Healthcare Platform for Online Consultation

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

Ananya Srivastav (181417)

UNDER THE SUPERVISION OF

Dr. Aman Sharma

To

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, Solan 173234, Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report is entitled as "Healthcare Platform for Online Consultation" in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering/ Information Technology submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of our own work carried out over a period from January 2022 to May 2022 under the supervision of Dr. Aman Sharma, Assistant Professor(SG), Computer Science. The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Ananya Srivastav, 181417

This is to certify that the above statement made by the candidate is true to the best of my knowledge

Shanky Jain

Dr. Aman Sharma

Assistant Professor (SG)

Computer Science

Dated:

Shanky Jain

Senior Product Specialist

HashedIn by Deloitte

Dated: 24/05/22

Acknowledgement

I would like to covey my heartfelt gratitude to everyone who made it possible for me to put this project together. I would like to convey my heartfelt gratitude to Dr. Aman Sharma, my final year project adviser, whose persistent encouragement ad helpful recommendations aided me in successfully completing my project, particularly throughout the drafting of this report.

I'd also like to express my gratitude to the Jaypee University of Information Technology team, who allowed me to use all of the necessary equipment and materials to accomplish my project "Healthcare Platform for Onlie Consultation". At last but not least I would like to extend my gratitude to my company's supervisor who assisted me with various areas of my project and provided solutions to my challenges. I am grateful for the advice I received from other faculty members too, particularly during my project presentation, which has helped my enhance my presentation skills. I appreciate their suggestions and remarks during the presentation.

Abstract

The majority of manual systems are being replaced and becoming automated as technology advances. In this context, I plan to design an appointment system that is simple, quick, and painless for both the doctor and the patient. Despite its status as a developing country, India boasts a big population of internet users. A nexus will be required if consumers want to connect to their selected doctor via the internet. To that end, I intend to create a website where people may schedule appointments. Ordinary people, both at home and abroad, will be able to receive instant aid without wasting time or effort. People can use this method to rapidly find out what the doctor's consultation time is and arrange their appointment whenever they like. If the list is appropriately structured, people will feel more at ease exploring their predicted specialty doctors.

The dissertation's purpose is to develop an online health-care platform. The purpose of developing this app is to create a system where a patient can easily select and schedule a doctor's appointment from the comfort of their own home. The second purpose is to use an online database management system to replace manual medical record keeping. Like all other programmes, this one has a client side and a server side. On the client side, React.js was used, and on the server, Node.js was used. For the time being, online appointment features have been introduced, but future work will focus on connecting pharmacies and laboratories to the system.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

One of the fastest-growing industries on the planet is health care. Prior to the last few years, most medical appointments were made over the phone or by visiting hospitals in person. Because this approach required individual engagement, the ability to book appointments was constrained by the availability of schedulers, phone lines, or a person's physical presence. As time passed, everyone wanted timeless and efficient medical care delivery because manual appointments (which require both individuals' physical presence) and long waiting lines had produced an inconvenient situation for healthcare facilities. As a result, a desire arose for an integrated health-care system that could provide seamless care to both outpatients and inpatients. The adoption of an online appointment system made health-care services more accessible and timely. As a result, online appointment scheduling is extremely important to hospitals and other medical organizations. Online appointment scheduling has become a widespread trend in recent years, and it is now considered one of the most significant operations in the healthcare industry.

Bailey (1952) saw scheduling as a compromise or trade-off between the waiting times of a doctor and a patient. Patients who are late or do not show up for appointments cause a doctor's time to be underutilised. In addition, gaps in appointment timings result in wasted time and underutilization of the doctor's time (Bailey, 1954). According to various researchers, long wait times are the leading cause of patient dissatisfaction. The period between a patient's request for an appointment and his examination is characterised by Cayirli (2003) as access time. Waiting time is defined by Veral as the period between the consultation and the scheduled appointment time, discounting the patient's early arrival (Veral, 2003).

Different researchers defined waiting/access time in different ways. In today's hurried lives, a well-designed appointment system is intended to improve patient satisfaction by reducing clinic and hospital costs and time. The demand for more efficient medical treatment develops in tandem with the world's population. An online appointment scheduling system allows a user to visit the doctor's website and arrange an appointment using online software. In this approach, online appointment scheduling tools assist doctors and patients, resulting in more effective healthcare delivery. There are many different types of online appointment solutions available today that are easy to set up and economical. The online scheduling system serves both doctors and patients because it delivers value-added services and various advantages. By eliminating the inconvenience of long wait times, it helps the patient feel appreciated. Online appointment solutions are becoming increasingly popular due to their low cost.

Medical emergencies are only one of the numerous challenges we confront every day. If someone is sick and needs to visit a doctor, they must first go to the hospital and wait for a doctor to become available. In order to get an appointment, the patient must also wait in line. The patient will not be notified of the cancellation until he or she arrives at the hospital if a doctor cancels an appointment due to an emergency. As a result, we must seek medical guidance whenever we are impacted by numerous disorders. Because everyone now has access to the internet, the online appointment system can be used by anybody to solve problems and inconveniences for patients. The goal of this project is to create a doctor patient management system that will help patients schedule medical appointments and achieve their objectives. This system allows doctors to manage their appointment slots online, and it also allows people to arrange appointments to fill vacant slots. This is the name-based counselling reservation system. Patients can book their favourite doctor through this system, which manages multiple doctors at the same time.

1.2 Problem Statement

Patients confront several challenges while trying to get an appointment with a doctor in their chambers or places. When people feel ill, they must first make an appointment with their chambers or hospital to visit a doctor for a checkup. It is a lengthy procedure that wastes people's time. People go to the doctor's chamber for a checkup on occasion, but the doctor is unavailable for various reasons. In the event of a medical emergency, it is the only way to get medical help. The problem statement clearly identifies the issue of contacting a doctor in the event of a medical emergency, thus if there is a means to schedule this appointment in a timely and convenient manner, it could be a solution. Because most individuals nowadays have access to the internet, having an online platform to contact a doctor during such moments could be a viable option. So, to execute this solution, I've intended to develop a web-based online doctor appointment application that allows users to arrange an appointment with a specialised doctor in case of a medical emergency.

1.3 Objectives

The purpose of this project is to provide a platform that allows patients and doctors to connect and interact freely, allowing patients to feel at ease. It also aims to alleviate the difficulties that patients face while booking appointments and keeping track of their medical records. Patients can select a doctor based on their professional biography and patient evaluations. The medical store administrator can see recommended prescriptions, and the laboratory can see clinical testing that the doctor has requested. This feature tries to get rid of prescriptions that are written on paper.

Our main objectives are to help people identify doctors and schedule appointments. Users can search for doctors, making finding a certain doctor easier. Gathering needs is critical to constructing a faultless system. The research will help us better understand people's requirements, the system we want to build, and how much ground we'll cover. All interactions between patients, clinicians, and administrative staff will be documented in the record. Thanks to the above text, anyone can understand the project at a glance.

The thesis' main purpose is to provide patients with high-quality medical treatment by bringing all of the city's medical practitioners onto a single platform where everyone can easily access them and make appointments. The major purpose of this project is to get the appointment in a short amount of time and without causing any inconvenience to the patients. I created this project for regular individuals who need to see a doctor and require an appointment quickly. They may book an appointment online and obtain a checkup at their preferred time. It will be really beneficial for everyone if they can acquire this solution at the same time via the internet. So, patients would be fascinated to use internet and get all medical solution in a web-based system

1.4 Project Outcome

This is a web-based paradigm that allows consumers to make patient appointments online using any web-connected device, such as a computer, laptop, smartphone, or tablet. The system will email a booking confirmation and recorded papers for the next requirement, as well as a meeting link for an online medical consultation, when you select a date and time.

Our system's adaptability allows it to be utilised for a variety of patient and doctor services and activities, including:

- **Saving time**

  Individuals save time by not having to take time out of their busy schedule to call their medical, healthcare, or wellness provider. Staff spends a lot of time on the phone booking and can't keep appointments on track, therefore booking online saves time. For instance, a standard phone booking system takes four minutes on average to book a hundred patients, however my approach will take less time.

- **Saving money**

  The personnel in the doctor's chambers are always willing to accept money in exchange for scheduling people. It is immoral to use this method to get a faster appointment. People will be able to see all of a doctor's available slots in our system, allowing him or her to create an easy appointment for them whenever they need it without having to pay extra money to the staff.

- **Maintaining serenity**

  If someone becomes ill and has to see a doctor, they must go to the chambers and wait for the doctor to become available. The patient must also queue to obtain an appointment. As a result, a chaotic environment is feasible. If the doctor cancels the appointment due to an emergency, the patients will try to create a commotion in the area. There will be no need to line for a long time under this system, and the available doctors will be accessible to the patients without causing any inconvenience to them.

1.5 Methodology

To build a successful online platform following implementation at various stages was done:

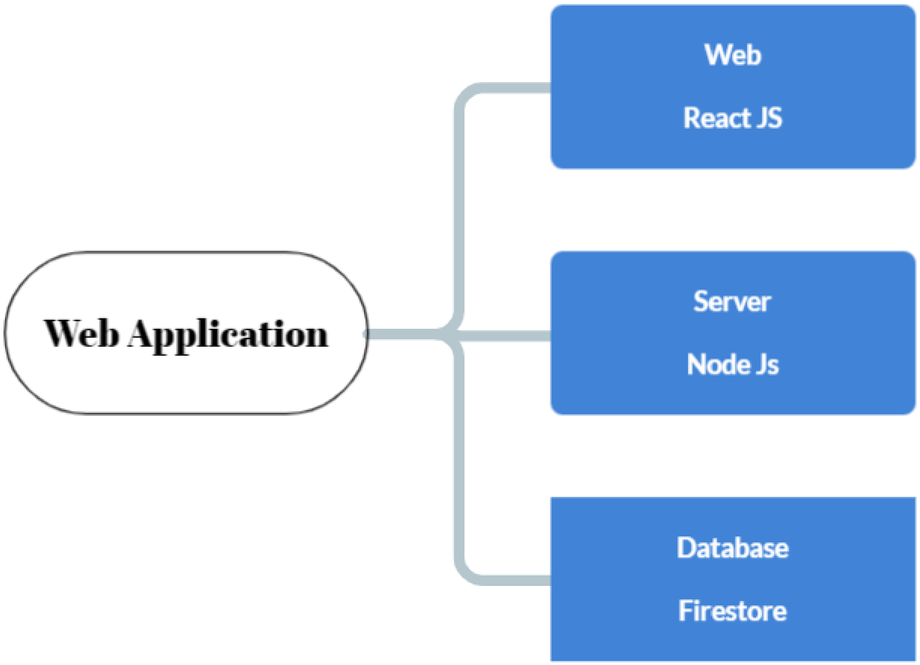


Fig 1
The above figure depicts the implementation of a web application

A. Web

Front-end development is a method for making online apps and websites have intuitive and business-focused user experiences. It measures business success by how well the website performs. An average user forms an opinion of a B2B company after spending 0.05 seconds on their website. As a result, the site's programming and structure, as well as design, functionality, and navigation experience, are critical.

- **Drive Performance**

  Visitors are irritated by slow online pages and applications and seek alternatives. Pages that load faster, on the other hand, increase visitor engagement, retention, and, ultimately, purchases. Conversion rates are higher when websites respond quickly. As a result, one of the business benefits of front-end development is improved performance.

- **Aligning Business Intent**

  The business's true objective must be reflected on the website. Customers should not be confused by the design or graphics. Instead, the positive user experience should explain why the market exists. It's all about developing pages with useful content that anticipates visitor questions. Visitors will bookmark your website and return if the information is updated on a regular basis. The overall goal of enterprise business front end design is to increase attention in a constantly changing and complex market.

- **Optimize Navigation**

  Visitors will find what they are looking for on your site thanks to intuitive navigation. It has a well-structured, clean, and well-planned site layout, as well as eye-catching images. It also assists enterprises in maximizing the business benefits of front-end development. This natural blend will allow tourists to get around quickly. It will also assist in gaining customer trust.

- **Visitor Retention**
  Well-designed interfaces create trust and confidence in your brand while also assisting you in meeting your communication objectives, resulting in increased traffic and conversion. It is one of the primary goals of developing a front-end application in an enterprise setting. Furthermore, they are able to emphasize the end-worth. User’s Sluggish websites with run-time faults, bad design, and graphics, on the other hand, will decrease user interest and make them less inclined to return.

B. Server

A web server accepts and fulfils client requests for static content (HTML pages, files, photos, and videos) from a website. Web servers just process HTTP requests and responses. The primary function of the Web server is to display site content, whereas the application server is responsible for logic and user interaction with the displayed content. The web server and the application server work in tandem, with one showing and the other interacting. The primary function of the Web server is to display site content, whereas the application server is responsible for logic and user interaction with the displayed content. The web server and the application server work together, with one showing and the other interacting.

