience but also fosters efficiency and maintainability in development.

2.2.2 INTEGRATION WITH BACKEND

The integration of the frontend with the backend is a pivotal component of the MedPrep Hospital Management System, serving as the backbone that enables seamless communication between the user interface and the underlying server-side logic and data storage. This integration is meticulously executed through the development of API endpoints using Express.js, a powerful Node.js framework renowned for its efficiency and scalability in building web applications.

Express.js plays a crucial role in facilitating communication between the frontend and backend components of MedPrep, providing a robust platform for defining and managing RESTful API endpoints. These endpoints serve as the gateway through which the frontend interacts with the backend services, enabling operations such as data retrieval, manipulation, and storage. By leveraging Express.js, MedPrep ensures that data flows smoothly between the user interface and the server, fostering a seamless and responsive user experience.

Moreover, the integration with the backend extends beyond mere data exchange, encompassing the execution of complex business logic and the enforcement of security measures. Express.js enables developers to implement intricate server-side functionalities, such as authentication, authorization, and data validation, ensuring the integrity and confidentiality of the system's data. Additionally, the framework's middleware architecture allows for the incorporation of various security mechanisms, such as encryption and access control, to safeguard sensitive information and mitigate potential security threats.

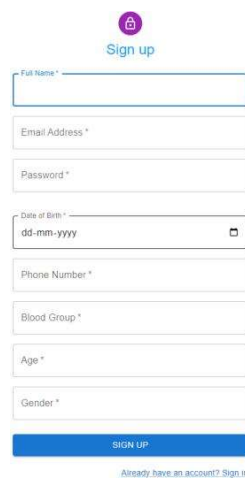
2.3 USER ACCESSIBILITY

User accessibility is a cornerstone of MedPrep's design, ensuring that patients, doctors, nurses, and administrators can engage with the platform effectively and effortlessly. The interface is meticulously crafted with simplicity and ease of use in mind, featuring clear navigation, intuitive layouts, and consistent design elements to accommodate users of all technical proficiencies. MedPrep's responsive design adapts seamlessly to various devices and screen sizes, enabling users to access the platform from desktop computers, tablets, or smartphones with equal convenience. Role-based access control ensures that each user group has access to the features and information relevant to their roles, while personalized dashboards provide tailored experiences for different user groups. Accessibility features such as high contrast modes, keyboard navigation support, and multi-lingual support further enhance usability, ensuring that all users, regardless of their abilities or language preferences, can engage with MedPrep comfortably and efficiently.

2.3.1 USER REGISTRATION

User registration in MedPrep is a streamlined process designed to onboard new users efficiently while ensuring security and data accuracy. Upon accessing the platform, users are guided through a registration form where they provide essential information such as their name, email address, contact details, and desired login credentials. MedPrep employs robust validation mechanisms to verify the authenticity of user-provided information, reducing the risk of errors and fraudulent registrations. Additionally, users may be required to verify their email address through a confirmation link sent to their registered email, enhancing account security and preventing unauthorized access.

Once the registration process is complete and the user's account is verified, they gain access to the full range of features and functionalities within MedPrep based on their assigned role and permissions. Whether they are patients, doctors, nurses, or administrators, users can log in securely to their personalized dashboard, where they can manage appointments, access medical records, communicate with healthcare providers, and perform other tasks relevant to their role. MedPrep's user registration process sets the foundation for a seamless user experience, empowering users to engage with the platform effectively and access the healthcare services they need with ease.



The image shows a user registration form titled "Sign up" with a purple lock icon. The form contains the following fields: Full Name *, Email Address *, Password *, Date of Birth * (with a placeholder "dd-mm-yyyy" and a calendar icon), Phone Number *, Blood Group *, Age *, and Gender *. At the bottom is a blue "SIGN UP" button and a link that says "Already have an account? Sign in".

Fig 2.2 User Registration

2.3.2 User Dashboard

Upon logging into MedPrep's system, users are seamlessly directed to their respective dashboards, each meticulously designed to cater to their specific roles and responsibilities within the healthcare management system. The Admin Dashboard stands out as the central nerve center, furnishing administrators with a plethora of comprehensive tools aimed at efficiently managing user data and ensuring the smooth operation of the platform.

In contrast, the Doctor Dashboard emerges as a vital tool for healthcare providers, equipping them with efficient resources to navigate patient care and workflows with utmost effectiveness. This dynamic platform offers a holistic overview of patient appointments, medical records, and treatment plans, serving as a pivotal command center for doctors to access critical patient information, review diagnostic results, and seamlessly collaborate with fellow healthcare professionals. By fostering seamless communication and collaboration among healthcare team members, the Doctor Dashboard not only enhances patient safety but also elevates the quality of care delivery, ultimately leading to improved patient outcomes and satisfaction.

Similarly, the Patient Dashboard serves as a gateway for patients to access their personal health information, manage appointment schedules, and engage in secure communication with their healthcare providers. Through intuitive interfaces and integrated health education materials, patients are empowered to take an active role in their healthcare journey, promoting informed decision-making and proactive engagement in preventive care measures. Moreover, the Nurse Dashboard plays a pivotal role in facilitating the coordination of patient care and efficient management of nursing tasks. With comprehensive overviews of patient statuses, medication schedules, and care plans, nurses can prioritize tasks effectively, leveraging integrated decision support tools to make informed decisions and ensure adherence to treatment protocols. Overall, these tailored dashboards within the MedPrep Hospital Management System not only streamline operations but also enhance collaboration, communication, and ultimately, the delivery of high-quality patient care.

