



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY-I UNIVERSITY BY UGC

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

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **Mr.K.S.Thanoj (41614096)** who carried out the project entitled "**Appointment booking web application**" under my supervision from January 2024 to May 2024.

Internal Guide
Dr.S.Nivetha M.E., Ph.D.,

Head of the Department
Dr. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on 7/5/24

Internal Examiner
External Examiner

DECLARATION

K.S.Thanoj (41614096), hereby declare that the Professional Training Report-II titled "Appointment booking web application" done by me under the guidance of Dr.S.Nivetha M.E.,Ph.D., is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in Cyber Security.

DATE: 7/5/24

PLACE: Chennai

Thanoj
SIGNATURE OF THE CANDIDATE

PROFESSIONAL TRAINING REPORT - II

entitled Appointment Booking Web Application

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering with
specialization in Cyber Security

by

GONGADA PRADEEP KUMAR [41614110]

K.S.THANOJ [41614096]

SHAIK ABDUL MUJEEB [41614089]

VATSAVAYI BHUVANESH KRISHNA VARMA [41614100]



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY -1 UNIVERSITY BY UGC

**Accredited with Grade “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)**

CATEGORY-I UNIVERSITY BY UGC

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

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **Mr. Shaik Abdul Mujeeb (41614089)** who carried out the project entitled "**Appointment booking web application**" under my supervision from January 2024 to May 2024.

**Internal Guide
Dr.S.Nivetha M.E., Ph.D.,**

**Head of the Department
Dr. S. VIGNESHWARI, M.E., Ph.D.,**

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **Shaik Abdul Mujeeb (41614089)**, hereby declare that the Professional Training Report-II entitled “**Appointment booking web application**” done by me under the guidance of **Dr.S.Nivetha M.E., Ph.D.**, is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in Artificial Intelligence.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** 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. S.Vigneshwari M.E., Ph.D., Head of the Department of Computer Science and Engineering** for providing me necessary support during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Internal Guide **Dr.S.Nivetha| M.E., Ph.D.,** for his/her valuable guidance, suggestions and constant encouragement which paved way for the successful completion of my phase-2 Professional Training.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

COURSE CERTIFICATE



ABSTRACT

If anybody is ill and wants to visit a doctor for check-up, he or she needs to visit the hospital and waits until the doctor is available. The patient also waits in a queue while getting appointment. If the doctor cancels the appointment for some emergency reasons then the patient is not able to know about the cancelation of the appointment unless or until he or she visits the hospital. As the mobile communication technology is developing rapidly, therefore, one can use the mobile's applications to overcome such problems and inconvenience for the patients. The proposed work in this paper is an Online Hospital Management Application that uses an android platform that makes the task of making an appointment from the doctor easy and reliable for the users. Android based online doctor appointment application contains two modules. One module is the application designed for the patient that contains a login screen. The patient has to register himself before logging in to the application. After logging in, the patient can select a hospital and can view the hospital details. The patient has the option of selecting a doctor from the list of doctors and can view the doctor's details. The patient can request for an appointment on his/her preferred day/time. The selected day/time slot will be reserved and patient will receive the notification of the successfully added appointment. The patient can view the location of the hospital on map. In addition, the patient can contact to the hospital and the doctor by making a call or may send an email to the doctor. There are considerable online scheduling tools in the internet, a few of which are trait loaded, simple to setup and economical For practitioners, online appointment reservation and scheduling delivers a lot of merit added benefits and services, like captivating the patient, composing the patient to feel welcomed, and being capable to save patients' details safely for future information. But the most admirable and useful preference is that online appointment reservation and scheduling is remarkably in expensive .Both doctors and patients can access the portal through their unique ID's.

Keywords: Online Appointment, Web Application, Healthcare, Scheduling

TABLE OF CONTENTS

Chapter No	TITLE	Page No.
	ABSTRACT	iv
	LIST OF FIGURES	vii
1	INTRODUCTION	1
	1.1 Main Purpose	
	1.2 Objective of the Project	
	1.3 Aim of the Project	
	1.4 Project Scope	
	1.5 Study Limitation	
2	LITERATURE SURVEY	3
	2.1 Waiting time	
	2.2 Appointment Delay	
	2.3 Managing Patient's Appointment	
	2.4 Online Booking	
	2.5 Existing System	
	2.5.1 Determination of Problem	
	2.5.2 Proposed System	
	2.6 Patient's Appointment Time	
	2.7 Application Technologies	
	2.7.1 Web Server	

3	SYSTEM ARCHITECTURE	9
3.1	Aim	
3.2	Software Requirements	
3.3	Features	
4	REQUIREMENT AND SYSTEM DESIGN	13
4.1	System Overview	
4.2	Flow chart	
4.3	Algorithm	
4.4	Class Diagram	
4.5	ER Diagram	
5	RESULTS AND DISCUSSION	25
5.1	System Testing	
5.2	User Interface Layout	
5.3	Discussion	26
5.4	Source Code	29
6	SUMMARY AND CONCLUSION	35
6.1	Summary	
6.2	Conclusion	35
6.3	References	37

LIST OF FIGURES

FIGURE NO	FIGURE NAME	Page No.
Fig : 4.1	Data Flow Chart	20
Fig : 4.2	Class Diagram	21
Fig : 4.3	ER Diagram	
Fig : 5.1	Home Page	
Fig : 5.2	Set Appointment	
Fig : 5.3	Search Doctor To Get Appointment	
Fig : 5.4	Appointment List	
Fig : 5.5	Contact Doctor	
Fig : 5.6	Payment For Appointment	
Fig : 5.7	My Appointment	

CHAPTER 1

INTRODUCTION

Health care is one of the fastest growing industry all over the world. Before the last few years, medical appointments were usually taken on the phone calls or by visiting the hospitals in person. This process needed the involvement of individuals so, the ability to take appointment was restricted to the availability of schedulers, phone lines or the physical presence of a person.

With the growth of time, everybody demanded timeless and efficient medical care delivery because manual appointments (that requires the physical presence of both individuals) and long waiting lines have formed an irritating situation for the healthcare institutions. So, it created a need for such an integrated health care system that could deliver seamless care to both outpatients as well as inpatients. The emergence of online appointment system offered timeless and efficient access to health care services. Therefore, for hospitals and other medical societies, online appointment booking has a great importance and a subject of interest (Koole, 2007).

Booking appointment online has become a new trend in the past few years and is considered as one of the key processes in the healthcare industry. Bailey (1952) considered scheduling system as a trade-off or a compromise between a doctor and patient's waiting times. Patients who get late for the appointments or who fails to come becomes the reason for the underutilization of a doctor's time. Idle time and underutilization of doctor's time are also resulted by gaps in the appointment times (Bailey, 1954).

Different researchers agreed that main patient dissatisfaction is caused by long waiting times. Cayirli (2003) defined access time as the time between patients' request for the appointment and the time he is checked up. According to Veral, waiting time is the time between consultation and the scheduled time while neglecting the early arrival of a patient (Veral, 2003). Different researchers defined waiting /access time in different ways. A well-designed appointment system supposed to improve patients' satisfaction by reducing cos

With the growing population need for more efficient ways to access a medical treatment is also growing. Through an online appointment scheduling system, a user gets access to the doctor's online webpage and can make an appointment with online software. Patient/user can also provide additional medical history in advance, giving adequate time to the doctor to prepare the necessary information for consultation. In this way, online appointment scheduling systems are helping doctors and the patients and making the healthcare delivery efficient.

Nowadays there are many kinds of online appointment tools available in the market which are easy to set up and not too much expensive. Online scheduling system offers value-added services and lots of benefits to the doctors and patients. It makes the patient appreciated by eliminating the hassle of long waiting times. Online appointment systems are also getting popular because of its low-cost availability.

