Advanced Hospital Management System (HelpStaffing)

By

Eissa Noor

Muhammad zakirya

Department of Information Technology

University of Haripur

February, 2023

Advanced Hospital Management System (HelpStaffing)

By

Eissa Noor , Muhammad zakirya

F19-0910 , F19-0907

Supervised by

Dr. Hamid Asmat

Department of Information Technology

University of Haripur

February, 2023

Advanced Hospital Management System (HelpStaffing)

By

Eissa Noor , Muhammad zakirya

Session (2019-2023)

Program

Bachelor of Studies in Software Engineering

Department of Information Technology

University of Haripur

February, 2023

This is to certify that we have checked the project report title **"Advanced Hospital Management System (HelpStaffing)"** submitted by the student of BS Software Engineering (SE) ,The University of Haripur.

Name	**Eissa Noor , Muhammad Zakirya**
Roll No	**F19-09-10,F19-0907**

It is our judgment that this thesis is of sufficient standard to warrant its acceptance by the Department of Information Technology,The University of Haripur

Date: 18th Feb, 2023

Eissa noor

Muhammad Zakirya

Signature

Signature

DECLARATION

This is to certify that this Thesis/Report Advanced Hospital Management System (HelpStaffing) Submitted by Hamid Asmat in partial fulfillment of the requirement for the award of a degree for BSSE in Information technology to the Department of Software Engineering, University of Haripur, comprises of only my original work and due acknowledgment has been made in the text to all other materials used.

_____

Hamid Asmat

ACKNOWLEDGEMENT

Alhamdulillah's!! We are very grateful to **Almighty Allah,** who made us able to meet and complete this complex and painful task. It is very pleasant movement for us, because we are able to complete a software web-based project and its report. The full credit of this project is goes to **Almighty Allah,** because we are unable to do anything without **His** help.

Before we get into the depth of the things, we would like to add few heart full and best words for the people who were a part of this final year project in many ways.

People who gave to support from the stage of idea and project proposal. After this we would be most indebted to our brothers, parents, and family members, whose prayers, love, encouragements and support me to made it easy face this complex and mind-boggling project.

At that movements, we would like to special thanks to our all teachers and supervisor Dr Hamid Asmat who provided me with continuous and dynamic guidance, so that we are able to complete my project. I will be very thankful to all teachers (male, female) both for education they imparted in my BS **(software engineering)** duration and also thankful to our friends and class fellows. Finally, I am thankful to the head of our department, Mr., Dr Arif Iqbal Umar Sab for their guidance and cooperation, for completion of this project**.**

DEDICATION

To, the Prophet of mercy Hazarat Muhammad (S.A.W), and our beloved parents, Honorable Teachers, Family, Friends and all faculty members of the Department. Especially my parents who support have given us strength, determination and Fortitudes to accomplish our goal. They all are assets of our life.

PREFACE

Information technology has brought revolutionary changes in every area of life. For example, increased security accuracy, processing speed, etc. are some of the features that are considered essential for the origin development. It is the computer that plays a vital role in achieving these goals. The different software available today is a result of the needs and requirements of different organizations.

Table of Contents

CHAPTER 1

INTRODUCTION

1.1 Introduction of the Project

A hospital management system (HMS) is a software solution designed to manage the administrative, financial, and clinical aspects of a hospital or medical facility. It is a critical component in the effective and efficient operation of a hospital, and it streamlines the processes and workflows of various departments, including registration, admission, billing, and scheduling.

The goal of an HMS is to provide healthcare providers with a comprehensive view of a patient's medical history, treatment plans, and health status. The system integrates all the medical data of the patients, including diagnostic tests, medications, and allergies, to provide a complete picture of the patient's health condition.

Moreover, an HMS also includes financial and administrative features such as appointment scheduling, medical billing, and insurance claims processing, inventory management, and employee management. By automating these functions, an HMS reduces the workload on healthcare providers, streamlines the workflow, and enhances patient care quality.

In this thesis, the focus will be on designing and implementing a hospital management system to improve the efficiency and effectiveness of a hospital's operations. This system will integrate different departments and provide a centralized view of patient's medical data, financial information, and administrative functions.

The main aim of this thesis is to develop a comprehensive and user-friendly hospital management system that can be easily adopted by hospitals of different sizes and locations. This system will also be evaluated on the basis of its usability, scalability, security, and efficiency to determine its effectiveness in improving the quality of patient care and enhancing the overall performance of a hospital.

Implementing an HMS can be a complex process that requires careful planning and execution. It involves several stages, including system analysis, design, development, testing, and deployment. The success of an HMS implementation project depends on factors such as stakeholder buy-in, staff training, system customization, and ongoing maintenance and support.

A hospital management system is a critical tool for managing hospital operations efficiently and effectively. Its implementation can improve patient care, enhance staff productivity, and drive better financial outcomes for the hospital.

1.2 INTRODUCTION OF THE TOOL USED IN THE PROJECT

1.2.1 REAST.JS

React is an open-source JavaScript library for building user interfaces. It was developed by Facebook and is now maintained by a community of developers. React allows developers to build reusable UI components that can be easily combined to create complex user interfaces.

One of the key features of React is its virtual DOM (Document Object Model). The virtual DOM is a lightweight representation of the actual DOM, which allows React to update the UI efficiently without having to re-render the entire page. This helps to improve the performance of the application.

