

E-HEALTH PORTAL

AN INTERNSHIP REPORT

Submitted by

**VARUN KONDREDDY [REG NO: RA1811003010692],
GUNDA SUMANTH [REG NO: RA1811003010701]**

Under the guidance of

Ms. MALAR SELVI G

(Assistant Professor, Department of Computing Technologies)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY



SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

Deemed to be University u/s 3 of UGC Act, 1956

S.R.M. Nagar, Kattankulathur, Kancheepuram District

MAY 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that 18CSP110L internship report titled “**E-HEALTH PORTAL**” is the bonafide work of “**VARUN KONDREDDY [REG NO: RA1811003010692], GUNDA SUMANTH [REG NO: RA1811003010701]**”, who carried out the internship work under my supervision along with the company mentor. Certified further, that to the best of my knowledge the work reported herein does not form any other internship report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

GUIDE

Ms. MALAR SELVI G
Assistant Professor
Dept. of Computing Technologies

PANEL HEAD

Dr. C.N.Subalalitha
Associate Professor
Dept. of Computing Technologies

HEAD OF THE DEPARTMENT

Dr. M. Pushpalatha
Professor
Dept. of Computing Technologies

INTERNAL EXAMINER

EXTERNAL EXAMINER



Internship Offer Letter

Dated: 2nd August, 2021

HighRadius Technologies Private Limited

Ektha Pearl, 4T Floor,
1st Lane White Fields,
BP Raju Marg,
Kondapur, Hyderabad, 500084
hereinafter referred to as "**HighRadius**"

AND

SRM Institute of Science & Technology
hereinafter referred to as "**SRM**"

AND

Name: Nagvarun Kondreddy

Location: Bhubaneswar

Roll Number: RA1811003010692

Graduation Year: 4th Year

enrolled as a student in the B.Tech Programme offered by SRM hereinafter referred to as "**Candidate**"

The above-named parties have agreed to the following:

1. Introduction:

The Highako program is an industry-institute partnership initiative of HighRadius for engineering undergraduates. Candidates offered participation in this program shall be trained at the Centre of Excellence established at KIIT's premises at Bhubaneshwar, and, based on the internship programme for which such Candidates are eligible, shall be paid a fixed stipend for the duration of their internship. Upon successful completion of their engineering qualification and satisfactory performance during the training period, Highradius may, at its sole discretion, further employ these candidates in accordance with HighRadius' existing terms and conditions of employment. The Highako program shall be considered by SRM for giving academic credit to the Candidates, in accordance with the guidelines determined by HighRadius and SRM.

2. Terms of the Training Program conducted by HighRadius

2.1. HighRadius Internship Programme:

Highradius Technologies Private Limited,
4th floor, campus 3, KIIT, Bhubaneswar, Khordha, Odisha, 751024
www.highradius.com

1

Highly Confidential

8/1/2021

DocuSigned by:

Nagvarun
8/1/2021



Internship Offer Letter

Dated: 2nd August, 2021

HighRadius Technologies Private Limited

Ektha Pearl, 4T Floor,
1st Lane White Fields,
BP Raju Marg,
Kondapur, Hyderabad, 500084
hereinafter referred to as "**HighRadius**"

AND

SRM Institute of Science & Technology
hereinafter referred to as "**SRM**"

AND

Name: Gunda Sumanth

Location: Bhubaneswar

Roll Number: RA1811003010701

Graduation Year: 4th Year

enrolled as a student in the B.Tech Programme offered by SRM hereinafter referred to as "**Candidate**"

The above-named parties have agreed to the following:

1. Introduction:

The Highako program is an industry-institute partnership initiative of HighRadius for engineering undergraduates. Candidates offered participation in this program shall be trained at the Centre of Excellence established at KIIT's premises at Bhubaneshwar, and, based on the internship programme for which such Candidates are eligible, shall be paid a fixed stipend for the duration of their internship. Upon successful completion of their engineering qualification and satisfactory performance during the training period, Highradius may, at its sole discretion, further employ these candidates in accordance with HighRadius' existing terms and conditions of employment. The Highako program shall be considered by SRM for giving academic credit to the Candidates, in accordance with the guidelines determined by HighRadius and SRM.

2. Terms of the Training Program conducted by HighRadius

2.1. HighRadius Internship Programme:

Highradius Technologies Private Limited,
4th floor, campus 3, KIIT, Bhubaneswar, Khordha, Odisha, 751024
www.highradius.com



ACKNOWLEDGEMENTS

I express my humble gratitude to **Dr. C. Muthamizhchelvan**, Vice Chancellor, SRM Institute of Science and Technology, for the facilities extended for the internship work through on campus placement and his continued support. I extend my sincere thanks to **Dr. Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her invaluable support. I wish to thank **Dr. M. Pushpalatha**, Professor & Head, Department of Computing Technologies, SRM Institute of Science and Technology, for her valuable suggestions and encouragement throughout the period of the internship work.

I am extremely grateful to our Panel Head, **Dr.C.N.Subalalitha**, Associate Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for her inputs during the monthly reviews. I register my immeasurable thanks to my Faculty Advisor, **Mr. H. KARTHIKEYAN**, Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for leading and helping me to complete my course.

My inexpressible respect and thanks to my guide, **Ms. MALAR SELVI G**, Assistant Professor, Department of Computing Technologies, and **Mr. Manas**, Project Lead, Highradius for providing me an opportunity to pursue my internship under his mentorship. He provided me the freedom and support to explore the work of my interest.

I sincerely thank staff and students of the Computing Technologies Department, SRM Institute of Science and Technology, for their help during my internship work. Finally, I would like to thank my parents, my family members and my friends for their unconditional love, constant support and encouragement [1].

Varun Kondreddy

Gunda Sumanth

ABSTRACT

Healthcare is a profession where precise record keeping and communication are vital, but where computing and networking technology is lagging behind other fields. Even though they agree that accurate record keeping and communication are crucial to successful healthcare, healthcare professionals and patients are generally uncomfortable with computers and believe that computers are not central to their purpose. Information is passed from one healthcare expert to another by paper notes or personal dialogue in today's healthcare. In the United States, for example, electronic contact between doctors and pharmacists is not common; instead, the doctor writes a prescription on paper and gives it to the patient. The patient brings the prescription to the pharmacy, stands in line to hand it over to a pharmacist, and then waits for the pharmacist to fill it. Prescriptions might be sent electronically from the physician to the pharmacy, and human computer interfaces for physicians, nurses, pharmacists, and other healthcare personnel could be voice-enabled to improve the process. According to Carmen Catizone of the National Association of Boards of Pharmacy, there are as many as 7,000 deaths in the United States each year as a result of erroneous prescriptions. According to a Washington Post story, up to 5 percent of the 3 billion prescriptions filled each year are inaccurate. These figures show that there is a pressing need to eliminate healthcare errors. In this paper, we present work on Creating an E-HEALTH PORTAL web application where the customer can look for Doctors, Test Labs, And Pharmacies near them. So that the customer can get the services such as appointment booking, Lab Tests, Buy medicines and Health insurance. Communication between patients, physicians, nurses, pharmacists, and other healthcare workers must be

effective and timely. The current communication systems, which are mostly focused on paper records and prescriptions, are antiquated, inefficient, and unreliable. The healthcare services supplied to a patient are frequently not coordinated when various healthcare experts and facilities are engaged. A physician often writes a prescription on paper and hands it to the patient. The patient brings the prescription to the pharmacy, stands in line to give it to the pharmacist, and then waits for the pharmacist to fill it. The pharmacist may be unable to read the doctor's handwriting; the patient may alter or falsify the prescription; or the doctor may be unaware of prescriptions provided by other doctors. SOA is used in our prototype distributed e-healthcare system to enforce basic software design principles and offer interoperability between diverse computer platforms and applications. Although our distributed e-healthcare system provides user-friendly interfaces for busy healthcare professionals and patients, security and privacy are especially crucial in this sector, so the prototype was created with security and privacy in mind. Only privileged users can see or alter data since the system authenticates users, logs session information, and links resources to the resource creator. E-HEALTH PORTAL can look for Doctors, Test Labs, And Pharmacies near them. So that the customer can get the services such as appointment booking, Lab Tests, Buy medicines and Health insurance.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	v
ABSTRACT	vi
LIST OF TABLES	x
LIST OF FIGURES	xii
ABBREVIATIONS	xiii
1 INTRODUCTION	1
1.1 Company Background	1
1.2 Work Responsibilities	2
2 TECHNOLOGIES	4
2.1 MONGODB	4
2.2 Express.js	5
2.3 React	6
2.4 Node Js	7
2.5 Merge Automations	8
2.6 Account Specific Analysis	10
3 SYSTEM ARCHITECTURE AND DESIGN	11
3.1 CLINIC MODULE	13
3.2 PHARMACY MODULE	13
3.3 ATOM/RSS MODULE	14
3.4 SPEECH MODULE	15
3.5 DATA FLOW DIAGRAM	15
3.6 Programming Environment	17
3.6.1 Programming Languages Used	17

3.6.2	Javascript Packages Used	18
3.6.3	Javascript Packages Description	18
3.6.4	Computer Specifications	20
3.7	Modules Description	20
3.7.1	Module 1: Book Appointments	21
3.7.2	Module 2: Book Lab Test	22
3.7.3	Module 3: Buy Medicine Module	24
3.7.4	Module 4: Login Module for customer/doctor	27
3.7.5	Module 5: Buy Health Insurance Module	33
4	CONTRIBUTIONS	35
4.1	Module 1 Implementation	35
4.2	Module 2 Implementation	36
4.3	Module 3 Implementation	37
4.4	Module 4 Implementation	38
4.5	Module 5 Implementation	39
5	LEARNING SKILLS AND EXPERIENCES	40
5.1	Non Technical	40
5.2	Technical	42
5.2.1	AEM	42
5.2.2	Figma	44
5.2.3	Java	46
5.2.4	Git	48
5.2.5	React Redux	49
6	CONCLUSION	50
REFERENCES		51

LIST OF TABLES

LIST OF FIGURES

1.1	Company Logo	1
1.2	MERN Stack	2
2.1	MONGODB	4
2.2	Express JS	5
2.3	React	6
2.4	NodeJS	7
2.5	Merge Automation	9
3.1	COMPLETE ARCHITECTURE	12
3.2	CLINIC MODULE	13
3.3	PHARMACY MODULE	14
3.4	DATA FLOW DIAGRAM-LEVEL0	15
3.5	DATA FLOW DIAGRAM-LEVEL1	16
3.6	DATA FLOW DIAGRAM-LEVEL1	16
3.7	Book Appointment Module UI 1	21
3.8	Book Appointment Module UI 2	21
3.9	Lab Test Module UI 1	22
3.10	Lab Test Module UI 2	22
3.11	Lab Test Module UI 3	23
3.12	Lab Test Module UI 4	23
3.13	Buy Medicine Module UI 1	24
3.14	Buy Medicine Module UI 2	25
3.15	Buy Medicine Module UI 3	26
3.16	Buy Medicine Module UI 4	27
3.17	Login UI 1	28
3.18	Customer Profile UI 1	29
3.19	Customer Profile UI 2	29

3.20	Customer Profile UI 3	30
3.21	Customer Profile UI 4	30
3.22	Doctor Dashboard UI 1	31
3.23	Doctor Dashboard UI 2	31
3.24	Doctor Dashboard UI 3	32
3.25	Doctor Dashboard UI 4	32
3.26	Doctor Dashboard UI 5	33
3.27	Buy Insurance Module UI 1	33
3.28	Buy Insurance Module UI 2	34
5.1	UN Sustainable Development Goals	40
5.2	Adobe Experience Manager	44
5.3	Figma Logo	46
5.4	Java Technology Logo	47
5.5	Git	48
5.6	Jenkins	49

ABBREVIATIONS

GDPR	General Data Protection Regulation
AML	Anti-Money Laundering
HDFS	Hadoop Distributed File System
DAG	Directed Acyclic Graphs
HiveQL	Hive Query Language
SQL	Structured Query Language
VM	Virtual Machine
YARN	Yet Another Resource Negotiator
RDD	Resilient Distributed Dataset
UI	User Interface
ASCII	American Standard Code for Information Interchange
API	Application programming interface
XCOM	Cross Communications
SMTP	Simple Mail Transfer Protocol
CLI	Command Line Interface

CHAPTER 1

INTRODUCTION

1.1 Company Background

HighRadius offers cloud-based Autonomous Software for the Office of the CFO. More than 700 of the world's leading companies have transformed their order to cash, treasury and record to report processes with HighRadius. Our customers include 3M, Unilever, Anheuser-Busch InBev, Sanofi, Kellogg Company, Danone, Hershey's and many more.

Autonomous Software is data-driven software that continuously morphs its behavior to the ever-changing underlying domain transactional data. It brings modern digital transformation capabilities like Artificial Intelligence, Robotic Process Automation, Natural Language Processing and Connected Workspaces as out-of-the-box features for the finance accounting domain. Finance business



Figure 1.1: Company Logo

stakeholders have been led to believe that they have only two choices: pick an application software vendor that digitizes a paper or Excel-based process to an electronic system of record, or, choose a middleware platform for AI or RPA to build and maintain in-house, domain-specific capabilities. In contrast, HighRadius Autonomous Software combines the best of both worlds to deliver measurable business outcomes.