1.1. Main Purpose

The main purpose of this project is to link and bring all major private medical clinics of the city (of my country) to a single platform. So that patients can easily get access to the doctor's profile and make online appointments. The second purpose is to create an online medical history database so that doctors and patients can freely exchange patients' medical history information much easier, faster, and safer.

1.2. Objective of the Project

The main objective of the thesis is to provide quality medical care to the patients by bringing all medical practitioners of the city to a single platform so that everybody can easily access them and make appointments. The second objective is to replace the current manual file keeping system with an online medical patient database system.

1.3. Aim of the Project

The aim of this project is to create a platform where patients and doctors can access /interact efficiently with each other and provide ease and comfort to the patients. It also aims to resolve the problems that patients have to face while taking appointments and keeping medical files. Patients can choose a medical practitioner based on their professional profile and other patient's reviews. While doctors can access and update a patient's medical record after every checkup.

Following features will be added in the future:

- Online follow up with doctors (specially for distant patients)
- Linking laboratory and pharmacy so that medical store administrator can view suggested prescription and laboratory can view clinical tests recommended by the doctor. This feature aims to eliminate paper-based prescriptions.

1.4. Project Scope

This system is implemented for all the individuals who want to get treated by the city practitioners. The users can participate only if they have created an account through the registration form and have provided their medical history. Once they get registered themselves further they would not need to update their record as it would be done automatically after each doctor's visit.

1.5. Study Limitation

The lack of time was the main the limitation. Secondly, because there is no such system in my country so there may be some difficulties to implement and convince people to use this system.

CHAPTER 2

LITERATURE SURVEY

2.1. Waiting time

Fernandes et al. (1994) defined waiting time as the period of time an individual wait until a specific action occurs. Waiting time was also defined as the time a patient entered a clinic or service point and gets consulted by a doctor with a prescription in his hand. There are two different ways in which the waiting time has been defined in the past. At first, waiting time is the time that starts when an individual comes for the appointment and it continues until the individual gets consulted by the doctor. In the second definition, it starts when an individual comes for taking appointment and ends when he got prescribed medicine after being consulted by the doctor.

With the passage of time, long waiting lines at doctor's clinics became a major problem in developing countries. In a South African health clinic, block appointment system was introduced as an experiment, in which waiting time for patients was measured for the period of one week before and after the implementation of the appointment system. During the experiment, interviews were taken from focus-group, staff, and patients and from results it was realized that acute medically ill patients with prior appointments had short waiting times as compared to the patients without appointments.

It has also been stated that the appointment system has no benefit for the patients not seeing the doctor or who are taking routine repeated medication. Later it was realized that the block appointment system offers shorter waiting time only for acutely ill patients and not for others.

2.2. Appointment Delay

It has been proved in the past research that there is a direct relationship between appointment delay and cancellation of the appointment. Appointment delay is defined as the time which starts when an individual request for an appointment and ends until he/ she gets consulted by the doctor. Longer appointment delays cause more appointment cancellations. So, the best way to reduce appointment cancellations or no-shows is to minimize the time gap between the appointment request and doctor's check- up/consultation time. Minimizing this gap is called open access (abbreviated as OA) or advance access policy which later became a popular practice and part of an active research also. Both positive and negative results were experienced by the researchers in their experiments. Some practitioners were in the favor of OA and highly recommended it, and on the other hand, there were some practitioners who disagreed and were against of OA implementation.

2.3. Managing Patient's Appointment

In a health care centre, an application which is used to manage and minimize patient waiting time is called managing patient's appointment system. Some health care clinics use this kind of applications while some do not. Medical clinics who use such applications usually have shorter waiting times as compared to those medical clinics who do not use any appointment application. Patients who supposed to wait more than an hour for their medical check-up feel disgraced and unfair. Patients can evaluate the quality of any medical centre based on the waiting time they have to experience. Therefore, it has become important to consider factors like "saving time" and "minimizing idle time" while developing any patient's appointment system. Whereas according to Klassen (2002), scheduling doctor's time and patient's appointment supposed to depend on the seriousness of the case and patient's need.

2.4 Online Booking

An online appointment system is a web-based system which is made up of independent components or web pages, working together for a common purpose. Such systems are available on the internet for the users to accomplish some purpose.

James (1999) defined internet as a system which provides an opportunity to millions of people to get access to a large amount of stored data and get connected to each other despite distances. Increased rate of missed appointments and patient dissatisfaction pushed to recognize the need for better-quality care services. With the advancement of information technology and the sensitive situation of medical care, online appointment system emerged as an important output for efficient and timely delivery of health care services. Earlier, appointment requests were used to make through the phone call, email or fax but with the advancement of technology and internet, the trend changed towards the proper online appointment systems.

According to Gruca (2004), a patient can make an appointment for a doctor either by going directly to the doctor's clinic or making a manual appointment through the phone call or sending the email. With the emergence of the internet, health care centres can connect and communicate with their patients efficiently. Therefore, many healthcare centres have started using online appointment systems. Such systems ensure efficiency and effectiveness.

2.5 Existing System

In Indian, there is a huge trend of private medical clinics and hospitals. Doctors run their own private clinics and consult patients during the evening or any time of the day depending on their availability. Some are popular and known to all while some are known by few people. This situation proves to be a challenge for new practitioners as they are known to very few people even if they have a good academic background. On the other side, patients also face difficulty in finding and choosing a nearby doctor. Unscheduled appointments, long waiting lines and keeping medical files in physical form are also the common problems faced by the people/ patients of Indian.

2.5.1 Determination of Problem Challenge for new practitioners

Starting a private clinic can be a challenge for new practitioners. For example, if a new doctor wants to open his clinic there is no platform exist, by which or from where people can get to know about his clinic.

The problem for a new individual in a city

In the same way, the same problem goes for any person new to the city as he is unknown, who is the best or closest doctor/clinic to go for a medical checkup.

Long waiting lines

To get doctor's consultation, patients come twice to the clinic once for taking the appointment and second time for the checkup. Some patients visit the doctor without any prior appointment resulting in a long waiting time. There is also a possibility that they don't consultation by the doctor even after a long waiting line.

Managing paper-based medical files

Patients must carry their medical related report or files every time they visit a doctor. And sometimes they lost their files too.

2.5.2 Proposed System

In "Online Polyclinic Appointment and Database Management System" I have tried to consider all of the above problems with the existing system in Indian. My objective was to provide a platform where all practitioners and patients can be brought together. In this application, all practitioners either new or working for a long time can sign up themselves. This way people can get to know and choose to visit any doctor of their choice. By using this application user will be able to know and access the professional profile of each registered doctor from all specialties. Doctor's profile includes

information regarding their professional experience, practicing license, educational background, clinic timings, working days, clinic accessibility and other patient reviews also. Users would be able to book appointment sitting at their homes. The user can choose a doctor close to them or have more professional experience or have more good reviews from the other patients. New practitioners can easily join the platform and start taking appointments without the need of any expensive advertisement.

When registering as a patient, the user can give all necessary medical history so that he doesn't need to keep or bring his medical file when going for a check-up.

2.6 Patient's Appointment Time

Scheduling appointments for the patients started long years ago. Primarily appointment system was developed to minimize doctor's idle time because it was thought that doctors time is more valuable as compared to the patient's waiting time. But later it was realized that the importance of minimizing the patient's waiting time is as important as the doctor's time. So now while developing an appointment system doctor's idle time and the patient's waiting time both factors are given equal importance. Patient's scheduling also includes improving quality health services, reducing doctors and nurses idle time and reducing patient's waiting time.