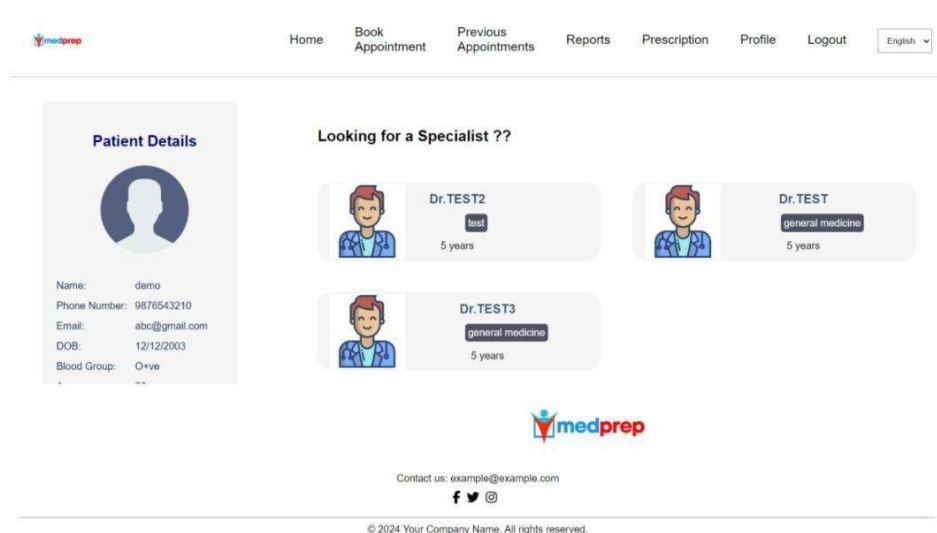


Fig 2.3 User Dashboard

2.3.3 APPOINTMENT MANAGEMENT

Booking appointments is a user-friendly process accessible through the Patient Dashboard. Patients can easily navigate to the appointment scheduling section, where they can view available time slots based on their healthcare provider's availability and specialty. They can then select a convenient appointment slot, input any relevant details or preferences, and confirm the booking.

Simultaneously, healthcare providers can view and manage their appointment schedules through the Doctor Dashboard. They have access to real-time updates on upcoming appointments, enabling them to plan their schedules effectively and allocate appropriate time for patient consultations. Additionally, providers can make necessary adjustments to appointment schedules, such as rescheduling or canceling appointments, based on patient needs or unforeseen circumstances. By streamlining the appointment management process, healthcare providers can optimize their workflow and ensure efficient patient care delivery.

Furthermore, the system enables users to view their previous appointments, providing a comprehensive overview of their medical history and healthcare interactions. Patients can access their appointment history through the Patient Dashboard, reviewing past appointments, including dates, healthcare providers, and any associated notes or treatments. Similarly, healthcare providers can access patients' appointment histories through the Doctor Dashboard, facilitating continuity of care and informed decision-making during patient consultations. Access to previous appointments allows both patients and healthcare providers to track progress, monitor treatment adherence, and make informed decisions about future care plans.

The screenshot displays the 'Book Appointment' form within the MedPrep application. The top navigation bar includes the MedPrep logo, a 'Home' link, and links for 'Book Appointment', 'Previous Appointments', 'Reports', 'Prescription', 'Profile', and 'Logout'. A language dropdown menu is set to 'English'. The form itself is titled 'Book Appointment' and contains several input fields: a 'Name *' text box, a 'Select Date *' dropdown (showing 'dd-mm-yyyy'), a 'Schedule Time *' dropdown (showing 'Select a Time slot'), a 'Select Specialist *' dropdown (showing 'Select a Doctor'), and a 'Reason For Appointment *' text box. A 'Make An Appointment' button is positioned below the form. The footer section includes the MedPrep logo, contact information 'Contact us: example@example.com', social media icons for Facebook, Twitter, and Instagram, and a copyright notice '© 2024 Your Company Name. All rights reserved.'

Fig 2.4 Appointment Management

2.3.4 ACCESS CONTROL

Access control in MedPrep's system is a crucial component designed to regulate and manage user permissions, ensuring that individuals only have access to the resources and functionalities necessary for their roles and responsibilities within the healthcare management system.

Through role-based access control (RBAC), MedPrep assigns specific roles to users, such as patients, doctors, nurses, and administrators, each with predefined sets of permissions. These roles determine the level of access users have to various features and data within the system, helping to maintain data confidentiality, integrity, and availability.

Administrators have the authority to define and manage user roles and permissions, assigning or revoking access rights as needed. They can customize access control settings based on organizational policies, regulatory requirements, and specific user needs, ensuring that sensitive information remains protected and only accessible to authorized personnel.

Furthermore, MedPrep implements authentication mechanisms to verify the identity of users before granting access to the system. This includes secure login processes such as username-password authentication, multi-factor authentication (MFA), and session management techniques to prevent unauthorized access and protect user accounts from potential security threats.

2.3.5 REPORTS AND PRESCRIPTION MANAGEMENT

MedPrep stands out for its multifaceted approach to healthcare management, offering a robust suite of features aimed at optimizing both administrative processes and patient care. One of its key strengths lies in its comprehensive reporting capabilities, which empower administrators and healthcare professionals to delve deep into hospital operations, patient demographics, and treatment outcomes. By providing detailed insights into various aspects of healthcare delivery, MedPrep enables informed decision-making, quality improvement initiatives, and regulatory compliance.

Furthermore, MedPrep revolutionizes prescription management by streamlining processes and introducing electronic prescription uploading capabilities. Healthcare providers can securely upload prescriptions directly into the system, ensuring accuracy and accessibility for both patients and fellow healthcare professionals. This electronic transmission not only enhances medication adherence but also reduces the risk of errors associated with manual prescription handling. Patients benefit from the seamless integration of prescriptions into their electronic health records, facilitating streamlined medication acquisition and fostering a patient-centric approach to healthcare delivery.

Moreover, MedPrep's innovative approach extends to report management, where healthcare providers can efficiently upload various types of medical reports directly into patient records. This feature significantly enhances the accessibility and organization of crucial medical information, such as imaging studies, laboratory results, and consultation notes. By centralizing medical reports within the system, MedPrep facilitates seamless collaboration among healthcare team members, reduces administrative burden, and promotes efficient healthcare delivery.