Efficiency and productivity enhancements are central to the value HighRadius provides to our customers. Regardless of what ERP, Accounts Receivable or Treasury Management system you are using, our products automate manually-intensive tasks, streamline communication, and allow standardization of processes to drive best practices into your receivables and treasury processes.

We empower our customers to be able to work more accurately and efficiently, forecast and manage cash, get paid faster, and improve key metrics like Days Sales Outstanding (DSO) and improve working capital availability. [1].

1.2 Work Responsibilities

After Joining the Company, everyone has to go through different types of mandatory training, which includes topics like HTML, CSS, JAVASCRIPT, React, etc. The most important one here is React. Since the Company primary tech stack is react, every developer in this company must learn react. React is a JavaScript library for building user interfaces.[2]. The primary work in this company is to build full stack projects using MERN stack. Full Form of MERN stack is MONGODB EXPRESS REACT NODEJS. MONGODB is a nosql framework which is used as a database, nodejs and express is used as backend language and react is used for frontend.

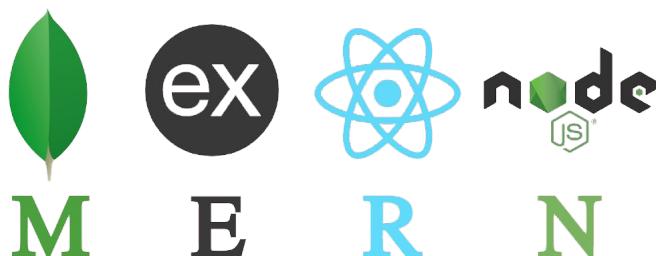


Figure 1.2: MERN Stack

As a new joiner to development Team, one has to go through the set of documentation to understand what the team does and how it helps the company achieve its goals. MERN stack is a collection of technologies that enables faster application development. It is used by developers worldwide. The main purpose of using MERN stack is to develop apps using JavaScript only. This is because the four technologies that make up the technology stack are all JS-based. The team's goal is to

build a health management system which we named as E-HEALTH PORTAL. This is achieved by building a website using MERN stack. As a developer in the team, one is responsible for creating efficient frontend and backend of the website. Once the website is complete the tester testes all the usecases of the website and approves the website.

The data Analytics also play a key role in this project as the data need to be analyzed and merged based on the patients interests. We use merge algorithms for the same in order to merge the claims and deductions which occurs during the payments.

CHAPTER 2

TECHNOLOGIES

As a Developer, one has to work with a lot of technologies. A developer has to work with lot of technologies it is because they have to use the technologies which is trending in the world. It is said that every one or two year new development technology arises. Since we know that right now the trending development technologies is MERN stack so in Highradius company we are using MERN stack. This project's tech stack includes MONGODB, EXPRESS, REACT and NODEJS.

2.1 MONGODB

MongoDB, the most popular NoSQL database, is an open-source document-oriented database. The term ‘NoSQL’ means ‘non-relational’. It means that MongoDB isn’t based on the table-like relational database structure but provides an altogether different mechanism for storage and retrieval of data. This format of storage is called BSON (similar to JSON format). [3].

SQL databases store data in tabular format. This data is stored in a predefined data model which is not very much flexible for today’s real-world highly growing applications. Modern applications are more networked, social and interactive than ever. Applications are storing more and more data and are accessing it at higher rates.



Figure 2.1: MONGODB

MongoDB is preferred over RDBMS in the following scenarios:

Big Data: If you have huge amount of data to be stored in tables, think of MongoDB before RDBMS databases. MongoDB has built-in solution for partitioning and sharding your database.

Unstable Schema: Adding a new column in RDBMS is hard whereas MongoDB is schema-less. Adding a new field does not effect old documents and will be very easy.

Distributed data Since multiple copies of data are stored across different servers, recovery of data is instant and safe even if there is a hardware failure. [3].

2.2 Express.js

Express.js is a small framework that works on top of Node.js web server functionality to simplify its APIs and add helpful new features. It makes it easier to organize your application's functionality with middleware and routing. It adds helpful utilities to Node.js HTTP objects and facilitates the rendering of dynamic HTTP objects. [4].



Figure 2.2: Express JS

Express gets installed via the Node Package Manager. This can be done by executing the following line in the command line

```
npm install express
```

The above command requests the Node package manager to download the required express modules and install them accordingly.

Let's use our newly installed Express framework and create a simple “Hello World” application.

Our application is going to create a simple server module which will listen on port

number 3000. In our example, if a request is made through the browser on this port number, then server application will send a ‘Hello’ World’ response to the client. [4].

2.3 React

React uses a declarative paradigm that makes it easier to reason about your application and aims to be both efficient and flexible. It designs simple views for each state in your application, and React will efficiently update and render just the right component when your data changes. The declarative view makes your code more predictable and easier to debug. A React application is made of multiple components, each responsible for rendering a small, reusable piece of HTML. Components can be nested within other components to allow complex applications to be built out of simple building blocks. A component may also maintain an internal state – for example, a TabList component may store a variable corresponding to the currently open tab.[5].



Figure 2.3: React

Let’s say one of your friends posted a photograph on Facebook. Now you go and like the image and then you started checking out the comments too. Now while you are browsing over comments you see that the likes count has increased by 100, since you liked the picture, even without reloading the page. This magical count change is because of Reactjs. React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It’s ‘V’ in MVC. ReactJS is an open-source, component-based front end library responsible only for the view layer of the application. It is maintained by Facebook.[6].

2.4 Node Js

Node.js is an open-source and cross-platform runtime environment for executing JavaScript code outside a browser. You need to remember that NodeJS is not a framework and it's not a programming language. Most people are confused and understand it's a framework or a programming language. We often use Node.js for building back-end services like APIs like Web App or Mobile App. It's used in production by large companies such as Paypal, Uber, Netflix, Walmart, and so on.[?].



Figure 2.4: NodeJS

Features of NodeJS:

1. It's easy to get started and can be used for prototyping and agile development.
2. It provides fast and highly scalable services
3. It uses JavaScript everywhere, so it's easy for a JavaScript programmer to build back-end services using Node.js
4. Source code cleaner and consistent.
5. Large ecosystem for open source library.
6. It has Asynchronous or Non-blocking nature.[?].

2.5 Merge Automations

This play is about how we can leverage tools like Jupyter Notebook to ultimately achieve the goal of 90 percent Merge. This automation increases the efficiency of HighRadius Client as the claim document will be automatically attached to the Deduction. We achieve the required automation in two steps - Executing the out-of-box in-house built algorithm in Jupyter to find patterns that translate into rules.

Finding account specific rules in case step 1 is not enough to reach the goal. Merge is a data related problem where you are trying to find a matching claim for a Deduction. Patterns exist in the data because each customer follows a strict process while raising a claim and referencing it on a Deduction. We use Jupyter as a tool to identify these patterns in the data and translate them into rules.

Sources of Data Data Fetching Merge Automation being a data-oriented analysis, it is necessary to understand the various types of entities involved in the analysis. This will give an understanding of the relevance of such entities and will also describe the source or location of such entities. In the present day, there are three types of entities important in the Merge Automation use case. The same is enlisted below,

1. Claim -t is raised by the buyer of goods asking for a reimbursement from the seller due to any contract-based allowance (Trade Promotion) or loss due to negligence (Non Trade Deduction - Shortage/Damage/Returns etc).
- 2.Pre-Deduction - The business rationale for creating a copy of the claim is to give the Deduction Analyst at the seller's end a chance to research the expected Deduction before it enters.
- 3.Deduction - This document usually comes in response to a short payment on an invoice. But there could be Deductions not related to invoices as well. Such entities are known as Standalone Deductions. These Deductions are usually a part of any Trade Promotion activity and are independent of invoices.

[4].

This approach has a limitation as we are dependent on the limited knowledge of

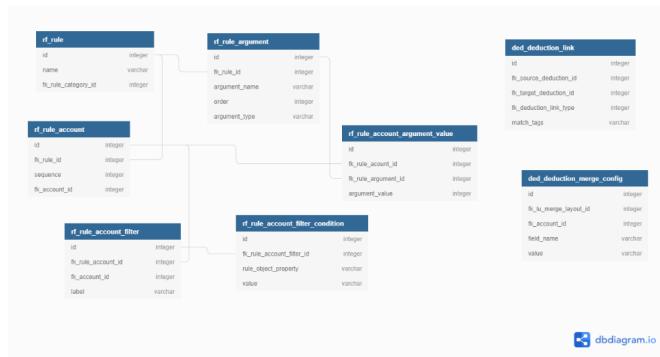


Figure 2.5: Merge Automation

the analyst. Hence we are moving towards a more data-oriented approach making the design more data driven. A Data-driven approach ensures patterns for all customers are covered. The algorithm can evaluate patterns existing in multiple attributes hence not limiting to just known attributes. The algorithm also reduces the chances of error.

1.Pre-Deduction to Deduction Merge Rules Configuration Following are basic configurations steps: Normalization of Data: Normalization of data rules to be added for massaging the data of specific fields.

2.Deductions to Claims Matching: For a given Deduction, these configurations help to identify the matching Pre-Deduction. Rules like Claim Number Exact Match with Invoice Number can be configured. Multiple rules using specific combinations of fields along with conditions like Exact Match, Contains can be configured.

3.Linking: Here the Deduction is linked to a Pre-Deduction by checking the Customer Group and Amount field. Linking can be configured based on a Tolerance

4.Identify the combination of pairs that should be chosen to be merged. This configuration is done by DTC. Here the combination of the above Matching and linking rules are selected based on which Merge should be conducted.

5.The system will identify the combination and merge based on the configuration [4].

2.6 Account Specific Analysis

After doing the analysis based on the intelligent algorithms present in the provided Jupyter Notebooks, one can also try the Account Specific Analysis and Rules Identification. Account Specific Analysis will help find additional rules that can increase the merge rates. It might so happen that the provided notebooks and the merge algorithms fail to capture a set of complex patterns and that is where the process of conducting the Account Specific Analysis will help. Although there is a very less likelihood of finding rules after running the Merge Analysis Notebooks provided, but yet this activity might turn out to be useful.[?].

After finding the patterns, we inspect the unmerged Deductions and unmerged Pre-Deductions to gain some insight as to why some Deductions did not get merged.

This process mainly involves identifying the pattern present on both the sides(Pre-Deduction and Deduction), and comparing the counts of these patterns. This is critical to see if the series existing on Deduction exists in the Pre-Deduction or not. If not, then it is clear that the data is not getting aggregated correctly.

While comparing the counts you will mainly come across the following scenarios:
The pattern is present on the Deduction side but not on the Pre-Deduction side, The pattern is present on both sides.

For both of these cases, it could be that the claims are missing or claims are not getting aggregated into Pre-Deductions. We can further confirm if the claims are really missing by checking the customer claim table. [?].

CHAPTER 3

SYSTEM ARCHITECTURE AND DESIGN

Clinical Module:

Input: Physician Login Page. Enter the username and password.

Process: Clinical staff services Add patient data / notes Forwards the prescription to the patient and pharmacy. Make referrals to other physician Patient services: Make appointments Deliver notification

Output: Select the menu bar. Patient Diagnosis data. Patient appointment[7].

Pharmacy module:

Input: Pharmacy Login page.

Process: Pharmacy Services: Receive / view / fill prescriptions Patient Services Enable renewals of prescriptions Provide notification Output: Menu bar (delivery date, today delivery, medicine details). Select any one from menu bar.

Atom/RSS module: Input: This module will run only in the Exception case and input is not needed (Web services not running). Process: The contents are converted as XML format and stored in local disk. Output: Show the Patient diagnosis data for this module.

Speech module: Input:Text to the diagnosis data Output:Text to speech only. This speech module will be available in mostly all indian languages. This module is very useful because many patients don't have time to write their query.

So, by speaking they can quickly tell their problem. Many indians are not able to write, so for them they can speak their query.