2.7 APPLICATION TECHNOLOGIES

The purpose of this project is to build an Online Appointment and Database Management System. It is important for the user to understand how this application works and knowing the technologies that are used to implement this project. For a better understanding, all steps are described in detail to give a full overview of the system.

2.7.1 Web Server

Role of a web server is to communicate between the client side and server side by storing, processing and delivering web pages to the client side. Usually, web browser initiates the communication using HTTP by sending the request of a specific resource and server gives the response with the content of that requested resource. For this project work, Apache HTTP server was chosen, and it exists on the WAMP service. Apache HTTP server is one of the popularly used web server software used in a lot of project works.

CHAPTER 3

AIM AND SCOPE

3.1 AIM

The aim of this project is to develop a comprehensive Doctor Appointment Management System (DAMS) that streamlines and enhances the process of scheduling and managing appointments between doctors and patients. The primary objectives include:

- 1. Efficiency Enhancement:** To create a user-friendly platform that optimizes the scheduling process, reducing waiting times for patients and ensuring efficient time utilization for doctors.
- 2. Accessibility and Convenience:** To design a system accessible through multiple platforms, including web and mobile applications, ensuring convenience for both patients and healthcare providers.
- 3. Appointment Scheduling Optimization:** To implement intelligent algorithms that prioritize urgent cases, minimize appointment conflicts, and provide flexible scheduling options based on the availability of doctors and patients.
- 4. Integration of Patient Data:** To securely integrate patient profiles, medical records, and appointment histories, enabling doctors to make informed decisions and provide personalized care.
- 5. Real-time Communication:** To incorporate features for real-time communication between doctors and patients, including appointment reminders, notifications, and updates on availability.
- 6. Scalability and Robustness:** To develop a scalable and robust system capable of handling a large volume of appointments while maintaining performance and reliability.
- 7. Security and Privacy:** To prioritize the security and privacy of patient information, implementing encryption protocols, access controls, and compliance with healthcare regulations such as HIPAA (Health Insurance Portability and Accountability Act).

By achieving these objectives, the Doctor Appointment Management System aims to revolutionize the healthcare scheduling process, improving accessibility, efficiency, and patient satisfaction while facilitating better communication and decision-making among healthcare professionals.

3.2 Software Requirements::

1.React: A JavaScript library for building user interfaces, offering a component-based architecture for creating interactive UIs. Developed by Facebook, React.js provides a declarative and component-based approach to building interactive UIs for web applications.

The core idea behind React.js is to break down the UI into reusable components, each representing a piece of the user interface. These components manage their own state and can be composed together to form complex UIs. React.js utilizes a virtual DOM (Document Object Model) to efficiently update the UI in response to changes in data or user interactions, minimizing the need for manual DOM manipulation and enhancing performance.

2. Redux Toolkit: A state management library for React applications, providing predictable state management with a single source of truth. Redux Toolkit is a set of tools and utilities designed to simplify the process of managing state in Redux-based applications. Redux itself is a predictable state container for JavaScript applications, commonly used with libraries like React for building user interfaces.

3.Ant Design: A UI library for React applications, offering a set of customizable and pre-designed components. Ant Design is a design language and a set of React components and resources for building high-quality, user-friendly web applications. Developed by the Alibaba Group, Ant Design follows the principles of design consistency, accessibility, and usability to provide developers with a comprehensive toolkit for creating modern and visually appealing user interfaces.

4.React Hook Form: A library for managing form state and validation in React applications, providing a simple and intuitive API. React Hook Form is a library for managing forms in React applications using React Hooks. It provides a flexible and efficient way to handle form state, validation, and submission without the need for complex state management libraries or boilerplate code.

5.Axios: A promise-based HTTP client for making HTTP requests, used for interacting with backend APIs. Axios is a popular JavaScript library used for making HTTP requests from web browsers and Node.js environments. It provides an easy-to-use API and powerful features for performing asynchronous HTTP operations, such as fetching data from APIs, submitting form data, and handling responses.

3.3.Features :

Dynamic Homepage: Conveniently gathers essential information on a single page.

Easy Appointment Setup: Seamlessly schedule appointments with doctors, featuring selectable available dates and time ranges.

Dynamic Appointment Management: Flexible time and date range selection for appointments.
Doctor Selection: Patients can choose their preferred doctor for appointments.

Industry-standard Code Quality: Utilization of top-notch code conventions, code splitting, and TypeScript in backend development.

Email Notifications: Automated email notifications for appointment setup and invoicing.

Doctor Dashboard: Dynamic dashboard for doctors to access patient information and provide online treatment with customizable prescriptions.

Patient Dashboard: Patients can track their treatment progress, view medications, prescriptions,

and appointment details from a dynamic dashboard.

Appointment Tracking: Track appointment status using a unique tracking ID directly from the homepage.

Dynamic Prescription and Invoice Generation: Customized prescriptions and invoices tailored to each appointment.

User Authentication: Dynamic authentication system allowing user sign-in, sign-up, password recovery, and email verification (for doctors).

Dynamic Doctor Filtering: Advanced filtering options for doctors and appointment scheduling, along with blogging and service features.

Ongoing Development: Continued enhancements with an evolving admin system.

Best Practices: Adherence to industry best practices, ensuring reusable and maintainable code.

CHAPTER 4

REQUIREMENT AND SYSTEM DESIGN

4.1. System Overview

We are implementing a web-based appointment system in Indian for booking an online appointment and keeping an online medical history. Users can sign up online, search for the nearby doctor and book appointment while sitting at their homes by using a web browser. Two different types of actors are using this system: user actor (patient), which can be registered to the system, search for the doctor and book an appointment. Administrative actor (doctor/ physician), which can log in to the system with a username and password, accept patient's appointment request and update medical record after each visit. The functionality and more features are explained in more detail in this chapter.

The main characteristics of the system are:

- Every individual either a doctor or a patient will have an independent id, name and a profile. Doctors are further categorized by the specialty of practice (i.e. Obs/Gyne, Urology, Internal Medicine etc.)
- The user (patient) can get, change or cancel appointment time and view medical record while the doctor can make changes in his calendar and accept or reject appointment request by log in to the site and accessing his/her personal page.

4.2. Data Flow Chart

To explain the better view and functionalities of the system, use case diagrams are chosen.