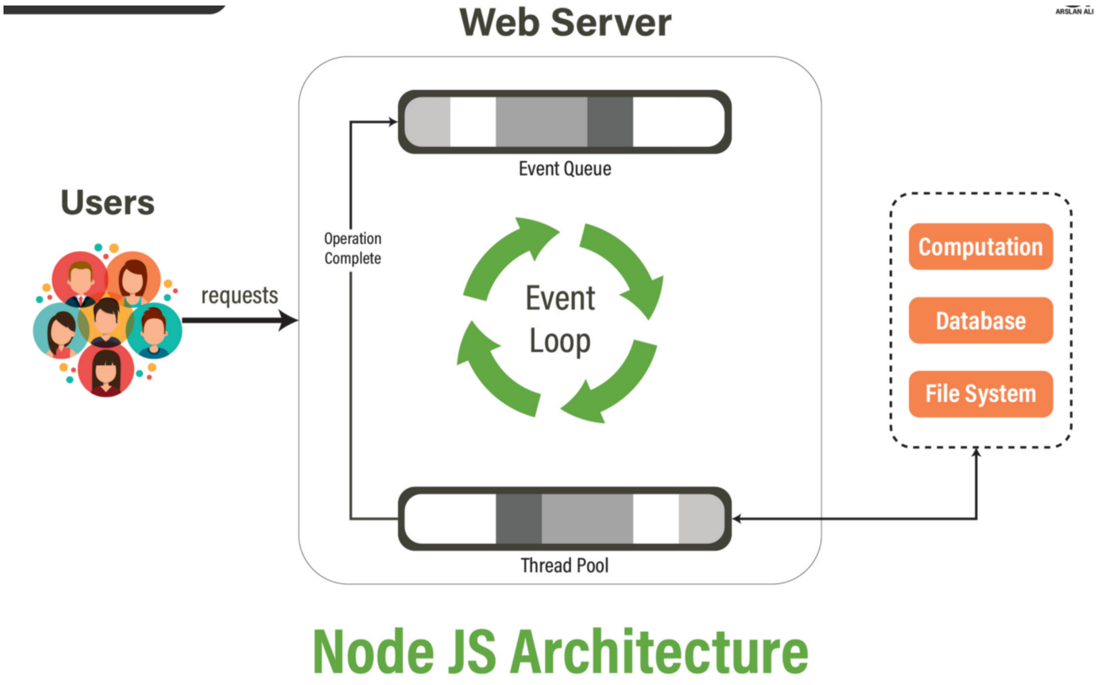


Fig 2
This picture depicts the web server structure of Node JS

(src: google.com)

C. Database

The database is crucial. A dynamic website requires primarily two components to function: the front-end and back-end of the site. Server-side programming is important for the operation of websites. It is quite important in the website's backend. Because the site's data is loaded there, it is actively fetching data from it. A database is a sort of software that is mostly used to manage website data. Its primary role is data storage, updating, creation, and deletion. Database handlers assist in the creation of databases in which only one software has access to manipulate the data. It also arranges it into tables with several rows and columns.

The importance of databases grows as the popularity of dynamic websites grows, because static websites just serve to display information to users. It allows consumers to get quick answers to their questions. It improves the accuracy and speed of the response. The database helps to improve the client-server relationship. One of its key advantages is data integrity. Because the websites may have a variety of databases. It improves the accuracy and user-specificity of the website. It allowed users to access web resources by requiring authentication. As a result, the site's security has improved. It mostly sets the tone for today's e-commerce environment.

Chapter 2

Literature Survey

We need to do some preliminary research before we can create a system. It aids in the investigation of a system's history. It also assists in detecting the defects in the current system, allowing for the necessary actions to be taken to improve the project by adding and updating new features.

My main goal for this project is to get the patient an appointment in the shortest amount of time possible. I created this project for regular individuals who require a doctor's consultation and want to book an appointment online and get their checkup at their preferred time. Instead of wasting time travelling to a clinic and trying multiple doctors to see which medicine works best, one can just go to the internet platform, select the preferred doctors from a large selection, and book in seconds.

2.1 Related works

There are some similar systems available right now, but not many and none are completely identical. Many aspects of the system are limited. I became interested in developing this system after studying a similar project. Some of my project's connected systems are listed here. I looked at a lot of websites that are related to medical health awareness, and the first one that got my eye was 'www.practo.com,' which is frequently utilised and the main inspiration for my project. In their system, users must find a doctor or hospital in a distant location and contact them to schedule an appointment. Because there is no user login option or personal profile, users are completely cut off from receiving more services in the future. For even doing the slightest booking user has to send a request only then a response will come from the website. Hence I improved my model by directly giving the access to user to book the doctor’s slot.

2.2 Comparative Studies

The internet is extremely important in our modern technological society. A Web-based system is another term for an online system. Because the world moves so quickly, there is a constant desire to communicate more quickly and effectively. There are no such things as things that come out without constraints, but we focused on doing the best we could. My major goal was to figure out what the patients' characteristics were. They are part of the system's greater community, and they deserve to get the most out of it. Both doctors and patients profit from adopting this technique.

A meeting can be scheduled in a number of ways. A person can either go to the hospital for consultation or book an appointment through a web-based system from home. To solve this problem, I developed an appointment system that improves patient satisfaction by ensuring that patients have consistent and timely access to care. As a result, I'd want to propose a system that would assist people in dealing with such a situation in their daily lives. This technique will deliver the greatest results while also saving them time.

2.2.1 Waiting time

Fernandes et al. (1994) defined waiting time as the amount of time an individual spends waiting for something to happen. Waiting time was defined as the time a patient arrived at a clinic or service point and was consulted by a doctor who had a prescription in his hand. In the past, there were two alternative techniques to determining waiting time. To begin, waiting time begins when a person arrives for an appointment and ends when the person is seen by a doctor. It starts when a person sets an appointment and ends when he is provided medication after speaking with a doctor in the second sense.

With the passage of time, long lines at doctor's clinics have become a major issue in developing countries. A block appointment system was deployed as an experiment in a South African health institution, with patients' waiting times measured for one week before and after the appointment system was adopted. Focus groups, staff interviews, and patient interviews were done as part of the trial, and the results suggested that acute medically sick patients with prior appointments had shorter wait times than those without. It has also been stated that the appointment system is useless for persons who do not see a doctor or who take routine prescriptions on a regular basis. Later research revealed that the block appointment technique benefits seriously ill patients more than everyone else.

2.2.2 Delayed Appointment

According to past research, there is a direct link between appointment lateness and cancellation. The time between when a patient requests an appointment and when the doctor sees him or her is referred to as appointment delay. Longer wait times result in more cancellations. As a result, the simplest technique for avoiding appointment cancellations or no-shows is to reduce the time between scheduling an appointment and the doctor's checkup/consultation.

The way of bridging this gap is known as open access (abbreviated as OA) or advance access policy, and it has since become widespread practise and a part of active research. In their testing, the researchers had both positive and negative results. Some practitioners were enthusiastic about OA and enthusiastically pushed its implementation, while others were skeptical.

2.2.3 Patient Appointment Management

A health-care application that manages and reduces patient waiting times is the managing patient's appointment system. Some health-care clinics use software like this, whereas others don't. When compared to medical clinics that do not use an appointment scheduling system, those that do usually have shorter wait times. Patients who must wait more than an hour for a medical examination feel ashamed and treated unfairly. The length of time patients must wait can be used to measure the quality of any medical facility. As a result, when designing a patient appointment system, it's become vital to consider things like "saving time" and "minimising idle time”. Klassen (2002) states that doctor's time and patient visits should be scheduled based on the seriousness of the situation and the patient's need, not on the arrival time.

2.2.4 Online Booking

A web-based appointment system is made up of individual components or web pages that work together to achieve a common purpose. Users can use the internet to access such systems and achieve their objectives. The internet, according to James (1999), is a technology that allows millions of people to connect over long distances and obtain access to a large amount of data. Providers have recognised the need for higher-quality care as a result of missed appointments and patient dissatisfaction.

The online appointment system has arisen as a vital output for the efficient and timely delivery of health care services, thanks to the advancement of information technology and the delicate nature of medical treatment. Appointment requests were previously made via phone, email, or fax, but as technology and the internet progressed, the tendency switched toward proper online appointment systems. A patient can book an appointment with a doctor by visiting directly to the doctor's clinic, calling, or sending an email, according to Gruca (2004). Health care facilities may now interact and communicate with their patients more effectively thanks to the internet. As a result, several healthcare facilities have begun to use online appointment scheduling systems. These kind of systems guarantee efficiency and efficacy.

2.3 Existing System

In India, private medical clinics and hospitals are getting more popular. Doctors run their own private clinics and see patients at any hour of the day or night, depending on their schedule. Some are well-known and popular, while others are quite unknown. New practitioners face a challenging situation, even if they have a solid academic background, because they are unfamiliar to most people. Patients, on the other hand, have a difficult time locating and selecting a nearby physician. Indian citizens and patients face challenges such as missed appointments, long lines, and the intangibility of medical records.

2.4 Determination of Problem

- **New practitioners face a challenge**

  Opening a private practise can be tough for new practitioners. If a new doctor wants to open a clinic, for example, there is now no platform through which or from which others might learn about it.

- **Difficulty for a newcomer**

  Any newbie to the city must figure out which doctor or clinic is the best or closest to see for a medical checkup.

- **Waiting Queues**

  To acquire a doctor's consultation, patients must visit the clinic twice: once to arrange an appointment and again for the checkup. Some patients visit the doctor without scheduling an appointment, resulting in a lengthy wait. There's a potential they won't get a doctor's consultation even after a long wait.

- **Traditional Medical Records**

  Patients must bring their medical records or files to every doctor's appointment. On occasion, they too misplaced their files.

2.5 Proposed System

I attempted to address all of the aforementioned difficulties with India's current system in "Healthcare Platform for Online Consultation." My ambition was to build a platform that would bring all practitioners and patients together. This application is open to all practitioners, whether new or seasoned. This allows customers to research and visit any doctor they like. Using this application, the user will be able to learn about and access the professional profiles of all registered doctors from all disciplines.

A doctor's profile includes information such as professional experience, practising licence, educational background, clinic hours, working days, clinic accessibility, and other patient ratings. Users will be able to book appointments from the convenience of their own homes. The user can choose a doctor who is nearby, has more professional experience, or has had positive feedback from prior patients. New practitioners can join the platform right away and start taking appointments without the need for costly promotion. When registering as a patient, the user can schedule a time slot and video call with the doctor without having to visit the clinic, which is especially useful if the user lives out of town and needs regular exams.

2.5.1 Appointment System for Patients

Time Appointments for patients have been set for years. Because it was felt that the doctor's time was more precious than the patient's waiting time, the appointment system was designed largely to reduce doctor idle time. It was eventually discovered, however, that reducing the patient's waiting time is just as significant as saving the doctor's time. As a result, both the doctor's idle time and the patient's waiting time are given equal weight when developing an appointment system. Patient scheduling results in higher-quality health services, less doctor and nurse idle time, and shorter patient wait times. Any patient who is registered can book a slot from the desired doctor for coming seven days and easily avail the medical treatment by sitting at home.

2.6 Challenges

Every job has some difficulties. As a result, this project also faces a modest number of challenges such as,

1. Because it is a web-based system, individuals must book appointments online using any web-connected device and if they do not, the programme will be rendered worthless.

2. Availability of slots can be limited, since the app will be showing slots for Monday to Friday from 9:00am to 5:00pm, so time restriction is one such challenge. If a patient wants urgent consultancy at midnight then that is not provided.

Chapter 3

System Development

3.1 Technical Requirements

The purpose of this project is to construct an online appointment scheduling web application. It is vital that the user understands how this application works as well as the technology used to complete this project. All phases are thoroughly described in order to provide a comprehensive picture of the system and aid comprehension.

Make sure that your computer can support the system requirements before installing any software or hardware. Computers, laptops, tablets, phones, and other internet-connected system components are all necessary. During development, we used the following software tools and platforms:

3.1.1. Web Front End

A professional Front-End developer will be familiar with the entire web development process, from conception to deployment. It will also be well-versed in industry trends as well as the most recent software applications and languages. In addition to technical abilities, the user interface should be straightforward and pleasing to the eye. It is the initial consideration for a user when deciding whether or not to proceed with the programme. The UI for this application is built with React.JS.

- **React.JS**

  ReactJS is a front-end JavaScript library. This JS package is used to create interactive and fast user interfaces for mobile apps and websites. React is a "view" that determines how an application appears and functions (towards the viewer side). React.JS is a JavaScript library for building user interfaces, particularly single-page apps. For online and mobile apps, React JS is used to handle the view layer. React JS enables the creation of reusable user interface components.

  ReactJS was invented by Jordan Walke, who was working on Facebook advertisements at the time. It is compatible with other JS libraries, such as AngularJS. It separates the user interface into various numbers of components, each with its own function and property. JSX is a JS syntactic extension that React uses to express how the user interface should appear. JSX is used for coding.

  JS + HTML = JSX

React makes creating interactive user interfaces a breeze. Create simple and appealing views for each component in your app or website, and react will assist you update and render the appropriate components as your data changes.

React JS is a programming language, not a framework. It's simply a library created by Facebook to address various issues with the creation of UI components. Developers can use React JS to create massive web applications that can alter data without having to reload the page. The primary goal of React JS is to make web pages load quickly and easily. It only works on application user interfaces. Other JavaScript libraries or frameworks, such as Angular JS, can be utilised with React JS in MVC. Each component in a React application is responsible for interpreting a short, reusable portion of HTML. Simple building blocks can be used to create complex applications by nesting components within other components. A component may also keep track of its own internal state; for example, a Tab List component may keep track of the currently open tab.

React JS has also been used to create popular social media sites such as Instagram, Facebook, Skype, Pinterest, Uber, and Netflix.

React Advantages

- Makes advantage of the virtual DOM JavaScript object. Because JavaScript virtual DOM is faster than traditional DOM, app speed will improve.