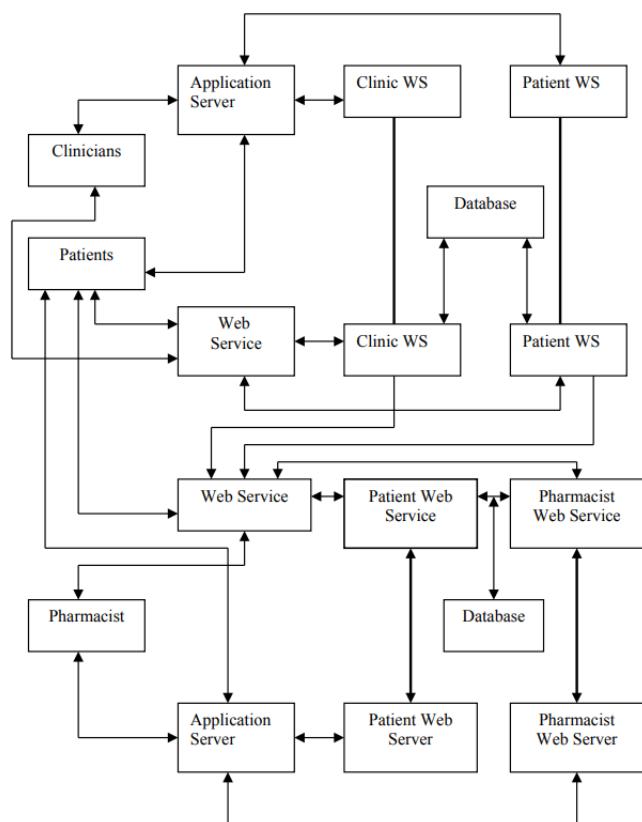


Figure 3.1: COMPLETE ARCHITECTURE

3.1 CLINIC MODULE

The Clinic module exposes two interfaces, a Web Server and a Web Service, for the clinic staff, the patients and the medical monitoring devices. The Web Server interface is intended for users who prefer to use a Web browser to access the healthcare services. Humans or devices to communicate with the e-healthcare system can use the Web Service interface. The Web Server uses the Web Services to access the data. .

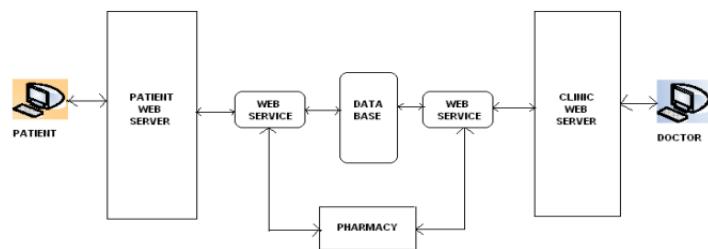


Figure 3.2: CLINIC MODULE

The Clinic module provides support for routine activities of the physician. It maintains information, such as the physician's appointments for a specific day/week, the patients that s/he has examined, notes related to the patients, etc. The Clinic module sends prescriptions from the physician to the desired pharmacies using the Web Service provided.

3.2 PHARMACY MODULE

The Pharmacy module exposes Web Server and Web Service interfaces. The Web Server interface allows the users to access the e-healthcare system at the pharmacy using a browser. The Web Service interface provides access for applications deployed at the pharmacy and can also be used by humans and devices. The Pharmacy module provides services to the pharmacist, patients and devices used at the pharmacy. The Pharmacy module keeps a record of the patient's prescriptions for the pharmacist's and the patient's reference. When the physician submits a new prescription to the pharmacy, the Clinic module at the physician's office communicates with the

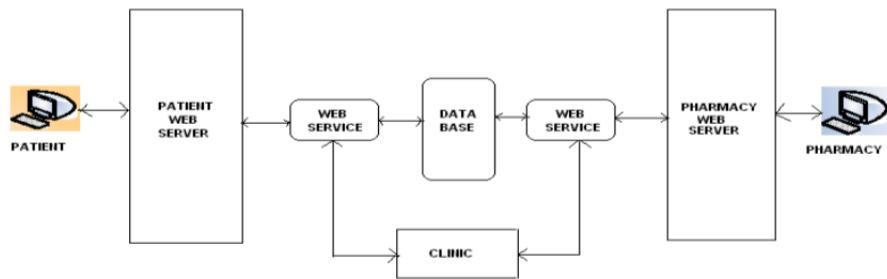


Figure 3.3: PHARMACY MODULE

Pharmacy module at the pharmacy. The pharmacist can view the outstanding prescriptions for the patients, as they are received from the physicians. The Pharmacy module updates the status of the prescriptions as the pharmacist fills them. The patient can determine, via the Web Server or Web Service, whether a prescription has been filled and is ready for pick up or delivery.

3.3 ATOM/RSS MODULE

Atom/RSS are syndication technologies, based on XML, that enable the sharing and communication of information between heterogeneous platforms by making the information self-describing. They allow a publisher to make information available to consumers on the Web, which retrieve that information subsequently. The information is delivered from the publisher to the consumer as an XML file, called an Atom/RSS feed.

This project develops a Consistent Data Replication (CDR) and Reliable Data Distribution (RDD) infrastructure that replicates information from one computer to another using Atom/RSS feeds. It uses this infrastructure to synchronize information on the physician's desktop or server computer with that on his/her PDA, allowing the physician to view that information when it is offline. At the start of the day, our software on the PDA retrieves the necessary updates from the Clinic Web Service on the desktop or server computer via a wired or wireless network. Any modifications to the information on the PDA are stored locally on the PDA. At the end of the day, our software on the PDA generates an update feed for the Clinic Web Service on the

desktop or server computer to read.

3.4 SPEECH MODULE

Natural Voices provides a simple and efficient way to produce natural sounding device-to-human voice interaction. It can accurately and naturally pronounce words, and speak in sentences that are clear and easy-to understand, without the feeling that a computer is talking to the human. Natural Voices supports many languages, male and female voices, and the SAPI, Voice XML and JSAPI interface standards. Using Natural Voices, created text to- speech software for the prototype device, which runs in the background and accepts messages in Voice XML format.

3.5 DATA FLOW DIAGRAM

Below there are 3 data flow diagram of our project. We have divided the data flow diagrams into different level so that it is easy to explain our project.

DATA FLOW DIAGRAM

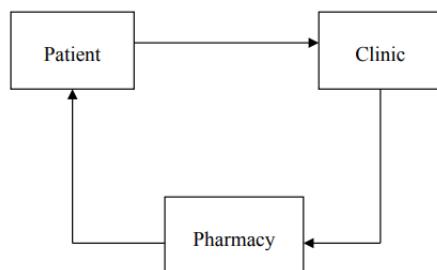


Figure 3.4: DATA FLOW DIAGRAM-LEVEL0

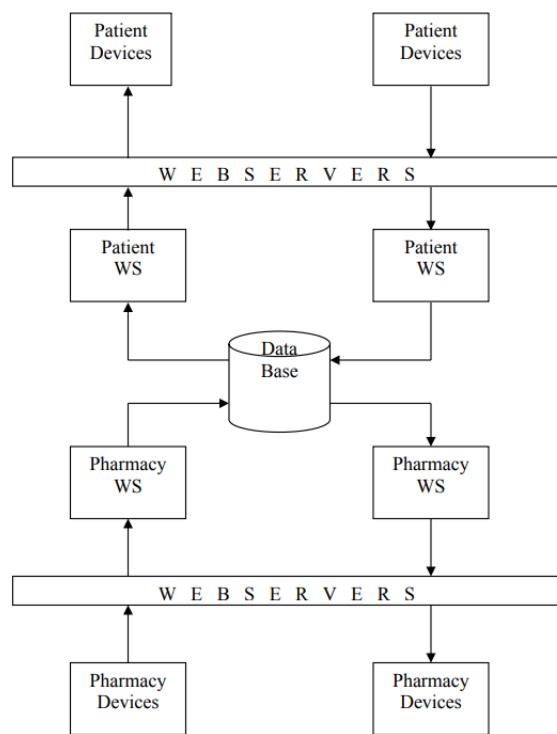


Figure 3.5: DATA FLOW DIAGRAM-LEVEL1

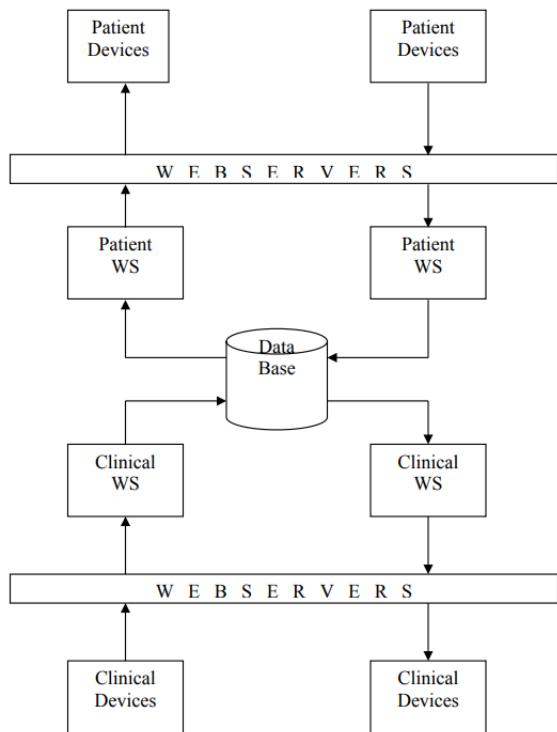


Figure 3.6: DATA FLOW DIAGRAM-LEVEL1

3.6 Programming Environment

3.6.1 Programming Languages Used

Javascript: JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

Python: Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

web development (server-side), software development, mathematics, system scripting.

3.6.2 Javascript Packages Used

- DOJO toolkit
- jQuery
- Google polymer
- JavaScript InfoVis toolkit
- D3.js
- Pixi.js
- SWFObject
- Three.js
- Velocity.js
- React js

3.6.3 Javascript Packages Description

1. **DOJO toolkit:** The Dojo is an open-source JavaScript library that helps develop cross-platform, JS, and Ajax-based web sites in a faster manner. DOJO has a vast set of APIs and modules. There are several parts in DOJO toolkit –
dojo – consists of the core modules dijit – user interface module that helps in setting up layout and widgets Util – includes utilities and build tools for testing, style checking, and more. dojox – this additional toolkit of dojo helps with charting, grids, and DTL. The modules in DOJO are independent and light-weight. They can be loaded asynchronously, which means a faster response. DOJO is simple to use and has a lot of community support and documentation.
2. **jQuery:** jQuery dramatically simplifies JS programming and is easy to learn and use. It is highly extensible and makes web pages load faster. jQuery wraps up a lot of standard functions making the job of the developer easy. A JS code of several lines could be just a method to be called in jQuery. It also has many plugins to perform different tasks. Some of the features of jQuery are CSS manipulation, HTML/DOM manipulation, HTML events, animations and effects, utilities, and AJAX. The best part of jQuery is the way it handles browser compatibility issues without the developer worrying about it. Some of the major IT companies like Microsoft, Netflix, and Google use jQuery. It is effortless to include jQuery in web pages.
3. **base64:** In Python the base64 module is used to encode and decode data. This module provides functions to encode binary data into printable ASCII characters and decode such encodings back to original binary data.

4. **Google polymer:** Created by Google, Polymer is a JS library that allows developers to reuse HTML elements and create custom elements using HTML, CSS, and JS to create more interactive applications. It is compatible with different platforms. Once you install Polymer using the command line interface or the Bower method, you can reuse already developed elements without worrying about how those were created. You can also build your custom elements using polyfills i.e., web component specifications. The custom elements can be distributed across the network and used simply by importing the required HTML. To install and use Polymer, you should be familiar with node.js, npm, Bower, Git, and Polymer CLI.
5. **JavaScript InfoVis toolkit:** If you want to create Data visualizations that are interactive and easy to pull together, InfoVis is the way to go. The toolkits some of the most advanced visualizations like Hypertree, TreeMap, RGraph, and more apart from the usual Bar, Pie, Area, and more.
6. **D3.js:** D3 stands for Data-Driven Documents. With D3, you can apply data-driven transformations to DOM objects. The keyword with D3 is ‘data-driven,’ which means documents are manipulated depending on the data received.
7. **Pixi.js:** Pixi js can create stunning digital content. This open-source, cross-platform 2D engine helps create games and interactive, animation-based websites.
8. **SWFObject:** This simple yet powerful JS library is mainly used to embed flash SWF files in HTML documents. It supports only SWF files, and no other media types can be embedded using SWFObject. There are two options – dynamic publishing and static publishing to add the markups. Many popular websites like Microsoft, YouTube, and more use SWFObject. To use this library, JS should be enabled on the user’s browser. SWFObject has many utility functions to do different tasks such as utilities to work with DOM or to identify if the latest version of the flash player is installed.

9. **Three.js:** Cross-browser JS library and API that allows for the creation of beautiful animations, Three.js relies on WebGL rather than conventional browser-plugins.
10. **Velocity.js:** Web animation has become a breeze with velocity.js. It is a fast performance engine that simplifies website animations. It is free, open-source software. Velocity is as fast (or more) as CSS and jQuery, especially on mobile devices.

3.6.4 Computer Specifications

The Computer Specifications are confidential and have not been shared by the company. However, I have replicated the entire project on my personal laptop using Virtual Machine (VM) technology. VM Specifications are as follows:

- **Operating System:** Windows 11
- **CPU:** Intel Core i7
- **RAM:** 8GB
- **Storage:** 120GB

3.7 Modules Description

The Project is divided into five important modules:

1. **Module 1:** Book Appointments
2. **Module 2:** Book Lab Test
3. **Module 3:** Buy Medicine
4. **Module 4:** Login Module for customer/doctor
5. **Module 5:** Buy Health Insurance

These five modules, together help build the entire website. Each of the module is necessary for application to be executed successfully.