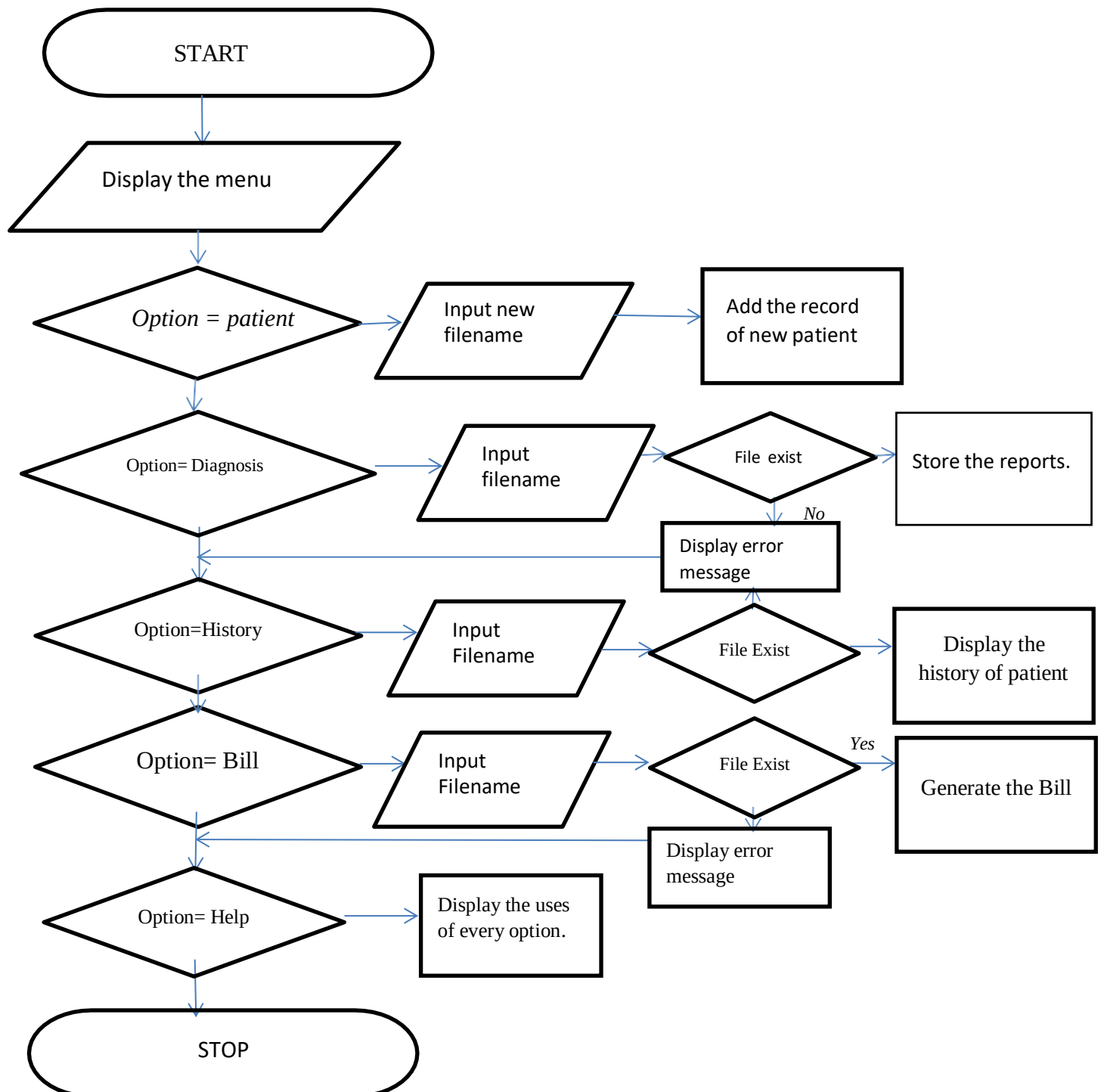


Figure 4.1: Data Flow Chart

4.3. Algorithm

Step 1: Enter the correct password.

Step 2: Menu will be displayed choose the required option you want to perform.

Step 3: Enter Option 1 i.e. the administration module which enables us to perform the following operation illustrate below:

- 1.Insert the patient information
- 2.Update the patient information
- 3.View the patient information
- 4.Delete the patient information

Step 4: If number of patient $\geq$ max patients the display patient can't be added and return

If person is outpatient go to step 6 else step 5.

Step 5: option 2 provides us to store patient personal information here we need to enter the following details:

1. Name :
- 2.Age :
- 3.Sex:
- 4.Address:
- 5.Weight:

Step 6 Now enter option 3 i.e outpatient module which take the information of the outpatient i.e illustrated below:

- 1.Enter the ID :
- 2.Enter the dept depending on the disease :
- 3.Room number:
- 4.Date of admitted date of discharge :

Step 7: Now press 4 to create patient report which ask us to

- 1.Enter patient ID number:
- 2.Enter Weight
- 3.Enter Name of Doctor appointed to
- 4.Date
- 5.Enter Precipitation to be followed.

Step 8: now enter 5 to generate bill of the patient,to generate bill

- 1.Enter the id number of patient
- 2.Name Doctor appointed to
- 3.Doctor fee
- 4.Health card number(not compulsory)
- 5.Room bill
- 6.Medicine bill
- 7.Lab bill
- 8.Operation theatre charges
- 9.Nursing bill

Step 9 : Now press enter which will display you the total cost you charged.

Step 10 :Exit

4.4. Class Diagram

The class diagram is chosen to explain the design phase of the system. A class diagram describes classes of the system, attributes, and operations and relationships of the classes in a better way. We can also say that class diagrams are used to justify the structure or behavior of use cases of the system. Class diagrams best explain the conceptual model of the system in terms of entities and their relationships. The class diagram looks like a shape of a rectangle, comprising three compartments stacked vertically. The first top box comprises the class name, the second middle box contains

the attributes of the class and third the last box contains the methods or functions performed by that class. The first compartment /box of the name is compulsory while rest of the two can be omitted to simplify the diagram. So, in any class diagram first compartment must be drawn while the second two compartments are optional.

The class "patient" contains multiple parameters (such as id, name, age, address), which depict the information of all the registered patients. The user class also contains the methods performed by these users such as get appointment, view/ create own medical record etc. In the same way, the class "doctor" has the parameters id, name, department, address possessing all the required information of the users registered as a doctor on to the system. Methods include accept/reject the appointment, check the patient, view a medical record of any patient etc. These methods are the functions performed by the users registered as a doctor on the system. The class "appointment" has the parameters of date and time, explaining what time or day patient user has requested for the appointment to the doctor. The class "department" has the parameters id and name and methods include add/ delete doctor and add or delete department. Every doctor user must belong to any department class. The "report" is another class containing methods like write report or prescribes medicine. In the end, class "admin" contains the parameters like id and name and methods of this class include manage users.

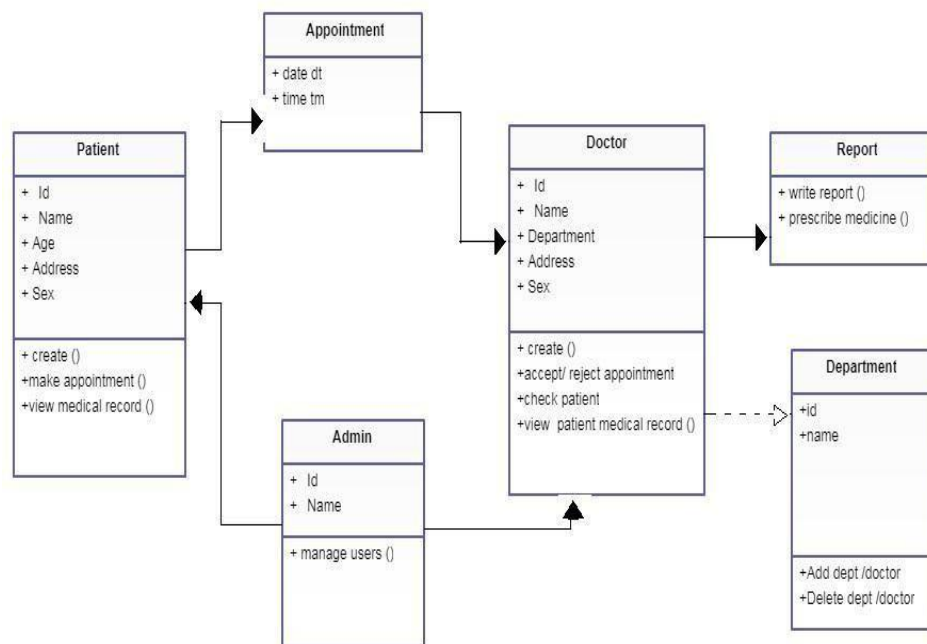


Figure 4.2: Class Diagram

4.5 ER Diagram

A basic ER model consists of objects called entities and specifies relationship among those entities. Purpose of this diagram is not to define any functionality rather show association and dependency among entities. ER diagram is drawn with "rectangular boxes" as entities and the "straight lines" showing the relationship between these boxes. An entity is an object or a thing that has an independent existence and can be easily differentiated from others. Each entity has some attributes like name, age, address, department etc. In the following diagram the doctor, patient, appointment, admin etc, all are different entities. So, an entity can be a person, animal, plant, event or a company.

