



North South University

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

PROJECT REPORT

DISABILITY CARE

JUNIOR DESIGN

CSE299

SECTION: 1

SUMMER 2022

Student	ID
Humayra Anjum	2012461042
Alif Elham Khan	2012658042
Mohammad Junayed Hasan	2013480042

Course Faculty Dr. Sifat Momen

Submission date : 31st August 2022

Declaration

We, hereby, declare that the work presented in this report is the outcome of the four months work performed by us with the feedback by Dr. Sifat Momen, Associate Professor, Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh. The work was spread over a span of a course, CSE 299, Junior Design Project, in accordance with the course curriculum of the department for the Bachelor of Science in Computer Science and Engineering program.

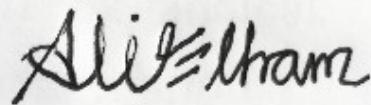
Student's name and Signature:



Humayra Anjum

ID: 2012461042

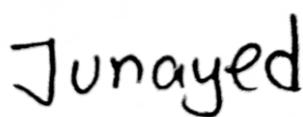
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh



Alif Elham Khan

ID: 2012658042

Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh



Mohammad Junayed Hasan

ID: 2013480042

Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh

Approval

The junior project report on 'Disability Care' has been submitted by Humayra Anjum (ID: 2012461042), Alif Elham Khan (ID: 2012658042) and Mohammad Junayed Hasan (ID: 2013480042), students of the Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh. This report partially fulfills the requirement for the degree of Bachelor of Science in Computer Science and Engineering in Summer 2022 and has been accepted as satisfactory.

Supervisor's Signature:

Dr. Sifat Momen
Associate Professor
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh

Department Chair's Signature:

Dr. Rajesh Palit
Professor and Chair (Acting)
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh

Acknowledgement

All praises are due to Almighty Allah, Most Gracious, Most Merciful who blessed us to be here at North South University for pursuing the bachelors degree.

We are grateful and wish our profound indebtedness to **Dr. Sifat Momen**, Associate Professor, Department of Electrical and Computer Engineering (ECE), North South University, Dhaka. We are blessed with the deep knowledge and keen interest of our supervisor in the field of design to carry out this project. His guidance, constant supervision, enthusiastic encouragement, sagacious advice and an effective surveillance throughout the entire period of the project have made it possible to complete this project.

We would like to express deepest appreciation towards **Dr. Rajesh Palit**, Professor and Chair (Acting) of Electrical and Computer Engineering Department, North South University, Dhaka, Bangladesh.

We would also like to thank each and everyone who were involved in the project from our friends, faculty members, fellow classmates, juniors and seniors who took the time to help and guide us during the span of the project.

We would also like to thank our family members, our parents, brothers and sisters who are the very reason of our existence. Without their unconditional love and support, this project could not be completed.

Humayra Anjum
Alif Elham Khan
Mohammad Junayed Hasan

Abstract

Disabled people are people who have mental or physical limitation so they depend on someone to support them in doing their daily life needs and jobs. Although disabled people are a minority and they are normally ignored, they are still a part of the society. The statistics show that the proportion of disabled people in the world rose from 10 percent in the seventies of the last century to 15 percent so far. The number of handicapped exceeds a billion people all over the world, occupied about 15 percent of the world's population, as a result of an aging population and the increase in chronic conditions such as diabetes, heart disease, blood and psychological diseases that are related with disabilities and impairments.

Awareness of disability-inclusive development is increasing globally. It is important to consider sustainable design for accessibility in all aspects of physical development. In addition to uplifting this morality, we deployed our solution in hoping to remove the inaccessibility, discriminatory prejudice and stigma in society. here has been a surge of disability independence so much so that 26 July is celebrated as World Disability Independence day. To boost this noble mission of disability independence, we have come up with software that bridges the gap between people with disability and healthcare services. The current existing traditional healthcare portals present a huge language and structural barrier for these people. One of the ways to assist them to be self-reliant is to improve accessibility. So we have made our system guided and easily graspable for these people in need.

Our system used website development methodologies to connect the disabled people with the hospitals, doctors, pharmacies and diagnostic centers all over the country just by pressing a few buttons. This system aims to glorify disability independence and make the life of disabled people better, self-sufficient and content.

Keywords: Disability, Website, Back end, Front end, Database, ReactJS, HTML, CSS, automation, Spring Boot.

Contents

1	Introduction	9
1.1	Identification of the problem	9
1.2	Importance	9
1.3	Our approach	9
1.4	Planning	10
1.4.1	Gantt Chart	10
1.5	Contribution	12
1.6	Selection of Tools	12
1.6.1	Frontend Tools	12
1.6.2	Backend Tools	12
1.6.3	Database Tools	12
1.6.4	Supported Environments	12
1.7	Structure	13
2	Project Specifications	14
2.1	Users and Their Roles:	14
2.1.1	Patient:	14
2.1.2	Doctor:	14
2.1.3	Pharmacy:	14
2.1.4	Diagnostic Center:	14
2.1.5	Hospital:	14
2.2	Use Cases of the System:	15
2.2.1	Patient:	15
2.2.2	Doctor:	16
2.2.3	Hospital:	17
2.2.4	Pharmacy:	17
2.2.5	Diagnostic Center:	17
3	Design	19
3.1	Front-end	19
3.1.1	Home	19
3.1.2	Signup	20
3.1.3	Login	20
3.1.4	Patient	21
3.1.5	Doctor	24
3.1.6	Pharmacy	26
3.1.7	Diagnostic Center	28
3.1.8	Hospital	29
3.2	Backend	32
3.2.1	Database: ER diagram	32
3.2.2	Whole model	33
4	Implementation	34
4.1	Front end	34
4.2	Back end	36
4.2.1	Framework - Spring Boot:	36

4.2.2	Database - Cloud Firestore:	38
4.2.3	Dependencies:	38
4.3	Connection	38
4.4	Deployment	38
5	Testing	39
6	Maintenance plan	40

List of Figures

1.1	A typical structure	10
1.2	Gantt Chart	11
1.3	Contribution Pie Chart	12
3.1	Landing page Left: Designed Version Right: Deployed Version	19
3.2	Signup page Left: Designed Version Right: Deployed Version	20
3.3	Log in page Left: Designed Version Right: Deployed Version	20
3.4	Patient Home page Left: Designed Version Right: Deployed Version	21
3.5	Patient Emergency Ambulance page Left: Designed Version Right: Deployed Version	21
3.6	Patient Medicine page Left: Designed Version Right: Deployed Version	22
3.7	Patient Tests page Left: Designed Version Right: Deployed Version	22
3.8	Patient Book Appointment page Left: Designed Version Right: Deployed Version	23
3.9	Patient Profile page Left: Designed Version Right: Deployed Version	23
3.10	Doctor Home page Left: Designed Version Right: Deployed Version	24
3.11	Doctor Due Appointment page Left: Designed Version Right: Deployed Version	24
3.12	Doctor follow up patient page Left: Designed Version Right: Deployed Version	25
3.13	Doctor Prescription page Left: Designed Version Right: Deployed Version	25
3.14	Doctor Profile page Left: Designed Version Right: Deployed Version	26
3.15	Pharmacy Home page Left: Designed Version Right: Deployed Version	26
3.16	Pharmacy Map page Left: Designed Version Right: Deployed Version	27
3.17	Pharmacy Profile page Left: Designed Version Right: Deployed Version	28
3.18	Diagnostic Center Home page Left: Designed Version Right: Deployed Version	28
3.19	Diagnostic Center Test Requested page Left: Designed Version Right: Deployed Version	29
3.20	Hospital Home page Left: Designed Version Right: Deployed Version	29
3.21	Hospital Available Doctors page Left: Designed Version Right: Deployed Version	30
3.22	Hospital Pending Appointment Requested page Left: Designed Version Right: Deployed Version	30
3.23	Hospital Test Report Deliveries page Left: Designed Version Right: Deployed Version	31
3.24	Hospital Profile page Left: Designed Version Right: Deployed Version	31
3.25	ER Diagram	32
3.26	Back-end Structure	33
4.1	The MVC-Architecture	34
4.2	Uses of ReactJS	35
4.3	Front end implementation	36
4.4	Spring Boot Layers	36
4.5	Spring Boot Flow Architecture	37
5.1	API handling using Postman	39
6.1	Maintenance Techniques	40

Chapter 1

Introduction

1.1 Identification of the problem

15% or roughly 1 billion of the world's population experience some form of disability[3]. In 2001, the WHO published a framework(International Classification of Functioning Disability and Health or ICF) that integrated previous models of disability [2]. They are more likely to experience adverse socioeconomic outcomes than persons without disabilities. Through lack of education and employment opportunities, disability may also increase risk of poverty. And all these enhances the barriers to full social and economic inclusion of persons with disabilities. Awareness of disability-inclusive development is increasing globally. It is important to consider sustainable design for accessibility in all aspects of physical development. In addition to uplifting this morality, we deployed our solution in hoping to remove the inaccessibility, discriminatory prejudice and stigma in society.

1.2 Importance

Disability is an emerging field within public health. Study suggests that people with such medical conditions have an urge to become self-dependent. And the accessibility of social sustainable for people with disabilities (PWDs) is in fact essential to ensure that they do not feel segregated from the community [7]. There has been a surge of disability independence so much so that 26 July is celebrated as World Disability Independence day. To boost this noble mission of disability independence, we have come up with software that bridges the gap between people with disability and healthcare services. The current existing traditional healthcare portals present a huge language and structural barrier for these people. One of the ways to assist them to be self-reliant is to improve accessibility. So we have made our system guided and easily graspable for these people in need.

1.3 Our approach

Nowadays websites are the most convenient way to present and disseminate information to the maximum number of people in the world [4]. A well built web platform can be an excellent place to connect users, make a system easy to use and solve a dynamic problem. Therefore, our approach for the solution of the aforementioned problem included development of a full-stack web application using relatively modern tools and kits. The following figure portrays how the approach typically works:

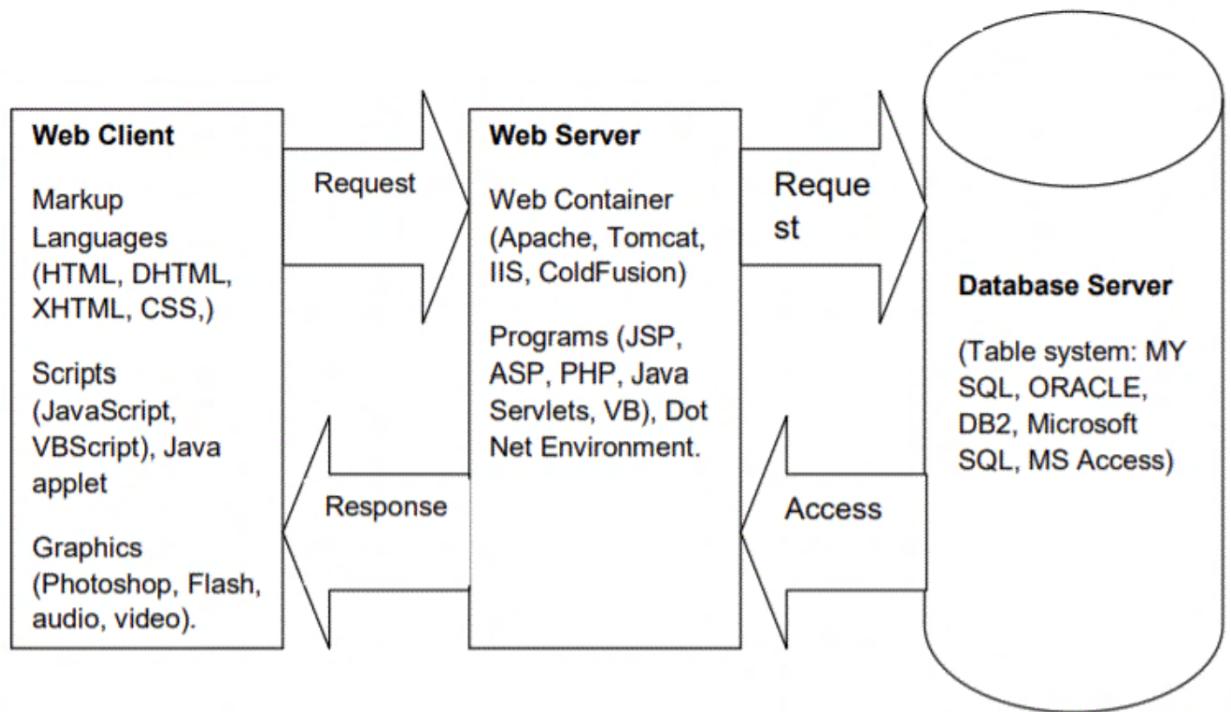


Figure 1.1: A typical structure

1.4 Planning

1.4.1 Gantt Chart

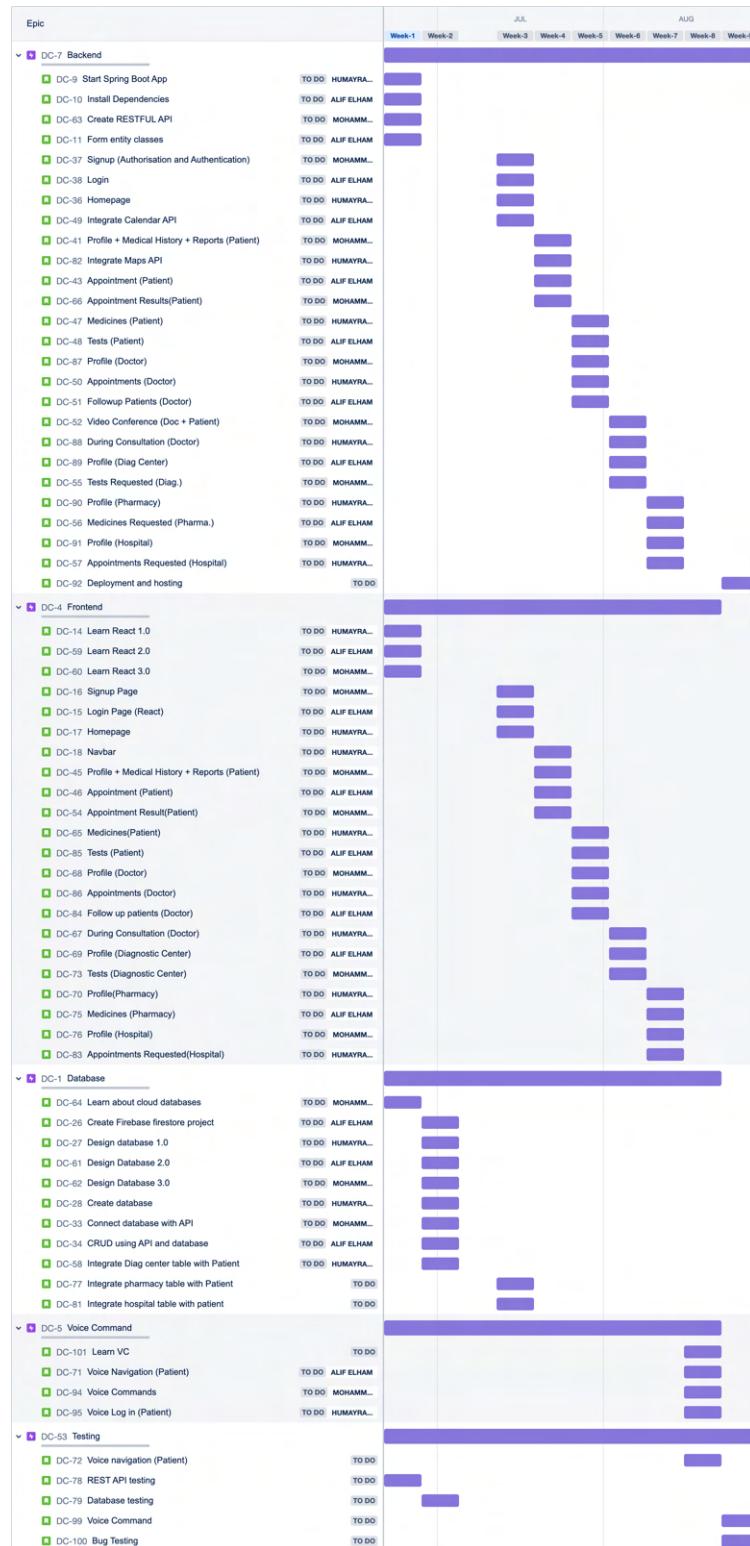


Figure 1.2: Gantt Chart

1.5 Contribution

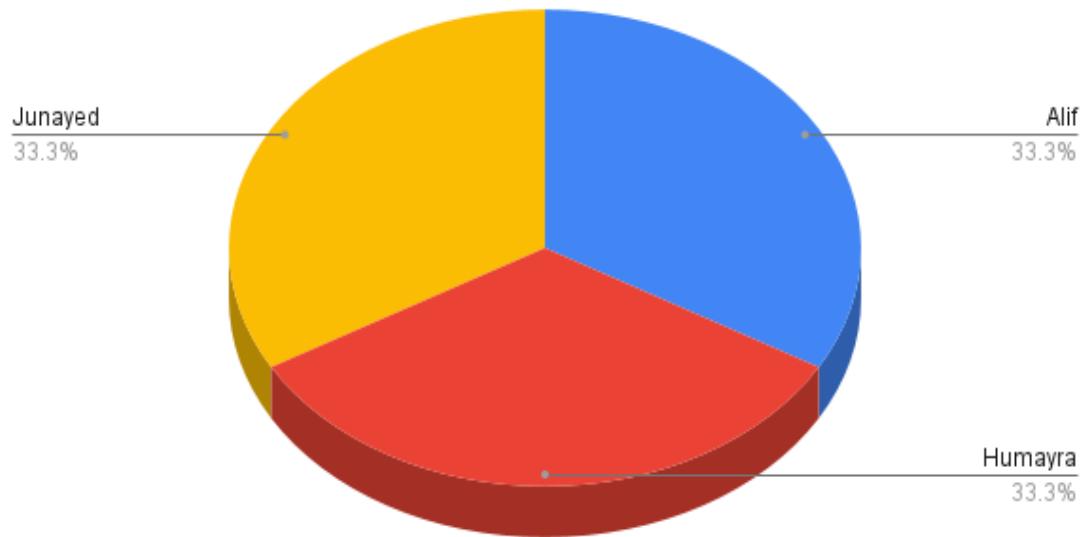


Figure 1.3: Contribution Pie Chart

1.6 Selection of Tools

1.6.1 Frontend Tools

- React
- Html
- CSS
- Javascript
- Third party libraries
- React router
- Axios

1.6.2 Backend Tools

- Java framework: Springboot

1.6.3 Database Tools

- Cloud Firestore Database

1.6.4 Supported Environments

- Any device with a web browser(Google Chrome, Brave, Safari, Microsoft Edge, Mozilla Firefox)

1.7 Structure

The rest of the report is structured in the following manner:

- **Project Specifications**

- **Users and Their Roles**
 - **Use Cases of the System**

- **Design**

- **Front End**
 - **Back End**

- **Implementation**

- **Front End**
 - **Back End**
 - * **Framework**
 - * **Database**
 - * **Dependencies**
 - **Connection**
 - **Deployment**

- **Testing**

- **Maintenance Plan**

- **Bibliography**

Chapter 2

Project Specifications

2.1 Users and Their Roles:

2.1.1 Patient:

Registered users aiming to get any form of medical service from the app will be considered patients. Patients will have unique PIDs as identifications to their profiles. The patient's medical history, diagnostic reports, and medicine supply data will be preserved in the profile along with their identification details such as name, DOB, location, contact information, and gender. Patients can book and attend doctor appointments.

2.1.2 Doctor:

Registered users who are considered doctors will be providing medical services and consultancy hours. Doctors will have unique DIDs as identifications to their profiles. The doctor's schedule, availability, and appointment data will be in their profile. They will be able to prescribe medicine, look up medical history, and assign tests after connecting to a patient via video conference.

2.1.3 Pharmacy:

Registered users aiming to supply medicines will be considered pharmacies. Each pharmacy will have a unique PHID as an identification to their profiles. The pharmacies will be notified of medicine requests whenever a doctor prescribes them to a patient. They will then deliver the requested medicines to the patient's location. The user will have an update of the patient's prescription every visit to the doctor.

2.1.4 Diagnostic Center:

The registered users who aim to provide medical test facilities will be considered Diagnostic centers. They will each have unique DCIPs as identifications to each of their profiles. The diagnostic centers will be able to receive notification from prescribed doctors about what tests need to be done on the patients. They will be able to schedule the medical tests according to the patient's suitable time. They can also collect test samples from the patient's location. They can directly send the test results to the doctor who prescribed the test.

2.1.5 Hospital:

The system will consider registered users who are aiming to provide all kinds of medical facilities as Hospitals. Hospitals will be able to provide emergency medical facilities such as instant video conferencing or sending help to a patient's location. Users of this kind can send transport facilities to the patient if requested and informed. Hospitals can also book doctor appointments for patients after checking availability of doctors.

2.2 Use Cases of the System:

The different use cases are grouped according to their actors below:

2.2.1 Patient:

- **Sign up / Register:** Patients can sign up into the system upon providing relevant information like name, email, phone number, location, gender, and DOB. They will be assigned a unique patient ID (PID) which can be used later to access their profile and information. The patient can also sign up for the system using their Google or Facebook accounts, after which the system will prompt for other necessary information.
- **Login:** A registered patient can log in to the system using their PID and OTP code sent to their contact number.
- **Logout:** A registered user can log out of the system.
- **Update Profile:** Patients can add a profile photo, change their current photo, and update medical history and reports. They can also update their email and phone number.
- **Locate User:** Patients can update their current location by selecting the ‘Locate Me’ option.
- **View Medical Reports:** Patients can go to their profile and view new medical reports under the medical reports section.
- **View Medical History:** Patients will be able to view their previous prescriptions, and medical and test reports under medical history in their profile.
- **Request Emergency Help:** Upon extreme emergencies, patients can request emergency service by clicking an easily accessible and visible “EMERGENCY HELP” button. Once clicked the system will notify the nearest hospital which will send their emergency healthcare professional and/or ambulance to the doorsteps of the patient immediately.
- **Request Urgent Consultation:** Upon emergencies, patients can request urgent consultation by clicking an easily accessible and visible “URGENT CONSULTATION” button. Once clicked the system will immediately search for available healthcare professionals and connect them in an instant video call. Doctors from the “favorite doctors” list will have the highest priority during the search.
- **Schedule Doctor’s Appointment (via Hospital):** Patients can request a doctor’s appointment from a hospital. This will be a guided process where a representative from the hospital will ask the patient about the medical problem and schedule an appointment with a doctor in that field. The appointment schedule will be automatically synced to the system calendar.
- **Schedule Doctor’s Appointment (Direct):** Patients can directly lookup for a doctor using their DID (Doctor ID) and view their available slots for appointments. They can request an appointment in a free slot and will receive a confirmation upon successful booking. The appointment schedule will be automatically synced to the system calendar.
- **Attend Scheduled Appointment:** The system will remind the patient 30 minutes prior to their scheduled appointment time via notification. A video call link will be generated and available on the notification and appointment page. By clicking on it the patient will be taken to the video call page.
- **Add favorite doctor:** At the end of a doctor consultation, through video call, a prompt will appear asking the patient if he/ she wants to add the doctor to their favorites list. A doctor on the favorite list can be easily reached for future appointments.
- **View Digital Prescription:** Patients can view the digital prescription prescribed by the doctor.
- **View Patient Progress (Requires Permission):** The doctor maintains a note to keep track of the patient’s progress. If the doctor makes it visible to the patient then they can view it.