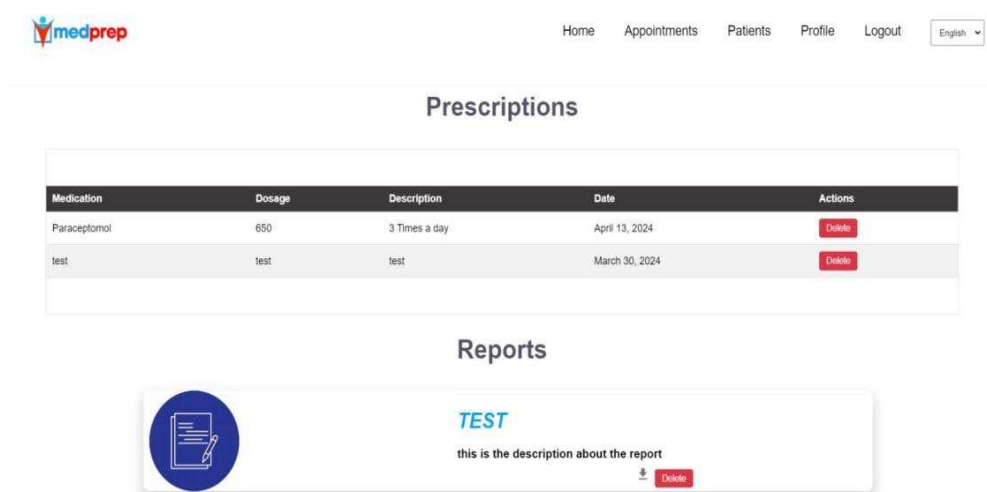


Fig 2.5 Prescription and Reports

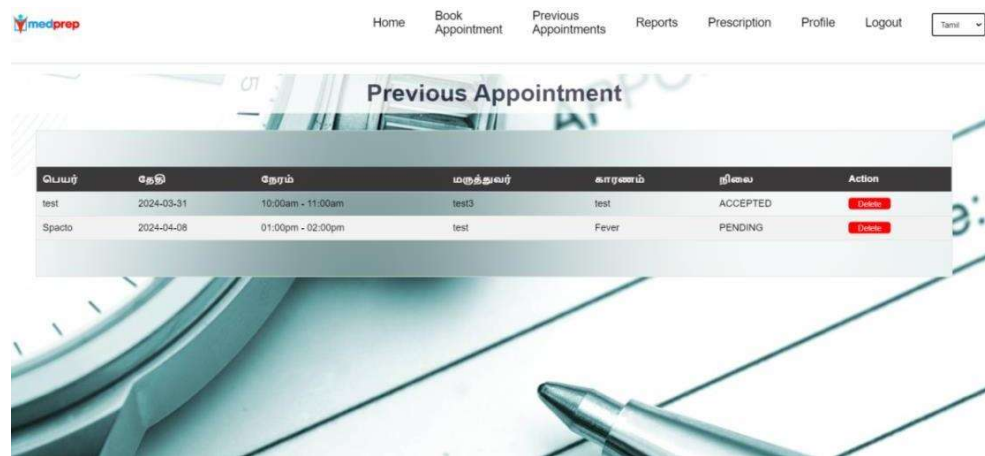
2.4 MULTI LANGUAGE SUPPORT

MedPrep's commitment to inclusivity and accessibility is not just a principle but a cornerstone of its user-centric approach to healthcare management. Central to this commitment is the implementation of a robust multi-language support feature, facilitated by the utilization of i18next for localization. This feature is designed to cater to the diverse linguistic backgrounds of patients, healthcare providers, and administrators, ensuring that individuals can interact with the platform in their preferred language.

The implementation of i18next for localization underscores MedPrep's recognition of the importance of linguistic diversity in healthcare delivery. By offering support for multiple languages, MedPrep acknowledges and respects the linguistic preferences of its users, thereby fostering a sense of inclusivity and belonging within the healthcare ecosystem. This approach aligns with MedPrep's overarching goal of providing equitable access to healthcare services for all individuals, regardless of their linguistic backgrounds.

Through the intuitive interface of MedPrep, users can seamlessly switch between languages, enabling them to navigate the platform with ease and familiarity. Whether accessing patient records, scheduling appointments, or generating reports, users can perform these tasks confidently in their chosen language, without encountering any language barriers. This not only enhances user experience but also promotes effective communication and engagement, ultimately contributing to improved patient outcomes and satisfaction.

In essence, MedPrep's adoption of i18next for localization exemplifies its commitment to creating a healthcare management platform that is truly inclusive and accessible to all. By prioritizing multi-language support, MedPrep ensures that language differences do not hinder access to quality healthcare services, thereby empowering individuals to engage with the platform in a manner that best suits their linguistic preferences and needs.



The screenshot shows the 'Previous Appointment' page of the MedPrep application. The page features a navigation bar with links: Home, Book Appointment, Previous Appointments, Reports, Prescription, Profile, Logout, and a language dropdown menu set to 'Tamil'. The main content area displays a table of previous appointments. The table has columns for Patient Name (பெயர்), Date (தேதி), Time (நேரம்), Doctor Name (மருத்துவர்), Symptom (காரணம்), Status (நிலை), and Action. Two appointments are listed: one for 'test' on 2024-03-31 at 10:00am - 11:00am, which is 'ACCEPTED', and another for 'Spacio' on 2024-04-08 at 01:00pm - 02:00pm, which is 'PENDING'. Each row has a 'Delete' button in the Action column.

பெயர்	தேதி	நேரம்	மருத்துவர்	காரணம்	நிலை	Action
test	2024-03-31	10:00am - 11:00am	test3	test	ACCEPTED	Delete
Spacio	2024-04-08	01:00pm - 02:00pm	test	Fever	PENDING	Delete

Fig 2.6 Multi Language

2.5 ADMINISTRATIVE FUNCTIONS

Administrators within our web application are pivotal figures tasked with managing user data and ensuring the platform's smooth operation. They wield essential capabilities and hold significant responsibilities, serving as the backbone of the system. This chapter explores the duties, tools, and responsibilities of administrators, highlighting their crucial role in maintaining the integrity and functionality of the platform.