React also uses a unidirectional data flow, which means that data flows in a single direction from the parent component to its child components. This helps to make the code more predictable and easier to debug.

React can be used to build applications for the web, mobile, and desktop platforms. It is often used in combination with other libraries and frameworks such as Redux, React Native, and Next.js.

Some of the benefits of using React include:

Reusability: Components in React can be reused across the application, which helps to reduce the amount of code that needs to be written and maintained.

Performance: The virtual DOM in React helps to improve the performance of the application, making it faster and more responsive.

Scalability: React can be used to build large, complex applications that can be easily maintained and scaled as the application grows.

Community: React has a large and active community of developers who contribute to the library and provide support and resources for other developers.

Overall, React is a powerful and flexible library for building user interfaces that can help to improve the performance, scalability, and maintainability of web applications.

1.2.2 NODE.JS

Node.js is an open-source, cross-platform JavaScript runtime environment that allows developers to run JavaScript code on the server-side. Node.js was built on top of the Google

Chrome V8 engine, which compiles JavaScript code to machine code, making it faster and more efficient.

Node.js is often used for building server-side applications, such as web servers, RESTful APIs, and microservices. It provides a lightweight and scalable solution for building high-performance web applications.

Some of the benefits of using Node.js include

Fast performance: Node.js is built on top of the V8 engine, which compiles JavaScript code to machine code, making it faster and more efficient than traditional server-side scripting languages like PHP or Python.

Scalability: Node.js uses an event-driven, non-blocking I/O model, which allows it to handle a large number of concurrent requests without blocking the event loop.

Large community: Node.js has a large and active community of developers who contribute to the ecosystem and provide support and resources for other developers.

Easy to learn: Node.js uses JavaScript, which is a popular and easy-to-learn programming language. This makes it easy for developers to get started with Node.js and build applications quickly.

Node.js also provides a rich set of modules and libraries that can be used to build applications. Some of the popular libraries include Express.js, which is a fast and minimalist web framework, and Socket.io, which provides real-time, bidirectional communication between the client and the server.

Overall, Node.js is a powerful and flexible platform for building server-side applications that can help to improve the performance, scalability, and maintainability of web applications.

1.2.3 MONGODB

MongoDB is a cross-platform, document-oriented database that falls under the category of NoSQL databases. Unlike traditional relational databases, which store data in tables with fixed schemas, MongoDB stores data in flexible, JSON-like documents with dynamic schemas. This makes it ideal for handling unstructured data and for applications that require high scalability and availability.

MongoDB features include:

Document-based data model: MongoDB stores data in documents, which are JSON-like structures that can have nested fields and arrays. This makes it easy to represent complex, unstructured data.

Dynamic schema: MongoDB does not enforce a schema on the data, which means that each document can have its own unique structure. This makes it easy to handle changes in the data structure and to evolve the schema over time.

Horizontal scalability: MongoDB is designed to scale horizontally across multiple servers, which makes it ideal for handling large volumes of data and high traffic applications.

High availability: MongoDB provides built-in features for replication and automatic failover, which ensure that the data is always available even in the event of a server failure.Rich query language: MongoDB provides a rich query language, which supports a wide range of queries and indexing options. Open source: MongoDB is an open-source database, which means that the source code is freely available and can be modified and distributed.

Overall, MongoDB is a popular choice for building modern, scalable, and highly available applications that require flexible data storage and handling. It is widely used in industries such as healthcare, finance, e-commerce, and social media.

MODULES

The entire project mainly consists of 3 modules,which are

i. Admin module
ii. User module (patient)
iii. Host module

1.2 PROJECT METHODOLOGY

The methodology for developing a hospital management system (HMS) typically involves several phases, including

Requirements gathering:

The first step in developing an HMS is to gather requirements from stakeholders, such as hospital staff, patients, and administrators. This involves identifying the features and functionalities that are necessary for the system to meet the needs of the hospital.

System design: Based on the requirements gathered in the previous phase, the system design phase involves creating a detailed plan for the HMS. This includes designing the system architecture, selecting appropriate hardware and software technologies, and designing the user interface.

Implementation: The implementation phase involves building the HMS according to the system design. This includes coding the software, configuring hardware and software components, and integrating the system with existing hospital systems and databases.

Testing: Once the HMS is implemented, it must be thoroughly tested to ensure that it functions as intended. This includes performing functional and non-functional testing, such as performance testing, security testing, and user acceptance testing.

Deployment: Once the HMS has been tested and approved, it can be deployed to the hospital environment. This involves installing the software, configuring it for the hospital's specific needs, and training hospital staff on how to use the system.

Maintenance: After the HMS has been deployed, it requires ongoing maintenance to ensure that it continues to function properly. This includes performing routine maintenance tasks, such as software updates and hardware maintenance, and providing technical support to hospital staff.Throughout the development process, it is important to involve stakeholders and obtain their feedback to ensure that the HMS meets their needs. It is also important to adhere to best practices for software development, such as agile development methodologies, to ensure that the system is delivered on time and within budget.