Entities consisting of similar attributes make the entity sets. These entities have some association among each other which make a relationship. These relationships can be "one to one" or "one to many" or "many to many". For example, a doctor and department can have "one to many" relationships, means one department can have many doctors but one doctor is related to only one department.

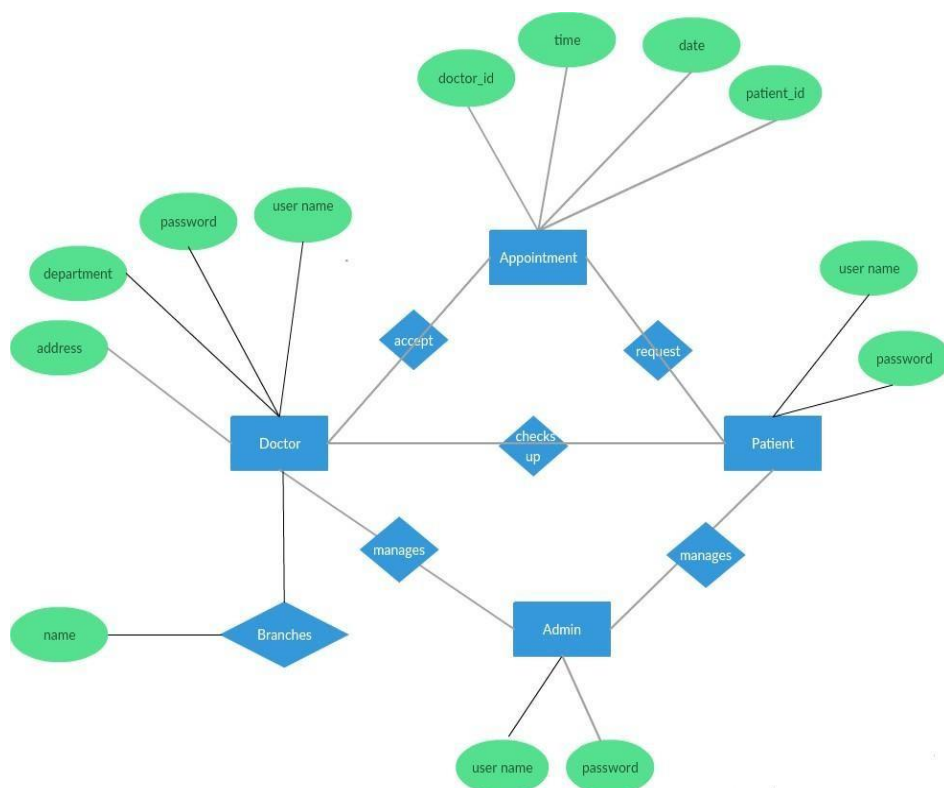


Figure 4.3:ER Diagram

CHAPTER-5

RESULTS AND DISCUSSIONS

Further discussions surrounding the Doctor Appointment Management System (DAMS) delve into its scalability, security, and potential for future enhancements. The system's scalability was a point of interest, as its robust architecture demonstrated the capability to handle a large volume of appointments while maintaining performance and reliability. This scalability is essential for accommodating the growing demand for healthcare services and the expansion of medical facilities. Additionally, the emphasis on security and privacy measures within the DAMS sparked discussions on compliance with healthcare regulations such as HIPAA. The encryption of patient data and implementation of access controls were noted as critical features ensuring the confidentiality and integrity of sensitive medical information. Looking ahead, discussions revolved around potential enhancements to further improve the DAMS's functionality and user experience. Suggestions included integration with telemedicine platforms for virtual appointments, predictive analytics for appointment forecasting, and continuous user feedback mechanisms for iterative improvements. Overall, the discussions highlighted the DAMS's success in modernizing appointment management practices while also identifying areas for future innovation and development. Usually developing any system comprises of three basic steps planning, design, and the testing phase. After finalizing the basic requirements that should be met by the system, it was also important to decide which technologies to use for the development of this system. For this purpose, I took help from the web and carefully studied already existed similar web applications. Use case diagram, class diagram, and ER diagram are used to better explain the system and its working. The purpose for drawing use case diagram was to explain not only functionalities but also to write down the requirements of each user very clearly. The class diagram was drawn to explain the relationship and behavior of entities.

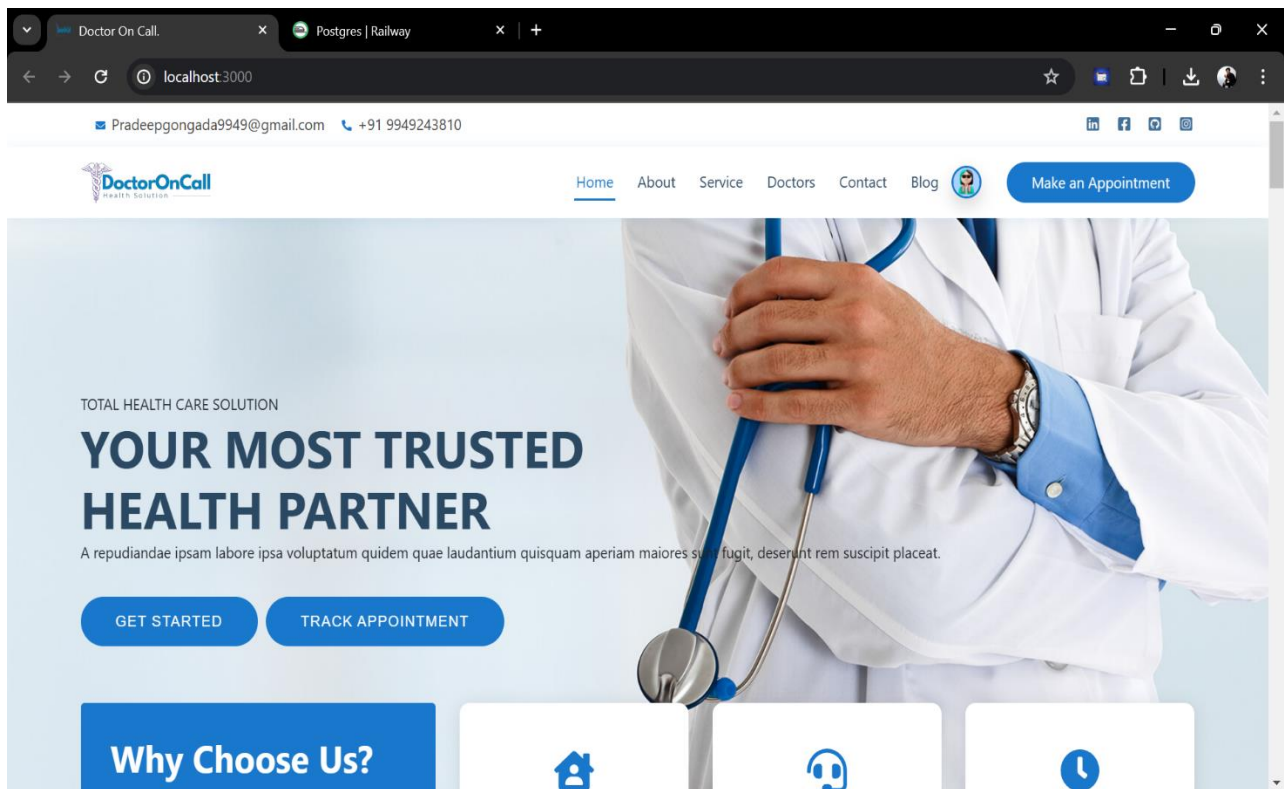


Figure 5.1: Home Page

Would you like to schedule an Interview? Pick a Date & Time

☐ With Doctor

☐ 30 Min

☐ Chennai, India
Flat Num:S2,Nest Vital Building,Semmencherry,Chennai,India

☐ Zoom Meeting

Pick a Time

May 2024	May 2024
5 Sunday	4 Saturday
3 Friday	2 Thursday

Pick a Time

Morning Time (8AM - 12PM)