2.5.1 ADDING DOCTOR AND NURSE

MedPrep redefines the onboarding process for new healthcare professionals, offering administrators efficient tools to expand their provider network and healthcare team seamlessly. Through an intuitive user interface and optimized workflows, administrators can effortlessly input comprehensive details for each new doctor and nurse, covering personal information, qualifications, specialties, certifications, and contact details. Furthermore, administrators have the flexibility to customize roles and permissions according to the unique responsibilities and access requirements of each healthcare professional.

By consolidating these operations within MedPrep's intuitive platform, healthcare organizations foster a collaborative and cohesive environment conducive to delivering high-quality care. The streamlined onboarding process not only enhances administrative efficiency but also promotes a unified healthcare team focused on meeting the needs of patients effectively. This cohesive approach supports seamless collaboration among healthcare professionals, ultimately contributing to the delivery of superior patient care services within the healthcare facility.

2.5.2 MANAGING DOCTORS AND NURSES

MedPrep stands as a pioneering solution in healthcare management, empowering administrators with efficient tools to oversee and manage doctor and nurse data within the system. With a focus on accuracy and accessibility, MedPrep streamlines the process of managing comprehensive profiles for each healthcare professional, ensuring that critical information such as personal details, contact information, and specialization areas are readily available and up-to-date.

Furthermore, MedPrep's role-based access control feature empowers administrators to assign specific roles and permissions to doctors and nurses, ensuring that access to patient records and system functionalities is appropriately managed and aligned with organizational policies and privacy regulations. This granular level of control not only enhances data security but also promotes accountability and compliance across the healthcare facility.

Patients and administrators alike benefit from MedPrep's seamless interface, which provides easy access to detailed profiles of healthcare providers within the system. Patients can make informed decisions when selecting healthcare providers, leveraging comprehensive information about qualifications, specialties, and contact details. Administrators, on the other hand, can rely on accurate and up-to-date records for scheduling and management purposes, fostering transparency and trust between patients, providers, and healthcare administrators.

2.6 BACKEND DEVELOPMENT

The backend architecture of our web application leverages Express.js, a highly versatile Node.js framework, and PostgreSQL, a robust relational database management system. This strategic combination establishes a robust foundation for efficient data management and sophisticated server-side logic. Express.js, renowned for its flexibility and scalability, serves as the backbone of our backend, facilitating seamless communication between the frontend and backend components. Its asynchronous nature and event-driven architecture enable rapid development of RESTful APIs, ensuring smooth interaction between the client and server.

Complementing Express.js, PostgreSQL stands as a powerful choice for data storage and management. As a relational database management system, PostgreSQL offers strong support for ACID (Atomicity, Consistency, Isolation, Durability) properties, ensuring data integrity and reliability. Its advanced features include support for complex queries, data types, and indexing mechanisms, enabling efficient retrieval and manipulation of structured data. Additionally, PostgreSQL's scalability and performance capabilities make it well-suited for handling large volumes of data and concurrent user requests.

Together, Express.js and PostgreSQL form a robust backend infrastructure that underpins our web application's functionality. The integration of these technologies provides scalability, reliability, and performance, essential for delivering a seamless user experience and supporting the application's growth.

2.6.1 BUILDING API ENDPOINTS

Building API endpoints is a fundamental aspect of developing a robust backend architecture for web applications, and in the case of our project, MedPrep, it serves as the backbone of communication between the frontend and backend components. The process involves creating routes within the Express.js framework to handle incoming requests from clients, execute the necessary server-side logic, and send back appropriate responses. These endpoints define the interface through which data is exchanged between different parts of the application, facilitating seamless interaction and data flow.

Express.js, being a minimalist and flexible framework for Node.js, offers a straightforward yet powerful way to define and manage API routes. Leveraging Express.js, our development team designs routes to handle various types of requests, such as GET, POST, PUT, and DELETE, corresponding to different CRUD (Create, Read, Update, Delete) operations. Each endpoint is carefully crafted to perform specific actions, such as retrieving user data, processing form submissions, or updating database records. By structuring endpoints logically and adhering to RESTful principles, we ensure consistency, scalability, and maintainability in our API design.

Furthermore, the implementation of API endpoints in MedPrep involves integrating with PostgreSQL, our chosen relational database management system, to perform data operations. This includes querying the database to fetch or update information, executing transactions, and handling errors gracefully. The endpoints are designed to handle authentication and authorization mechanisms to secure sensitive data and control access based on user roles and permissions.

2.6.2 DESIGNING DATABASE

Designing the database architecture for MedPrep involves careful consideration of the application's data requirements and the choice of appropriate tools to fulfill those needs. With Prisma as the ORM tool and MongoDB as the chosen database, we ensure a flexible and efficient data storage solution that aligns with the project's objectives.

Prisma serves as a powerful tool for database management, offering features such as schema definition, data modeling, and query generation. Leveraging Prisma's schema definition capabilities, we define the structure of our data models, specifying entities, relationships, and constraints.

This schema-centric approach provides a clear and concise representation of the application's data architecture, enabling efficient management and maintenance of the database. In conjunction with Prisma, MongoDB offers a NoSQL database solution that excels in handling unstructured or semi-structured data. MongoDB's document-oriented model aligns well with the dynamic nature of healthcare data, allowing for flexible schema designs and seamless scalability. Through Prisma's integration with MongoDB, we leverage MongoDB's features such as document storage, indexing, and aggregation pipelines to optimize data storage and retrieval performance.

Prisma's schema definition capabilities enable us to define these entities and their respective fields, ensuring data integrity and consistency. We establish relationships between entities, such as one-to-many or many-to-many relationships, to represent the connections between different data elements accurately. Additionally, we optimize the database schema for efficient query performance, such as indexing, denormalization, and data partitioning.

2.6.3 USER AUTHENTICATION

User authentication stands as a cornerstone within our web application, serving as a pivotal mechanism to secure sensitive data and ensure that access is granted only to authorized individuals. In the dynamic landscape of online platforms, where privacy and data integrity are paramount concerns, our authentication system plays a critical role in fortifying the integrity of user interactions and safeguarding against unauthorized access.

At its core, our authentication system operates on the principles of robust security measures and seamless user experience. Through a combination of industry-standard protocols and tailored implementation, we ensure that users can securely access their accounts while mitigating risks associated with unauthorized access or data breaches.