3.7.1 Module 1: Book Appointments

Appointment module in our project arranges the schedule of doctors due to the patients' application. It helps to organize the availability of medical specialists at any convenient time. Some hospital can even offer remote visits when you need immediate assistance.

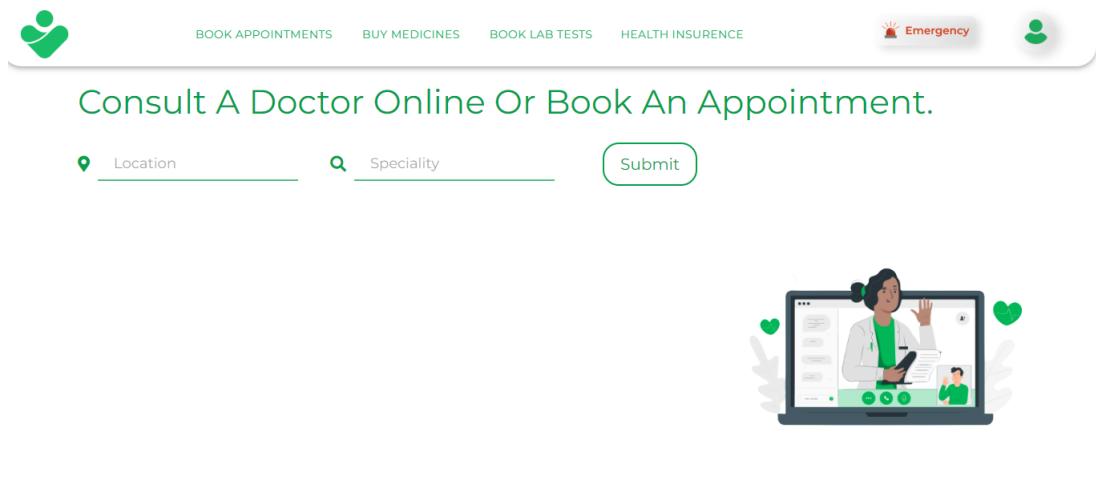


Figure 3.7: Book Appointment Module UI 1

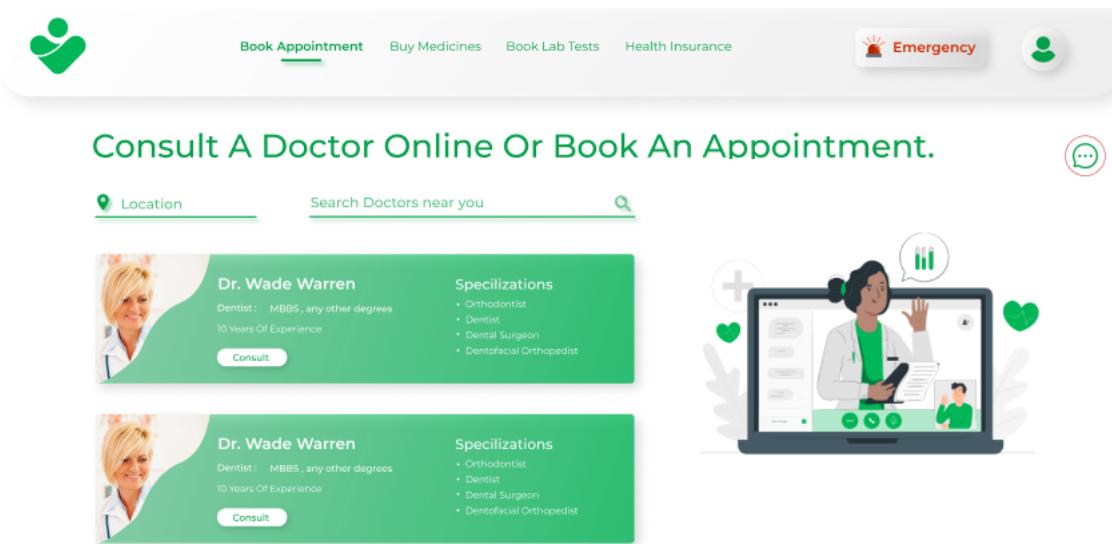


Figure 3.8: Book Appointment Module UI 2

3.7.2 Module 2: Book Lab Test

Lab management module shows the test results of the particular patient. The lab data can be viewed by the staff and generated for the patients' reports. It is usually integrated with other hospital information system modules for the better overall functionality of the system.

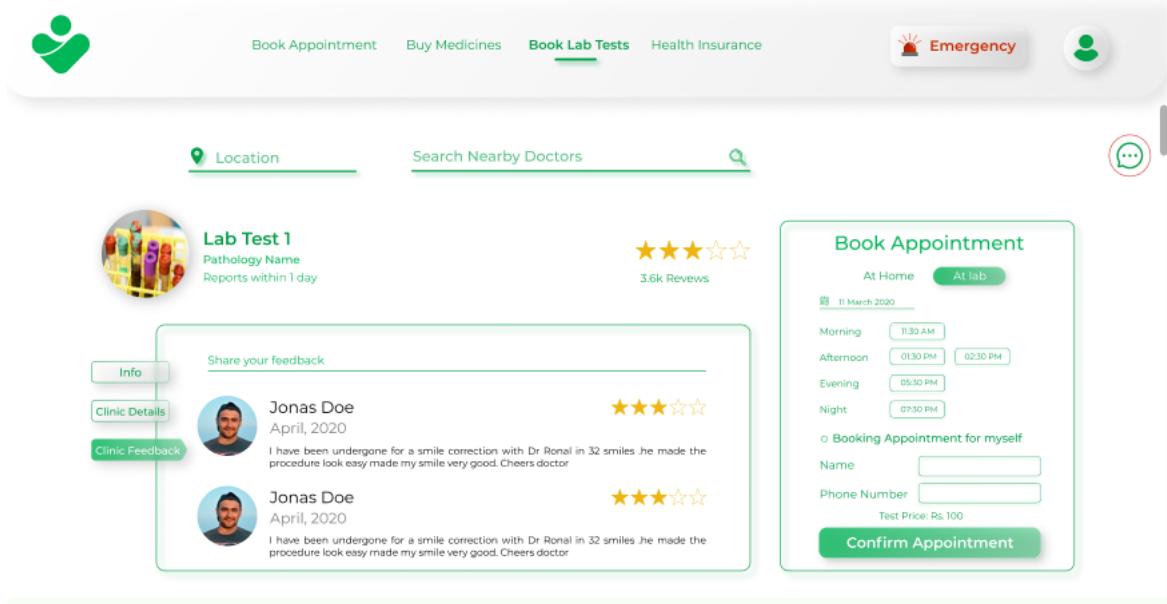


Figure 3.9: Lab Test Module UI 1

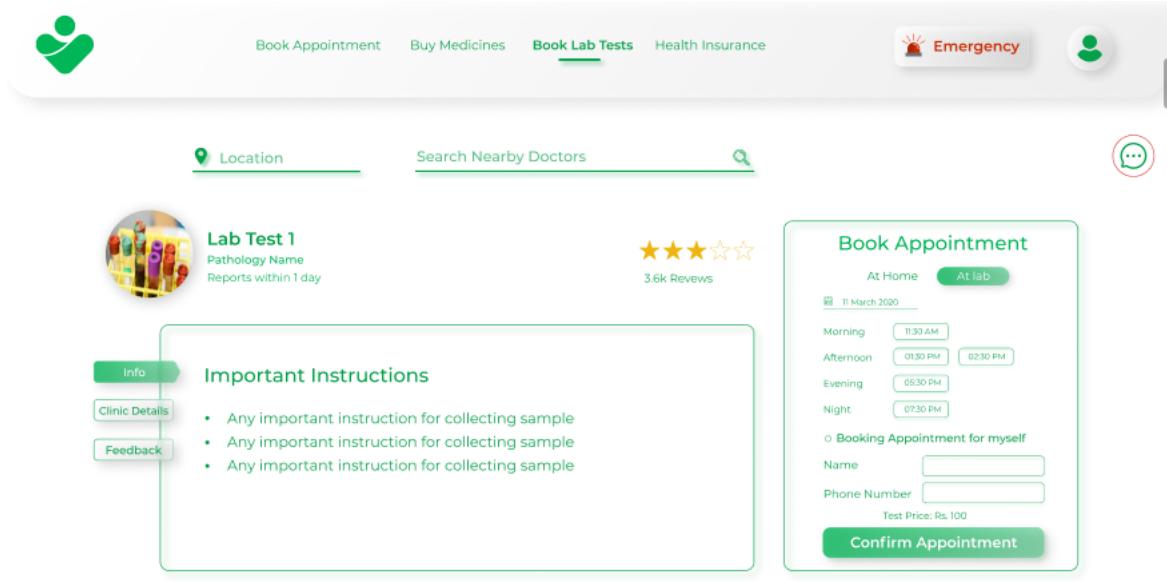


Figure 3.10: Lab Test Module UI 2

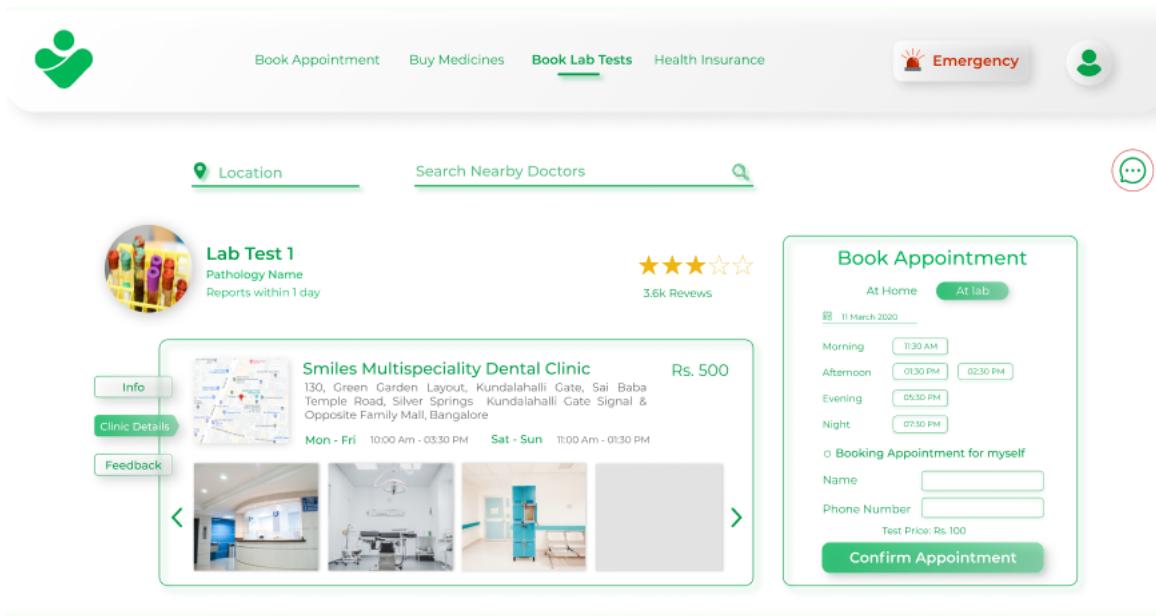


Figure 3.11: Lab Test Module UI 3

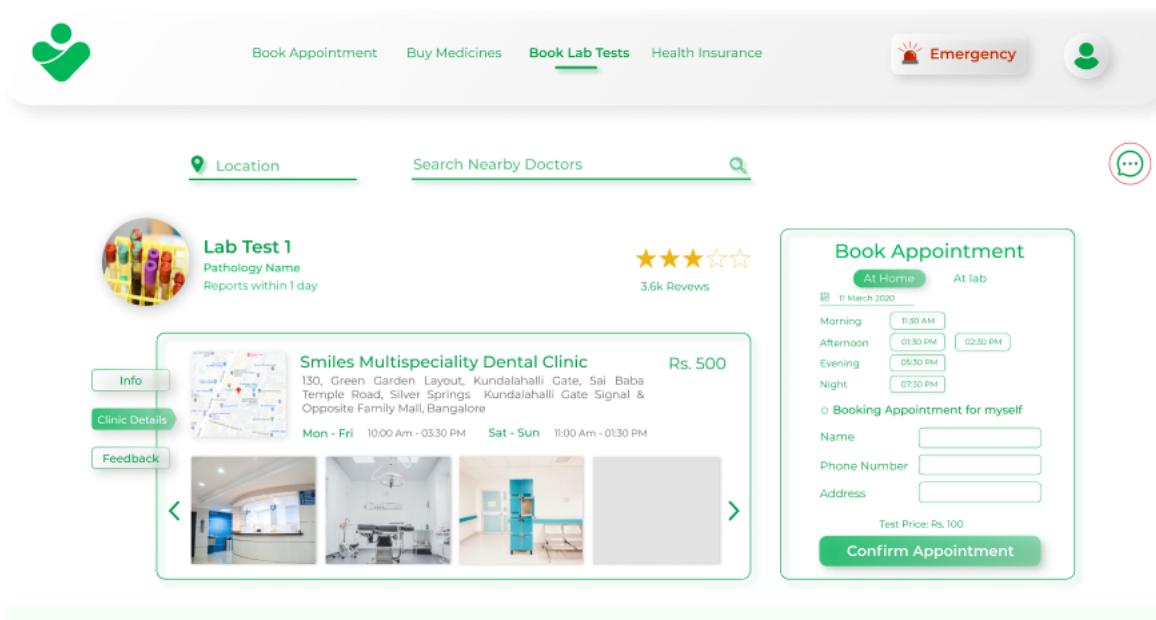


Figure 3.12: Lab Test Module UI 4

3.7.3 Module 3: Buy Medicine Module

Medicine management module contains the list of drugs that usually used for the specific treatment. It keeps records of every patient' drugs used during their treatment. In this module we will show medical shops to the users on basis of their location and the medicine they need. We have given them an input field for entering their location and the type of medicine they need. We will also give an option to the user to upload the prescription of their medicines so that they dont need to search for the medicine they need. Once the prescription is uploaded, our application will automatically find the nearest medical store and give them the prescription so that the medical store can deliver the required medicines.