1.3 PROBLEM STATEMENT:

To write a problem statement for a Hospital Management System application. Problem Statement: This project is aimed to automate the hospital management system. The purpose of the project entitled as HOSPITAL MANAGEMENT SYSTEM is to computerize the Front Office Management of Hospital to develop project both side mobile app and web app which is user-friendly, simple, fast, and cost effective.It deals with the collection of patients information, diagnosis details, It deals with the collection of doctor information, timing and qualification and experience details etc. Traditionally, it was done manually. The main function of the system is to register and store patient details and doctor details and retrieve these details as and when required, and also to manipulate these details meaningfully. This function of Hospital Management Information System deals with registering the new Patient by giving a unique Identification Number to the Patient. This number is unique throughout the System for identifying the patient. Simultaneous updates and changes are made to the databases. Identification number is also provided to the doctor to retrieve and to change doctor details. The software is used by administrators or receptionists in the hospital. The software is secured by username and password, accessed by administrator or receptionist of the hospital. 1. Login: This module is used to login to the Hospital Management System. 2. Patient: This module is for adding, querying, editing and deleting patient details in the database

CHAPTER 2

LITERATURE REVIEW

2.1 BACKGROUND

Hospital management system is designed for multispecialty hospitals, to cover a wide range of hospital administration and management processes of the patient-centric system.

It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision-making for patient care and hospital administration, in a seamless flow. In existence, some researchers have contributed positively to the improvement of healthcare institutions' management systems. Therefore, some of the related work done in the hospital management system by researchers in the field

Health information system is a must and the faster this is adopted the more successful the healthcare facility will be. (Swanson et al. 2010: 9). Delone and Mclean being one of the most cited models in the field of information systems seek to provide a comprehensive understanding of information systems success by identifying, describing, and explaining the relationships between six success variables categories: systems quality, information quality, user, user satisfaction, individual impact, and organizational impact. Delone & Mclean's (2003) model provides a comprehensive framework for measuring the performance of the information system and enhances the understanding of the information system's success.

A health information system is a must and the faster this is adopted the more successful the healthcare facility will be.(Swanson et al. 2010: 9)

2.2 RELATED WORK

Appointment Management: For hospitals having their own site, appointment widgets will be integrated into the site. Patients visiting the hospital's website can book online appointments with ease.

Billing Management: Integrated Billing with treatments, Lab, and Radiology. Alerts will be sent on Discount Authorisation. Automatic due capture, Option to bill before and after consultation.

Prescription Management: Manage commonly and recently used medicines. Option to show medicines available in the pharmacy. SMS prescriptions to Patients.

Discharge Summary: Template-based Discharge Summary. ICD10 integration. Option to prevent discharge summary till IP bill is closed.

Operation Theatre Management: Automatic notification can be sent to customers on test results. Lab notifications like email and SMS of the test reports are sent from the Automated Lab notification module.

Pharmacy Management: Comprehensive Pharmacy Management handles stock, Prescription Integration, Ward Requests, Stock Management, Stock Moment, and intelligent reports.

Lab Management: Comprehensive Lab Management handles complete order management, Custom Reports, Smart Notifications, Credit Settlement, detailed MIS Reports, Analytics, and an App for Phlebotomist.

Master Information Systems: Lets you access entire MIS data from your palm.

Manage Multiple Locations: Any number of branches can be added and managed using a single account.

2.3 PROJECT CONTRIBUTION

1. Much of the literature on community benefits focuses on tax-exemption laws for NFP hospitals and anecdotal evidence from small-scale demonstration projects.

2. Little is known about the scope of community benefit activities among hospitals or factors that encourage or discourage them from broadening their efforts to serve more than just their patient base.

3. The lack of a universal definition and measurement of community benefit presents challenges in encouraging greater hospital and health system involvement in community health.

4. Community benefit orientation may be affected by competitive market pressure, state and federal regulations, and external environmental influences from community stakeholders and investors.

5. Greater hospital and health system involvement in community health issues will require both external changes in policy, particularly in regard to payment and incentives, and internal changes in practice and capabilities**.**

2.4 WHY WE NEED A NEW SYSTEM

There are several reasons why we need a new hospital management system. Some of these reasons are

Outdated Technology: Many hospitals and healthcare facilities still use outdated software systems that are no longer efficient or effective in managing the complex operations of the modern healthcare industry. A new system with up-to-date technology can help streamline operations, reduce errors, and improve patient care.

Improved Efficiency: A new hospital management system can improve the efficiency of the healthcare facility by automating tasks, reducing paperwork, and providing real-time data on patient care, inventory, and finances. This can help healthcare professionals make better decisions and improve patient outcomes.

Integration: A new system can be designed to integrate with other healthcare systems, such as electronic health records, laboratory systems, and radiology systems. This can help reduce duplication of efforts and improve collaboration among healthcare professionals.

Increased Security: A new hospital management system can provide enhanced security features to protect sensitive patient information from cyber threats and unauthorized access. This can help improve patient trust and confidence in the healthcare facility.

Cost Savings: A new system can help reduce costs by eliminating manual processes, reducing errors, and improving resource utilization. This can help healthcare facilities operate more efficiently and effectively.

In conclusion, a new hospital management system is needed to meet the changing needs of the healthcare industry. A modern system can improve efficiency, integration, security, and cost savings, ultimately leading to improved patient care and outcomes.