2.7 Deployment and Hosting

The final phase of development involves deploying the application to a hosting environment where it can be accessed by users. This section discusses the deployment process and hosting choices.

Continuous Integration and Deployment (CI/CD): We implemented CI/CD pipelines to automate the deployment process. Code changes were automatically tested and deployed to the production environment, reducing manual errors and ensuring a consistent release cycle.

Security was a top priority during deployment. Firewalls, intrusion detection systems, and SSL certificates were utilized to protect the application from potential threats

CHAPTER 3

RESULTS AND DISCUSSION

USER REGISTRATION:

This page allows users to either log in with their registered email, their role and password or sign up for a new account.

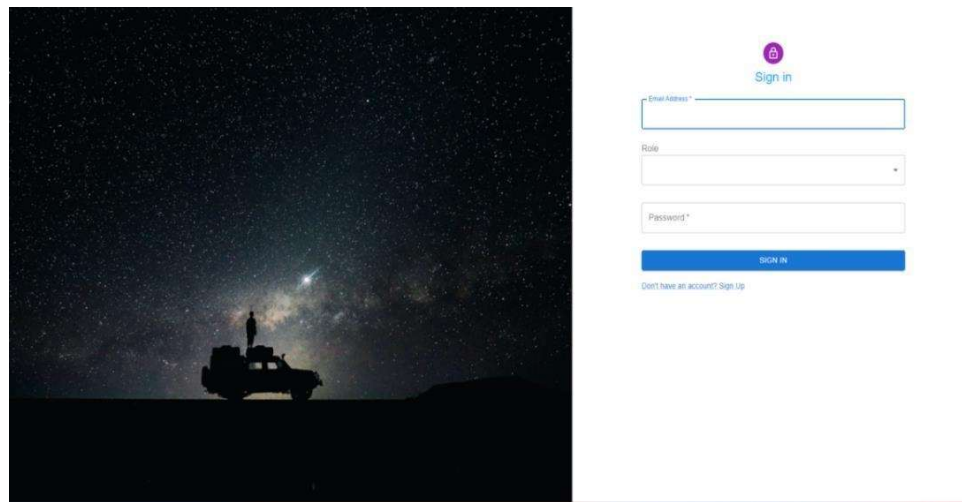


Fig 3.1 Multi User Login Page

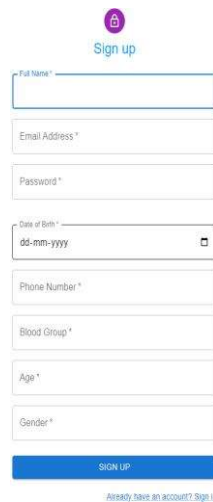
The image shows a 'Sign up' form on the right side of a dark background. The background features a silhouette of a person standing on a vehicle, looking up at a starry night sky with the Milky Way visible. The form has a purple lock icon and the text 'Sign up' in blue. It includes input fields for 'Full Name *', 'Email Address *', 'Password *', 'Date of Birth *' (with a date picker icon), 'Phone Number *', 'Blood Group *', 'Age *', and 'Gender *'. Below these fields is a blue 'SIGN UP' button. At the bottom of the form, there is a link that says 'Already have an account? Sign in'.

Fig 3.2 Patient Signup

DASHBOARD:

A dashboard is a visual representation of key performance indicators, metrics, and data insights presented in a centralized interface. It provides users with at-a-glance information, allowing them to monitor the health and performance of systems, processes, or businesses in real-time

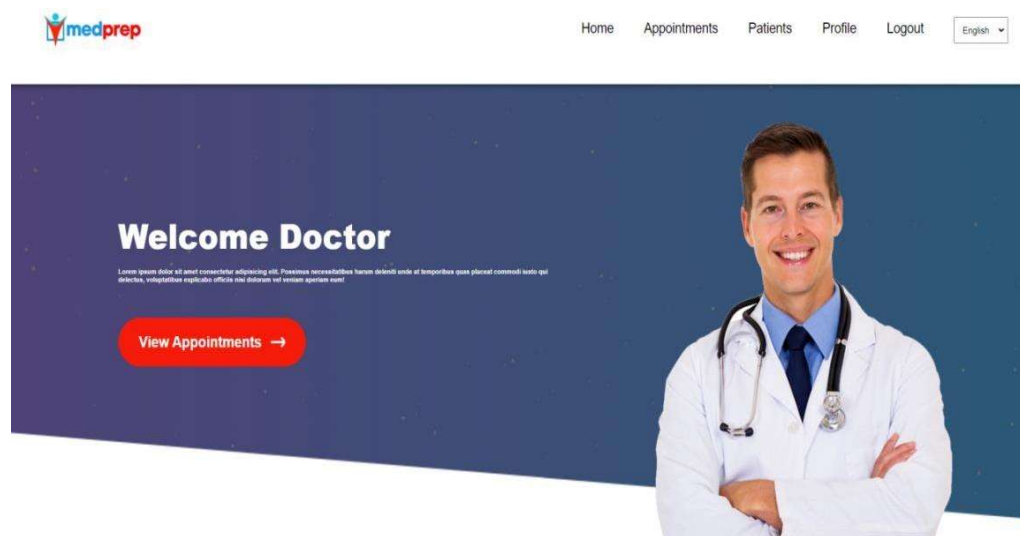


Fig 3.3 Home Page

APPOINTMENTS:

This page serves as a platform for users to efficiently schedule appointments according to their specific needs and preferences. Users can access the page to set up appointments for various purposes, such as medical consultations, meetings, or service appointments.