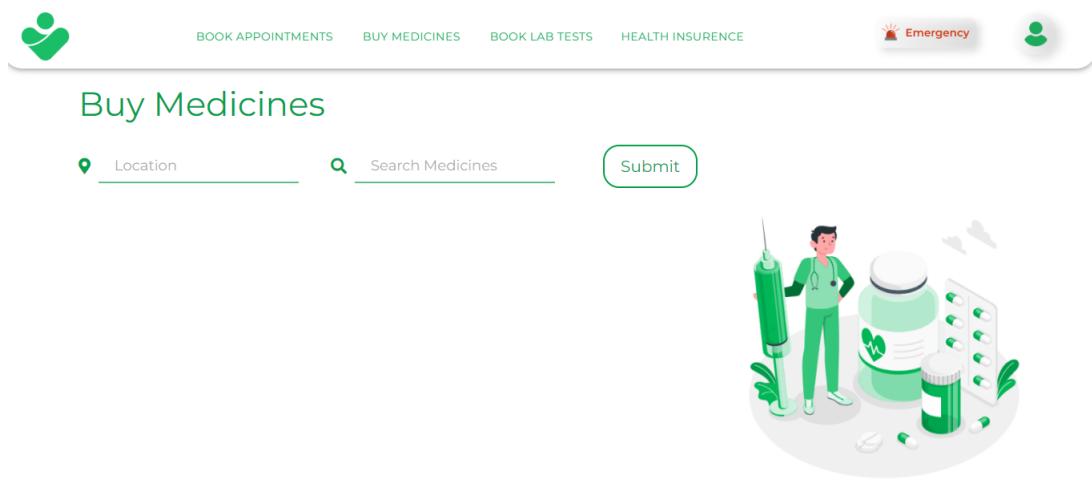


Figure 3.13: Buy Medicine Module UI 1

Medicine Name

* Prescription Required

★★★★★

₹ 10/tablet

Quantity

Buy Now

Add to Cart

Ingredients :
Amet minim mollit non deserunt ullamco est sit aliqua sint veniam consequat sunt nostrud amet.

Dosage :
Amet minim mollit non deserunt ullamco est sit aliqua sint veniam consequat sunt nostrud amet.

Estimated Delivery :
within a week from the day of order placed

Description **Reviews** **FAQ's**

- Manufactured By : Company Name
- Established since 1995, India's one of the largest pharmaceutical productions
- Company Headquaters : Company Address
- Mfg. Date : 11/2019
- Exp. Date : 07/2022
- For Complaints and Feedback email us at : company@email.com

Figure 3.14: Buy Medicine Module UI 2

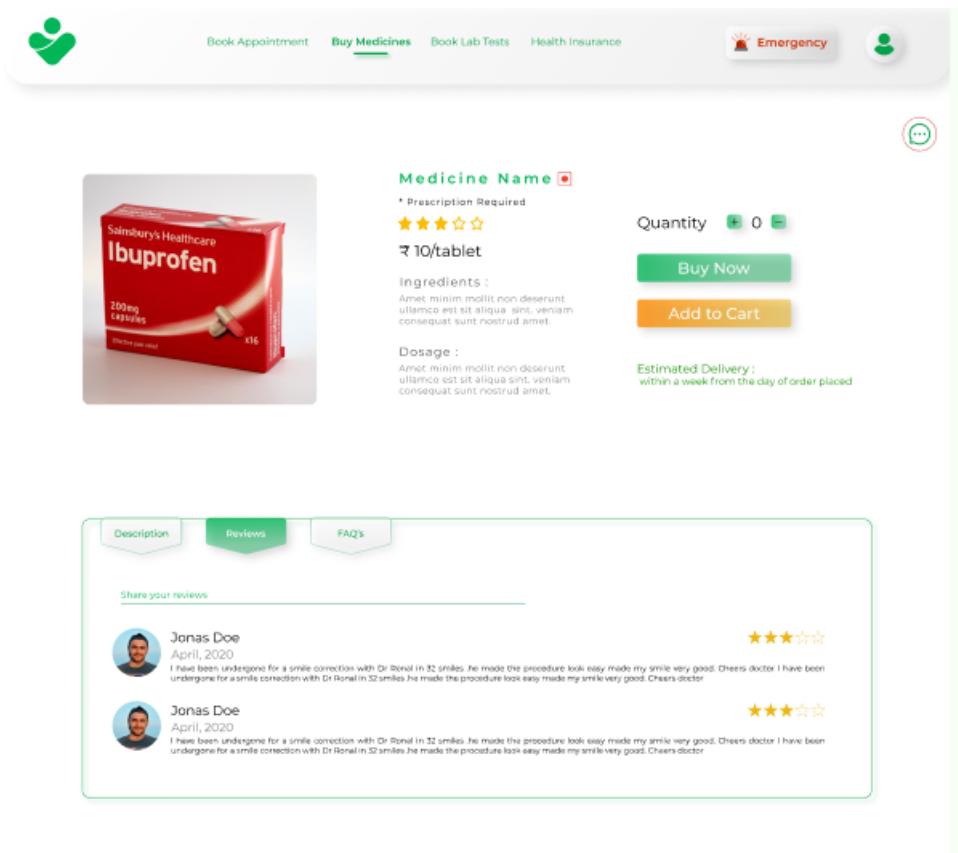


Figure 3.15: Buy Medicine Module UI 3

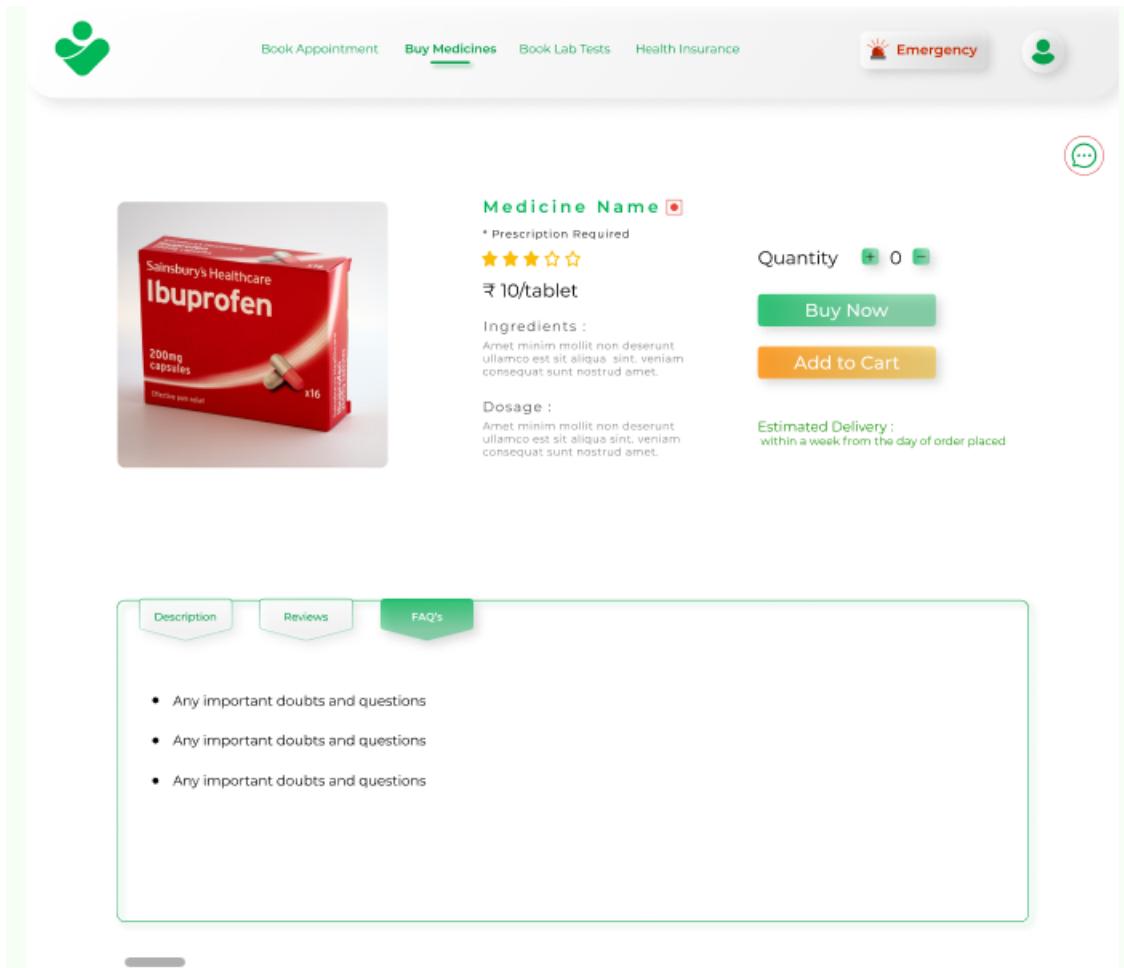


Figure 3.16: Buy Medicine Module UI 4

3.7.4 Module 4: Login Module for customer/doctor

In this module there will be login and signup page for customers and doctors. The doctors that are registered in our platform will be shown to customers when they want to book a doctor. When customer or doctor register in our platform they will have their own dashboard.

Users can use their gmail account, mobile number, instagram account, skype account to login in our application. Patients can see their uploaded report, previous orders, previous appointments etc. in their respective dashboard. Doctors can see their appointments in their respective dashboard.