- It works with different frameworks and may be used on both the client and server sides.

- Component improves readability, making it easier to manage larger apps.

React Framework: Material UI

This is a collection of components made using Google's material design principles. Many adjustable and accessible UI widgets are included in Material UI. Because the components are self-contained, they will only inject the styles that they need to display, which can improve the performance of your application.

This library is supported by a strong community and active maintainers. It currently has more than 65k stars on GitHub, making it one of the most well-known component libraries available.

Material UI is a light, user-friendly, and simple design and layout that makes it simple to create stunning apps. It may brag about having a consistent and easy-to-use interface that represents the Google design team's considerable understanding. This library will be the right answer for you if you want to create a consistent, light, and appealing interface in a short period of time without losing performance or accessibility.

- **Redux**

For JavaScript programmes, Redux is a state management system. To put it another way, Redux makes it simple to manage your applications. It assists you in managing your data and the appearance of your applications. Managing the state of the Model-View architecture in small applications is simple. However, this can be particularly difficult for large applications. Redux simplifies state management by dealing with the difficulties.

A unidirectional data flow is built into React. There are some basic principles that must be adhered to:

1. Only one root state exists.
2. This root state can only be updated by particular events such as user interactions and network changes.
3. You can only conduct one update at a time.
4. View updates to reflect changes in the state.

Redux is built on three pillars**: Store, Action**, and **Reducers**.

State - A store is an object that holds the state tree of an application. Because the composition happens at the reducer level, a Redux app should only have one store.

getState() : gives the current status of the state

dispatch() : the only method to update the application state

subscribe() : adds change listener to state

unsubscribe() :It comes in handy when you don't want to call your listener method every time the state changes.

Actions - An action is a simple object that conveys a desire to alter the current state. Actions are information payloads that transfer data from your app to your store. Any data, whether through user interface events or network callbacks, must eventually be dispatched as actions. A type field must be present in actions to identify the type of action being executed.

Reducers - Reducers define how the state of the application changes in response to store activities. Actions merely explain what happened, not how the state of the programme changes. A reducer is a function that takes two states and an action and returns a new state with the action. Combine all of the app's reducers into a single index reducer with the combineReducers() function. This makes maintenance a lot easier.

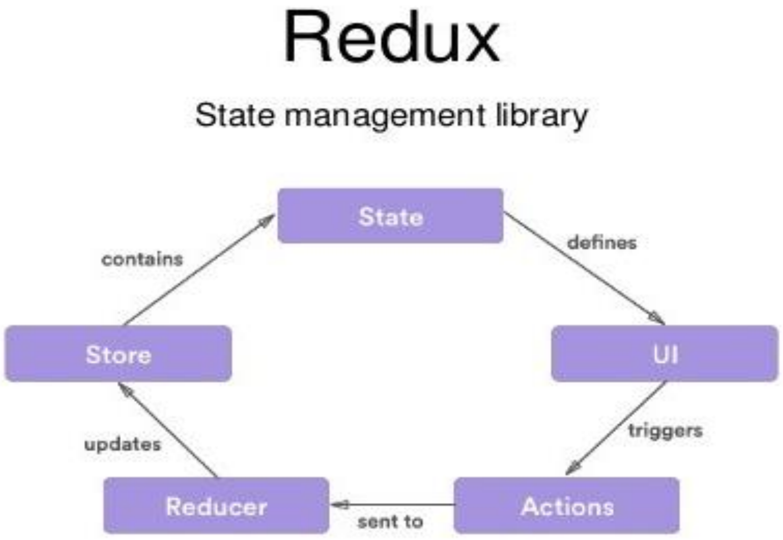


Fig 3
This picture depicts the state management library of redux

(src: google.com)

- **React with Redux**

When a value is used by numerous components in your React project, Redux is usually employed. If you only need to use a state in one component and no other components are depending on it, Redux isn't necessary. Redux keeps track of a whole application's state in a single immutable state tree (object) that can't be altered directly. A new item is formed when something changes (using actions and reducers). Redux makes it simple to keep track of your application's state. Another way to look at it is that it assists you in managing the data you show and how you respond to user activities. Because having a state in a data-handling component is convenient and requires you to think about the structure.

To some extent, attaching state to the props of a child component works. This allows your components to communicate with one another, but when you have a lot of data that is/could be utilised everywhere, it's much better to use a solution like redux to provide app-wide access. A lot of props, ugly jsx-code, and debugger-hell are the alternatives. Redux is also a clear technique for centrally managing API/database answers. Those calls should not be sprinkled all across your components.

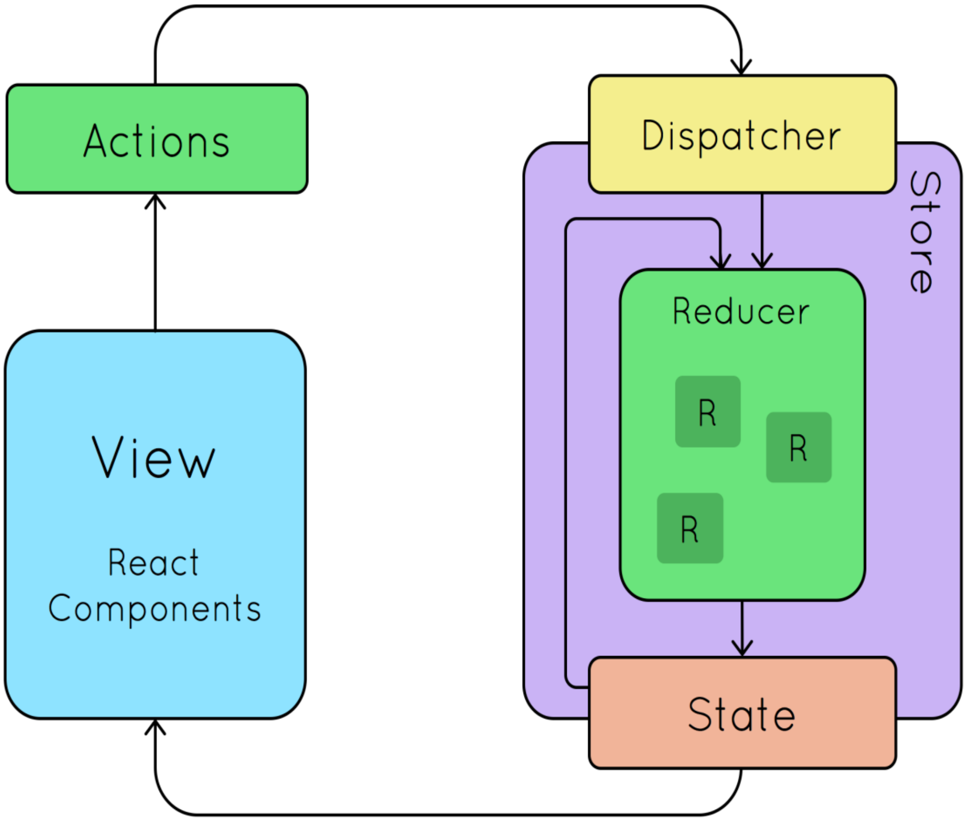


Fig 4
This picture depicts the state management library of redux with react components

(src: google.com)

3.1.1. Web Back End

Back-End The task of web development is to coordinate data transfer between the server and the users. The important priorities are the creation of server-side logic, the defining and management of the central database, and ensuring high performance and responsiveness to front-end queries. It's also in charge ofa integrating the application's front-end elements created in the UI section. To merge front-end and back-end technologies, a fundamental understanding of front-end technologies is also required.

- **Node JS**

  By storing, processing, and distributing web pages to the client, a web server facilitates communication between the client side and server side. A request for any specific resource is generated by the server through HTTP communication where in return server provides the requested resource.

Node.js shines in real-time online applications that use web sockets and push technology. We now have online applications with real-time, two-way connections, where both the client and server can initiate communication, allowing them to freely exchange data, after more than 20 years of stateless-web based on the stateless request-response paradigm. This is in sharp contrast to the conventional online response paradigm, in which the client is always the one to initiate communication. Furthermore, it's completely built on the open web stack (HTML, CSS, and JS) and runs on port 80.

- **Express JS**

ExpressJS is a NodeJS framework that makes it easier to build server-side web applications. Its features include simplicity, minimalism, adaptability, and scalability, and it inherits NodeJS' performance. In a word, ExpressJS was the NodeJS equivalent of Bootstrap for HTML/CSS and responsive web design.

It simplified NodeJS writing and provided programmers with new options to enhance their server-side development. ExpressJS is by far the most well-known NodeJS framework, to the point where most people refer to NodeJS as NodeJS+ExpressJS.

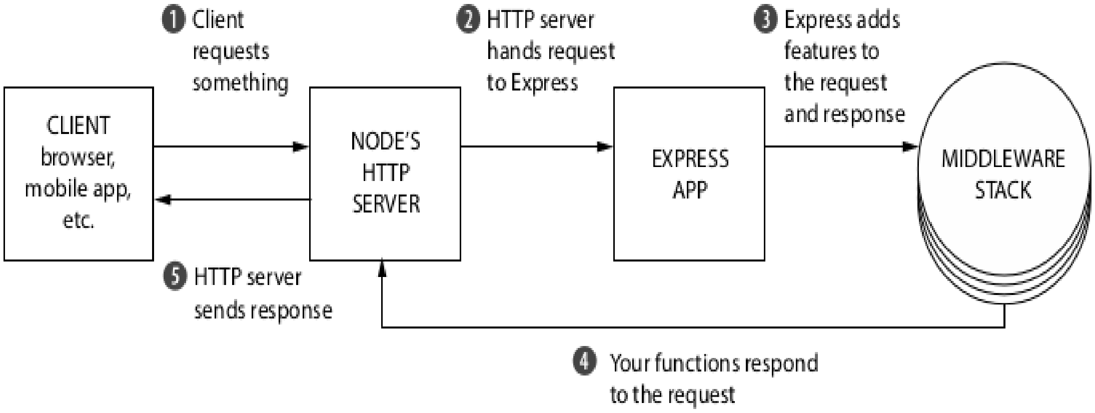

1 Client requests something
2 HTTP server hands request to Express
3 Express adds features to the request and response
CLIENT browser, mobile app, etc.
NODE'S HTTP SERVER
EXPRESS APP
MIDDLEWARE STACK
5 HTTP server sends response
4 Your functions respond to the request

Fig 5
This picture depicts the overview of express.js with node server

(src: google.com)

- **Firestore**

Despite being newer, Cloud Firestore does not replace Firebase Real-time Database. Cloud Firestore is a highly adaptable and scalable NoSQL cloud database. It is used to store and sync data for client and server-side programming. For mobile, web, and server development, Google Cloud Platform and Firebase are used. It, like the Firebase Real-time Database, is constantly syncing our data with the client app via real-time listeners. By offering offline support for mobile and web, we can create responsive apps that run regardless of network latency or Internet access. Cloud Firestore also works in tandem with Google Cloud Platform and other Firebase products, including cloud features.

Cloud Firestore, a cloud-hosted NoSQL database, is easily accessible through the native SDK in our iOS, Android, and web apps. In addition to REST and RPC APIs, Cloud Firestore has native Node.js, Java, Python, and Go SDKs. After leveraging Cloud Firestore's NoSQL data model, we may store data in documents using field mappings for values. Collections are storage containers for documents. Our data is organised and queried using these containers. Documents can include a wide range of data types, from basic characters and integers to complex nested structures. Within a document, we may create sub-collection, resulting in a hierarchical data structure that expands with our database. Because the Firestore data architecture works with any data structure, it's ideal for my needs.

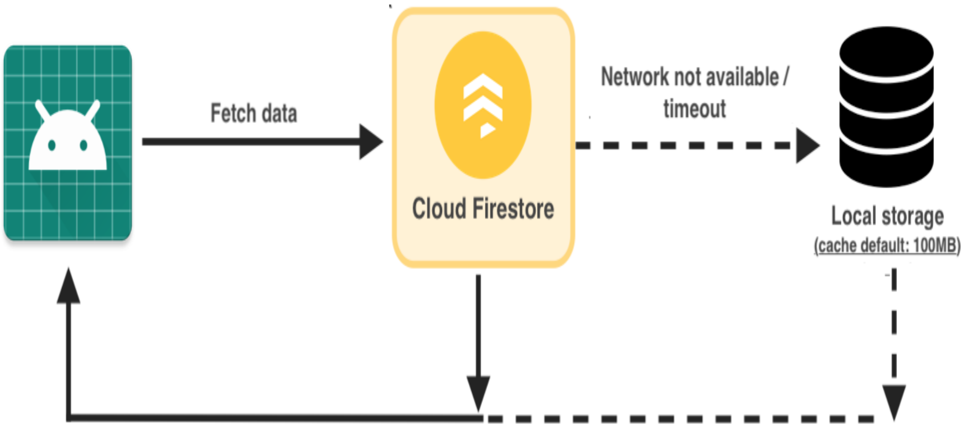


Fig 6
This picture depicts the working of firestore

(src: google.com)

- **Socket.io**

Socket.IO is a JavaScript-based real-time web application library. It enables real-time and bidirectional communication between web clients and servers. It consists of two parts: a client-side library for the browser and a server-side library for node.js. Both components have APIs that are almost identical. Web sockets are used by Socket.io to transfer data across users who are all connected to the same web server. There are no negotiation mechanisms with web sockets, and the connection remains open as long as the users are registering for service with the web server. As previously stated, the payload is substantially smaller than that of the http/https protocol.