- **Schedule preferable time for delivery:** Patients will be able to add a preferable time of the day in the system calendar to notify when they will be able to receive medicine deliveries from the pharmacy.
- **Schedule preferable time for medical test:** Patients will be able to add a preferable time of the day in the system calendar to notify the diagnostic center when they will be free for medical tests.

2.2.2 Doctor:

- **Sign up/ Register:** Doctors can sign up into the system upon providing relevant information like name, email, phone number, location, gender, and DOB. They will be assigned a unique Doctor ID (DID) which can be used later to access their profile and information. They can also sign up for the system using their Google or Facebook accounts, after which the system will prompt for other necessary information.
- **Login:** A registered doctor can log in to the system using their Doctor ID and OTP code sent to their contact number.
- **Logout:** A registered user can log out of the system.
- **Update Profile:** Doctors can add a profile photo, degrees, and certificates. They can also update their email and phone number.
- **Update Schedule on Calendar:** Doctors can update their schedules in the system calendar by integrating Google calendar. The system will extract the appointment slots available for patients from the available times set by the doctor. Once an appointment vacancy is booked by a patient, the system will update the calendar and attach a video conference link to it.
- **Join Patient Call:** Upon the arrival of a scheduled appointment, doctors will receive notifications with a link to the video conference. By following that link the doctor will be able to join the conference and a digital prescription will be launched.
- **View Patient Information:** Doctors can view the medical history and reports of patients during regular and emergency consultations.
- **Add Digital Prescription:** During video consultation, a digital prescription will be available where the doctor can prescribe medicines and tests. They can also write about the progress of patients. This prescription will also contain links to recent prescriptions of the patient.
- **Prescribe Medicines:** Doctors can prescribe medicines to patients in a digital prescription. Upon the completion of the prescription, the medicine requests are sent to the nearest pharmacy offering those medicines.
- **Request Medical Tests:** Doctors can prescribe medical tests to patients in a digital prescription. Upon the completion of the prescription, the test requests are sent to the nearest diagnostic center offering those tests.
- **Add and View Follow-up Patients:** At the end of a consultation, a doctor can choose if a patient needs to have a follow-up. The doctor will select a date for follow-up. Follow-up patients appear on the follow-up list. From this list, doctors can send emails to patients and view pending reports. They can also cancel follow-ups if deemed not necessary.
- **View Pending Reports:** As soon as the diagnostic center has filed test reports, doctors will be notified and updated with the test reports of a patient they had requested. The follow-up patients list will have a column named 'Requested Tests' where the tests with pending reports will be labeled as pending until they are made available to the doctors. The status will be changed to available once they are viewable by the doctor. The doctor can filter the list on the availability of reports.

2.2.3 Hospital:

- **Sign up/ Register:** A hospital admin can sign up in the system upon providing relevant information like hospital name, emails, contact numbers, hospital location, password, and so on.
- **Login / Logout:** A hospital admin, after signing up for their account, can log in to the system and log out of the system using the provided email/ phone number and password.
- **Receive Emergency Alert:** The hospital panel will be open to receive any kind of emergency alert from patients anytime. Such alerts are received with the location of the patient. The medical emergency alerts will be shown pending until an ambulance is assigned and sent off to the doorstep of the patient. When the vehicle will be assigned and sent off, the alert will become green and removed.
- **Manage Appointments:** Hospitals will receive appointment requests from patients and then connect them to a suitable doctor. The status of such a request will show pending in the “appointment requests” list until a doctor’s appointment has been assigned. Upon assigning an appointment, the calendars of both the patient and the doctor get updated and the patient receives a notification.
- **Manage Help Desk:** Hospital admins will manage their help desk and respond to any queries sent by patients.

2.2.4 Pharmacy:

- **Deliver Medicines:** Once confirmed by the patient, the medicines will be delivered to the doorsteps of the patient while prioritizing the ‘available for delivery’ time set by the patient.
- **Receive Medicine Resupply Requests:** If the patient needs resupply of some medicine, the system calendar will notify the pharmacy 2 days prior to the ending of the patient’s current supplies. The adjusted quantity of the medicines will be notified to the pharmacy and they will notify the patient to confirm the supply.
- **Receive New Medicine Requests:** Whenever a doctor prescribes medicines to a patient, the nearest pharmacy will be notified. If this pharmacy has the supplies then they will send a notification to the patient asking them to confirm delivery. If the nearest pharmacy does not have the supplies then they defer it to the next nearest pharmacy by selecting the ‘defer’ option.
- **Login / Logout:** A registered pharmacy can log in to the system and log out of the system using the provided email/ phone number and password.
- **Sign up/ Register:** A pharmacy admin can sign up in the system upon providing relevant information like pharmacy name, emails, contact numbers, pharmacy location, and password.

2.2.5 Diagnostic Center:

- **Sign up/ Register:** An admin of the diagnostic center can sign up in the system upon providing relevant information like the diagnostic center name, emails, contact numbers, location, and password.
- **Login / Logout:** A DC admin, after signing up for their account, can log in to the system using the provided email/ phone number and password. They can also log out of the system.
- **Update Available Tests:** A DC admin will be able to update the available tests in the center. Upon the addition of new technology for tests, the admin can add a newly available test. Upon a test being temporarily unavailable, the admin will be able to remove the test from the available tests list.
- **Receive Medical Test Request:** Whenever a doctor prescribes a patient with a test, the nearest diagnostic center providing such a test is notified. The test name and patient details (name and address) are listed in the pending tests list.

- **Submit Report:** As soon as the test report is ready, the diagnostic center can submit the test report of a patient in the requested tests list. The status of the test changes from pending to submitted. The test report is sent to both the doctor and the patient.

Chapter 3

Design

3.1 Front-end

The front-end is designed according to the users of the system. Here are the comparisons between our estimated and deployed design:

3.1.1 Home

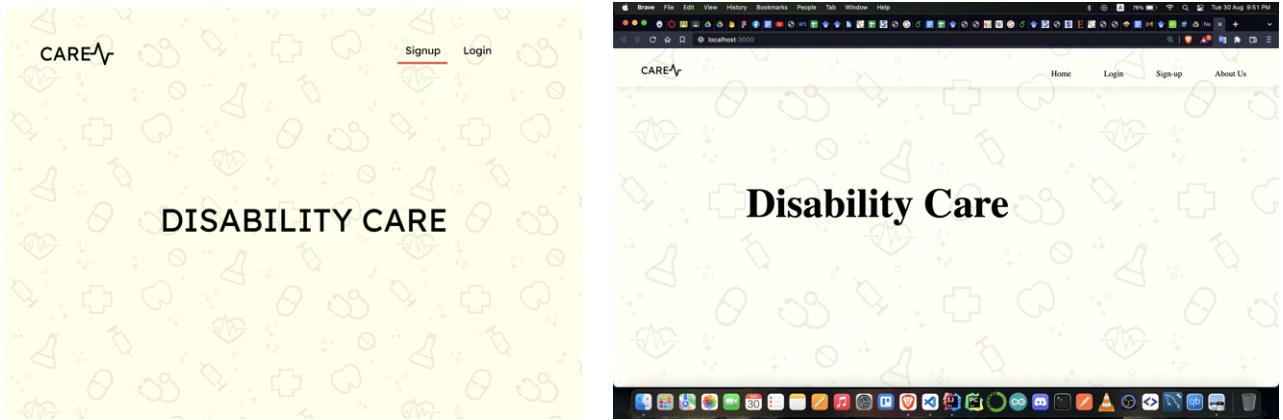


Figure 3.1: Landing page
Left: Designed Version Right: Deployed Version

3.1.2 Signup



Figure 3.2: Signup page
Left: Designed Version Right: Deployed Version

3.1.3 Login



Figure 3.3: Log in page
Left: Designed Version Right: Deployed Version

3.1.4 Patient

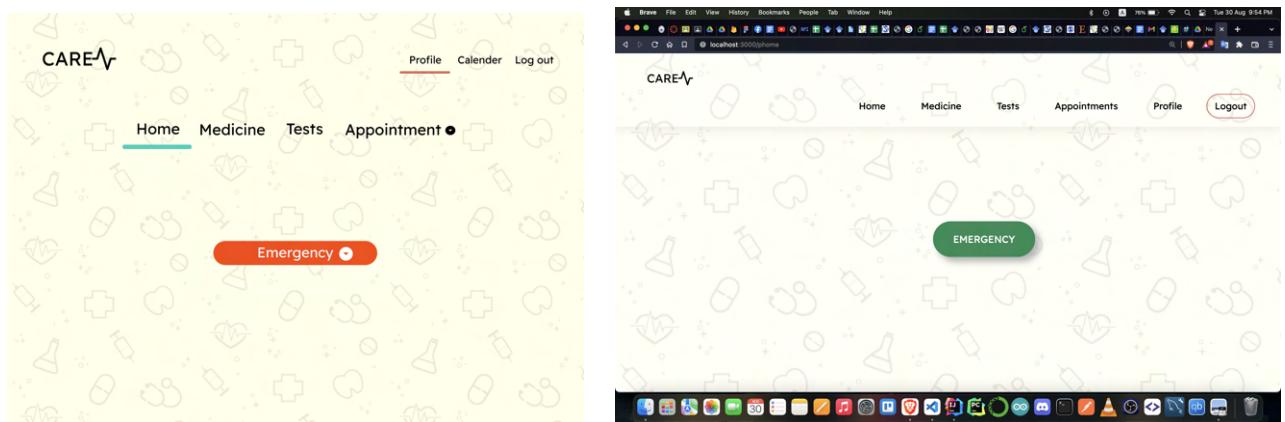


Figure 3.4: Patient Home page
Left: Designed Version Right: Deployed Version

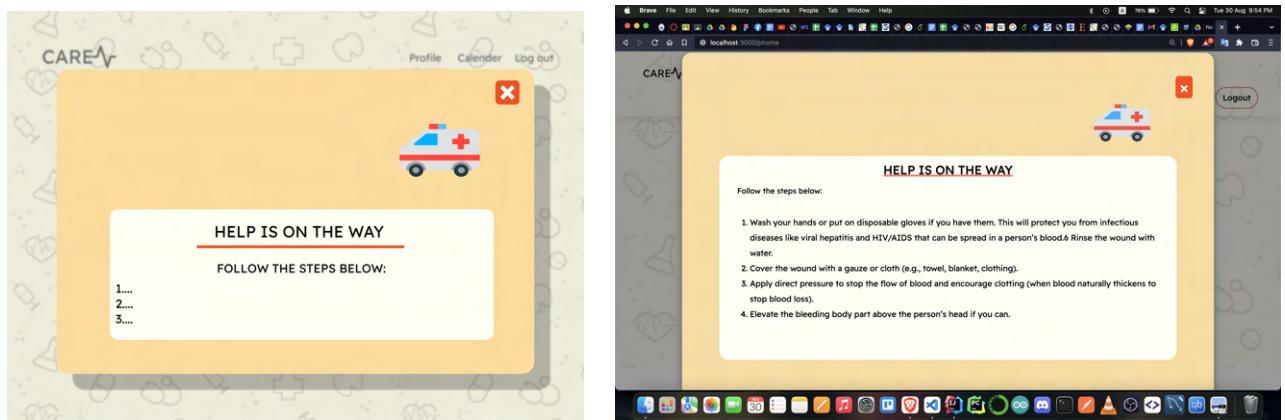


Figure 3.5: Patient Emergency Ambulance page
Left: Designed Version Right: Deployed Version