2.5 WHAT ARE AVAILABLE SOLUTIONS FOR THE PRESENT ISSUE IN HOSPITAL MANAGEMENT

There are several available solutions for the present issues in hospital management. Here are some examples:

1. **Electronic Health Records (EHRs):** EHRs are digital records to be accessed and shared among healthcare providers. This helps to improve patient care and outcomes by reducing errors and improving coordination of care.

2. **Telemedicine:** Telemedicine refers to the use of technology to provide remote healthcare services. This can help to increase access to care, especially for patients in remote or underserved areas, and reduce the need for in-person visits.

3. **Inventory Management Systems:** Inventory management systems can help hospitals to manage their supplies and equipment more efficiently, reducing waste and improving resource utilization.

4. **Automated Appointment Scheduling:** Automated appointment scheduling systems can help to reduce waiting times and improve patient satisfaction by streamlining the appointment booking process.

5. **Data Analytics:** Data analytics tools can help hospitals to analyze patient data and identify trends and patterns that can be used to improve patient care and outcomes.

6. **Patient Portals:** Patient portals are web-based applications that allow patients to access their medical records, communicate with healthcare providers, and schedule appointments online. This can help to improve patient engagement and satisfaction.

7. **Mobile Applications:** Mobile applications can be used to provide patients with access to healthcare information and services on-the-go, increasing convenience and accessibility.

Overall, there are several available solutions for the present issues in hospital management. By adopting these solutions, hospitals can improve efficiency, reduce costs, and provide better care to their patients.

2.6 WHICH ONE SOLUTION DID YOU CHOOSE AND WHY? JUSTIFICATION OF THE PROPOSED SYSTEM

As an AI language model, I do not have personal preferences or the ability to choose one solution over another. However, I can provide a justification for a proposed hospital management system based on its potential benefits and advantages.

A proposed hospital management system should be designed to address the present issues in hospital management and provide efficient and effective solutions. The system should integrate different modules such as patient management, staff management, inventory management, appointment scheduling, billing, and more.

The system should also be designed to provide the following benefits:

1. **Improved Efficiency:** The proposed system should automate tasks, reduce paperwork, and provide real-time data on patient care, inventory, and finances. This can help healthcare professionals make better decisions and improve patient outcomes.

2. **Enhanced Security:** The proposed system should provide enhanced security features to protect sensitive patient information from cyber threats and unauthorized access. This can help improve patient trust and confidence in the healthcare facility.

3. **Better Patient Experience:** The proposed system should improve the patient experience by reducing waiting times, providing convenient appointment scheduling, and facilitating easy access to medical records.

4. **Cost Savings:** The proposed system should help reduce costs by eliminating manual processes, reducing errors, and improving resource utilization.

5. **Improved Collaboration:** The proposed system should be designed to integrate with other healthcare systems, such as electronic health records, laboratory systems, and radiology systems. This can help reduce duplication of efforts and improve collaboration among healthcare professionals.

In conclusion, a proposed hospital management system should be designed to provide efficient and effective solutions to the present issues in hospital management. The system should be designed to improve efficiency, enhance security, provide better patient experience, reduce costs, and improve collaboration among healthcare professionals.