08:00 AM	08:30 AM	09:00 AM
09:30 AM	10:00 AM	10:30 AM
11:00 AM	11:30 AM	

After Noon Time (1PM - 5PM)

01:00 PM	01:30 PM	02:00 PM
02:30 PM	03:00 PM	03:30 PM

Fig : 5.2 Set Appointment

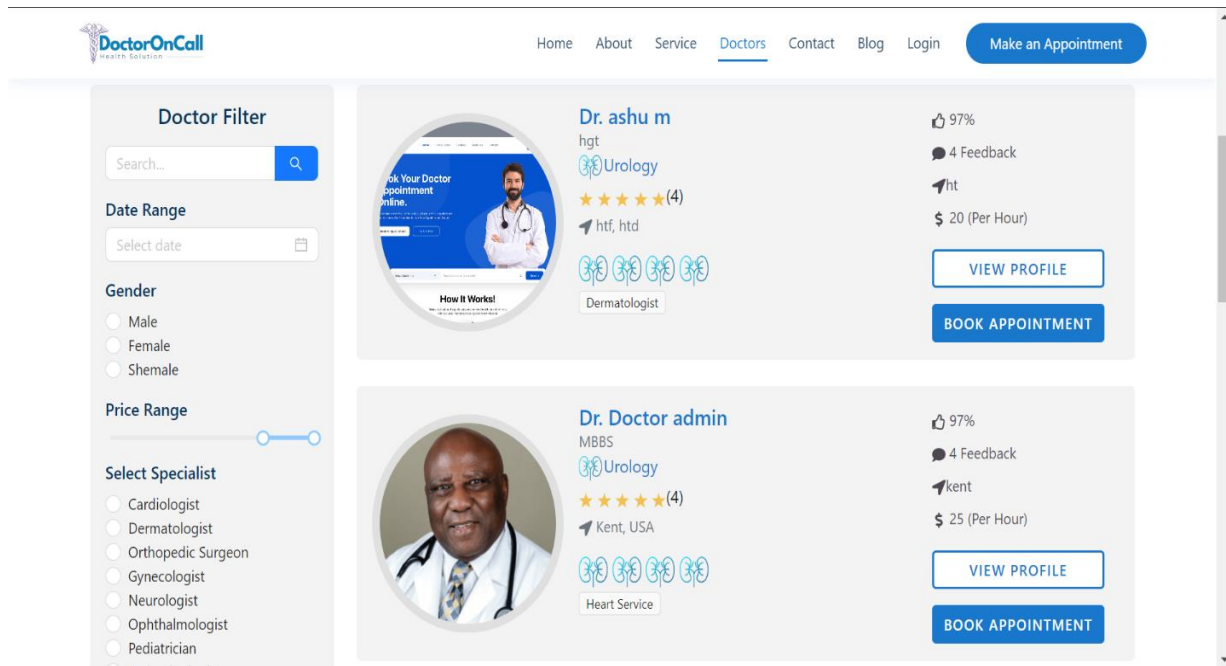


Figure 5.3: Search Doctor To Get Appointment

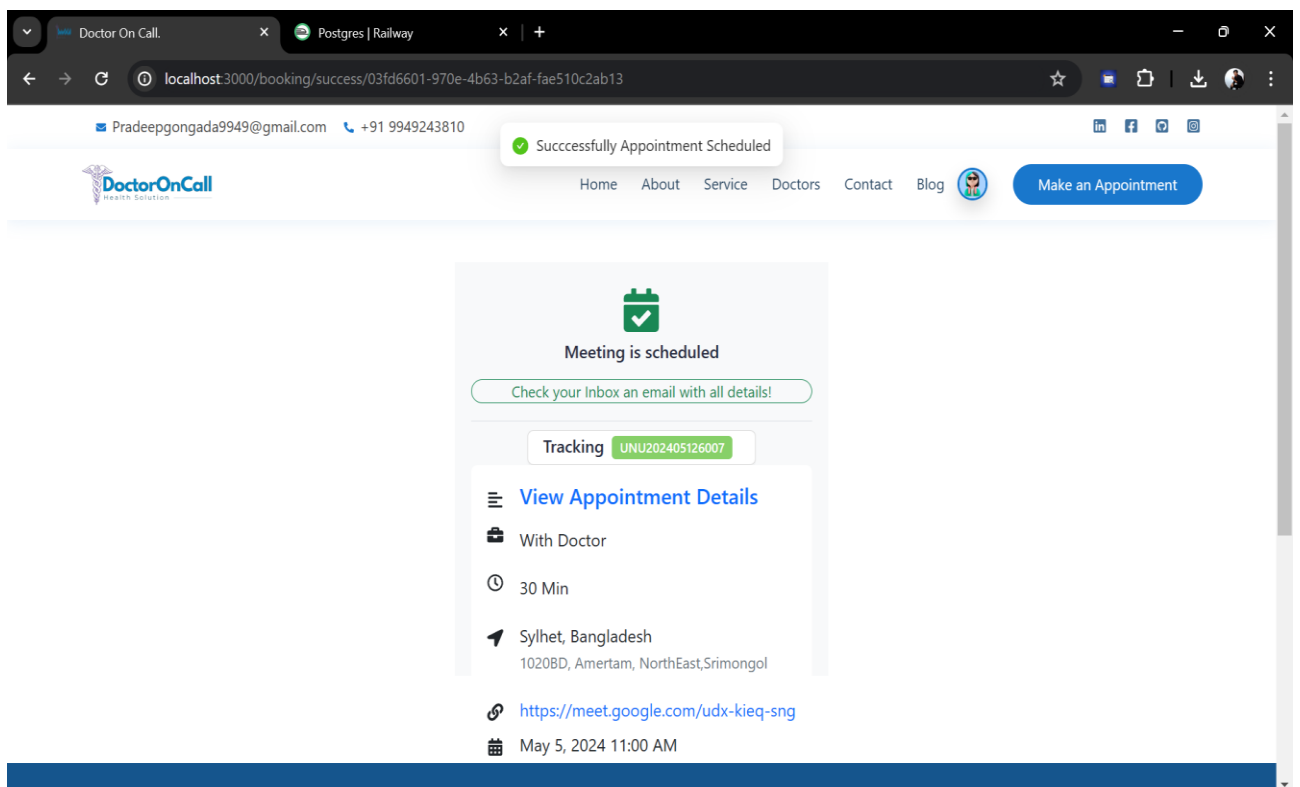


Figure 5.4: Appointment Scheduling

Figure 5.5: Doctor Contact

Item	Amount
Consulting Fee	\$60
Booking Fee	\$10
Vat (Including 15%)	\$9
Total	\$79