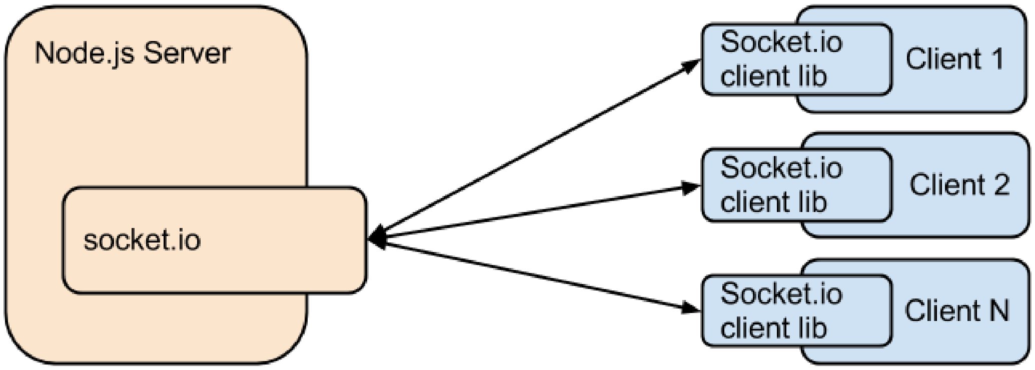


Fig 7
This picture depicts the working of socket.io in node server with multiple clients
(src: google.com)

- **Cloudinary**

Through easy-to-use REST APIs, Cloudinary's Programmable Media solution allows you to programmatically add picture and video upload, transformation, optimization, and delivery capabilities to your apps. Cloudinary also provides a number of SDK libraries that wrap the APIs, allowing you to easily integrate Programmable Media features into your existing application code.

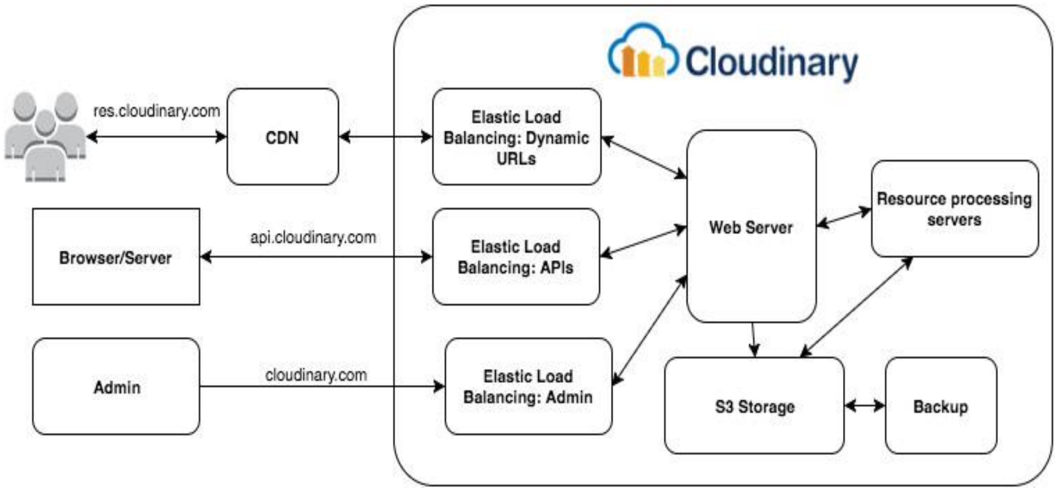

Cloudinary
res.cloudinary.com
CDN
Elastic Load Balancing: Dynamic URLs
Browser/Server
api.cloudinary.com
Elastic Load Balancing: APIs
Web Server
Resource processing servers
Admin
cloudinary.com
Elastic Load Balancing: Admin
S3 Storage
Backup

Fig 8

This picture depicts the working of cloudinary api in server side

(src: google.com)

3.2 Development of the Algorithm

Web development approaches exist beneficial for a huge target of audience as they can access the visible form of the application and find solution to their daily life problems. The UI of any website is kept easy and user friendly so any user irrespective of age, gender etc can access it. To obtain this simple result three parts are followed for the development of algorithm of a full stack project. First includes the UI part which is called the front end of any web application, it is the utmost important part as all visuals are present in this part so to target the audience this part needs to be most innovative. Next is the back end part where server runs and all the data is fetched and stores effectively. Last is the database part which is the core of any web application as so much data is available to us on daily basis and the correct form of storage is very necessary. Hence, whatever data reaches the web application needs to be stored in a simple and effective manner for further use.

Below image gives an insight of these three parts are connected, in the image the database storage is given as Mongo db where in my project I have used firestore by firebase.

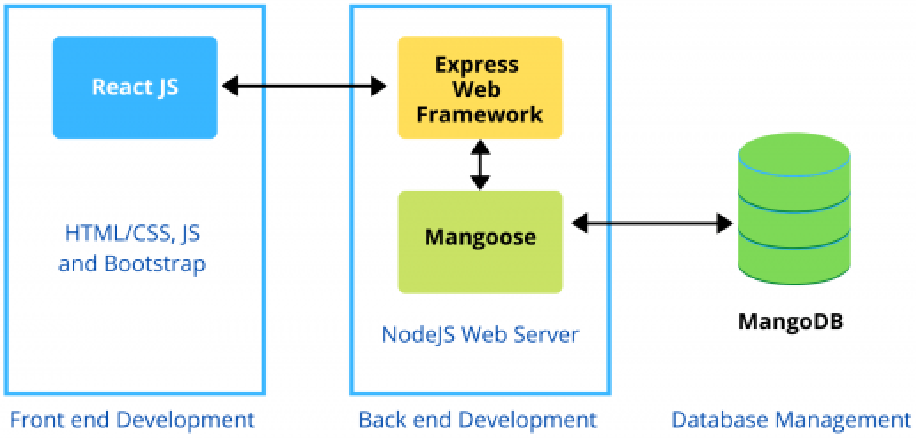


Fig 9
This picture depicts the basic working of a web application

(src: google.com)

Chapter 4

Performance Analysis

In India, I am introducing a web-based appointment system for arranging appointments and preserving medical records. Users may sign up online, search for a nearby doctor, and make an appointment all while sitting at home using a computer browser. Two types of actors utilise this system: user actors (patients) who may register, search for a doctor, and book an appointment. Administrative actor (doctor/physician) who may utilise a username and password to log in, accept patient appointments, and update medical records after each visit. This chapter goes through the functionality and other elements in greater depth.

The following are the system's primary characteristics:

- Every user will have their own unique profile with valid information. Doctors are also divided into specialties (such as obstetrics and gynaecology, urology, and internal medicine).

- By logging on to the site and visiting his or her own page, the user (patient) can view, edit, or cancel appointments while the doctor will have access to his or her appointment’s schedule and also be eligible to cancel it.

4.1 Use case

To demonstrate the system's better vision and functionalities, use case diagrams are selected. The use case diagram is used to record and specify the system's needs and functionality. Users can better understand how they interact with the system by using use case diagrams.

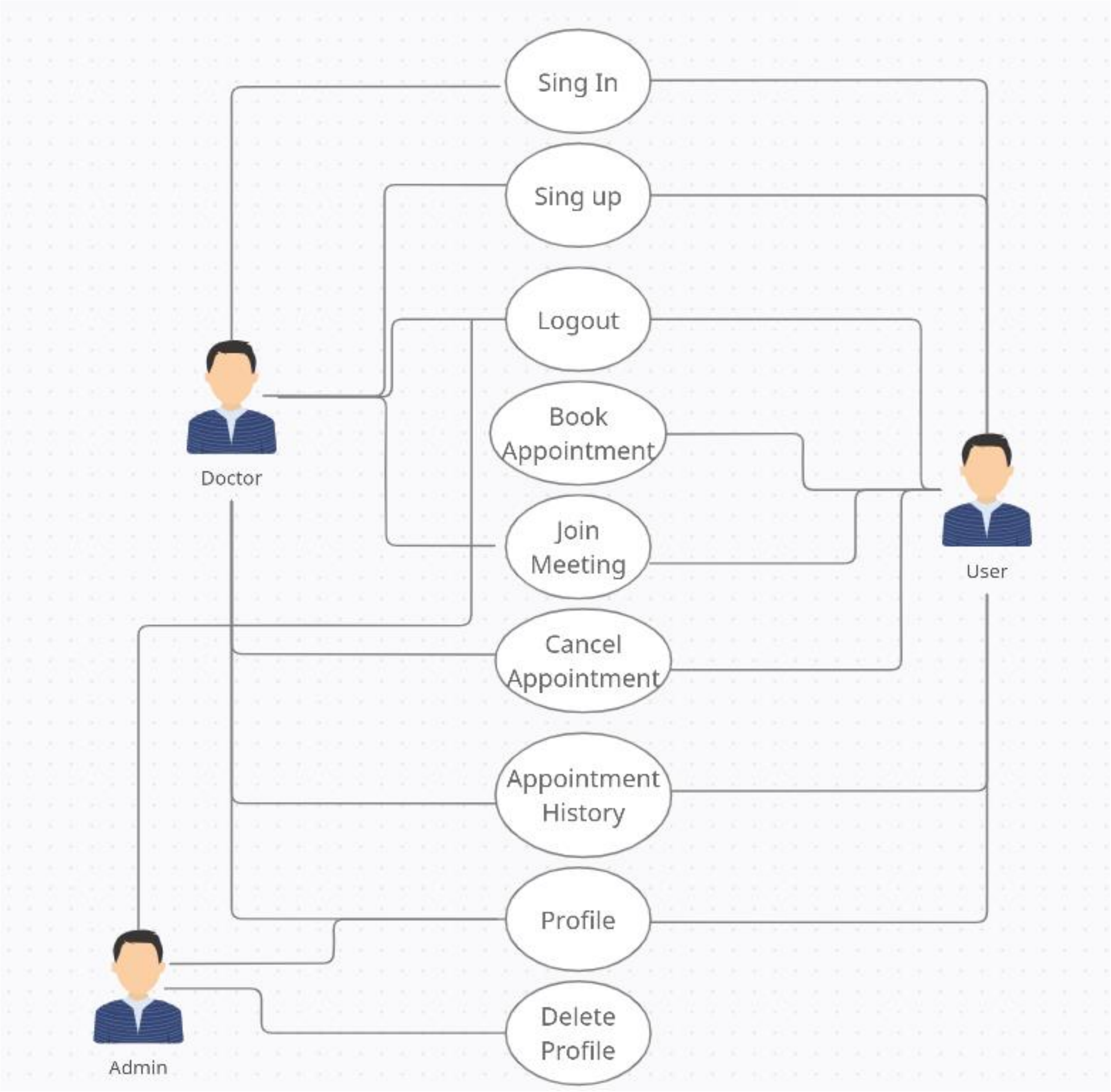


Fig 10
Use case model

4.1.1 User use case

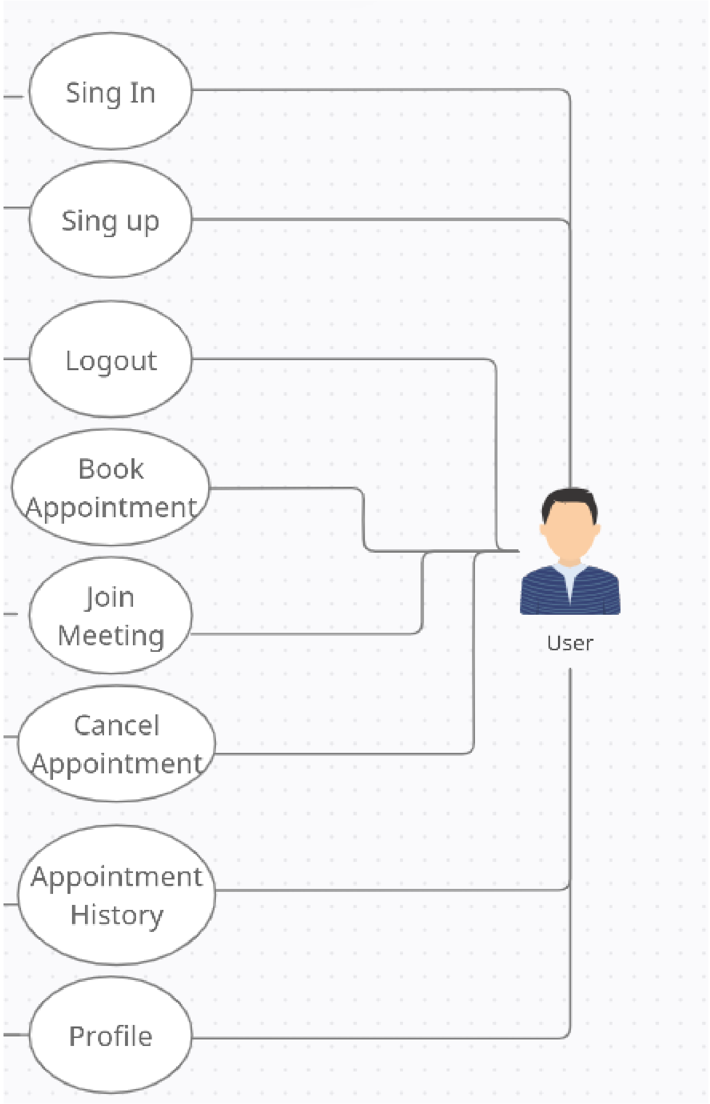


Fig 11
Use case model for the patient

- **Use-Case 1: Sign up**

  Primary-actor: Any user or patient

  Description: User needs to register before entering the web app

Precondition: All mandatory fields must be filled

Basic use-case flow: Registration will only be done once user has updated all his information and successfully passed through the authorization, following information is required for a user to register

- First name
- Last name
- Email
- Password

Main scenario: Initially an authorization page will come asking for user to sign up where a new user can register then only he/she can pass through the authorization page by successfully logging in. User need to fill the sign up form with all valid details besides which he/she can only enter the platform. User needs to provide authentic information. Only when the form is validated and credentials match with each other the user will be able to sign in else error will be displayed on the screen.