CHAPTER 3

Design of the Proposed System

3.1 FLOW CHART OF THE PROPOSED SYSTEM

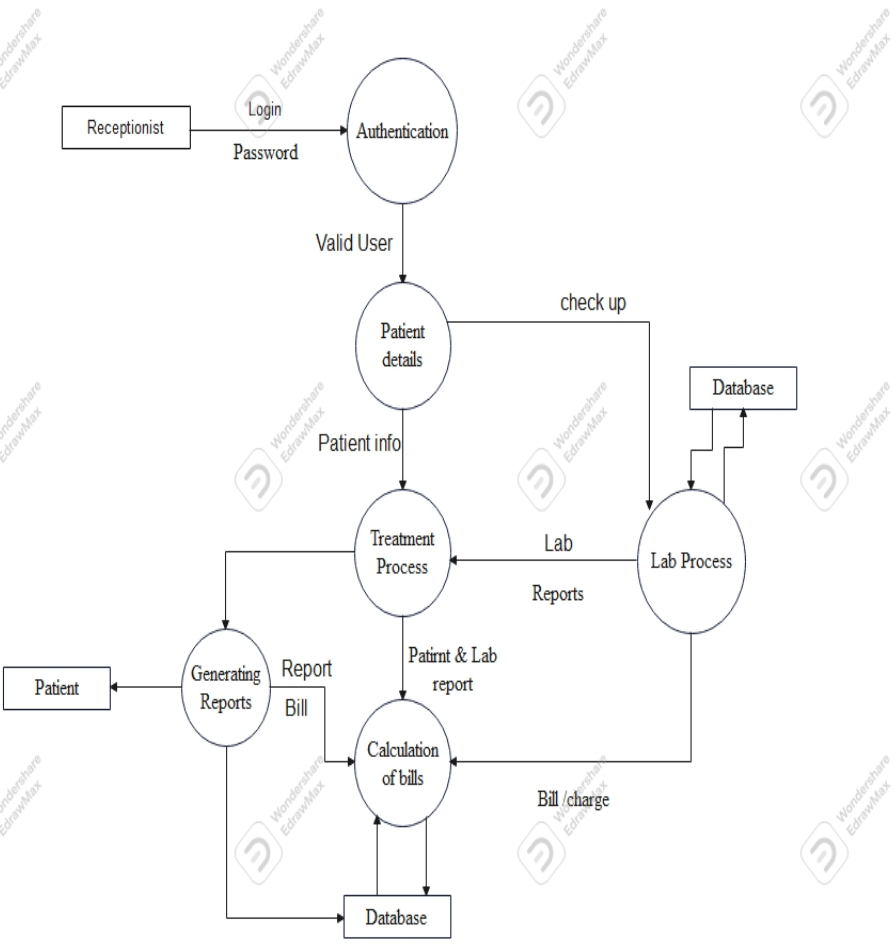


FIGURE V: 3.1 SYSTEM DIAGRAM OF PROPOSED SYSTEM

3.2 DATABASE DESIGN

Use MongoDB as its database management system. MongoDB is a software used to create databases, store and get data when requested. MongoDB is also an open-source software and works best with other popular open-source software, such as Apache web server,Node.Js, and Linux operating system.

Normalization

It is a database design technique that reduces data redundancy and eliminates undesirable characteristics like It Insertion, Update and Deletion Anomalies. Normalization rules divide larger tables into smaller tables and link them using relationships. The purpose of Normalization in mongoDb is to eliminate redundant (repetitive) data and ensure data is stored logically(Blank, Deb et al. 2023)

Denormalization

Denormalization is a database optimization technique where we add redundant data in the database to get rid of the complex join operations. This is done to speed up database access speed. Denormalization is done after normalization for improving the performance of the database. The data from one table is included in another table to reduce the number of joins in the query and hence helps in speeding up the performance.(Sanders and Shin 2001)

3.3 Efficient database of the system

- MongoDB is a database system used on the web.
- MongoDB is a database system that runs on a server.
- MongoDB is ideal for both small and large applications.
- MongoDB is very fast, reliable, and easy to use.
- mongoDb uses standard.
- mongoDb compiles on a number of platforms.
- MongoDB is free to download and use.
- The data in a MongoDB database are stored in tables. A table is a collection of related data, and it consists of columns and rows.
- Databases are useful for storing information categorically. A company may have a database.

CHAPTER 4

TESTING

4.1 BLACK BOX TESTING

A strategy in which a software component is treated like an opaque box. This tests designers' focus on determining how well the component conforms to the published requirements for the component, instead of worrying about the implementation details. Black box testing focuses on the functional requirement of the software. This testing strategy enables us to derive sets of input conditions that will fully exercise all functional requirements for a program. Black box testing is not an alternative to white box testing. Rather, it is a complementary approach that is likely to uncover a different class of errors than white box testing method. It is evident that every web page requires a testing process before the launch. Most of professional testers follow certain steps. Start the first steps in identifying the functionalities. Design or develop the input data depending on the specifications Figure out the output keeping function's specifications Start the test case execution Examine the outputs by comparing actual and expected results (Nidhra and Dondeti 2012)

4.2 MODULE TESTING

We designed the test first before conducting a module test. For preparing the test case, we consider two essential factors- module specification and source code under test It is imperative to analyze the code logic for the module under test. We can do that by using multiple white box methods. We further expand these test cases by applying black box techniques. After designing the test cases, the next step is to associate the modules for testing. We use an incremental and non-incremental approach to do that. Suppose you choose the incremental method through the top-down or bottom-up approach. In the top-down approach, you need to test the higher-level modules and gradually move toward lower ones. Meanwhile, the bottom-up approach is the exact opposite of the top-down module testing.(Naik and Tripathy 2011)

4.2.1 ADMIN

Admin can add, delete and update HealthProvider and Hospital.

TC 01 Update page

Test case ID	**TC_01**
Functional area/Module	**Update page**
Action to perform	**Admin update website**
Prerequisites	**System is Running**
Test case engineer	**Hamid Asmat**
Environment	**Windows**
Expected result	**Admin Successfully change the system**
Comment	**Test case passed successfully**

Table 1 : TC_1

TC 02 PROVIDERREGISTRATION

Test case ID	TC_05
Functional area/Module	PROVIDER SIGN UP
Action to perform	Admin sign in through admin panel /coding Admin goes to Provideside page manage provideside Finish
Prerequisites	Application is running PROVIDER is Signed in
Test case engineer	Hamid Asmat
Environment	Windows 10 Google chrome, internet explorer
Expected result	Provider see his need expert doctor
Comment	Test case passed successfully