Figure 5.6: Payment For appointment

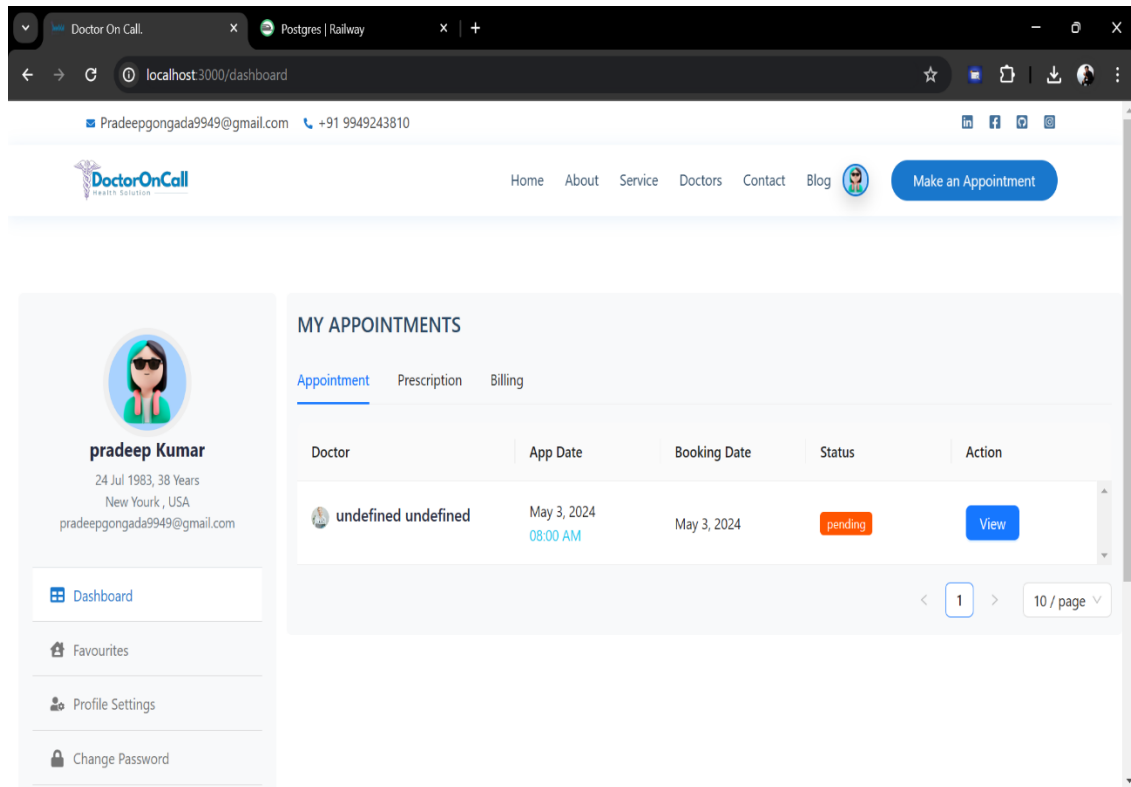


Figure 5.7: My Appointment

SOURCE CODE

```
@import
url('https://fonts.googleapis.com/css?family=Poppins:300,400,500,700,900');

* {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
}

a {
  color: #05335c;
  text-decoration: none !important;
}

a:hover {
  color: rgb(20, 104, 177);
  text-decoration: none;
}

body {
  overflow-x: hidden;
  font-family: 'Poppins', sans-serif;
  height: 100%;
  color: #272b41;
  background-color: #f8f9fa;
}

h1,
```

```
h2,
h3,
h4,
h5,
h6 {
  color: #272b41;
  font-weight: 500;
}

.h1,
h1 {
  font-size: 2.25rem;
}

.h2,
h2 {
  font-size: 1.875rem;
}

.h3,
h3 {
  font-size: 1.5rem;
}

.h4,
h4 {
  font-size: 1.125rem;
}

.h5,
h5 {
  font-size: 1rem;
}

.h6,
h6 {
  font-size: 0.875rem;
}

a {
  color: #2E3842;
  text-decoration: none;
}

a:hover {
  color: #09dca4;
}
```

```

a:hover,
a:active,
a:focus {
    outline: none;
    text-decoration: none;
}

.ant-radio-group {
    display: flex !important;
    flex-direction: column !important;
}

.card-label>label {
    background-color: #f8f9fa;
    color: #959595;
    display: inline-block;
    font-size: 13px;
    font-weight: 500;
    margin: 6px auto auto 8px;
    padding: 0 7px;
}

.card-label>input {
    background-color: #fff;
    border: 1px solid #dbdbdb;
    border-radius: 4px;
    box-shadow: 0 1px 3px 0 rgba(0, 0, 0, .05);
    display: block;
    height: 50px;
    margin-top: -13px;
    padding: 5px 15px 0;
    transition: border-color .3s;
    width: 100%;
}

.btn-get-started {
    font-family: "Raleway", sans-serif;
    text-transform: uppercase;
    font-weight: 500;
    font-size: 14px;
    letter-spacing: 1px;
    display: inline-block;
    padding: 12px 35px;
    margin-top: 30px;
    border-radius: 50px;
    transition: 0.5s;
    color: #fff;
    background: #1977cc;
}

```

```

}

.btn-get-started:hover {
    background: #3291e6;
}

.section-title h2 {
    text-align: center;
    font-size: 32px;
    font-weight: bold;
    padding-bottom: 20px;
    position: relative;
    color: #2c4964;
}

.section-title h2::before {
    content: "";
    position: absolute;
    display: block;
    width: 120px;
    height: 1px;
    background: #ddd;
    bottom: 1px;
    left: calc(50% - 60px);
}

.section-title h2::after {
    content: "";
    position: absolute;
    display: block;
    width: 40px;
    height: 3px;
    background: #1977cc;
    bottom: 0;
    left: calc(50% - 20px);
}

.more-btn {
    display: inline-block;
    border: 1px solid #3291e6;
    padding: 6px 30px 8px 30px;
    color: #3291e6;
    border-radius: 50px;
    transition: all ease-in-out 0.4s;
}

.more-btn .icon {
    font-size: 14px;

```

```

}

.more-btn:hover {
  color: #fff;
  background: #3291e6;
}

.custom-blog-card:hover {
  transform: scale(1.1);
}

.custom-blog-card {
  transition: 0.5s ease-in;
  max-width: 18rem;
}

.text-title {
  margin: 0;
  font-size: 20px;
  font-weight: 600;
  text-transform: uppercase;
  color: #2c4964;
}

.change-photo-btn {
  background-color: #1977cc;
  border-radius: 50px;
  color: #fff;
  cursor: pointer;
  display: block;
  font-size: 13px;
  font-weight: 600;
  margin: 0 auto;
  padding: 5px 10px;
  position: relative;
  transition: .6s ease-in;
  text-align: center;
}

.change-photo-btn:hover {
  background-color: #094b85;
  border-radius: 50px;
  color: #fff;
}

.change-photo-btn input.upload {
  bottom: 0;
  cursor: pointer;
}

```

```

    filter: alpha(opacity=0);
    left: 0;
    margin: 0;
    opacity: 0;
    padding: 0;
    position: absolute;
    right: 0;
    top: 0;
}
import './App.css';
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
import Home from './components/Home/Home/Home';
import SignInForm from './components/Login/SignInForm';
import DoctorBooking from './components/Booking/DoctorBooking/DoctorBooking';
import BookingSuccess from './components/Booking/BookingSuccess';
import BookingInvoice from
'./components/Booking/BookingInvoice/BookingInvoice';
import DoctorProfile from './components/Doctor/DoctorProfile/DoctorProfile';
import Appointments from './components/Doctor/Appointments/Appointments';
import MyPatients from './components/Doctor/MyPatients/MyPatients';
import Reviews from './components/Doctor/Reviews/Reviews';
import Schedule from './components/Doctor/Schedule/Schedule';
import ProfileSetting from
'./components/Doctor/ProfileSetting/ProfileSetting';
import ChangePassword from
'./components/Doctor/ChangePassword/ChangePassword';
import AdminDashboard from './components/Admin/Dashboard/Dashboard';
import AdminAppointments from './components/Admin/Appointments/Appointments';
import Doctors from './components/Admin/Doctors/Doctors';
import Patients from './components/Admin/Patients/Patients';
import Profile from './components/Admin/Profile/Profile';
import Transactions from './components/Admin/Transactions/Transactions';
import Specialites from './components/Admin/Specialites/Specialites';
import AdminReviews from './components/Admin/Reviews/Reviews';
import PatientFavouriteDoctor from
'./components/Doctor/PatientFavourite/PatientFavourite';
import DoctorInvoice from './components/Doctor/Invoice/DoctorInvoice';
import SearchDoctor from './components/Doctor/SearchDoctor/SearchDoctor';
import Blogs from './components/Doctor/Blogs/Blogs';
import BlogsEdit from './components/Doctor/Blogs/BlogsEdit';
import AddBlog from './components/Doctor/Blogs/AddBlog';
import Blog from './components/Blog/Blog';
import BlogDetails from './components/Blog/BlogDetails';
import Contact from './components/Contact/Contact';
import About from './components/About/About';
import Service from './components/Service/Service';
import AppointmentPage from './components/Appointment/AppointmentPage';
import TrackAppointment from './components/TrackAppointment/TrackAppointment';