Designed Version (Left):

Medicine Names	Dosage
Med 1 : Zimax	: 500 mg : once a day before eating : 1 week
Med 2 : Alcet	: 5 mg : twice a day after eating : 3 days
Med 3 : Revotril	: 0.5mg : once a day after eating : 1 month
Med 4 : Motigut	: 10 mg : once a day after eating : 3 month
Med 5 : Nexum	: 20 mg : once a day after eating : 3 month

Deployed Version (Right):

Med 1: Napa	Appointment ID: 1
Med 2: Tufnil	Doctor ID: 0
Med 3: asdasdasd	Date: 2022-12-22T00:00:00.000+00:00
Dose 1: 2	
Dose 2: 4	
Dose 3: 5	

Figure 3.6: Patient Medicine page
Left: Designed Version Right: Deployed Version

Designed Version (Left):

Test Names	Time
1 : ANA	: Fasting
2 : BETA HCG	: After eating
3 : HbsAg	: Before/ After eating
4 : LDL	: After eating
5 : T4	: Fasting

Deployed Version (Right):

Test 1: sdsdfsd	Test ID: #5
Test 2: sfdsdfs	PID: 0
Test 3: sdsdfsf	DID: 0
Instruction 1: Beshi rokto niben	Date: 2022-08-12T00:00:00.000+00:00
Instruction 2: kom rokto niben	
Instruction 3: medium rokto	

Figure 3.7: Patient Tests page
Left: Designed Version Right: Deployed Version

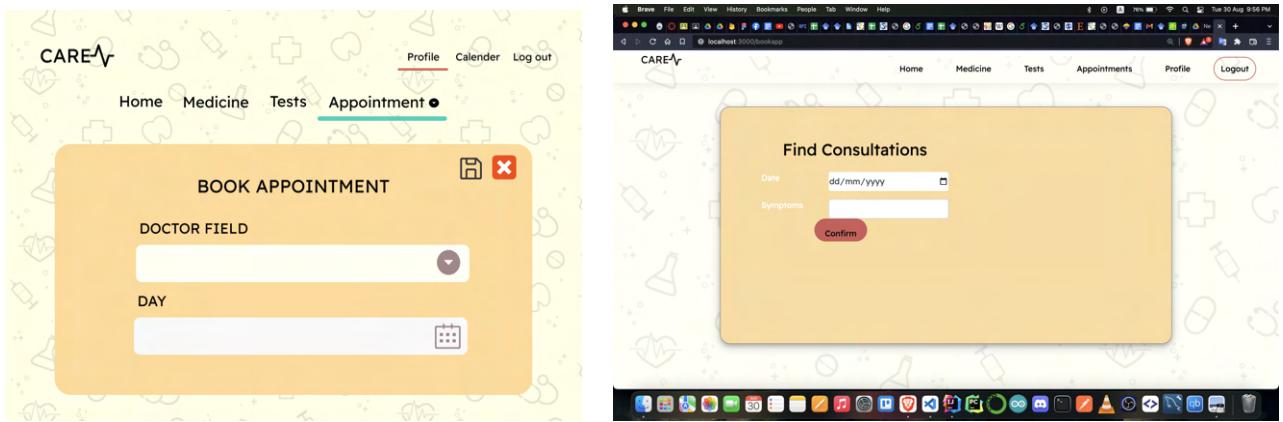


Figure 3.8: Patient Book Appointment page
Left: Designed Version Right: Deployed Version

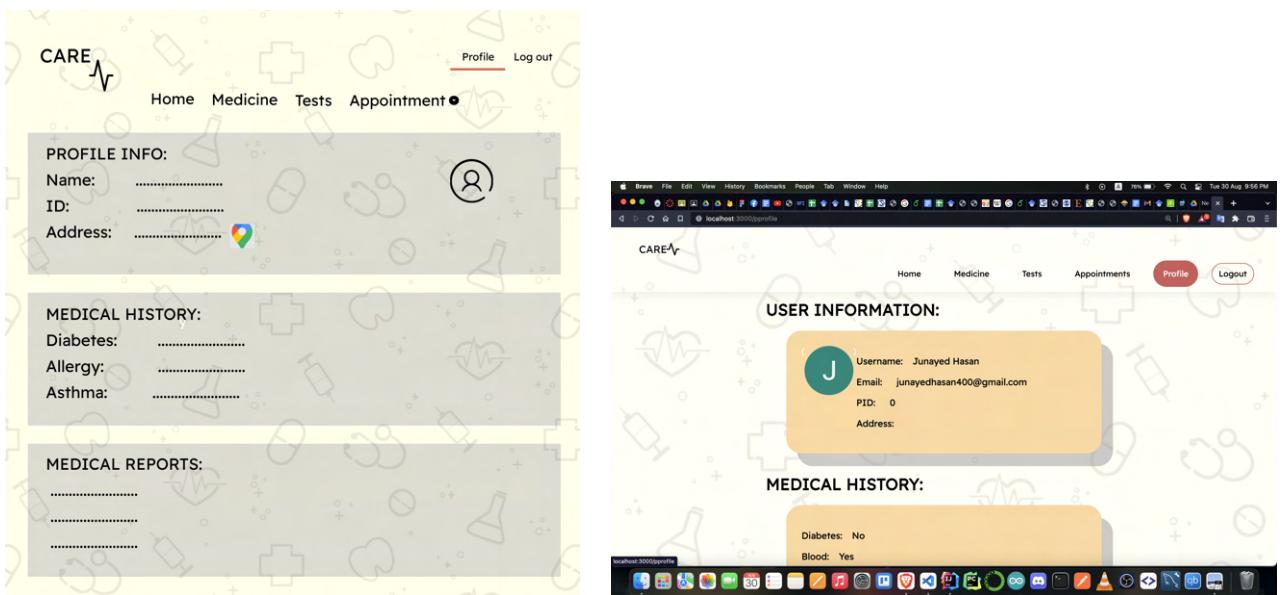


Figure 3.9: Patient Profile page
Left: Designed Version Right: Deployed Version

3.1.5 Doctor

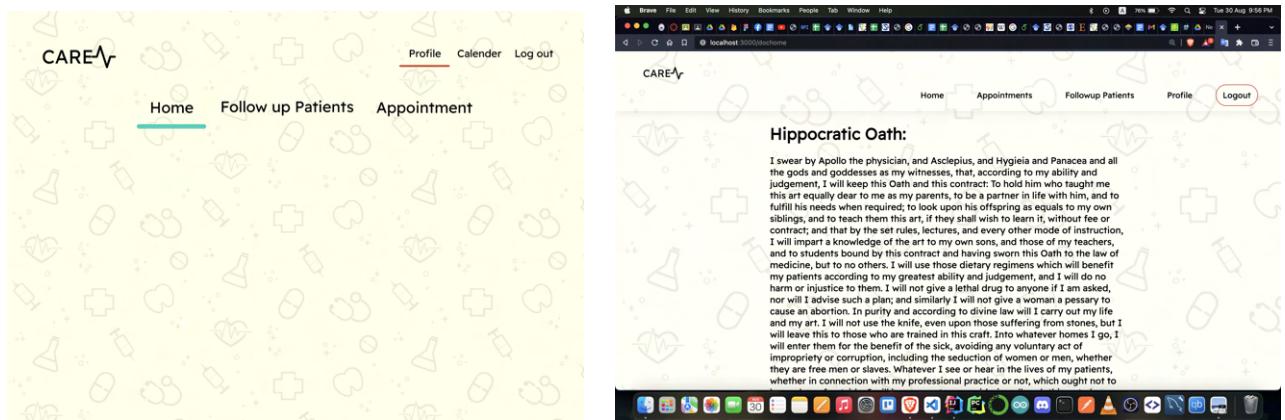


Figure 3.10: Doctor Home page
Left: Designed Version Right: Deployed Version

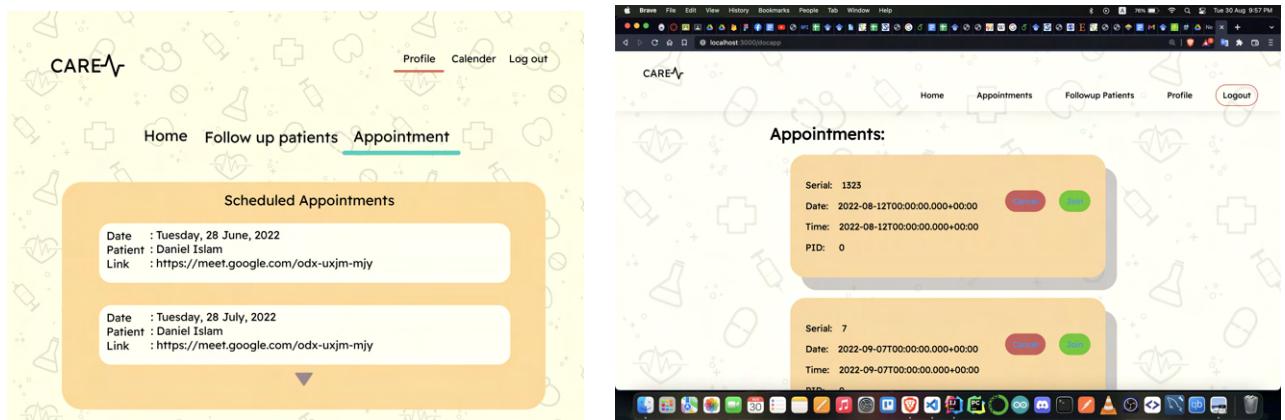


Figure 3.11: Doctor Due Appointment page
Left: Designed Version Right: Deployed Version

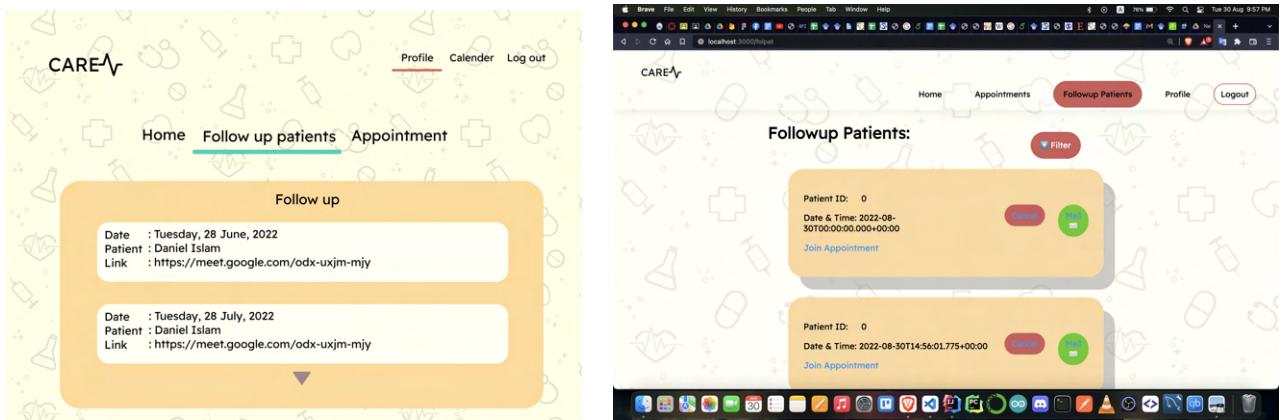


Figure 3.12: Doctor follow up patient page
Left: Designed Version Right: Deployed Version