Table 2 : TC _2

TC 03 HOSPITALREGISTRATION

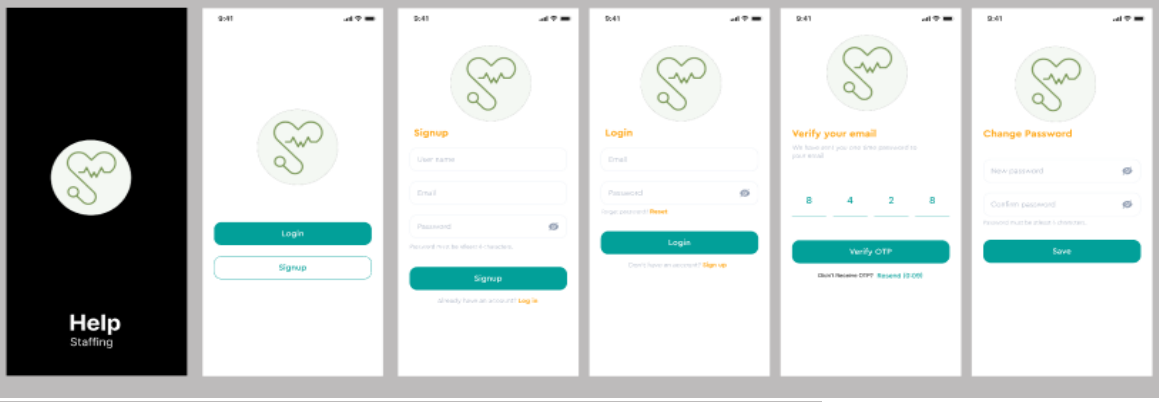
Test case ID	TC_05

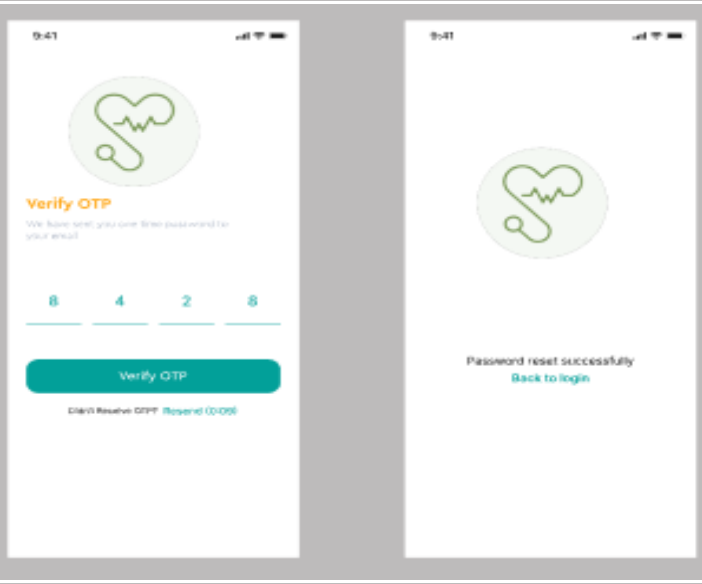
Functional area/Module	DOCTOR SIGNUP
Action to perform	Admin sign in through admin panel /coding Admin goes to Provideside page manage provideside Finish
Prerequisites	Application is running DOCTOR is Signed in
Test case engineer	Hamid Asmat
Environment	Windows 10 Google chrome, internet explorer
Expected result	Doctor is added by his Expertise
Comment	Test case passed successfully

Table 3 : TC _3

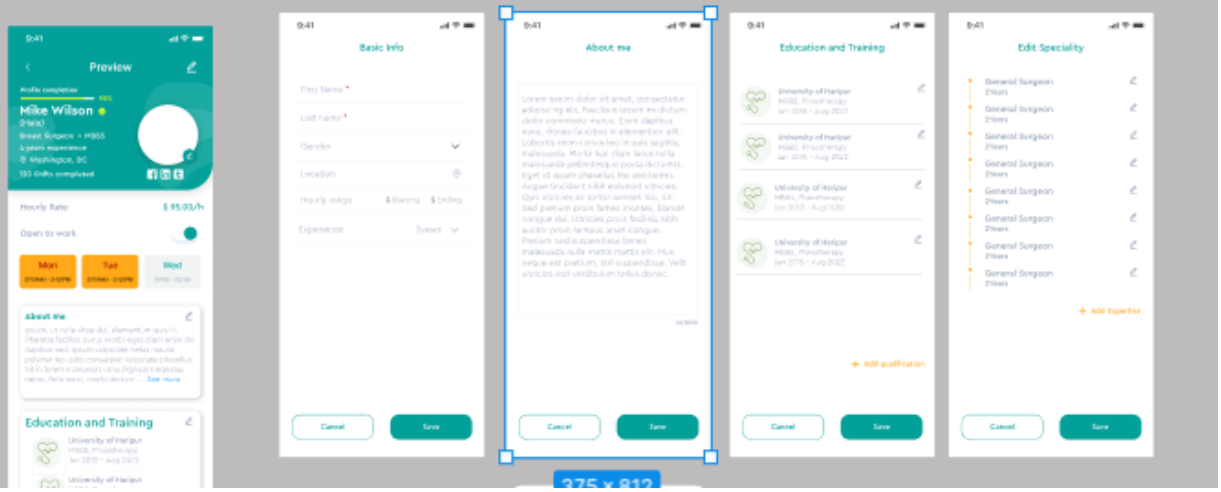
4.3 SCREENSHOT OF HEALTH PROVIDER PAGES APP