```

```

import Treatment from './components/Doctor/Treatment/Treatment';
import Prescription from './components/Doctor/Prescription/Prescription';
import PrescriptionView from
'./components/Doctor/Prescription/PrescriptionView';
import TreatmentEdit from './components/Doctor/Treatment/TreatmentEdit';
import ViewAppointment from
'./components/Doctor/Appointments/ViewAppointment';
import ForgotPassword from './components/Login/ForgotPassword';
import Dashboard from './components/Doctor/Dashboard/Dashboard';
import PrivateOutlet from './components/Shared/PrivateOutlet';
import NotFound from './components/UI/NotFound';

function App() {
  return (
    <Router>
      <Routes>
        <Route element={<PrivateOutlet />} />
        <Route path='/dashboard/blogs' element={<Blogs />} />
        <Route path='/dashboard' element={<Dashboard />} />
        <Route path='/dashboard/my-patients' element={<MyPatients />} />
        <Route path='/dashboard/reviews' element={<Reviews />} />
        <Route path='/dashboard/schedule' element={<Schedule />} />
        <Route path='/dashboard/appointments' element={<Appointments />} />
        <Route path='/dashboard/appointments/:id' element={<ViewAppointment
/>} />
        <Route path='/dashboard/prescription' element={<Prescription />} />
        <Route path='/dashboard/prescription/:id' element={<PrescriptionView
/>} />
        <Route path='/dashboard/appointment/treatment/:id'
element={<Treatment />} />
        <Route path='/dashboard/appointment/treatment/edit/:id'
element={<TreatmentEdit />} />
        <Route path='/dashboard/change-password' element={<ChangePassword
/>} />
        <Route path='/dashboard/profile-setting' element={<ProfileSetting
/>} />
        <Route path='/dashboard/favourite' element={<PatientFavouriteDoctor
/>} />
        <Route path='/dashboard/invoices' element={<DoctorInvoice />} />
      </Route>
      <Route path='/login' element={<SignInForm />} />
      <Route path='/' element={<Home />} />
      <Route path='/blog' element={<Blog />} />
      <Route path='/blog/:id' element={<BlogDetails />} />
      <Route path='/contact' element={<Contact />} />
      <Route path='/about' element={<About />} />
      <Route path='/service' element={<Service />} />

```

```

        <Route path='/reset-password/:userId/:uniqueString'
element={<ForgotPassword />} />
        <Route path='/appointment' element={<AppointmentPage />} />
        <Route path='/track-appointment' element={<TrackAppointment />} />
        <Route path='/doctors' element={<SearchDoctor />} />
        <Route path='/doctors/profile/:id' element={<DoctorProfile />} />
        <Route path='/dashboard/blogs/:id' element={<BlogsEdit />} />
        <Route path='/dashboard/blogs/create' element={<AddBlog />} />
        <Route path='/booking/:doctorId' element={<DoctorBooking />} />
        <Route path='/booking/success/:id' element={<BookingSuccess />} />
        <Route path='/booking/invoice/:id' element={<BookingInvoice />} />
        { /* Admin Dashboard */ }
        <Route path='/admin/dashboard' element={<AdminDashboard />} />
        <Route path='/admin/appointments' element={<AdminAppointments />} />
        <Route path='/admin/doctors' element={<Doctors />} />
        <Route path='/admin/patients' element={<Patients />} />
        <Route path='/admin/profile' element={<Profile />} />
        <Route path='/admin/reviews' element={<AdminReviews />} />
        <Route path='/admin/transaction' element={<Transactions />} />
        <Route path='/admin/specialites' element={<Specialites />} />

        <Route path='*' element={<NotFound/>}/>
    </Routes>
  </Router >
);
}
export default App;

import React from 'react';
import {createRoot} from 'react-dom/client';
import App from './App';
import 'bootstrap/dist/css/bootstrap.min.css';
import { Provider } from 'react-redux';
import { store } from './redux/store';

const root = createRoot(document.getElementById('root'));

root.render(
  <React.StrictMode>
    <Provider store={store}>
      <App />
    </Provider>
  </React.StrictMode>
);

```

CHAPTER-6

SUMMARY AND CONCLUSION

6.1 Conclusion

This project deals with the Corporate Medicare Management. This project is very helpful to both Medicare staff as well as to the public. It is having mainly Administration and Client modules.

The growing quality demand in the hospital sector makes it necessary to exploit the whole potential of stored data efficiently, not only the clinical data, in order to improve diagnoses and treatments, but also on management, in order to minimize costs and improve the care given to the patients.

This project adds the details of every individual patient and the staff appointed to them, it stores the schedule of doctors and their operation timings. It is a user friendly system which can be used by any person. It can also store the duration of a patient till the time of discharge. It helps in the satisfaction of the user and the public itself.

Doctor Appointment System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

The aim is to automate its existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically the project describes how to manage for good performance and better services for the clients.

6.2 Future Enhancement

Every project whether large or small has some limitations no matter however diligently developed. In some cases limitations is small while in other cases they may be broad also. The new system has got some limitations. Major areas where modifications can be done are as follows:

- Our system is not online so further it can be improved.
- The security is limited so some additional arrangement could be made to provide more security to the system. Including user input and sentiment analysis can also increase the system's ability to adapt and give meaningful recommendations. Additionally, by incorporating real-time data such as weather, traffic, and the accessibility of tourist attractions, the system may be made more dynamic and responsive. Another area for further study is the use of social media data, which can provide informative information about user preferences and travel trends. Additionally, the system might be able to provide consumers with compelling, immersive travel experiences by investigating the application of cutting-edge technologies like virtual reality and augmented reality. Overall, these research directions can help develop a hybrid tourist recommendation system that is more informed and effective.

REFERENCES

- <http://www.javaworld.com/javaworld/jw-01-1998/jw-01-Credentialreview.html>
- Database Programming with JDBC and Java by O'Reilly
- Head First Java 2nd Edition
- <http://www.jdbc-tutorial.com/>
- Java and Software Design Concepts by Apress
- <https://www.tutorialspoint.com/java/>
- <http://www.javatpoint.com/java-tutorial>
- <https://docs.oracle.com/javase/tutorial/>
- <http://www.wampserver.com/en/>
- <http://www.JSP.net/>
- <http://www.tutorialspoint.com/mysql/>
- <httpd.apache.org/docs/2.0/misc/tutorials.html>
- JavaScript programming language, JS tutorial. Read 03.06.2016.
<http://www.w3schools.com/js/>
- MySQL, Open source database. Read 05.06.2016.
<https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>
- 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.
- <http://airccse.org/journal/jcsit/6414ijcsit05.pdf>
- http://www.w3schools.com/html/html_intro.asp