Exception: Invalid details entered in registration form

- **Use-case 2: Logging in**

Primary-actor: Any user or patient

Description: User needs to enter valid credentials which will be authenticated before the user can enter the platform.

Precondition: Registered user can only sign in

Basic use-case flow: Username should match with password for authentication

Main scenario: Once a user has successfully registered, he/she has to login to get inside the dashboard page of the web application. Only authenticated credentials would take the user to the main home page else error would be displayed on the screen saying invalid credentials.

Exception: Credentials invalid or unmatched username and password

- **Use-case 3: Finding a Doctor**

Primary-actor: Any user or patient

Description: Registered user can look for a doctor as per the requirement, user will be accessible to list of all doctors as well as list of doctors divided by specialization

Precondition: Only successfully logged in user can find

Basic use-case flow: User can search the doctor based on the specialization

Main scenario: Whenever a user has successfully logged in. On the homepage user can fid an option of Doctors where a list of all the doctors will be available for the user. Apart from the common list, user will also be able to find doctors categorized by their specialization. So the user can easily find the doctor as per his or her medical emergency

Exception: There might be no available doctor for a specific specialization

- **Use-case 4: Book Appointment**

Primary-actor: Any user or patient

Description: Once the desired doctor is selected by user, he/she will be able to book an available slot under the doctor's profile

Precondition: Only successfully logged in user can book a slot

Basic use-case flow: User will be able to book a slot under the selected doctor based on the availability of time

Main scenario: Once the user has selected a doctor based on hi/her medical requirement, time slots differentiated by twenty minutes will be displayed to the user under the doctor's card. User will be able to book any available slot in offline and online mode. For online booking user will get a meeting link and for offline booking a pdf token along with otp will be given to the user.

Exception: No slots might be available under the slected doctor

- **Use-case 5: Appointments**

Primary-actor: Any user or patient

Description: Successfully logged in user will also have access to all his previous records including upcoming and past appointments.

Precondition: Only successfully logged in user can access this page

Basic use-case flow: Under my profile section, user can look into appointments page where upcoming appointments as well as history of previous appointments will be available to the user.

Main scenario: Once a user goes to my profile page, user can see a list of upcoming appointments where he/she will have an option to either join the meeting link for online appointment or download the pdf containing otp for offline appointment. The user will also get an option to cancel upcoming appointments. History of all his/her past appointments will also be available to the user including cancelled appointments.

Exception: Server can be slow which might delay the data being called by api

- **Use-case 6: Logging out**

Primary-actor: Any user or patient

Description: Any user can log out from his or her profile

Precondition: Only successfully logged in user can log out

Basic use-case flow: User can easily log out by clicking the log out button

Main scenario: Once the user clicks on a log out button, user will be successfully logged out and bought to the authorization or login page.

Exception: No exception

4.1.2 Doctor use case

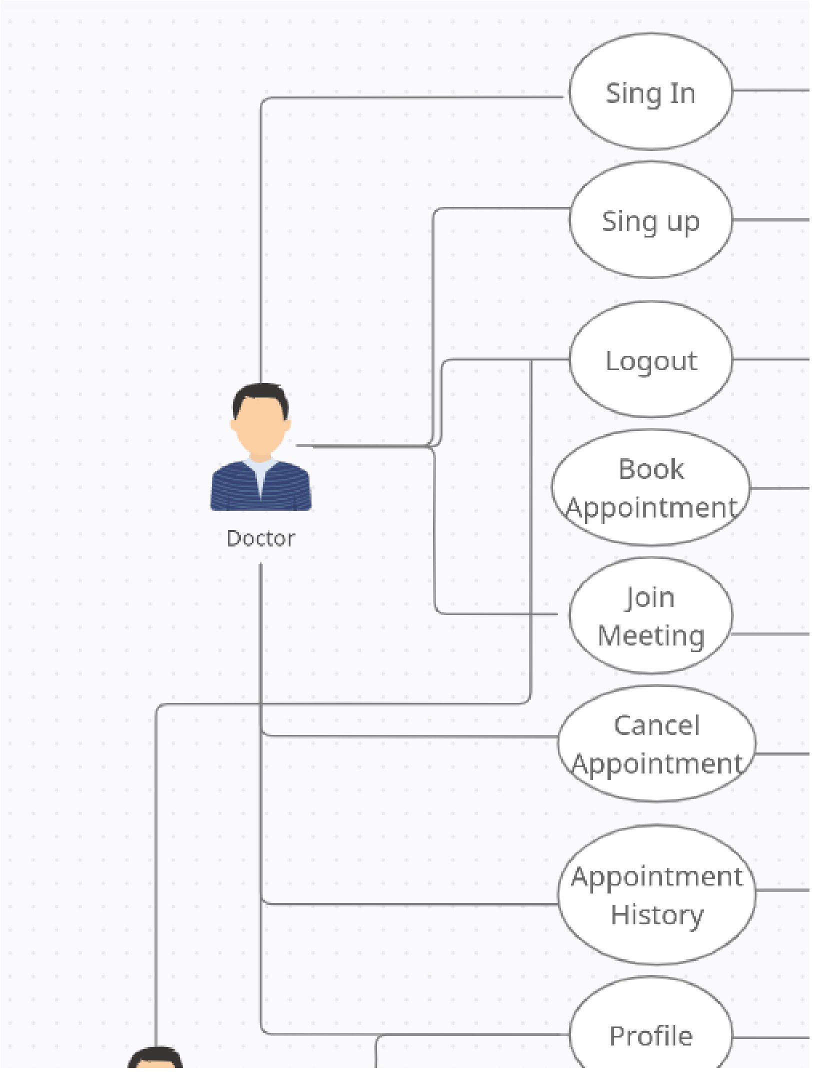


Fig 12
Use case model for the doctor

- **Use-Case 1: Sign up**

  Primary-actor: Any user or patient

  Description: User needs to register before entering the web app

  Precondition: All mandatory fields must be filled

  Basic use-case flow: Registration will only be done once user has updated all his information and successfully passed through the authorization, following information is required for a user to register

- First name
- Last name
- Email
- Password
- Qualification
- Specialization
- Address
- Pin code
- Consultation fee

Main scenario: Initially an authorization page will come asking for user to sign up where a new user can register then only he/she can pass through the authorization page by successfully logging in. User need to fill the sign up form with all valid details besides which he/she can only enter the platform. User needs to provide authentic information. Only when the form is validated and credentials match with each other the user will be able to sign in else error will be displayed on the screen.

Exception: Invalid details entered in registration form

- **Use-case 2: Logging in**

  Primary-actor: Any user or patient

  Description: User needs to enter valid credentials which will be authenticated before the user can enter the platform.

  Precondition: Registered user can only sign in

  Basic use-case flow: Username should match with password for authentication

  Main scenario: Once a user has successfully registered, he/she has to login to get inside the dashboard page of the web application. Only authenticated credentials would take the user to the main home page else error would be displayed on the screen saying invalid credentials.

  Exception: Credentials invalid or unmatched username with password

- **Use-case 3: Appointments**

Primary-actor: Any user or patient

Description: Successfully logged in user will also have access to all his previous records including upcoming and past appointments.

Precondition: Only successfully logged in user can access this page

Basic use-case flow: Under my profile section, user can look into appointments page where upcoming appointments as well as history of previous appointments will be available to the user.

Main scenario: Once a user goes to my profile page, user can see a list of upcoming appointments where he/she will have an option to either join the meeting link for online appointment or download the pdf containing otp for offline appointment. The user will also get an option to cancel upcoming appointments. History of all his/her past appointments will also be available to the user including cancelled appointments.

Exception: Server can be slow which might delay the data being called by api

- **Use-case 4: Logging out**

Primary-actor: Any user or patient

Description: Any user can log out from his or her profile

Precondition: Only successfully logged in user can log out

Basic use-case flow: User can easily log out by clicking the log out button

Main scenario: Once the user clicks on a log out button, user will be successfully logged out and bought to the authorization or login page.

Exception: No exception

4.1.3 Admin Use Case

- **Use-case 1: Logging in**

Primary-actor: User as admin

Description: User who owns administrative rights can only login

Precondition: Valid username and password

Basic use-case flow: Username should match with password for authentication

Main scenario: Once a user has successfully registered, he/she has to login to get inside the dashboard page of the web application as an admin. Only authenticated credentials would take the user to the main home page else error would be displayed on the screen saying invalid credentials.

Exception: Credentials invalid or unmatched username and password

- **Use-case 4: Delete Profile**

  Primary-actor: User as admin

  Description: Successfully logged in user will also have access manage other users like doctors and user where he/she can view, edit and delete as per the requirement.

  Precondition: Only successfully logged in user as admin can access this page.

  Basic use-case flow: User or admin can view, edit or delete records including profiles and appointments.

  Main scenario: Once a user is successfully logged in as admin he/she will have the right to manipulate the database as per the requirement. User will have full access to each profile.

  Exception: No exception

- **Use-case 4: Logging out**

  Primary-actor: User as admin

  Description: User can log out from the platform

  Precondition: User can easily log out by clicking the log out button

  Basic use-case flow: User can easily sign out

Main scenario: Once the user clicks on a log out button, user will be successfully logged out and bought to the authorization or login page.