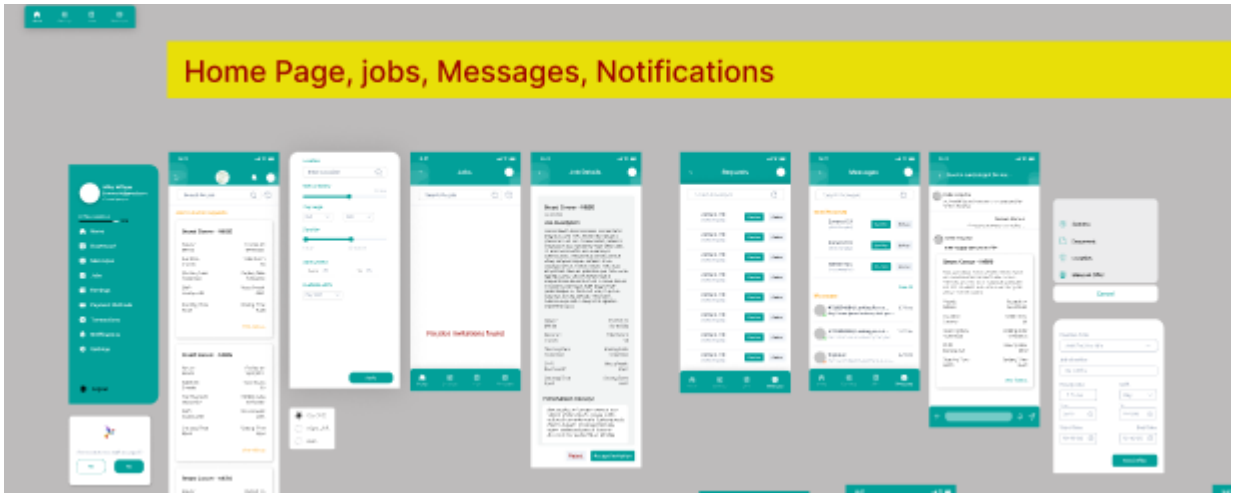
Splash Screen, Registration

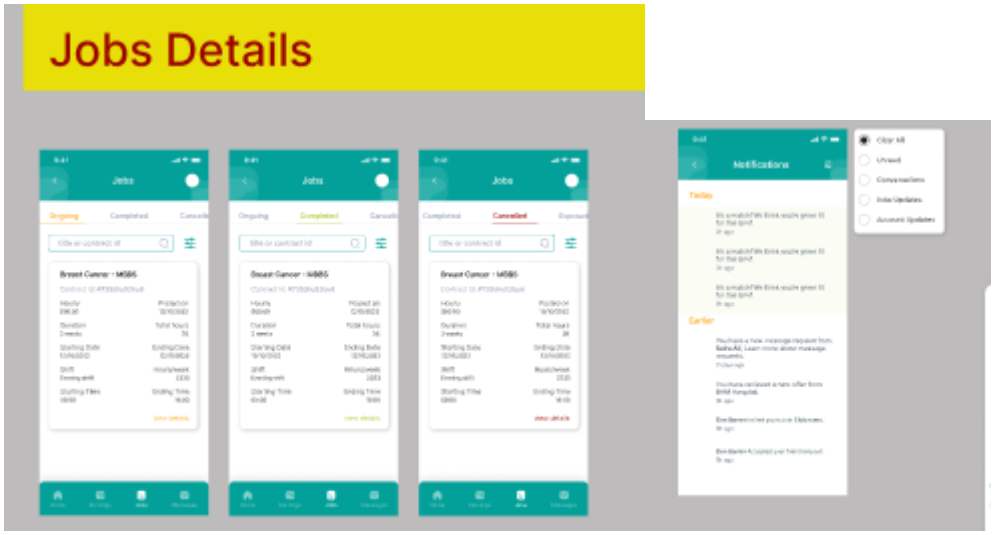