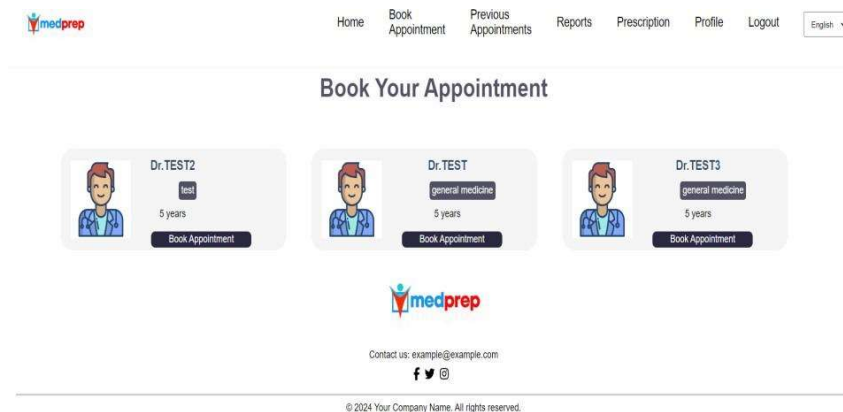


Fig 3.4 Book Appointment

PRESCRIPTION AND REPORT:

The prescription and reports page serves as a centralized hub for accessing medical prescriptions and diagnostic reports. Users can view and download their prescriptions from healthcare providers, including medication details and dosage instructions. Additionally, users can access diagnostic reports such as lab tests, imaging results, and medical summaries, providing comprehensive health information and facilitating informed decision-making regarding their healthcare needs.

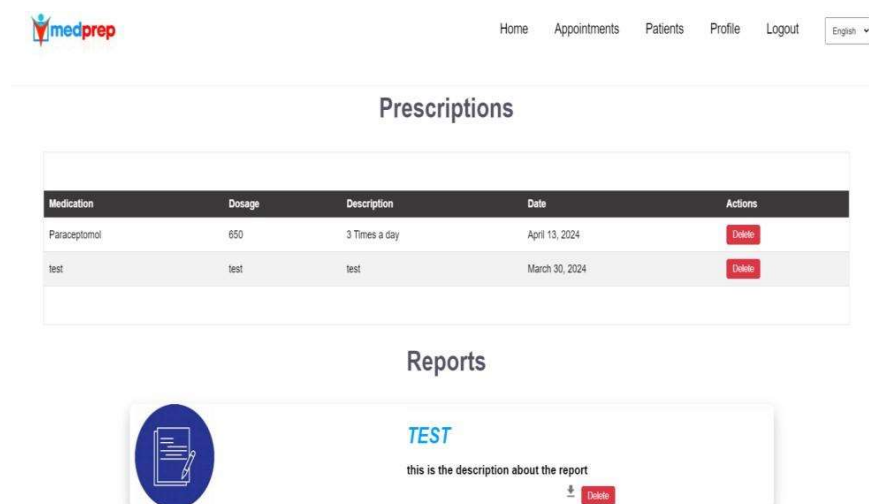


Fig 3.5 Prescription and Reports

MULTI LANGUAGE SUPPORT:

The multi-language page enables users to access content in different languages, catering to a diverse audience. Users can select their preferred language from a dropdown menu or language switcher, dynamically translating the content into their chosen language. This feature enhances accessibility and user experience, allowing individuals from various linguistic backgrounds to engage with the website or application seamlessly.

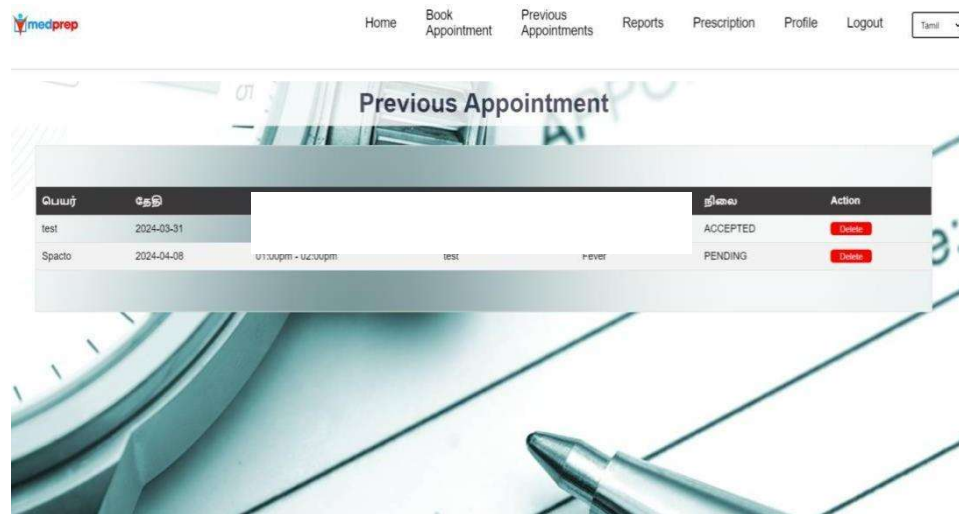


Fig 3.6 Multi-Lingual Support

CHAPTER 4

CONCLUSION AND FUTURE WORKS

4.1 CONCLUSION

MedPrep epitomizes the epitome of contemporary healthcare management systems, revolutionizing the landscape with its comprehensive features and user-centric design. Throughout the exploration of its functionalities, it becomes evident that MedPrep is not merely a tool for administrative convenience, but a catalyst for transforming patient care. By seamlessly integrating appointment scheduling, report management, prescription tracking, and multi-lingual support, MedPrep bridges the gap between patients and healthcare providers, fostering a collaborative environment that prioritizes accessibility and inclusivity.

Moreover, MedPrep's commitment to data security and privacy underscores its dedication to safeguarding sensitive medical information, ensuring compliance with regulatory standards and enhancing user trust. With its robust authentication mechanisms and secure data management protocols, MedPrep provides users with peace of mind, allowing them to focus on delivering high-quality care without compromising confidentiality.

Furthermore, MedPrep's intuitive interface and customizable features empower healthcare institutions to tailor the platform to their specific needs, optimizing workflow efficiency and enhancing productivity. Whether it's managing doctor and nurse data, viewing patient information, or generating comprehensive reports, MedPrep offers a seamless experience that caters to the unique requirements of each user.

4.2 FUTURE WORKS

1. Enhanced Mobile Accessibility:

We plan to develop dedicated mobile applications for MedPrep, enabling users to access the platform seamlessly from smartphones and tablets, enhancing convenience and accessibility.