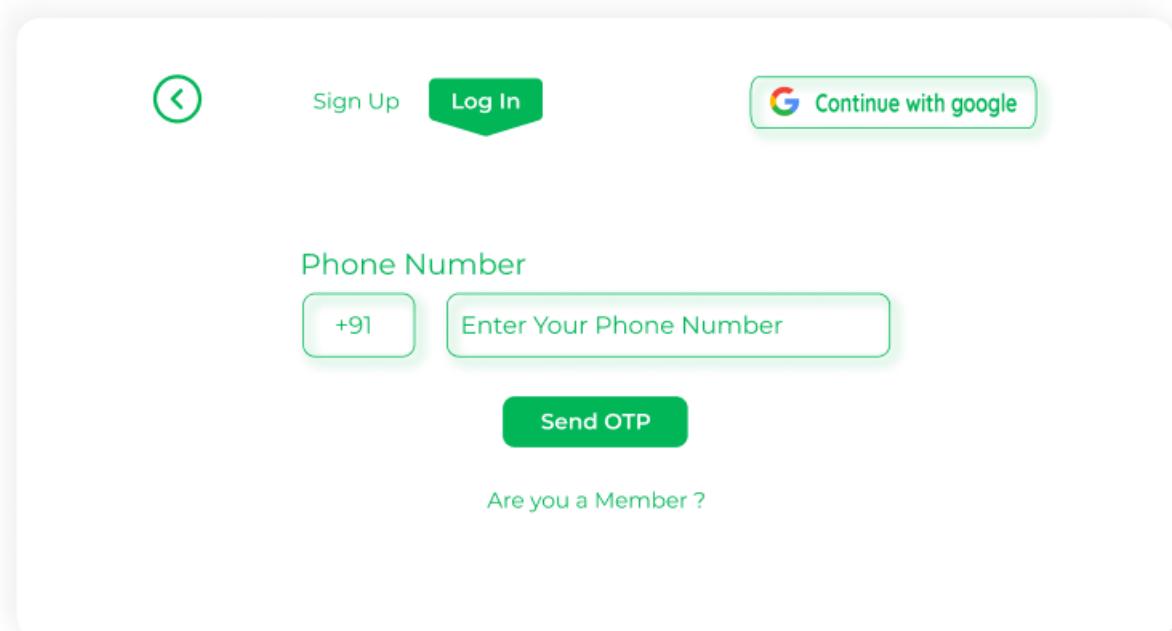


Figure 3.17: Login UI 1

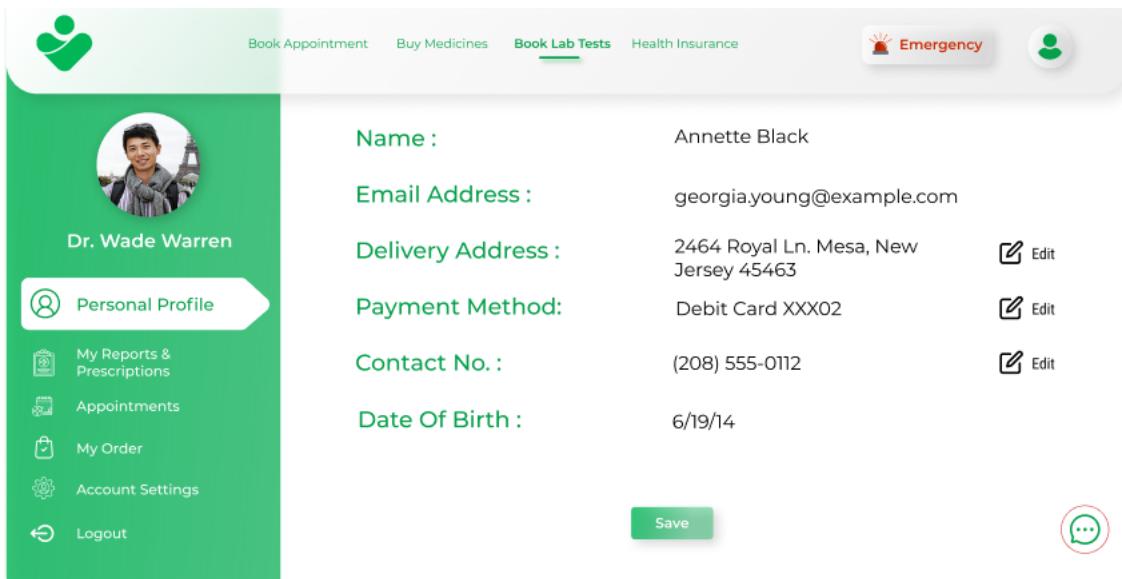


Figure 3.18: Customer Profile UI 1

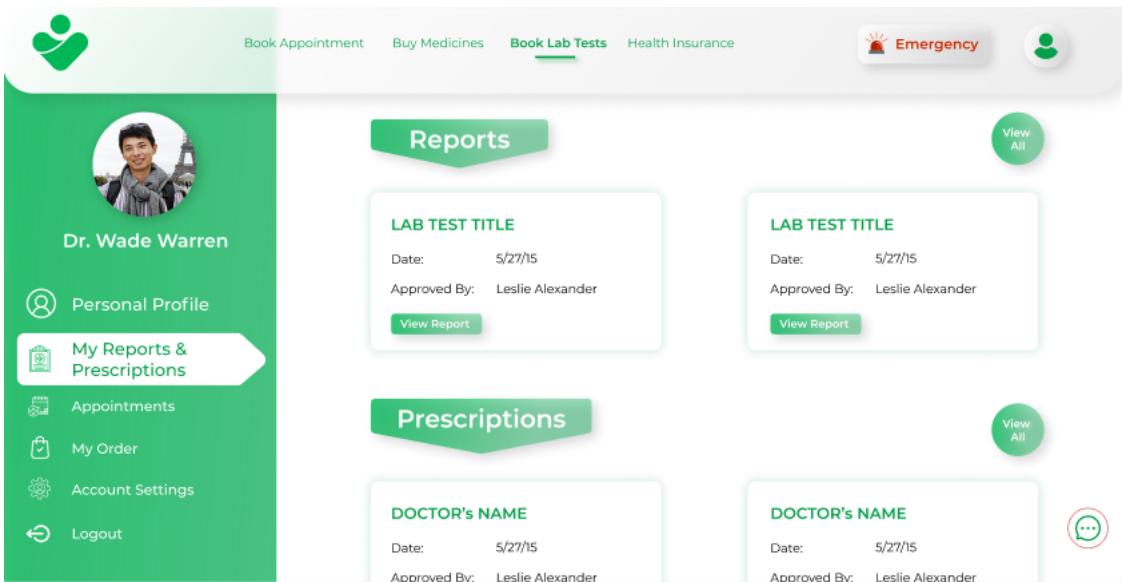


Figure 3.19: Customer Profile UI 2

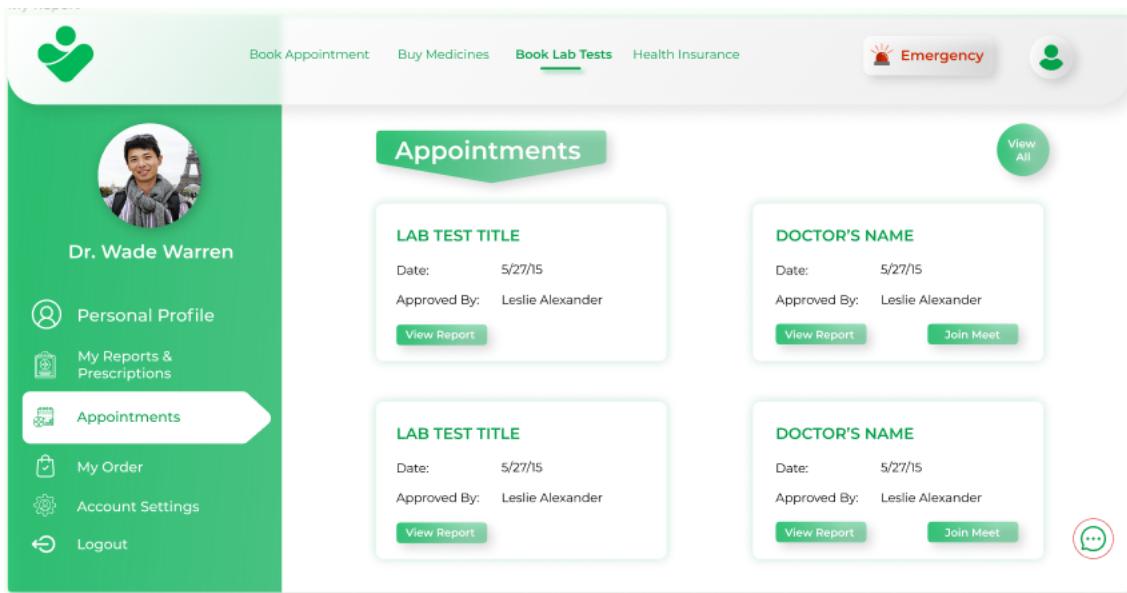


Figure 3.20: Customer Profile UI 3

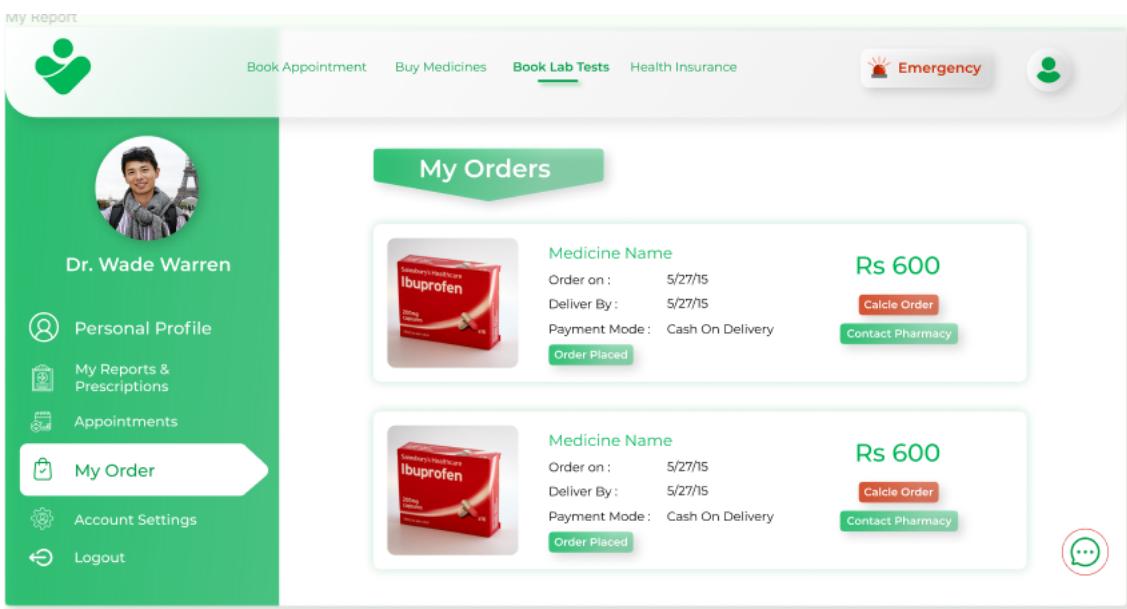


Figure 3.21: Customer Profile UI 4

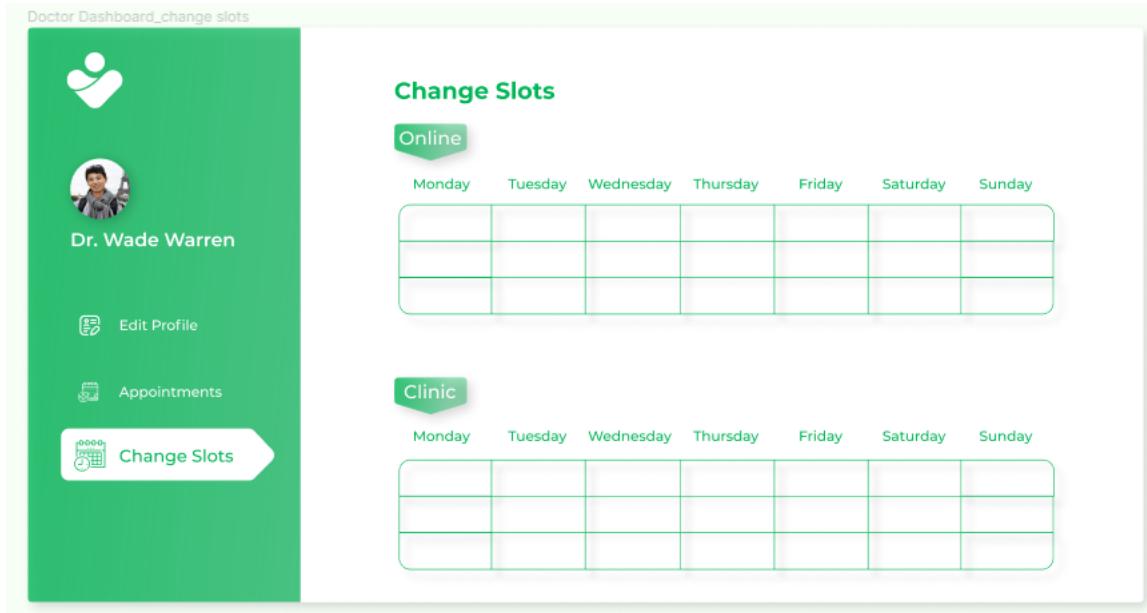


Figure 3.22: Doctor Dashboard UI 1

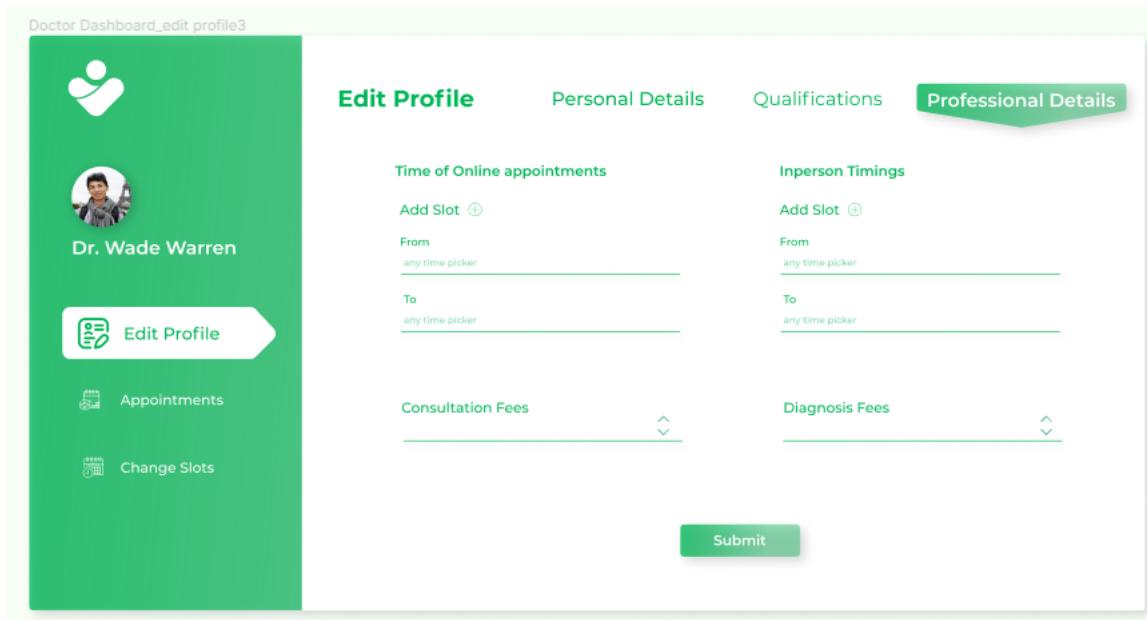


Figure 3.23: Doctor Dashboard UI 2

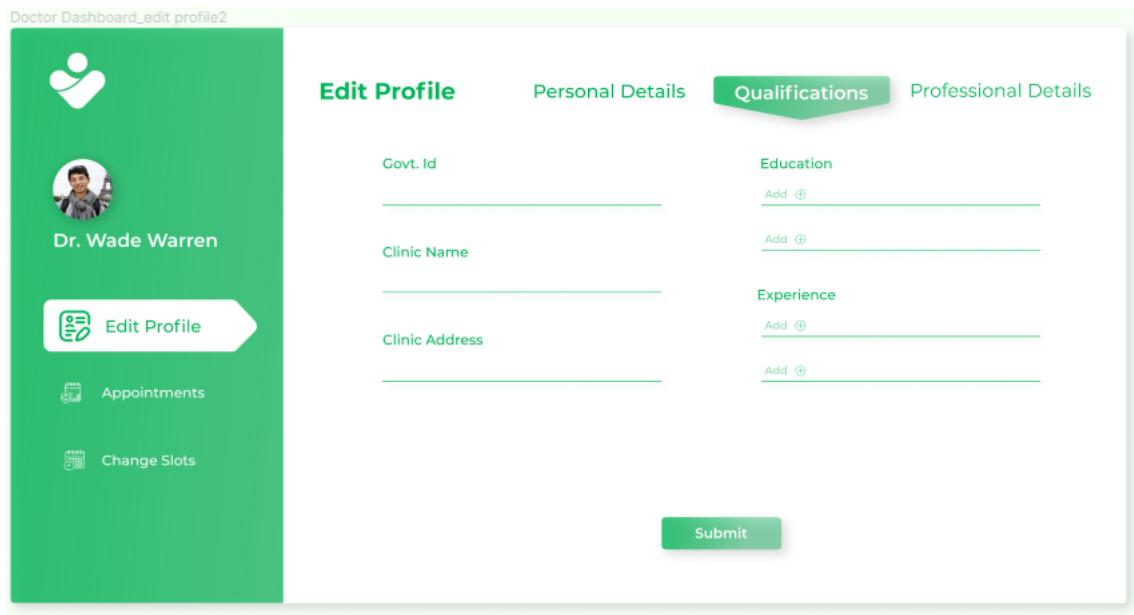


Figure 3.24: Doctor Dashboard UI 3

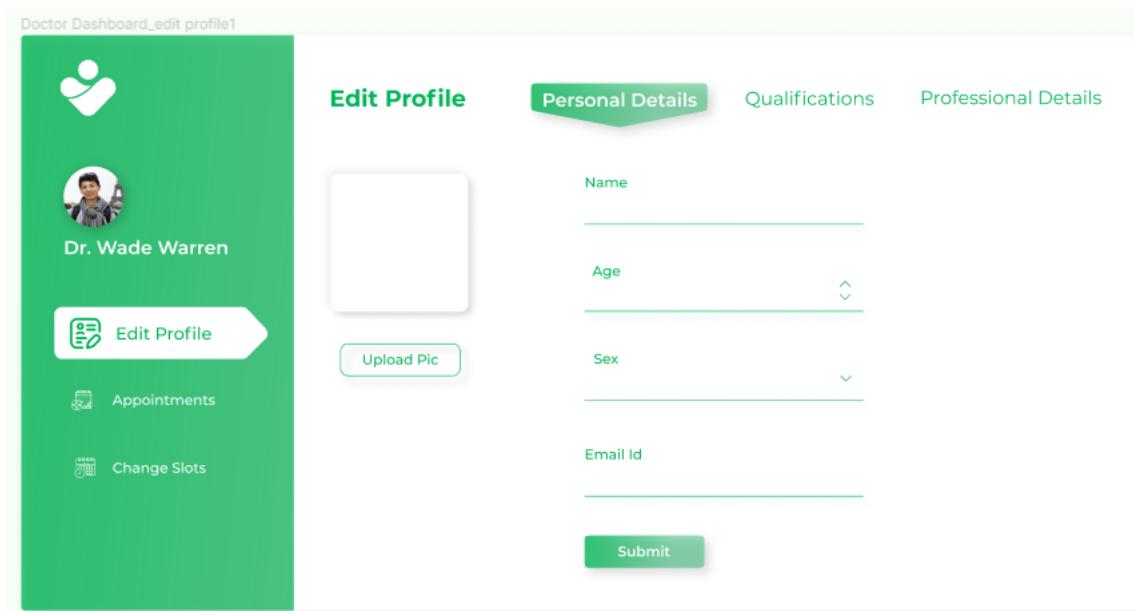


Figure 3.25: Doctor Dashboard UI 4

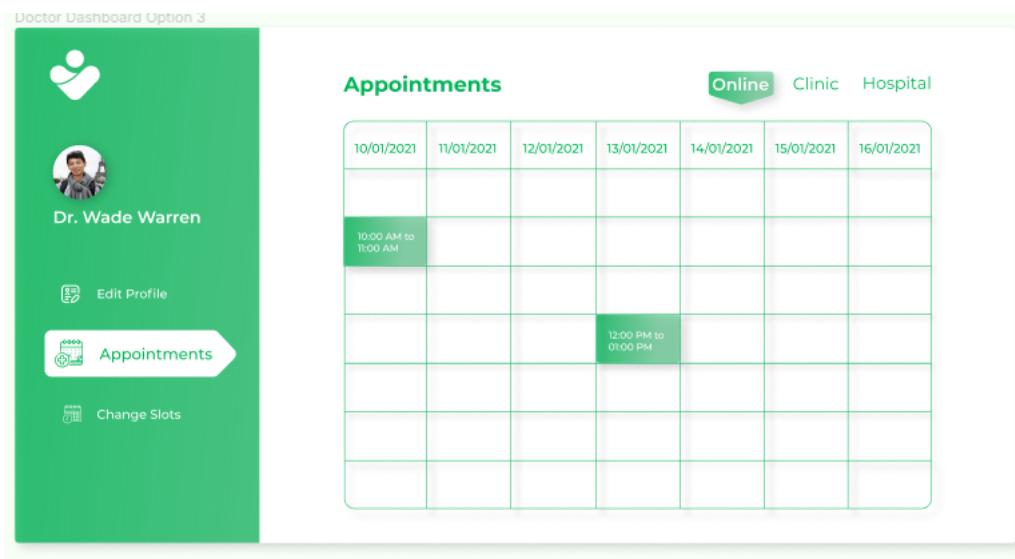


Figure 3.26: Doctor Dashboard UI 5

3.7.5 Module 5: Buy Health Insurance Module

In this module you can enter your location and find all the available health insurance plans which are there nearby you. In this module we will bring all the health insurance plans which are there in india. In this module there will also be a recommendation system where user will get recommended which health plan is best for them.

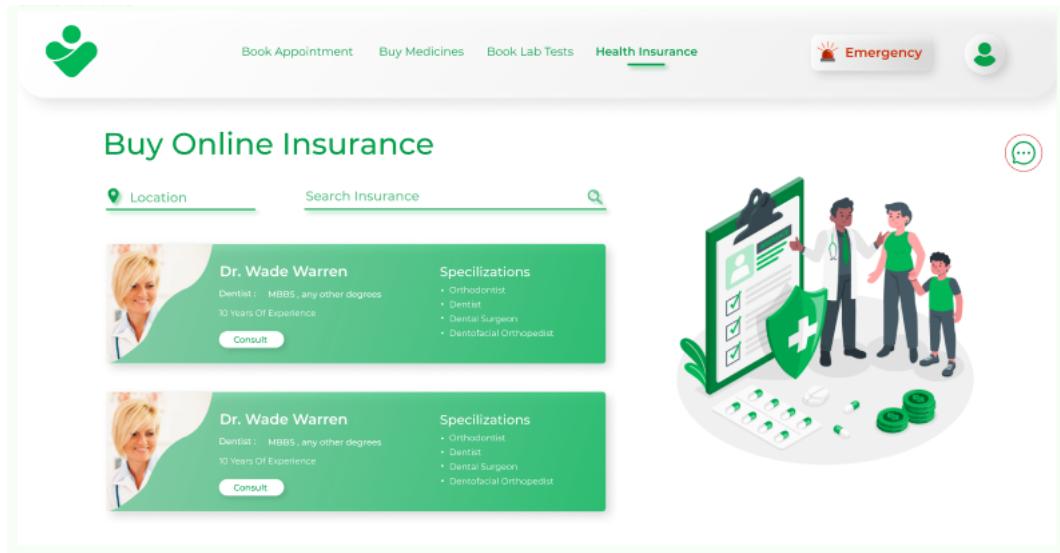


Figure 3.27: Buy Insurance Module UI 1

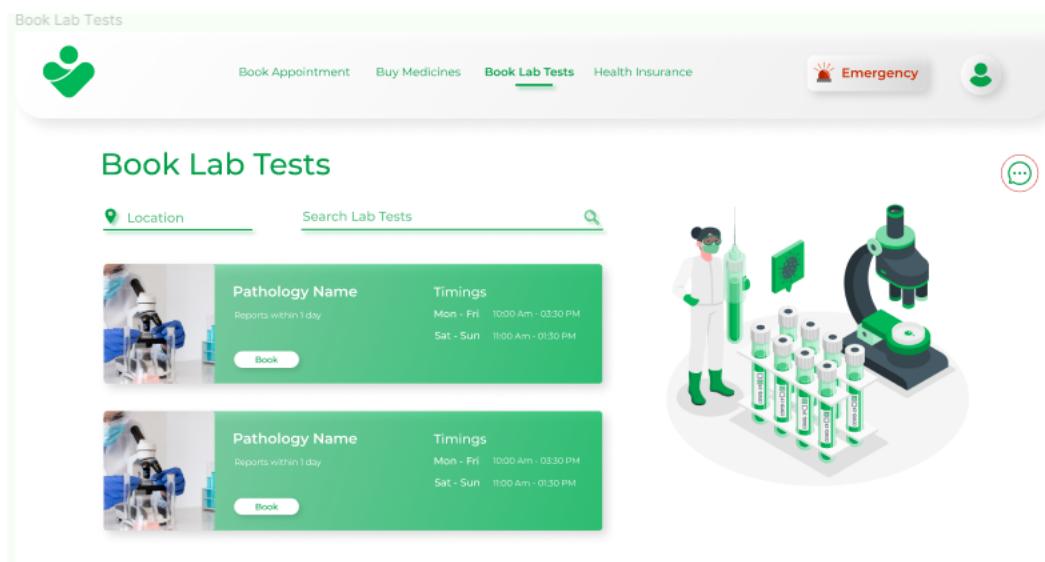


Figure 3.28: Buy Insurance Module UI 2

CHAPTER 4

CONTRIBUTIONS

I and my batch mate are doing the entire project from scratch with equal contribution and our mentor's help. We are building all five modules for this project from the beginning.

4.1 Module 1 Implementation

We came up with a solution to first create the User Interface, user interface means how the website will look after it is complete. For building UI we use a tool called figma. First we learned how to use the figma tool and then started building UI screens using figma. Once we are done with UI screens then our UI will be approved by our manager and if he did not like the UI we have to think a unique design and have to change the entire design. Once the UI is approved then we start developing the UI screens using MERN stack. The Figma link in which UI screens were made and the original code is confidential and hence it's not being shared in this report.