Exception: No exception

4.2 Flow of Algorithm

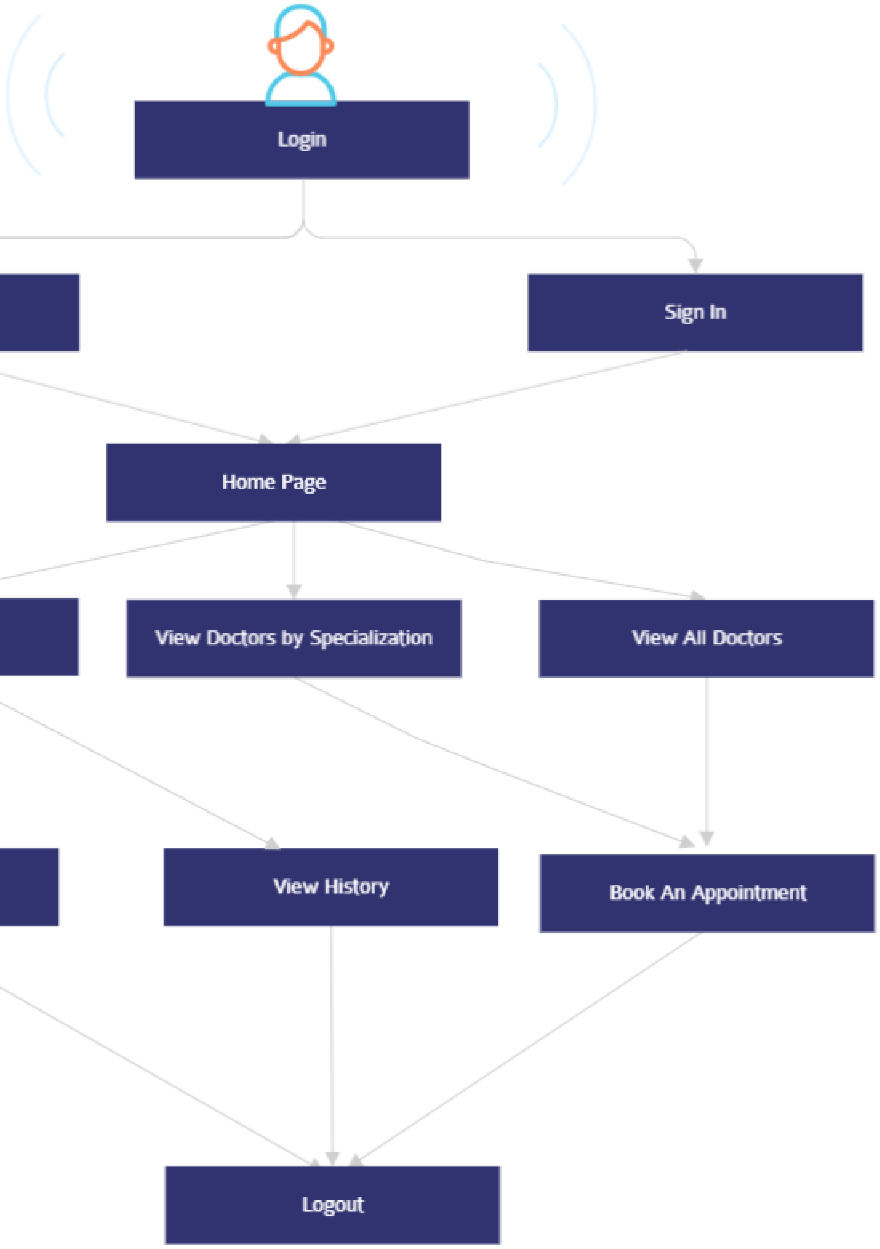


Fig 13
Activity Diagram of User/Patient

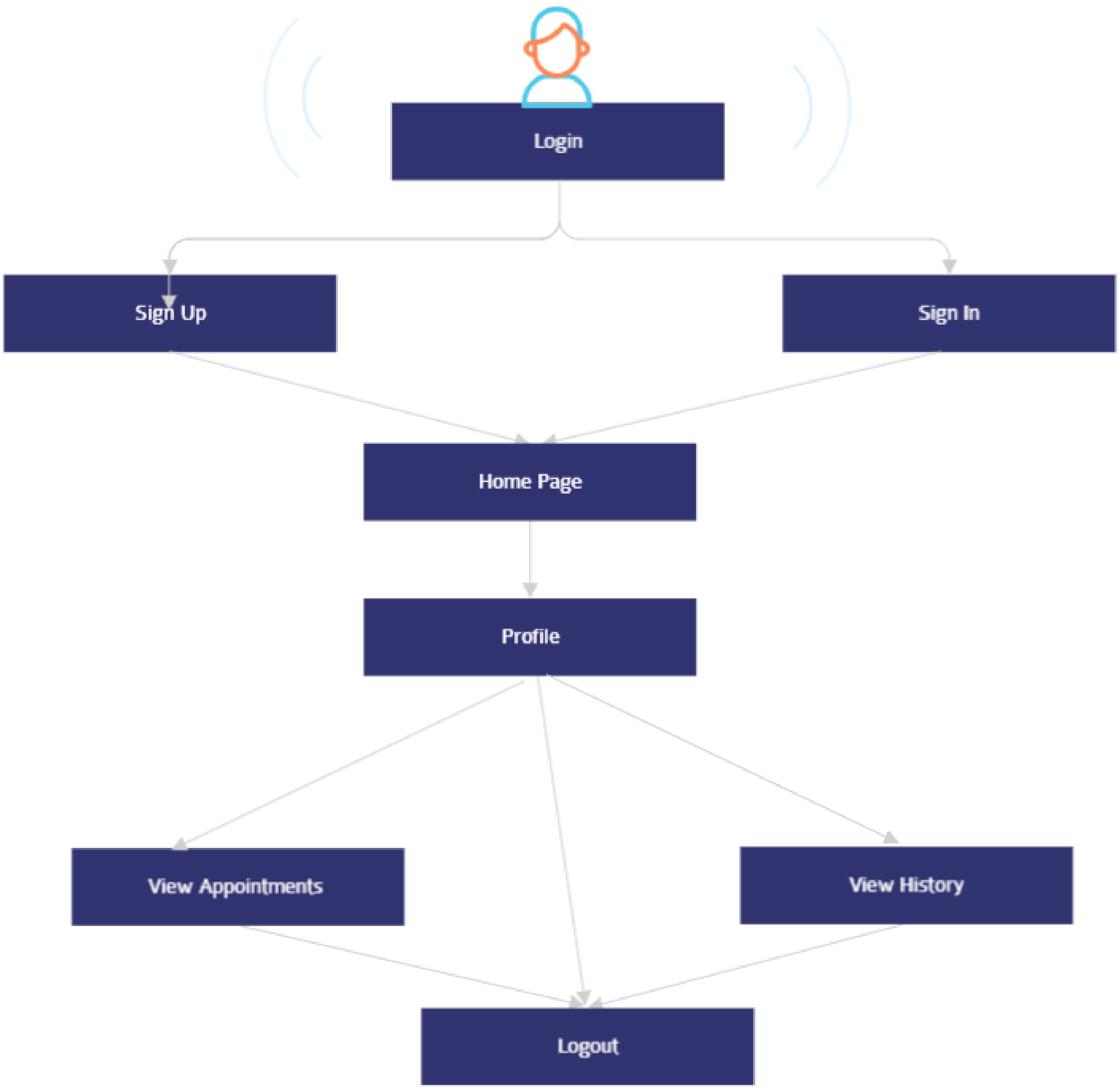


Fig 14
Activity Diagram of User/ Doctor

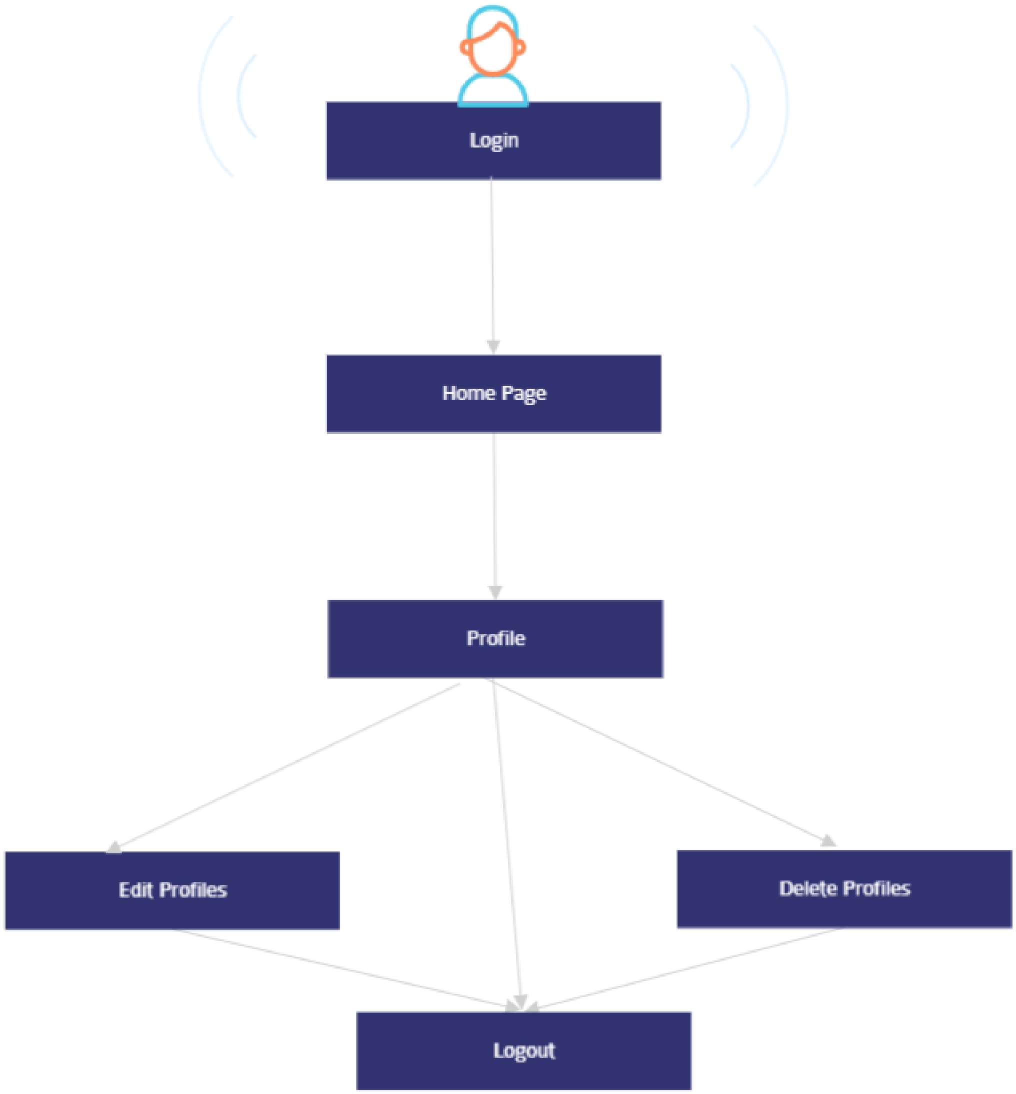


Fig 15
Activity Diagram of User/ Admin

Chapter 5

Results

5.1 Implementation Results

In today's system, interactions are common. The capacity of a system to be dynamic and interesting for users is crucial. Making a system interactive is also critical. As previously said, I want to create a user-friendly and dynamic platform. It should be done in such a way that the user is drawn in.

Below screenshots depicts the working model of my website:

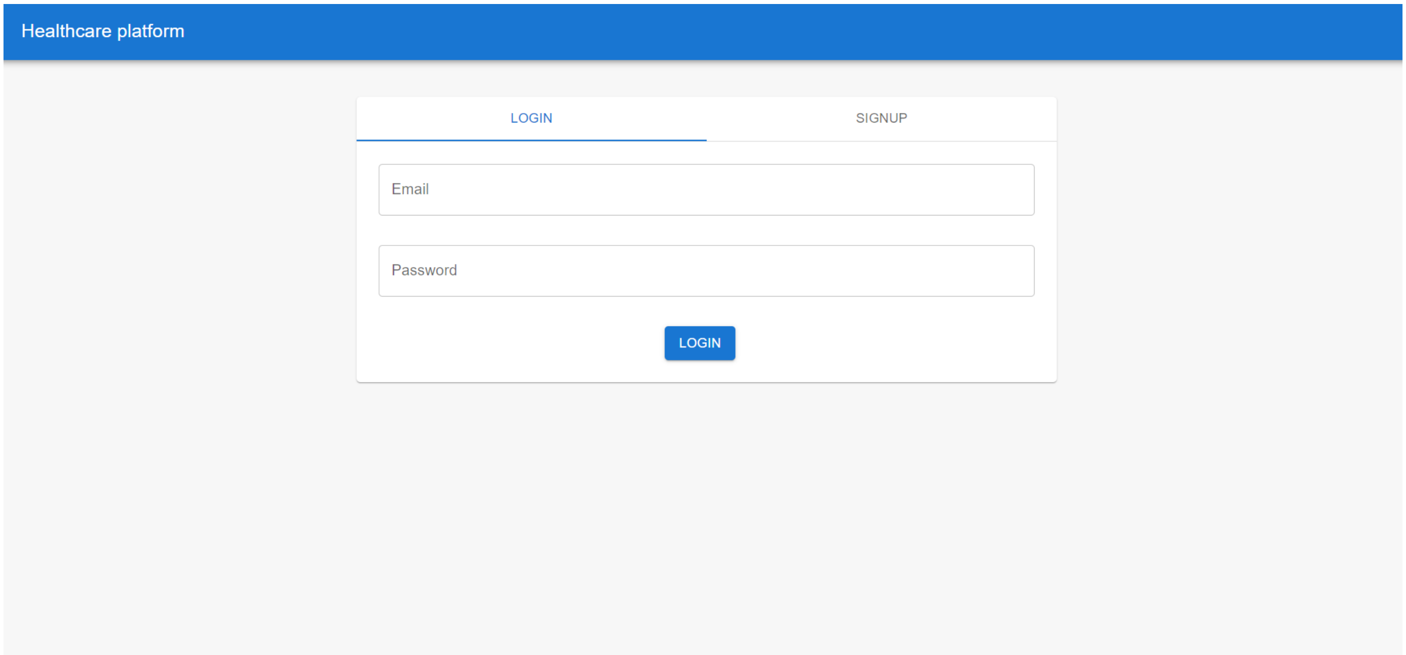


Fig 16
Authorization Page

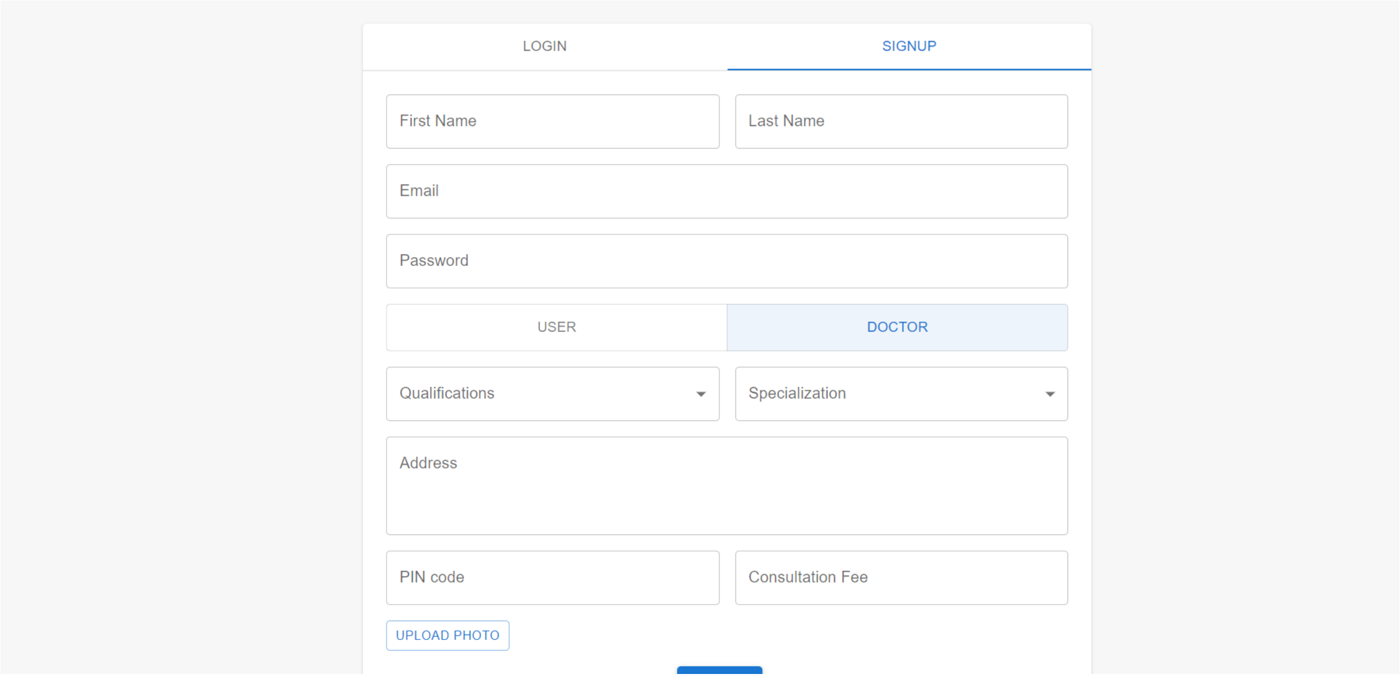
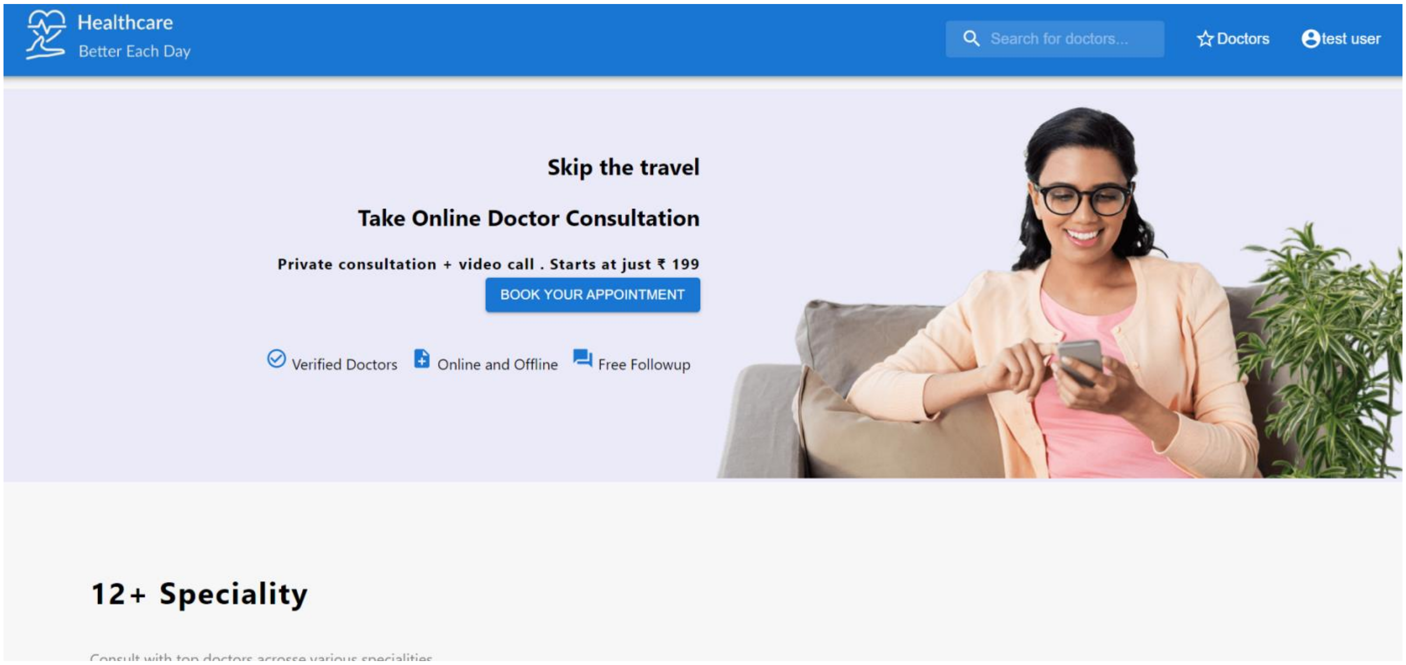


Fig 17
Registration Page



Skip the travel

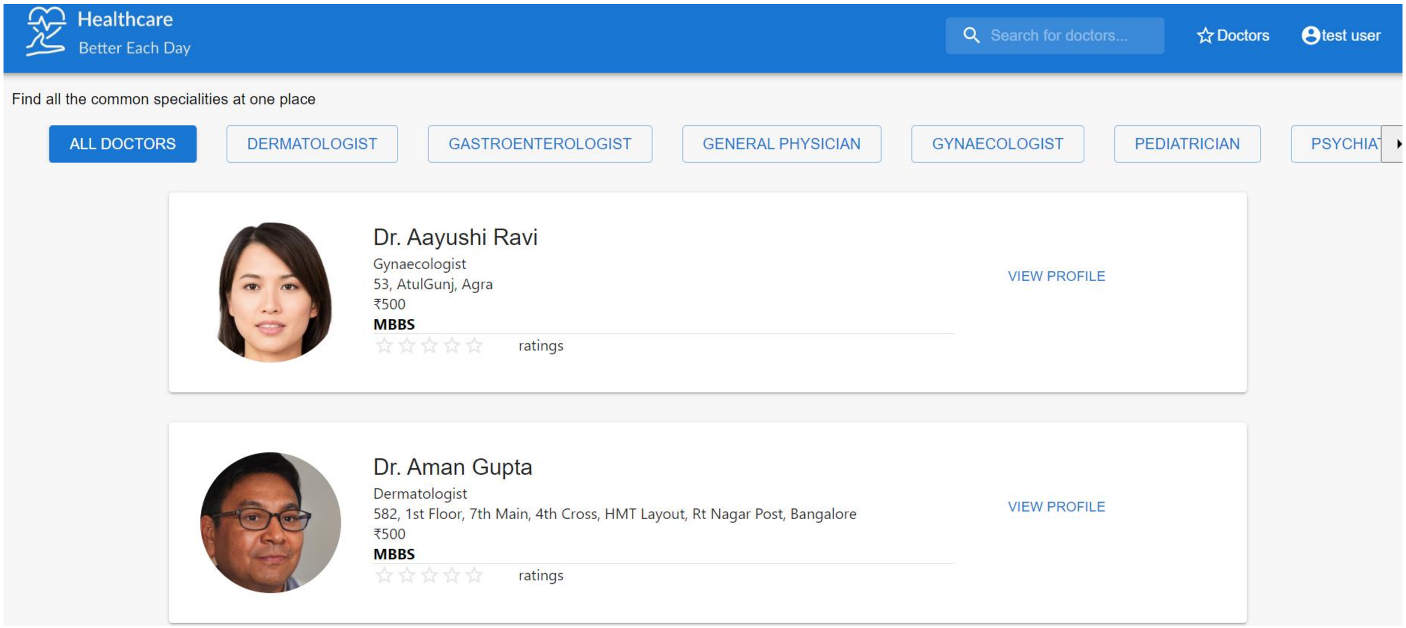
Take Online Doctor Consultation

Private consultation + video call . Starts at just ₹ 199

12+ Speciality

Consult with top doctors acrosse various specialities

Fig 18
Home Page

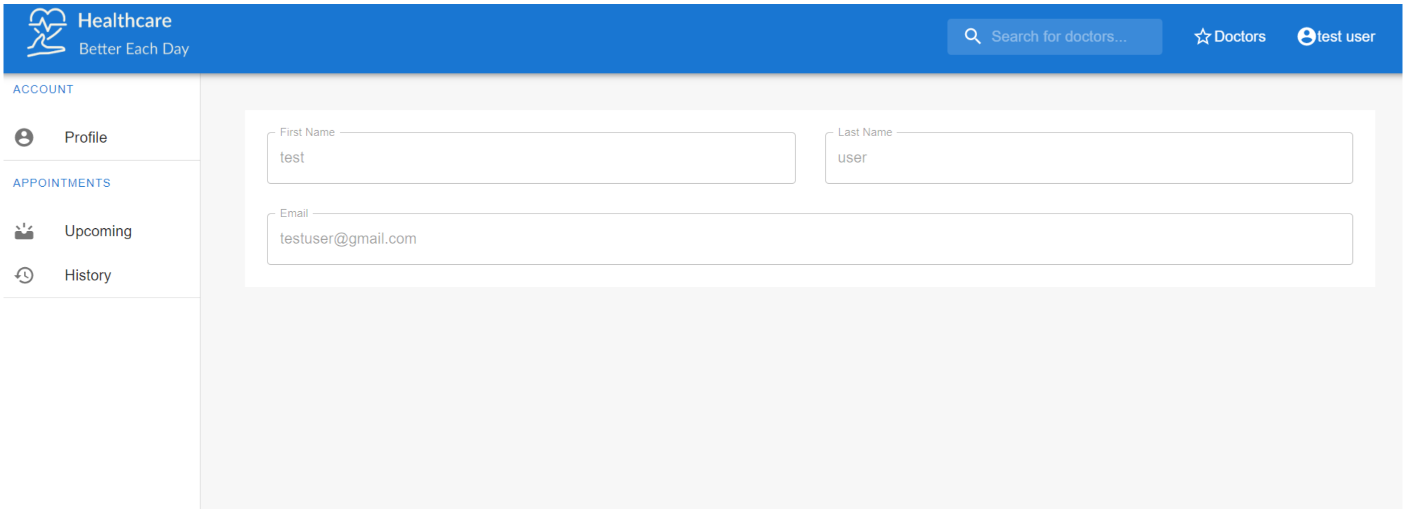


Find all the common specialities at one place

Dr. Aayushi Ravi
Gynaecologist
53, AtulGunj, Agra
₹500
MBBS
☆☆☆☆☆ ratings

Dr. Aman Gupta
Dermatologist
582, 1st Floor, 7th Main, 4th Cross, HMT Layout, Rt Nagar Post, Bangalore
₹500
MBBS
☆☆☆☆☆ ratings