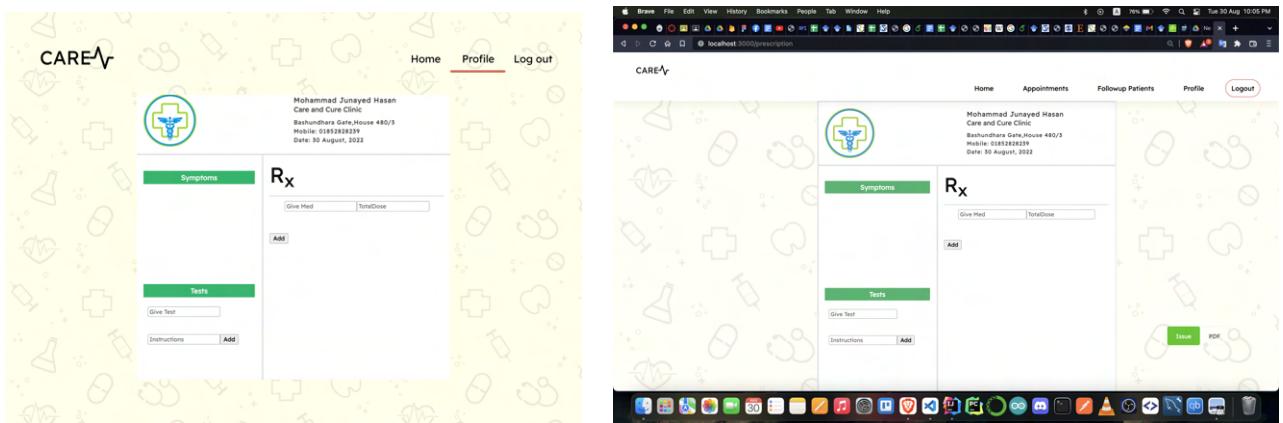


Figure 3.13: Doctor Prescription page
Left: Designed Version Right: Deployed Version

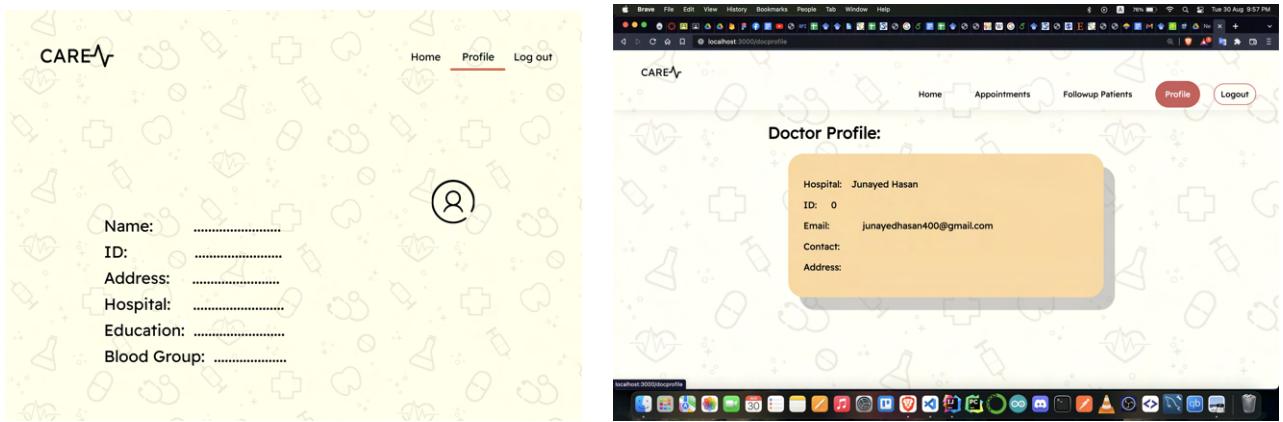


Figure 3.14: Doctor Profile page
Left: Designed Version Right: Deployed Version

3.1.6 Pharmacy

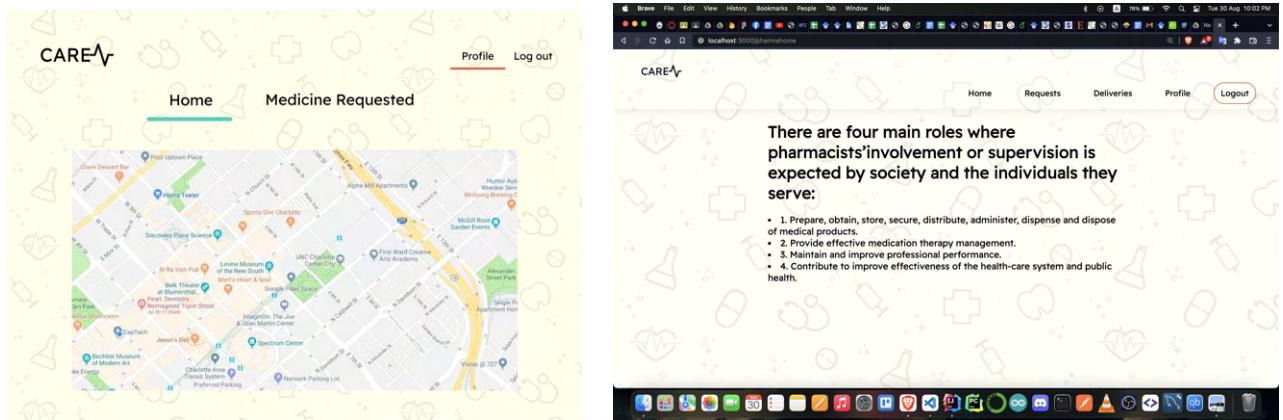


Figure 3.15: Pharmacy Home page
Left: Designed Version Right: Deployed Version

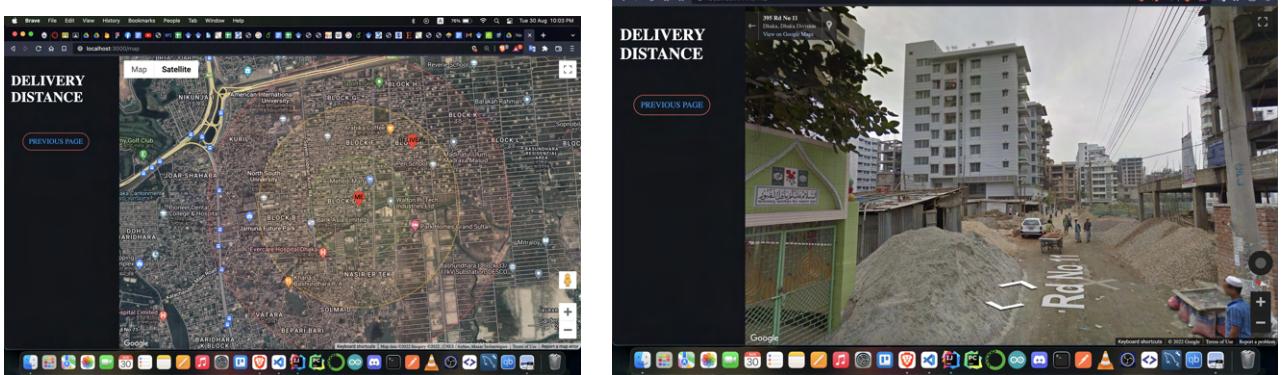
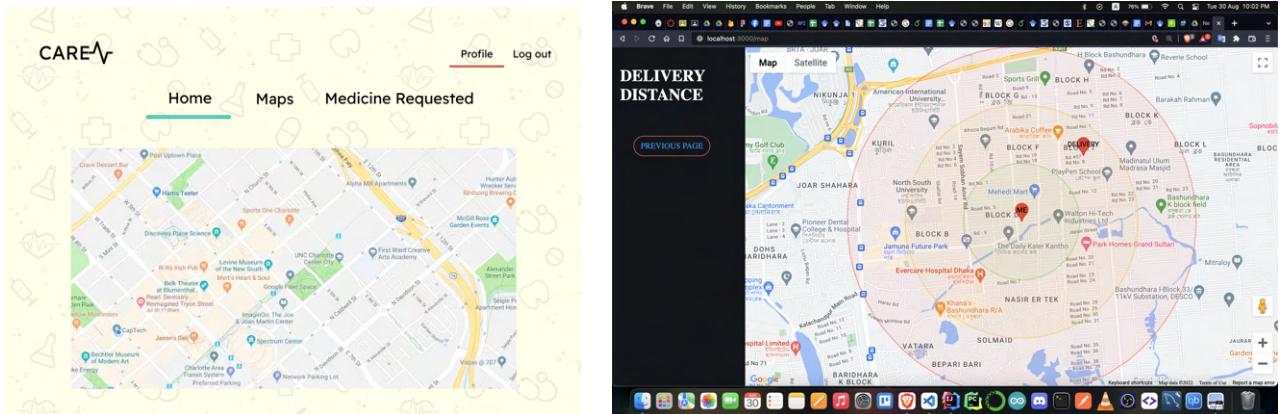


Figure 3.16: Pharmacy Map page
Left: Designed Version Right: Deployed Version

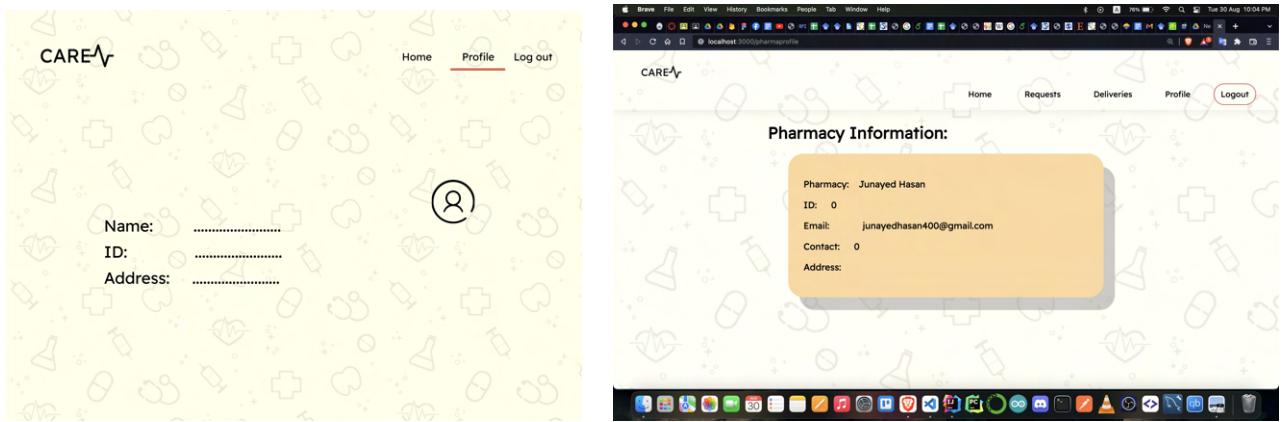


Figure 3.17: Pharmacy Profile page
Left: Designed Version Right: Deployed Version