2. Telemedicine Integration:

Integration with telemedicine platforms will enable MedPrep users to conduct virtual consultations and appointments, expanding access to healthcare services and improving patient engagement.

3. AI-driven Insights:

We aim to leverage artificial intelligence and machine learning algorithms to analyze data within MedPrep, providing actionable insights for healthcare providers to optimize treatment plans, resource allocation, and patient outcomes.

4. IoT Integration:

Integration with Internet of Things (IoT) devices, such as wearable health monitors and smart medical devices, will enable MedPrep to collect real-time patient data, facilitating remote monitoring and proactive healthcare management.

5. Blockchain for Data Security:

Exploring blockchain technology for enhanced data security and privacy, ensuring immutable and transparent record-keeping while maintaining confidentiality and integrity of medical records within MedPrep.

6. Patient Engagement Tools:

Development of interactive patient engagement tools within MedPrep, such as health trackers, medication reminders, and educational resources, to empower patients to take an active role in managing their health.

7. Genomic Data Integration:

Integration of genomic data analysis capabilities into MedPrep, enabling healthcare providers to incorporate genetic information into personalized treatment plans and disease management strategies.

8. Expansion of Language Support:

Continuously expanding language support within MedPrep to cater to diverse linguistic populations, ensuring that language barriers do not hinder access to quality healthcare services.

9. Predictive Analytics for Disease Prevention:

Implementation of predictive analytics models within MedPrep to identify at-risk patient populations and proactively intervene with preventive measures, reducing disease burden and healthcare costs.

REFERENCES

- F. Lubrano, F. Stirano, G. Varavallo, F. Bertone, and O. Terzo, "Hams: an integrated hospital management system to improve information exchange," in *Advances in Intelligent Systems and Computing*, 2021.
- M. Kleppmann, "Designing Data-Intensive Applications," 3rd ed. O'Reilly, 2017
- O. Obe and L. S. Hsu, "PostgreSQL: Up and Running," 3rd ed. O'Reilly, 2017
- Paras V. Kothare, Yogesh K. Gedam and Ratnadeep R. Deshmukh, "Ranking Algorithm RA-SVM for Domain Adaptation: A Review", *International Journal of Scientific & Engineering Research*, vol. 5, no. 4, pp. 503-507, 2014, ISSN 22295518.
- Pratik R. Mantri and Prof. Mahip M. Bartere, "Review on Adaption of Ranking Model for Domain Specific Search", *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 4, pp. 103-110, 2014, ISSN 2320-088X.
- P. I. Listyorini, A. Dader, M. Z. Ulhaq, and A. S. Nugraini, "Literature Study: Factors Affecting the Implementation of Hospital Management Information Systems in Hospital," *Proceedings of the International Conference on Nursing and Health Sciences*, vol. 3, no. 1, 2022,
- Robert C. Martin, "Clean Code: A Handbook of Agile Software

APPENDICES

A. SAMPLE CODE

```
const App = () => {
  return (
    <>
      <BrowserRouter>
        <Routes>
          <Route path="/" exact element={<Home />} />
          <Route path="/signup" exact element={<SignUp />} />
          <Route path="/doctor" exact element={<DoctorDashboard />} />
          <Route path="/nurse" exact element={<NurseDashboard />} />
          {/* Patient Dashboard */}
          <Route path="/patient" exact element={<PatientDetails />} />
          <Route path="/nurse/patientlist" exact element={<PatientList />} />
          <Route path="/nurse/patients/:patientId" exact
            element={<IndividualPatientNurse />} />
          <Route path="/patient/book/:id" exact element={<Book />} />
          <Route path="/patient/bookApp"
            exact element={<PatientDashboard />} />
          <Route path="/patient/prev" exact element={<PrevApp />} />
          <Route path="/patient/Reports" exact element={<Reports />} />
          <Route
            path="/patient/:activepage"
            exact
            element={<PatientProfile />}
            />
          <Route
            path="/patient/:activepage"
            exact
            element={<PChangePassword />}
            />
          {/* <Route path='/patient/Details'
            exact element={<PatientDetails/>}/> */}
          <Route path="/doctor/:activepage"
            exact element={<DoctorProfile />} />
          <Route
            path="/doctor/prescription-form-doctor/:patientId"
            exact
            element={<PrescriptionForm />}
            />
          <Route
            path="/doctor/upload-form-doctor/:patientId"
```

```

    exact
    element={<ReportUpload />}
  />
  <Route
    path="/patient/prescription"
    exact
    element={<Prescription />}
  />
  <Route path="/patient/docinfo" exact element={<ViewDetails />} />

  <Route path="/Forgot" exact element={<Forgot />} />
  <Route path="/Admin/AddDoctor" exact element={<AddDoctor />} />
  <Route path="/Admin/AddNurse" exact element={<AddNurse />} />
  <Route path="/Admin" exact element={<AdminDashboard />} />
  <Route path="/Admin/ListDoctor" exact element={<ListDoctor />} />
  <Route path="/Admin/ListNurse" exact element={<ListNurse />} />
  <Route
    path="/doctor/appointment"
    exact
    element={<AppointmentDoc />}
  />

  <Route path="/doctor/patients" exact element={<PatientlistDoc />} />
  <Route path="/doctor/patients/:patientId" exact
    element={<IndividualPatient />} />
  <Route path="/nurse/:activepage" exact element={<NurseProfile />} />
  <Route
    path="/nurse/:activepage"
    exact
    element={<NChangePassword />}
  />
</Routes>
</BrowserRouter>
</>
);};

export default App;