4.2 Module 2 Implementation

We came up with a solution to first create the User Interface, user interface means how the website will look after it is complete. For building UI we use a tool called figma. First we learned how to use the figma tool and then started building UI screens using figma. Once we are done with UI screens then our UI will be approved by our manager and if he did not like the UI we have to think a unique design and have to change the entire design. Once the UI is approved then we start developing the UI screens using MERN stack. The Figma link in which UI screens were made and the original code is confidential and hence it's not being shared in this report.

4.3 Module 3 Implementation

We came up with a solution to first create the User Interface, user interface means how the website will look after it is complete. For building UI we use a tool called figma. First we learned how to use the figma tool and then started building UI screens using figma. Once we are done with UI screens then our UI will be approved by our manager and if he did not like the UI we have to think a unique design and have to change the entire design. Once the UI is approved then we start developing the UI screens using MERN stack. The Figma link in which UI screens were made and the original code is confidential and hence it's not being shared in this report.

4.4 Module 4 Implementation

We came up with a solution to first create the User Interface, user interface means how the website will look after it is complete. For building UI we use a tool called figma. First we learned how to use the figma tool and then started building UI screens using figma. Once we are done with UI screens then our UI will be approved by our manager and if he did not like the UI we have to think a unique design and have to change the entire design. Once the UI is approved then we start developing the UI screens using MERN stack. The Figma link in which UI screens were made and the original code is confidential and hence it's not being shared in this report.