3.1.7 Diagnostic Center

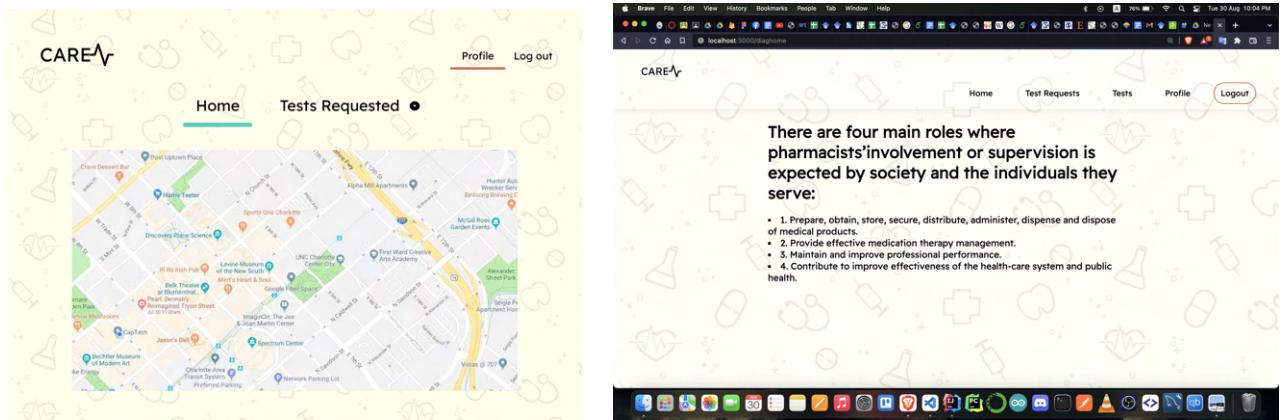


Figure 3.18: Diagnostic Center Home page
Left: Designed Version Right: Deployed Version

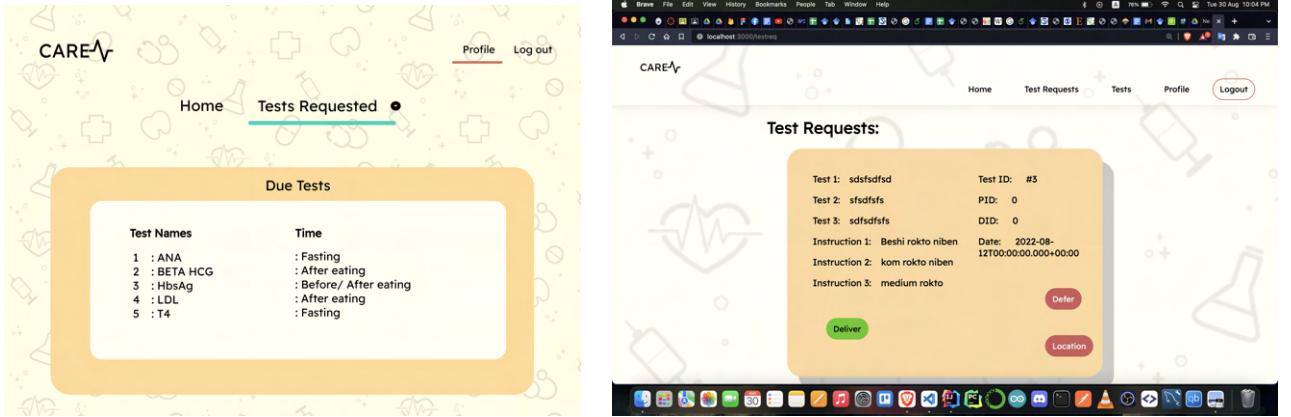


Figure 3.19: Diagnostic Center Test Requested page
Left: Designed Version Right: Deployed Version

3.1.8 Hospital

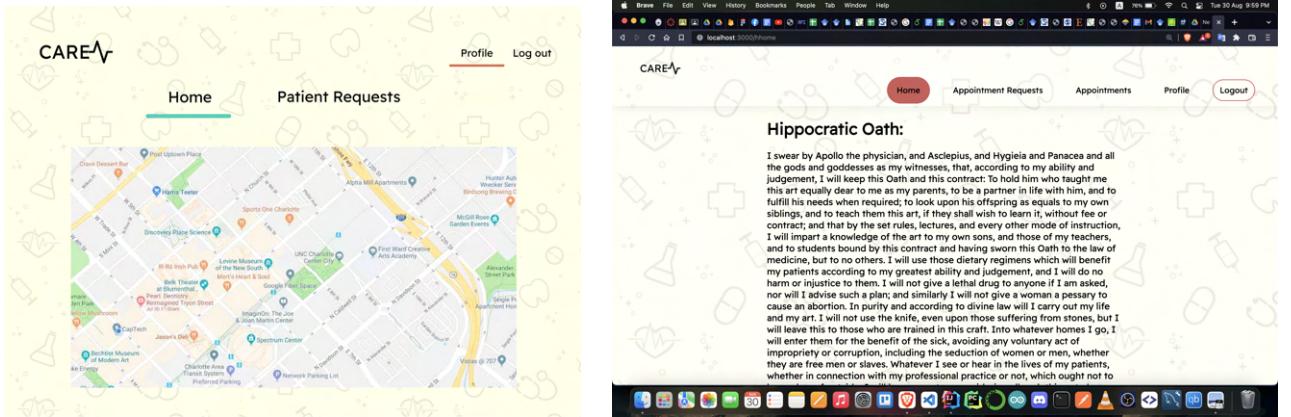


Figure 3.20: Hospital Home page
Left: Designed Version Right: Deployed Version

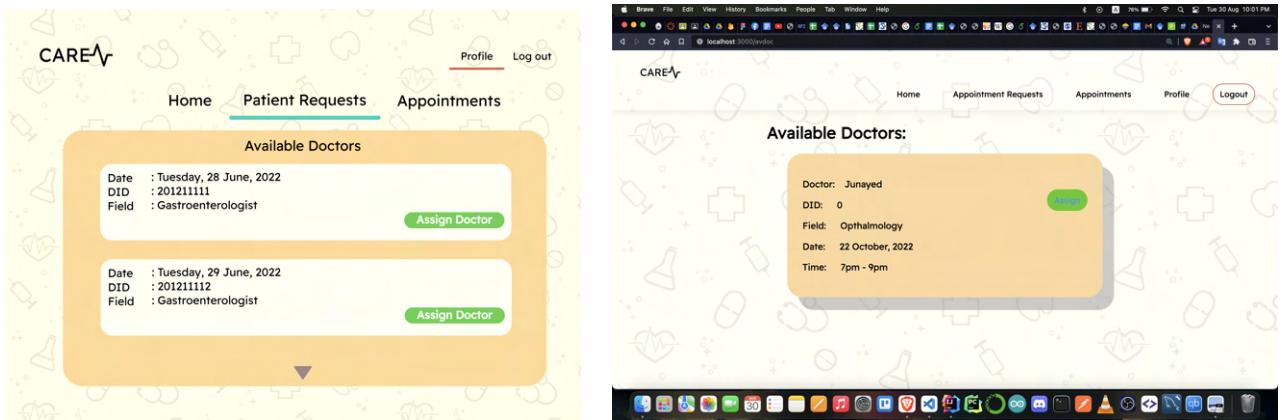


Figure 3.21: Hospital Available Doctors page
Left: Designed Version Right: Deployed Version

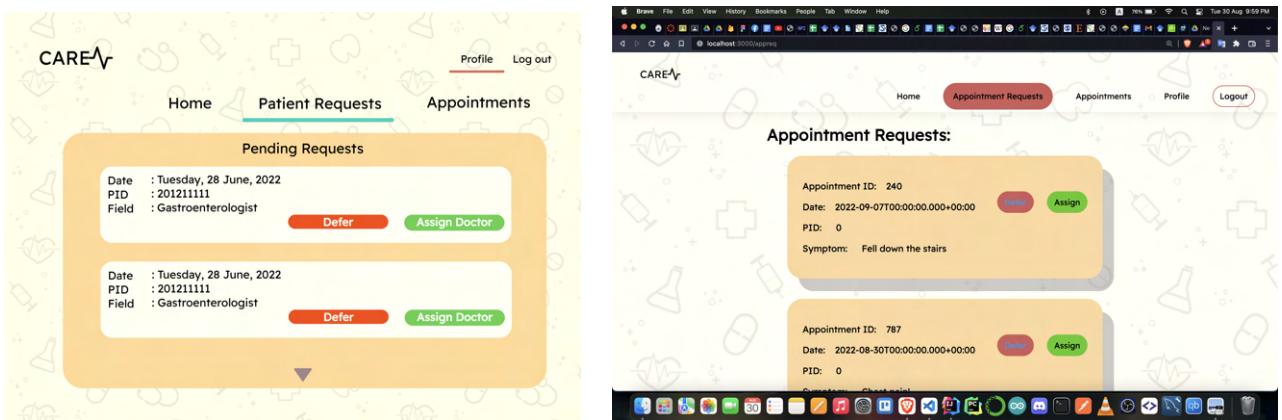


Figure 3.22: Hospital Pending Appointment Requested page
Left: Designed Version Right: Deployed Version

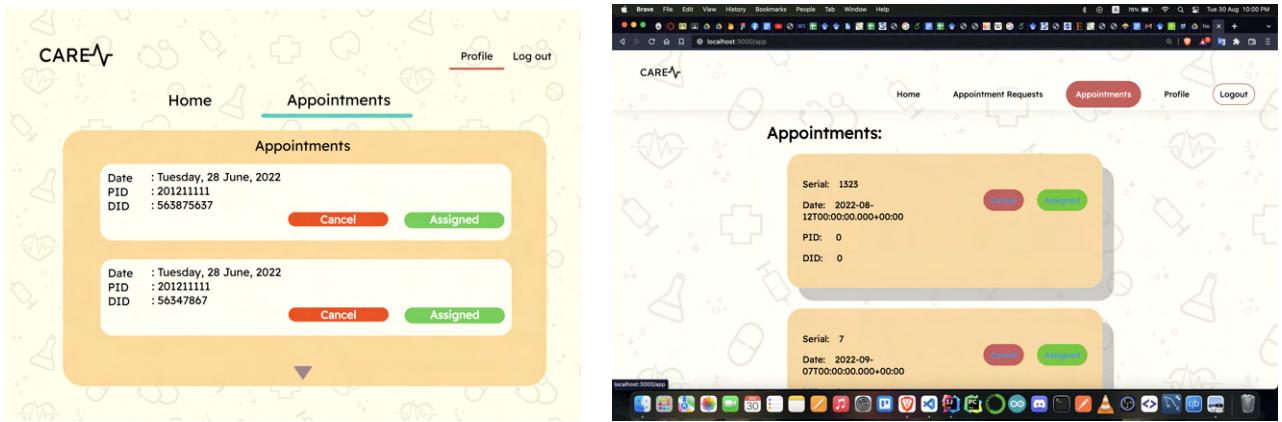


Figure 3.23: Hospital Test Report Deliveries page
Left: Designed Version Right: Deployed Version