```

// This is your Prisma schema file,
// learn more about it in the docs: <https://pris.ly/d/prisma-schema>

```
generator client {  
  provider = "prisma-client-js"  
}  
  
datasource db {  
  provider = "postgresql"  
  url      = env("DATABASE_URL")  
}  
  
enum AppointmentStatus {  
  PENDING  
  REJECTED  
  ACCEPTED  
}  
  
enum FileType {  
  pdf  
}  
  
model Admin {  
  id      String @unique @default(uuid()) @db.Uid  
  fullName String  
  phone   String @db.VarChar(10)  
  email    String @unique  
  password String  
  createdAt DateTime @default(now())  
  updatedAt DateTime @updatedAt  
}  
  
model Nurse {  
  id      String @unique @default(uuid()) @db.Uid  
  fullName String  
  phone   String @db.VarChar(10)  
  email    String @unique  
  password String  
  experience String  
  createdAt DateTime @default(now())  
  updatedAt DateTime @updatedAt  
}  
  
model Patient {  
  id      String @unique @default(uuid()) @db.Uid  
  fullName String  
  email    String @unique  
  phone    String @db.VarChar(10)
```

```

dob      String
bloodGroup String
password String
gender   String
age      String
appointments Appointments[]
reports   Reports[]
prescriptions Prescription[]
// doctors DoctorOnPatients[]
createdAt DateTime @default(now())
updatedAt  DateTime @updatedAt
}

model Appointments {
  id          String @unique @default(uuid()) @db.Uuid
  doctorId    String @db.Uuid
  doctor      Doctor @relation(fields: [doctorId], references: [id])
  patientId   String @db.Uuid
  patient     Patient @relation(fields: [patientId], references: [id])
  patientName String
  doctorName  String
  reason      String
  appointmentDate String
  appointmentTiming String
  status      AppointmentStatus @default(PENDING)
  createdAt   DateTime @default(now())
  updatedAt   DateTime @updatedAt
}

model Prescription {
  id          String @unique @default(uuid()) @db.Uuid
  patientId   String @db.Uuid
  patient     Patient @relation(fields: [patientId], references: [id])
  medication  String
  dosage      String
  description  String
  createdAt   DateTime @default(now())
  updatedAt   DateTime @updatedAt
}

model Doctor {
  id          String @unique @default(uuid()) @db.Uuid
  fullName    String
  phone       String @db.VarChar(10)
  email       String @unique
  password    String
  specializations String
  experience  String
  appointments Appointments[]

```

```

    // patients    DoctorOnPatients[]
    createdAt     DateTime    @default(now())
    updatedAt     DateTime    @updatedAt
  }

  model Reports {
    id           String  @unique @default(uuid()) @db.Uuid
    patientId    String  @db.Uuid
    patient      Patient @relation(fields: [patientId], references: [id])
    title        String
    fileId       String
    // file      File[]
    description   String?
    createdAt    DateTime @default(now())
    updatedAt    DateTime @updatedAt
  }

  model File {
    id           String  @unique @default(uuid()) @db.Uuid
    type         FileType @default(pdf)
    data         Bytes
    // reportId   String? @unique @db.Uuid
    // reports    Reports? @relation(fields: [reportId], references: [id])
    createdAt    DateTime @default(now())
    updatedAt    DateTime @updatedAt
  }

import * as dotenv from "dotenv"

import app from "./server";
import logger from "./logger";
import { createAdmin } from "./handlers/admin";

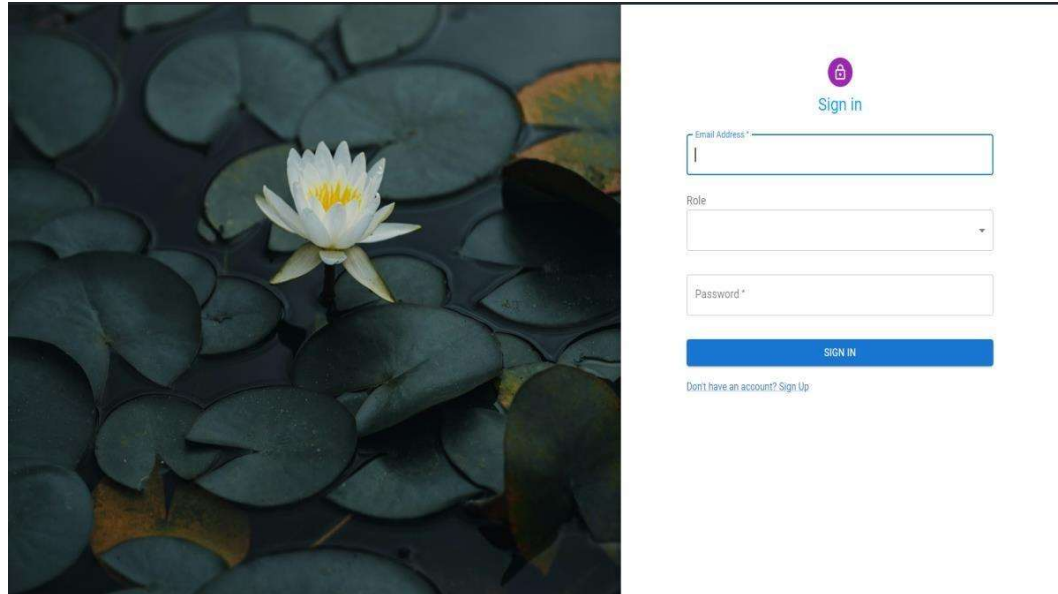
dotenv.config()
const port = 2222 as number;

(async (): Promise<void> => {
  try {
    await createAdmin();
  } catch (error: any) {
    logger.error(`Error creating admin: ${error.message || error.toString()}`);
  }
})();

app.listen(port, () => {
  logger.info(`listening on http://localhost:${port}`);
})

```

B. SCREEN SHOTS



Nurse List

	Name: TEST1 Email: abcd@gmail.com Phone: 123567893 Experience: fresh  Delete
	Name: TEST2 Email: abc@gmail.com Phone: 123567897 Experience: fresher  Delete



Appointments

Name	Date	Time	Reason	Status	Actions	Report	Prescription
test	2024-03-31	10:00am - 11:00am	test	ACCEPTED	Approve Reject	Upload Report	Add Prescription



Contact us: example@example.com



My profile

[Account Settings](#)[Change Password](#)

Personal Information

Name *

demo

Phone *

9876543210

Email *

abc@gmail.com

Age *

20

Gender *

Date Of Birth *