4.5 Module 5 Implementation

We came up with a solution to first create the User Interface, user interface means how the website will look after it is complete. For building UI we use a tool called figma. First we learned how to use the figma tool and then started building UI screens using figma. Once we are done with UI screens then our UI will be approved by our manager and if he did not like the UI we have to think a unique design and have to change the entire design. Once the UI is approved then we start developing the UI screens using MERN stack. The Figma link in which UI screens were made and the original code is confidential and hence it's not being shared in this report.

CHAPTER 5

LEARNING SKILLS AND EXPERIENCES

We have learned a lot of new concepts till now, from both technical and non-technical perspectives. We are grateful to the company for providing me with such a great learning experience.

We started with zero experience in this company, as this was my first internship. We have gained quite a good amount of experience not only on the technical side but also on how everything works in the corporate world.

5.1 Non Technical



Figure 5.1: UN Sustainable Development Goals

Company provided us a list of mandatory courses that we had to go through before working on a project. These training included topics like Code of Conduct, GDPR Basics, Competition Law, etc. I have finished all the mandatory training courses assigned to me before the due date. The List of all the mandatory training are as follows:

- **Sustainable Progress:** It provided us with basic training around Danske Bank's role in society and our ability to drive sustainable progress by creating sustainable long-term value for the stakeholders of the company.
- **Conflict of Interest:** It helped me understand what constitutes a conflict of interest and market abuse and what actions I needed to take in order to manage difficult situations where I might come into possession of inside information or spot any incidents of potential market abuse.
- **GDPR Basics:** I learned the basics of GDPR, and how GDPR can be used as a tool to identify and protect personal information. I learned the key fundamentals of being compliant with the GDPR at Danske Bank.
- **Sharing Concerns:** I learned how to share concerns in different situations, and it gave me a better understanding of why Whistleblowing system exist and how and when to use it.
- **Competition Law:** The purpose of this course was to ensure effective competition, which helps to ensure that all the people who work for the bank enjoy a wider range of higher quality products/services at a lower price and the rules are therefore a cornerstone of a well-functioning market economy that ultimately benefits them.
- **Risk Management:** Learned how each and everyone working for the bank are Risk Managers and learned how strong risk management skill help us to deliver high quality services to the customers and achieve our objectives. Learned how we can approach manager, if we sense something going wrong.
- **Code of Conduct:** I learned about the principles that constitute Code of Conduct and our purpose and culture commitments. It helped me understand what good conduct is, why good conduct is important and what company expects from us.
- **Financial Crime:** Learned about financial crime, money laundering, tax evasions. Learned why it is essential to fight financial crime and how AML systems can be used to fight such crimes.

Besides these courses, company also provided sessions on 'Introduction to Banking and Financial Markets' and a workshop on 'Campus to Corporate'. I learned a lot about banking, retail banking, corporate banking etc. In Financial Markets i got introduced to many concepts such as Trade finance, Foreign Exchange, etc. In 'Campus to Corporate'

workshop, I learned about what changes to expect from this transition. Learned how to behave professionally in corporate world. Learned how to be an effective team player, writing effective emails, Ownership and Accountability, etc.

5.2 Technical

The company provided us access to Udemy courses, even before joining the company officially. Company gave us plenty of time to explore and learn the new concepts that we needed to learn to work on the project once we join the company. Since we were joining development team, we was asked to learn MERN stack concepts (MongoDB, Express Js, React and NodeJs). My Mentor, Mr. Manas, also spent quite a time with me, to help me understand the project and he also thought me new concepts such as git, BitBucket, etc., which helped me a lot to build this project.

5.2.1 AEM

Adobe Experience Manager (AEM) as a Cloud Service is the latest offering of the AEM product line, helping you continue to provide your customers with personalized, content-led experiences. It provides cloud-native agility to accelerate time to value and is extensible to meet your unique business requirements. You can build on past investments and innovations by preserving and extending all your use cases and functionalities.

AEM as a Cloud Service lets you capitalize on the AEM applications in a cloud-native way, so that you can:

Scale your DevOps efforts with Cloud Manager: CI/CD framework, autoscaling, API connectivity, flexible deployment modes, code quality gates, service delivery transparency, and guided updates.

Enable developers to add automation to application development practices.

Deliver content quickly and efficiently on a global scale, using a built-in Content Delivery Network (CDN) and other network-layer best practices.

Leverage a dynamic architecture that auto-scales, thus removing infrastructure considerations.

Stay on top of threats and security-risk mitigation, using automated tests to scan for common vulnerabilities.

Ensure maximum resilience and efficiency backed by optimized performance topologies.

Take advantage of AEM as a Cloud Service's deep integration with the Adobe Experience Cloud to provide better customer experiences with online marketing and web analytics products.

Utilize tools that help accelerate the migration tasks, such as code refactoring, transfer of content, and more.

With AEM as a Cloud Service,

Developers can concentrate on configuring and extending AEM with the new cloud-based development pattern.

Content Authors can take advantage of the latest innovations delivered to them regularly.

System Administrators can minimize manual tasks in configuration and infrastructure maintenance.

Marketing Professionals can achieve faster time-to-value.

AEM achieves these goals by adopting the main characteristics of modern cloud services:

It is always on:

The entire service architecture has been revisited so that you do not experience any downtime; for neither the content management, nor the content delivery capabilities. It is always at scale:

All instances of AEM as a Cloud Service are created equal; so the service architecture will automatically scale, up and down, depending on your needs. It is



Figure 5.2: Adobe Experience Manager

always current:

AEM as a Cloud Service implements a new, continuous delivery pipeline for the AEM codebase, with automated updates up to several times a month. This solves one of the main challenges of AEM applications, by keeping you current on the most recent version. It is always evolving:

AEM as a Cloud Service evolves on a daily basis, based on the projects implemented by our customers. Content, code and configurations are constantly reviewed and vetted against best practices, allowing us to guide you on how to achieve your business goals.

5.2.2 Figma

The Team Library, a feature that makes Figma so uniquely powerful for teams who closely collaborate, is a way for designers to create, maintain, and share Components and Styles across all of their designs. The concept of components is nothing new to developers, as it's a framework that's existed within engineering for some time now as a way to build interfaces. However, for many of us designers, components are still a fairly new way of thinking. Many of us are still working to learn best practices for not

only creating and using components but also sharing their function with developers.

To begin to understand more about what it looks like when developers are inside of Figma files, I couldn't imagine better people to talk with than Lauren LoPrete, Natalie Annin, and Ben Munge from Expedia. Lauren and Natalie are both designers and work closely with Ben, a Senior Software Engineer who is working on the company's iOS applications. Lauren has spent much of her time at Expedia working on design systems and documentation within Figma and Natalie most recently designed and implemented dark mode component variants for the team's library.

Because developers at Expedia have access to in-progress design files, it's more important than ever to communicate how reusable components in designs are being used. One way to relay this information, and a feature that the team relies on, is Figma's ability to navigate to the master component's location from an instance.

When a component instance is selected, inside of the properties panel on the right is an option to "go to master component," which when clicked, will open the file where the component originated from. Lauren shared with me that having access to where these components were created makes it really easy for developers at Expedia to understand more about them, such as whether or not the component is a part of the larger design system or if it was created specifically for the project. The ability to view the master component also makes it possible to understand, which, if any, overrides were applied by the designer who used the component.

As developers get access to design files, which are filled with dozens of in-progress frames, iterations, components, and more, it's good practice to communicate which parts of the files are ready for implementation and which are still being worked on. File organization and communication were consistent trends among each of the design teams at Expedia, Dropbox, and Cash App and there were a few ways, in particular, that each team approached it.

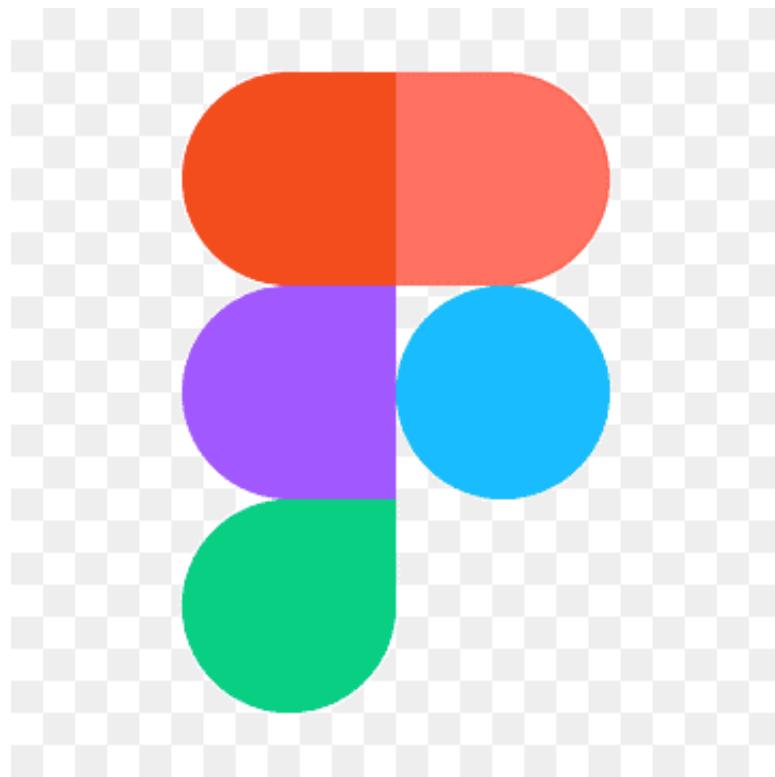


Figure 5.3: Figma Logo

5.2.3 Java

Java is one of the most popular and widely used programming language and platform. A platform is an environment that helps to develop and run programs written in any programming language. Java is fast, reliable and secure. From desktop to web applications, scientific supercomputers to gaming consoles, cell phones to the Internet, Java is used in every nook and corner.

Java is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Software Development Domain. I will list down some of the key advantages of learning Java Programming:

Object Oriented In Java, everything is an Object. Java can be easily extended since it is based on the Object model.

Platform Independent Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.



Figure 5.4: Java Technology Logo

Simple Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.

Secure With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.

Architecture-neutral Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.

5.2.4 Git

I learned about Git and how Git serves as a free and open source distributed version control system which is optimised for managing small to extremely big projects. I learned how Git as a software system can be used to monitor changes to any group of files and how it is used to coordinate work among programmers who are collaboratively developing source code during software development.



Figure 5.5: Git

I learned basic git commands such as:

- **git pull:** I learned how to clone repositories from BitBucket git repository using this command.
- **git add -all:** I learned how this command can be used to add all the changes made to all the files in local project directory.
- **git commit -m "commit message":** I learned how this command can be used to commit all the changes made to the code with a commit message.
- **git push:** I learned how it can be used to push the changes made to the code locally to the BitBucket git repository server.

5.2.5 React Redux

Redux is a predictable state container for JavaScript apps.

It helps you write applications that behave consistently, run in different environments (client, server, and native), and are easy to test. On top of that, it provides a great developer experience, such as live code editing combined with a time traveling debugger.

You can use Redux together with React, or with any other view library. It is tiny (2kB, including dependencies), but has a large ecosystem of addons available.



Figure 5.6: Jenkins

CHAPTER 6

CONCLUSION

The healthcare problem is being a huge worry in each and every family where patients are not able to choose the best doctors and spends the money hugely for medical treatments at later stages. The problems come in organizing all the hospitals and services so that any patient will be able to opt for the best doctor. Also, Lab tests are very important for a doctor to give a specific treatment. So, patients can't be able to find the nearest lab tests required for their particular health problem. Our motive is to make the process easier for a patient by improving their mental health by making it easier on searching for the hospitals, booking appointments, booking lab tests, Buying Medicines. Health insurance is really very necessary for everyone but people are not aware of the process in applying for specific kind of insurance in their locations so, we also built a service where they can buy the online insurance available near to them. This platform acts as problem solver between hospitals and patients with overall necessity a patient requires during his life.

It has been an incredible three months for me in this company. We were able to learn a lot from my mentor and the company. The workshops conducted by my mentor helped me elevate my skills. By only working for three months as an Intern, I have gained quite an amount of Industrial experience, as well.

Some companies do not allow working on live projects at the start of the internship, but that was not the case for us. Within 2-3 weeks of training and exposure to the company's work, I was put in a live project, where I was able to learn and work as well. My mentor also helped me in times of need. It is a great place to start my career with, We learned and enjoyed a lot while working for the company.

REFERENCES

- [1] Highradius, “Highradius Corporation,” <https://www.highradius.com/>
- [2] React, “React,” <https://reactjs.org/docs/getting-started.html>
- [3] MongoDB, “MongoDB,” <https://www.mongodb.com/docs/>
- [4] Express, “Express Documentation,” <https://expressjs.com/>
- [5] React, “React Documentation,” <https://reactjs.org/docs/getting-started.html>
- [6] Git, “Git Documentation,” .
- [7] Figma, “Figma Documentation,” <https://www.figma.com/best-practices/guide-to-developer-handoff/file-organization/>