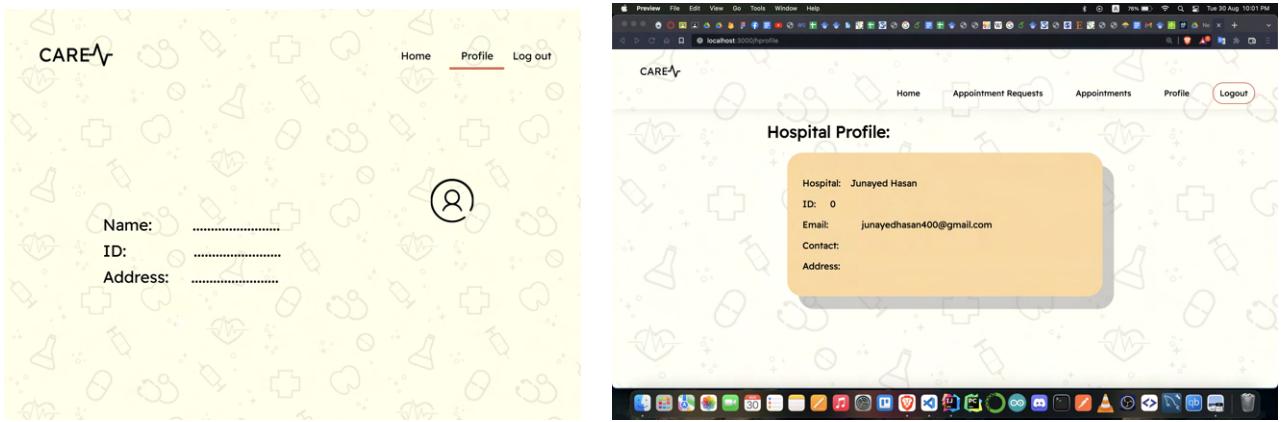


Figure 3.24: Hospital Profile page
Left: Designed Version Right: Deployed Version

3.2 Backend

The backend design is done as the following manner:-

3.2.1 Database: ER diagram

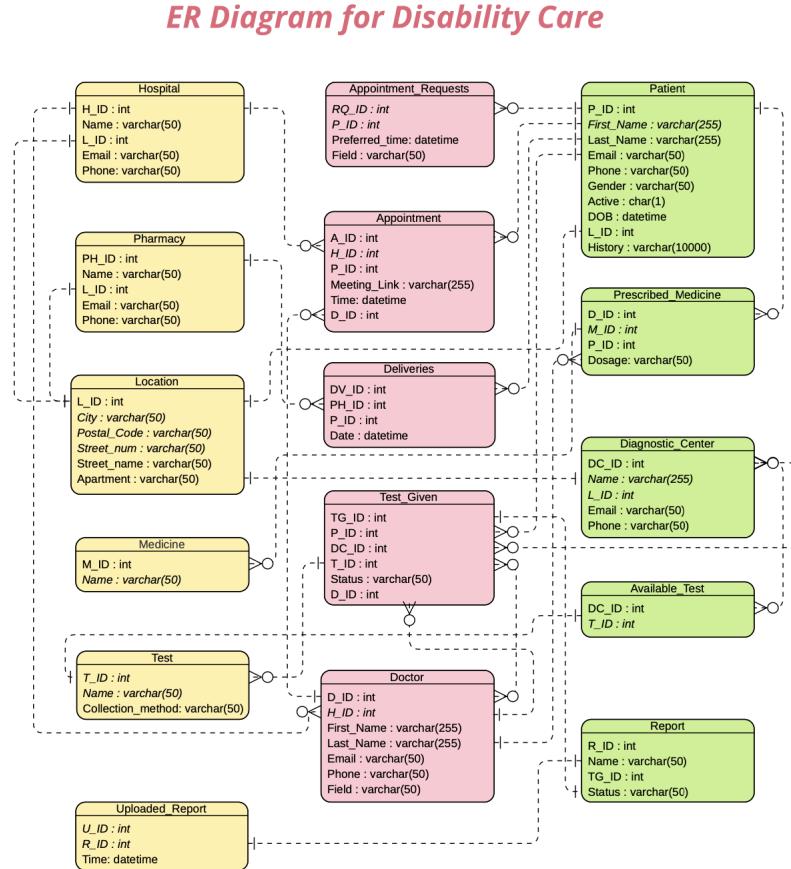


Figure 3.25: ER Diagram

3.2.2 Whole model

ReactJS + Spring Boot CRUD Full Stack

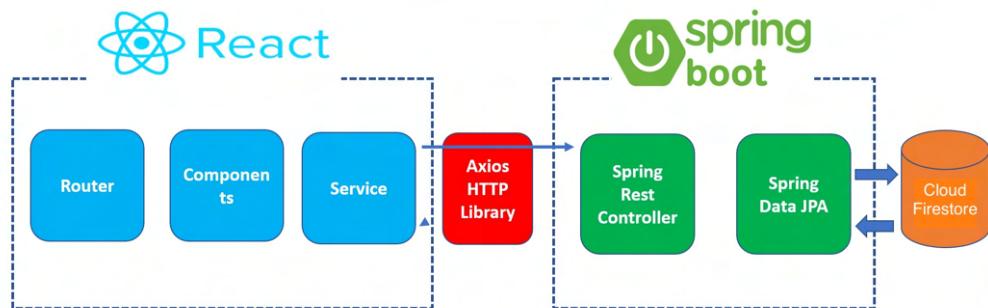


Figure 3.26: Back-end Structure

Chapter 4

Implementation

4.1 Front end

- **Library - ReactJS:** ReactJS is a component based library which is deployed for the development of interactive user interfaces. Currently it is the most popular front-end JS library. It incorporates the view (V) layer in M-V-C (Model View Controller) pattern. It is supported by Facebook, Instagram and a community of individual developers and organisations. React basically enables development of large and complex web based applications which can change its data without subsequent page refreshes. It targets to provide better user experiences and with blazing fast and robust web apps development. ReactJS can also integrated with other JavaScript libraries or frameworks in MVC, such as AngularJS [1]. The MVC architecture is shown in the following figure:

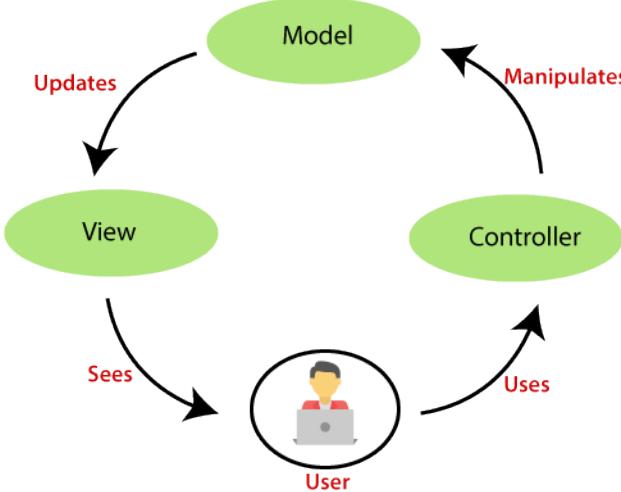


Figure 4.1: The MVC-Architecture

ReactJS was used as the main front end library in the implementation and building our web application. Using this library enabled us to make our website responsive, reactive and dynamic. This library came with many additional features and benefits that no other existing libraries can provide, such as Lightweight DOM(Document Object Model) For Better Performance, Easy Learning Curve (ELC), JSX, efficient performance, one way data flow, virtual DOM and so on. Moreover, it enabled us to make the system quite impressive that would grab the attention of a large customer base. The many benefits of reactjs can be visualized in the following figure:

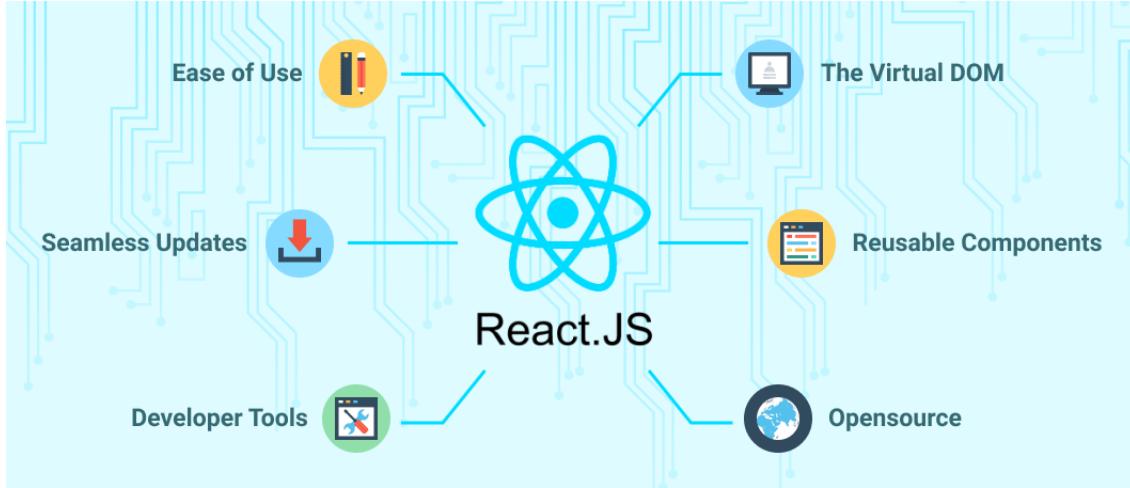


Figure 4.2: Uses of ReactJS

- **Markup Language:** HTML was used as the markup language in our frontend. The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser [5]. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. We rendered HTML inside reactjs library as components and used Javascript and CSS in parallel.
- **Styling:** The styling of the markup language that was rendered was done using raw CSS (Cascading Style Sheets) and Bootstrap 5.

CSS is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. The markup language was styled and wrapped with CSS and JavaScript to give our front end a classic view.

To summarize, the overall front end can be structured in the following manner:

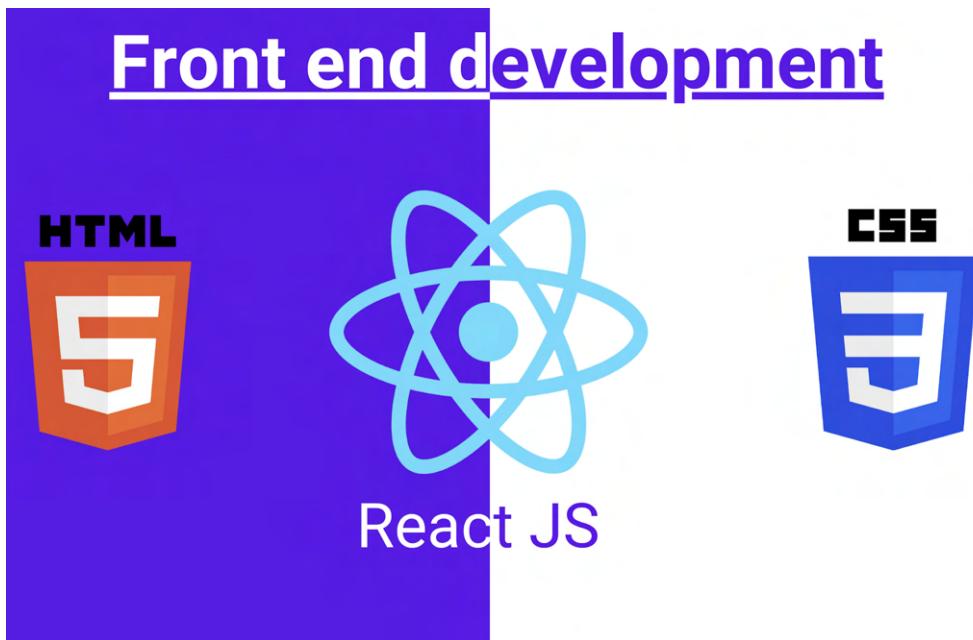


Figure 4.3: Front end implementation

4.2 Back end

4.2.1 Framework - Spring Boot:

The back end of the application was done using Spring Boot framework. Spring Boot is an open-source micro framework that provides Java developers with a platform to get started with an auto configurable production-grade Spring application. The back end using Spring Boot is divided into four distinct layers as shown below:

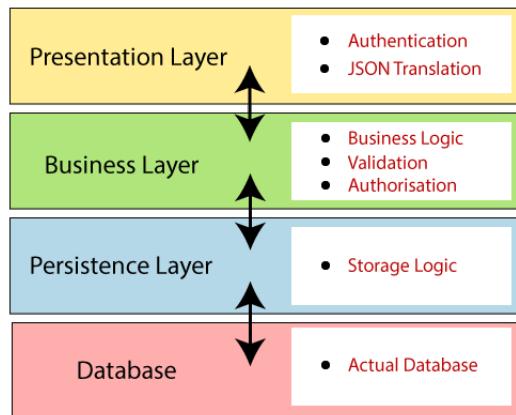


Figure 4.4: Spring Boot Layers

- **Presentation Layer:** The presentation layer handles the HTTP requests, translates the JSON parameters to object, and authenticates the request and transfer it to the business layer. In short, it consists of views of the front end part.
- **Business Layer:** The business layer handles all the business logic. It consists of service classes and uses services provided by data access layers. It also performs authorization and validation.
- **Persistence Layer:** The persistence layer contains all the storage logic and translates business objects from and to database rows.
- **Database:** In the database layer, CRUD (create, retrieve, update, delete) operations are performed.

On the other hand, the flow of Spring Boot back end can be divided into three distinct parts:

- **Controller:** Generates APIs, creates API endpoints and takes HTTPS requests from front end directly. CRUD operations are carried out in this part of the flow.
- **Service:** This part of the flow handles the business layer and implements all the required services by the back end implementing the logical requirements. The HTTPS requests by the front end are taken in by the API endpoints of the controller part, and sent to the service layer to be processed and connect to the database.
- **Model:** This part of the back end creates models and entities, tables, data fields and structures to be replicated in the database. It defines the primary keys, the field attributes and data-types of the fields in the database.

The Spring Boot flow architecture can be thus, represented as follows:

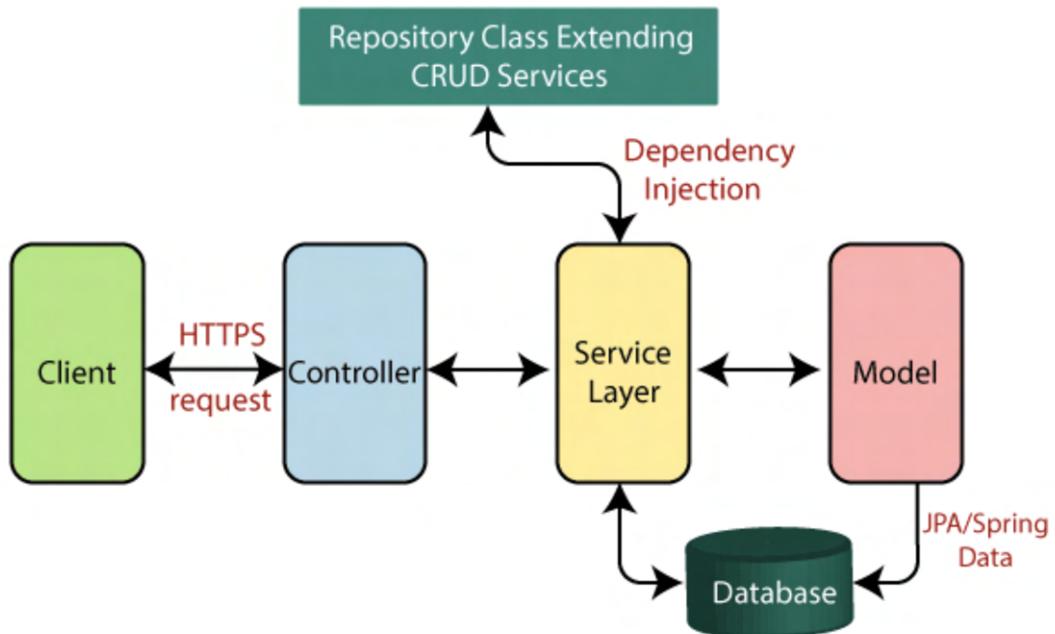


Figure 4.5: Spring Boot Flow Architecture

4.2.2 Database - Cloud Firestore:

Firestore is a NoSQL document database built for automatic scaling, high performance, and ease of application development. While the Firestore interface has many of the same features as traditional databases, as a NoSQL database it differs from them in the way it describes relationships between data objects. The entities are saved as documents and the data are saved as collections in this database. It is a cloud platform and hence no local services are required to use it.

Firebase database comes with not only CRUD facilities but also various additional facilities like Machine Learning, Statistical Analysis of data, real-time database, storage, hosting, deployment and so on.

4.2.3 Dependencies:

- **Spring Security:** Spring Security is a Java/Java EE framework that provides authentication, authorization and other security features for enterprise applications.
- **Spring Security OAuth2 Client:** It is a dependency that enables oauth2 authentication and authorization of users in the backend.
- **JJWT Api:** This dependency was used to provide tokens for token-based sessions of users upon login and sign up. Moreover, this dependency came with a feature to encode the passwords provided by users and enable authentication by token-comparing.
- **Firebase Admin:** This was used to connect the back end with the cloud firestore database for data access, modification, delete and retrieval.
- **Spring Boot Test:** This dependency enabled auto testing of our application by generating cases based on provided parameters. This is discussed further in the testing portion of the report.

4.3 Connection

The front end and back end connection was made using a dependency library in the reactjs front end called 'Axios'. Axios is a promised-based HTTP client for JavaScript. It has the ability to make HTTP requests from the browser and handle the transformation of request and response data. It was used to connect to the API endpoints generated by the back end and perform CRUD operations according to need. While accessing the API endpoints, authorization and authentication was ensured; hence, data was secured and in turn, the system.

4.4 Deployment

The web application was hosted and deployed online via Heroku. Heroku is a free platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud. The generated url for our website was: <https://disabilitycare.herokuapp.com/>

Chapter 5

Testing

Software testing is the act of examining the artifacts and the behavior of the software under test by validation and verification. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation [6]. There are many third-party dependency libraries available that can be integrated with the system to auto-test by generating random test cases and corner cases depending on class attributes and method parameters. One such dependency that was used in our back end is Spring Boot Test. This library is responsible for testing the whole project completed with Spring boot framework.

Apart from auto-testing the classes and methods, many manual testing methodologies were carried out as well. The generated APIs were tested using manual CRUD operations from Postman. Postman is an API platform for developers to design, build, test and iterate their APIs.

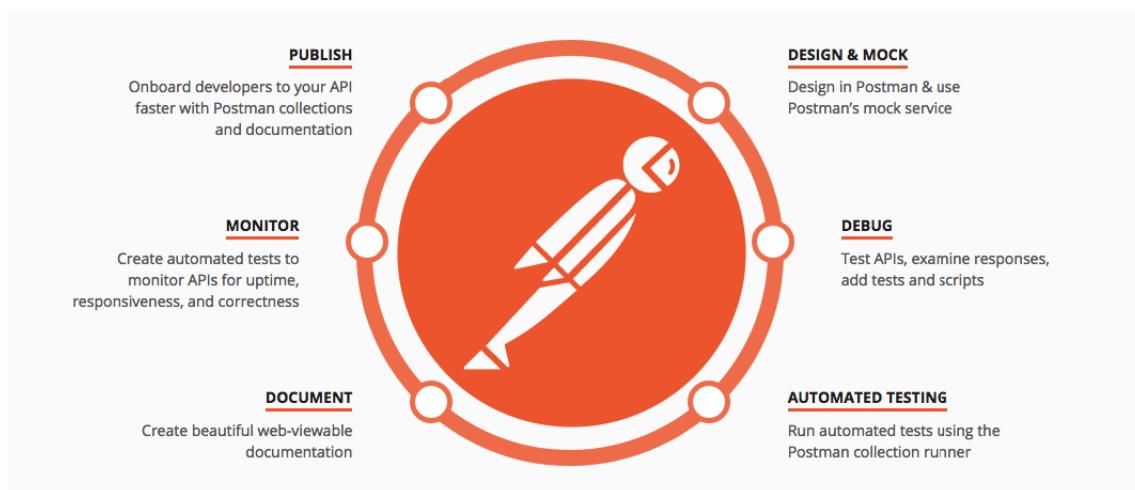


Figure 5.1: API handling using Postman

Chapter 6

Maintenance plan

Software maintenance is the process of changing, modifying, and updating software to keep up with customer needs. Software maintenance is done after the product has launched; for several reasons including improving the software overall, correcting issues or bugs, to boost performance, and more. The four types of maintenance techniques typically followed are: Corrective Software Maintenance, Preventative Software Maintenance, Perfective Software Maintenance and Adaptive Software Maintenance. The



Figure 6.1: Maintenance Techniques

combination of these four maintenance techniques give us seven critical steps in maintenance which are targeted to be followed in the future:

- **Identification and Tracing:** The process of determining what part of the software needs to be modified (or maintained). This can be user-generated or identified by the software developer itself depending on the situation and specific fault.
- **Analysis:** The process of analyzing the suggested modification including understanding the potential effects of such a change. This step typically includes cost analysis to understand if the change is financially worthwhile.
- **Design:** Designing the new changes using requirement specifications
- **Implementation:** The process of implementing the new modules by programmers.

- **System Testing:** Before being launched, the software and system must be tested. This includes the module itself, the system and the module, and the whole system at once.
- **Acceptance Testing:** Users test the modification for acceptance. This is an important step as users can identify ongoing issues and generate recommendations for more effective implementation and changes.
- **Delivery:** Software updates or in some cases new installation of the software. This is when the changes arrive at the customers.

Bibliography

- [1] Sanchit Aggarwal. Modern web-development using reactjs. *International Journal of Recent Research Aspects*, 5(1):133–137, 2018.
- [2] Gloria L Krahn, Deborah Klein Walker, and Rosaly Correa-De-Araujo. Persons with disabilities as an unrecognized health disparity population. *American journal of public health*, 105(S2):S198–S206, 2015.
- [3] World Health Organization et al. *World report on disability 2011*. World Health Organization, 2011.
- [4] Yash Pal Singh Sandeep .*. *A Study on the Importance of Web Design, Development and Security in WSDM towards Web Development*. Journal of Advances and Scholarly Researches in Allied Education, 2017.
- [5] Wikipedia contributors. Html — Wikipedia, the free encyclopedia, 2022. [Online; accessed 31-August-2022].
- [6] Wikipedia contributors. Software testing — Wikipedia, the free encyclopedia, 2022. [Online; accessed 31-August-2022].
- [7] Halmi Zainol, Haryati Mohd Isa, Siti Rashidah Md Sakip, and Ainaa Azmi. Social sustainable accessibility for people with disabilities at public transport stations through sustainable development goals in malaysia. *Environment-Behaviour Proceedings Journal*, 3(9):89–94, 2018.