Fig 19
List of Doctors



First Name: test

Last Name: user

Email: testuser@gmail.com

Fig 20
My Profile Page

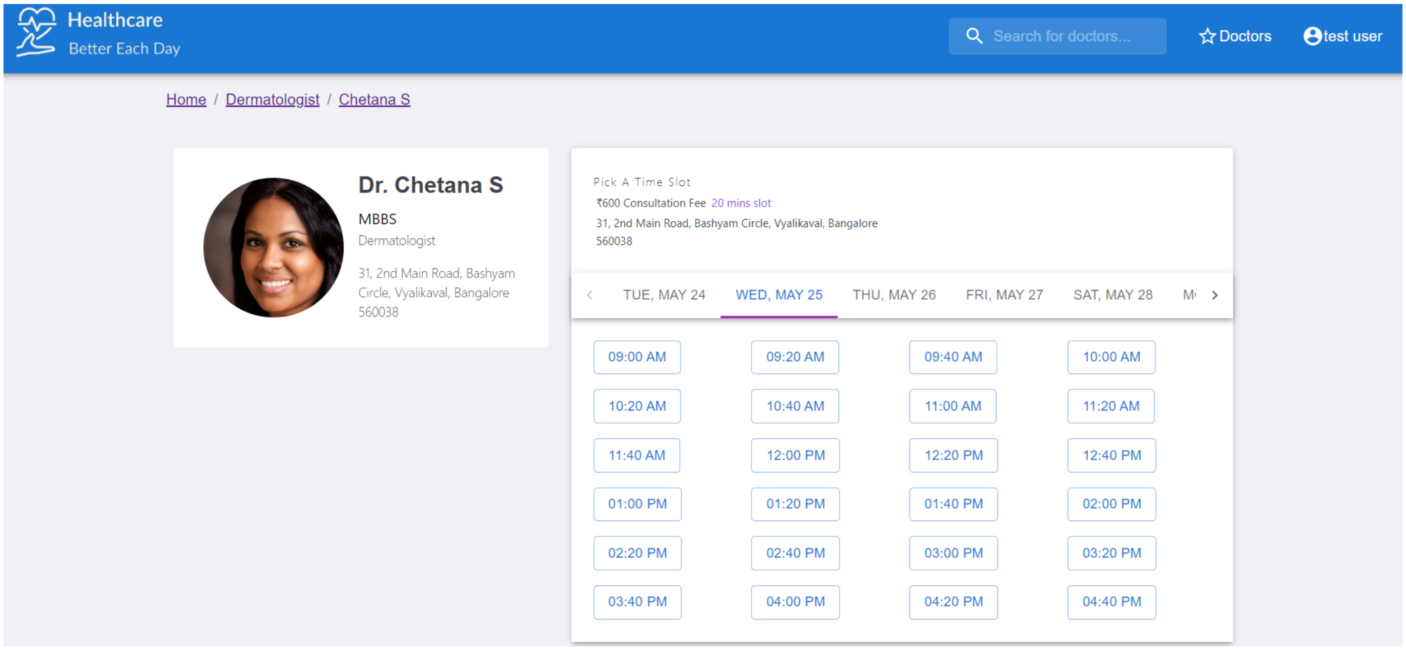
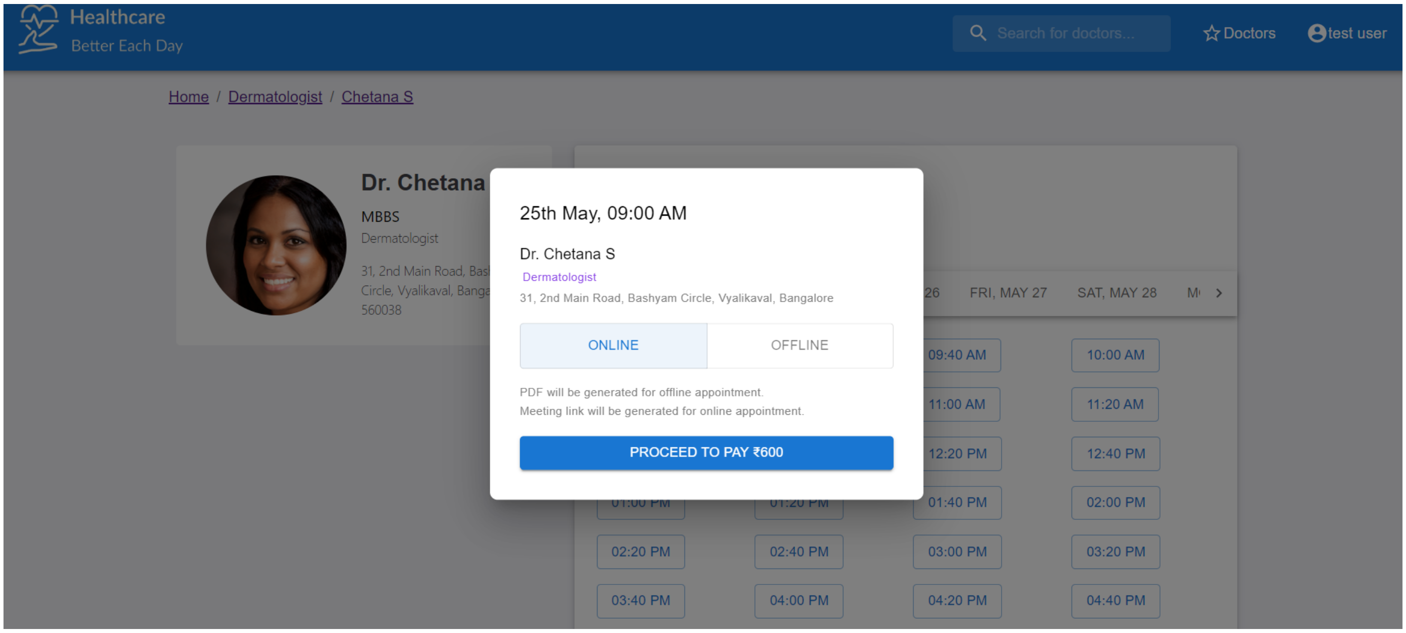
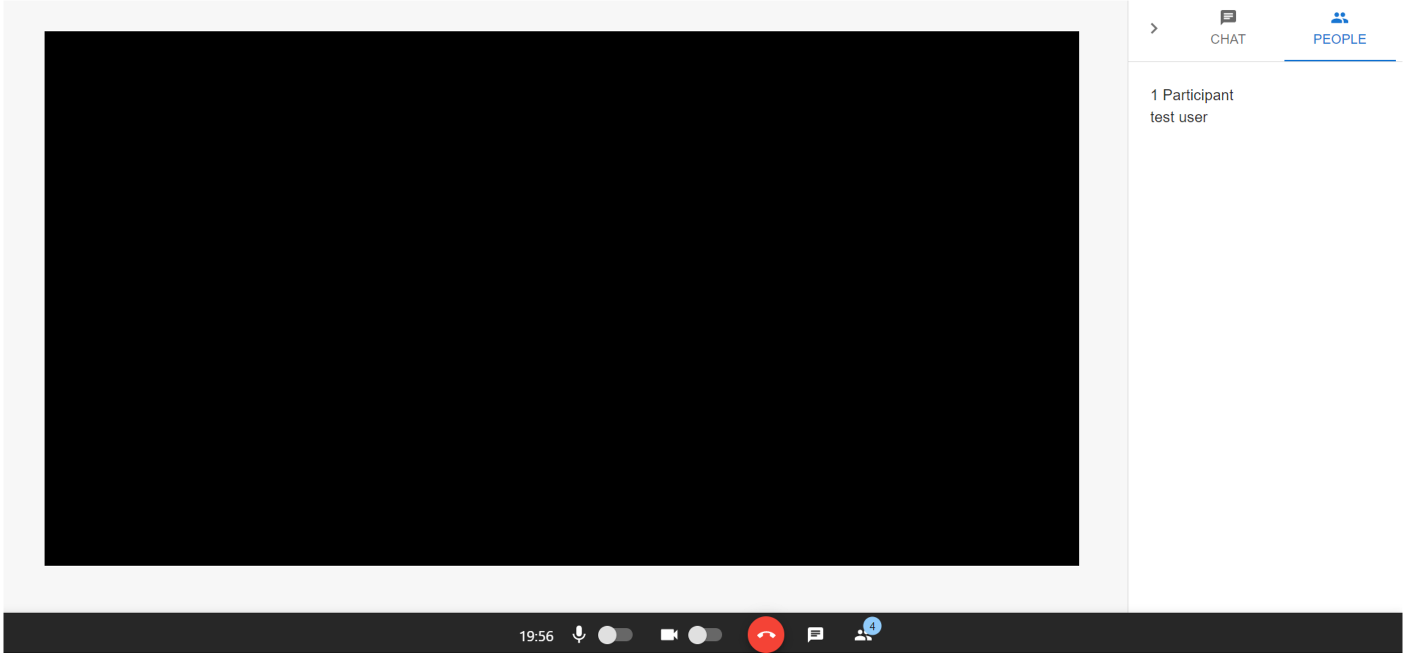


Fig 21
Slot Booking Page



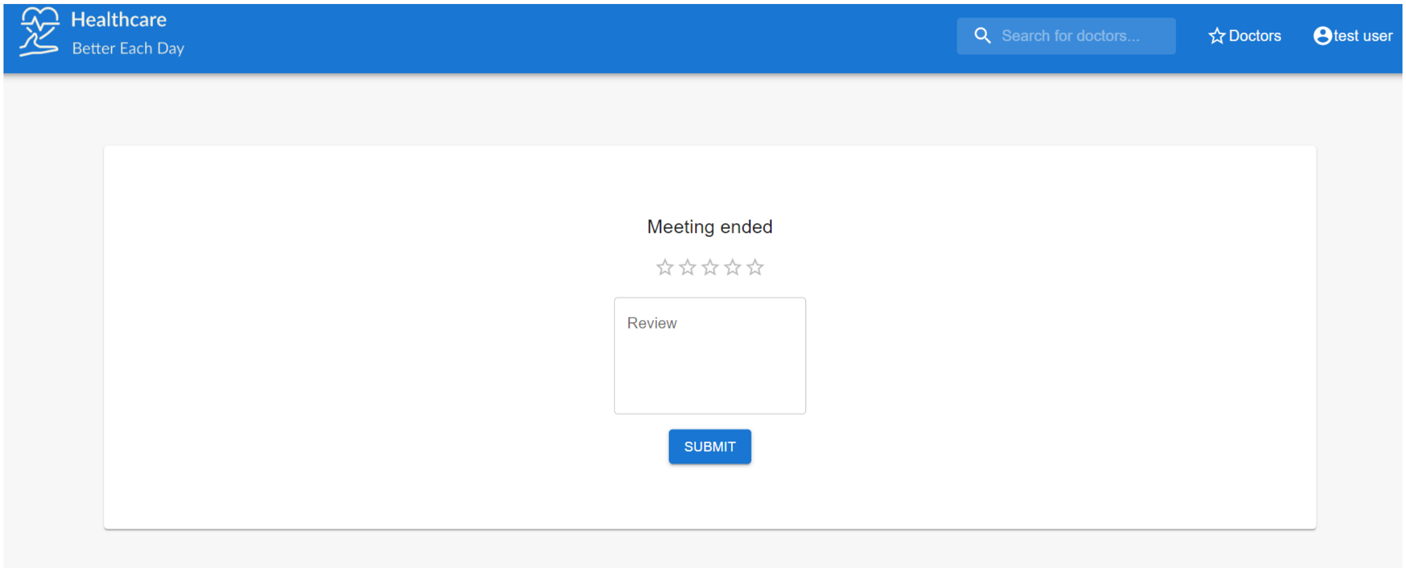
PDF will be generated for offline appointment.
Meeting link will be generated for online appointment.

Fig 22
Online & Offline Slot Option



1 Participant
test user

Fig 23
Online Consultation Meeting Page



Meeting ended

Fig 24
Review Submission Page

5.2 Test Results and Report

S. No.	Test Case	Input Value	Expected Outcome	Actual Outcome	Result
1	Invalid email	Without '@' symbol	Invalid email	Invalid email	Passed
2	Mandatory fields	Blank fields	Fill all fields	Fill all fields	Passed
3	Invalid password	Less than 8 characters	Invalid pass word	Invalid password	Passed
4	Special character	Special character in name	Special character not allowed	Special character not allowed	Passed

Table 1: Sign up and Sign in

S. No.	Test Case	Input Value	Expected Outcome	Actual Outcome	Result
1	Display homepage	Google chrome, Microsoft edge	Successful UI displayed	Successful UI displayed	Passed
2	Invalid Credentials	Unmatched username and password	Login Failed	Login Failed	Passed
3	Blank passwords	Blank field	Mandatory field	Mandatory field	Passed
4	Logout	Logout	Successfully logged out	Successfully logged out	Passed

Table 2: User Information

5.3 Conclusion

It's a lot of fun working on the doctor and patient appointment system. I was presented with various unpleasant duties after completing the assignment. The healthcare system in our culture is becoming increasingly important. As a result, I've made the decision to build this system. I looked at a few other systems to see if they might help me construct this system. I inquired as to the nature of their difficulties. They were overjoyed to accept this arrangement since it allowed them to unwind in this fast-paced environment. Despite my achievements, completing this project offered significant challenges. Because it's a web-based system, both the doctor and the patient must follow the rules for the system to function; otherwise, the system would fail.

5.4 Application

With the growing world in the direction of technology, we all can agree that more digitalized the life gets more easy it becomes in our daily lives. Medical emergency is an uncalled situation which can occur to anyone at anytime. So having an online solution to it has a huge application in our daily lives. The target customers of this app is not just the patients but also the doctors, which have a huge community. From big organization to personal usage, this web application can be used vastly and their so no location barrier as it provides online consultation too which can be accessible from any part of the world. Any adult can easily keep a check of their health with this web application and since it stores the history of all previous appointments, one need not to hit and try with different doctors to see which one works for them the best. No matter where in the world the person is, one can easily book an online slot and get the consultation within twenty minutes.

5.5 Future Work

The online system is subject to frequent development. It gets better every day, making life simpler for everyone. This system has huge scope for a great relationship between doctor and patients. For the development of this system in future I would be adding more feature including advanced functionalities. Also the user interface can be improved by making to more appealing to the users. Although this web application is currently enough user-friendly but when it comes to UI there is always a scope of improvement.

I would also want my system to have features apart from medical appointment like purchasing medicines online or booking diagnostic tests. Alone getting medical treatment is not enough, there are future steps to be followed like getting the tests done and purchasing the prescribed medicines. So for my system to be a standalone solution for the patients these features become necessary. Also the medical history of each user can be stored which will be accessible to the doctor before consultation, it would increase the efficiency of the treatment. Follow up feature is one more feature that becomes necessary if a patient is suffering from a serious issue. Many more such features can be added based on requirement of the present times.

References

[1] Alex Chaffee (2000-08-17). "What is a web application (or "webapp")?". Retrieved 2008-07-27.

[2] Bailey, N. (1952). A study of queues and appointment systems in hospital outpatient departments, with special reference to waiting times. Journal of the Royal Statistical Society, A14, 185-199.

[3] Bailey, N. (1954). Queuing for medical care. Applied Statistics 3, 137-145

[4] Cayirli, T, E. Veral, and H. Rosen. (2006). Designing appointment scheduling systems for ambulatory care services. Health Care Management Science 9, 47–58.

[5] Cayirli, T, E. Veral, and H. Rosen. (2004). Assessment of patient classification in appointment systems. 1st Conference of the POMS College of Service Operations, New York, NY, USA.

[6] Fernandes C, Daya M, Barry S, Palmer N (1994). Emergency department Patients who leave without seeing a Physician: The Toronto Hospital experience. Ann. Emer. Med., 24: 1092-1096.

[7] https://theseus.fi/bitstream/handle/10024/121504/Aljaf_Begard.pdf?sequence=1

*[8] Hypertext Mark-up Language, Introduction. Read 29.05.2016.
http://www.w3schools.com/html/html_intro.asp*

[9] Klassen, K. J., T. R. Rohleder. 2004. Outpatient appointment scheduling with urgent clients in a dynamic, multi-period environment. Internat. J. Service Indust. Management 15(2) 167–186.

[10] Koole, G., and G. Kaandorp. 2006. Optimal outpatient appointment scheduling, Working Paper, Department of Mathematics, Vrije Universiteit Amsterdam, The Netherlands, http://www.math.vu.nl/˜koole/articles/hcms07b/ (checked on May 24, 2007).

[11] James Duncan Davidson, Danny Coward (1999-12-17). Java Servlet Specification ("Specification") Version: 2.2 Final Release. Sun Microsystems. pp. 43–46. Retrieved 2008-07-27.

[12] http://airccse.org/journal/jcsit/6414ijcsit05.pdf

[13] JavaScript programming language, JS tutorial.

[14] http://www.w3schools.com/js

[15] MySQL, Open source database.

[16] https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html