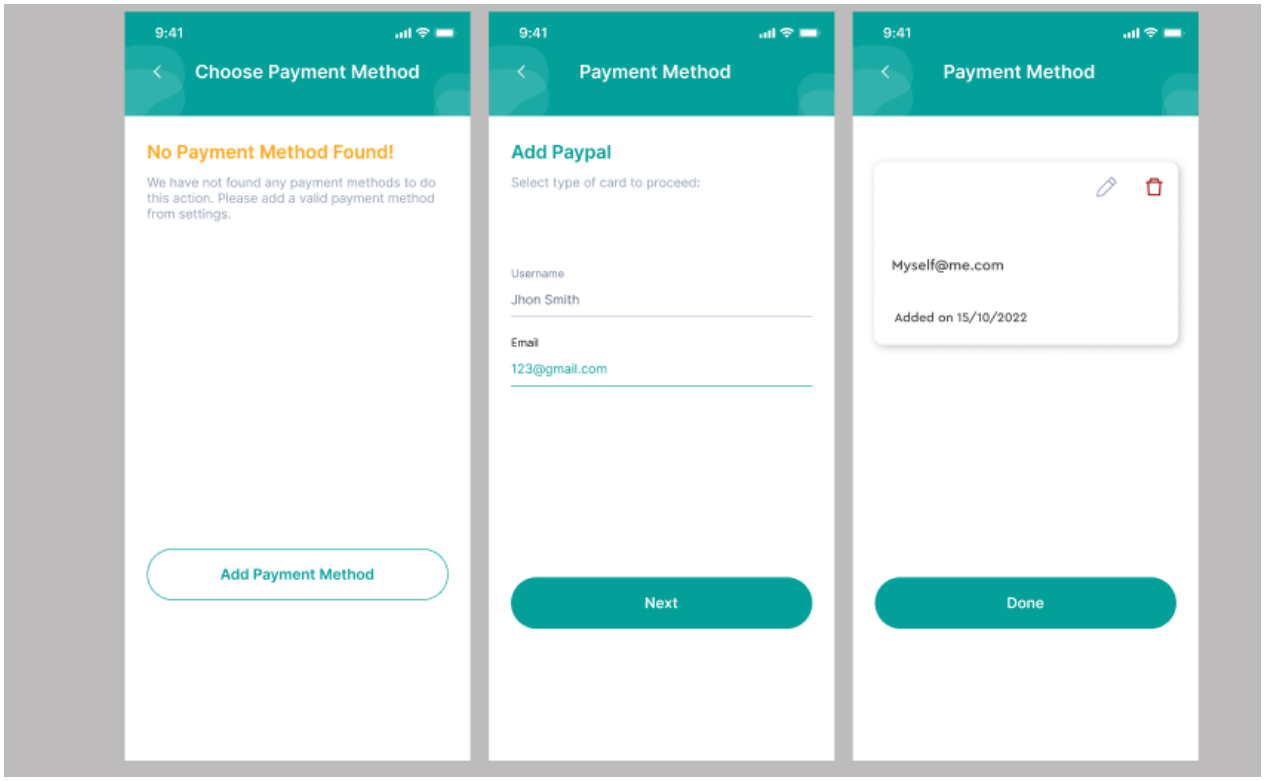


Profile View, Edit

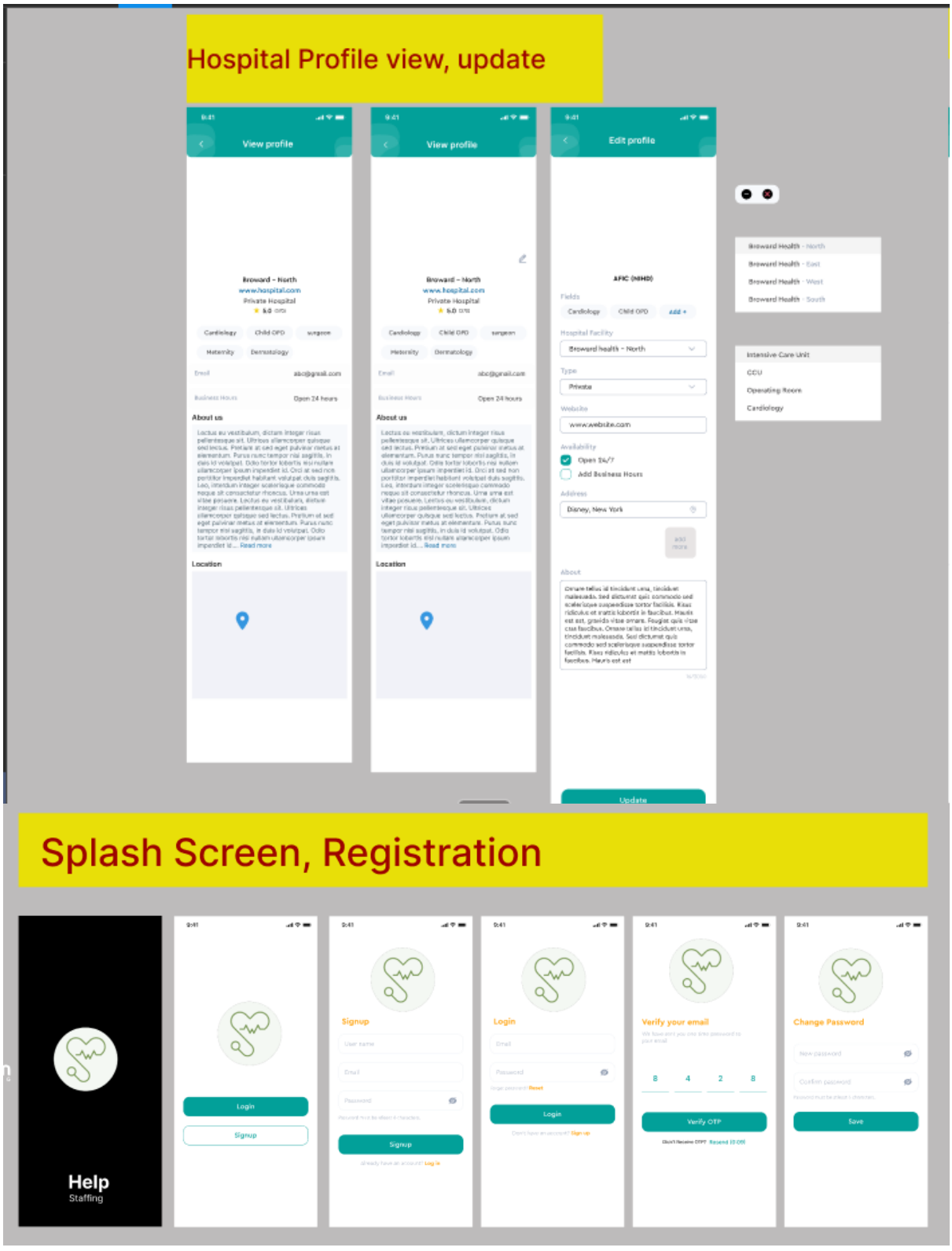


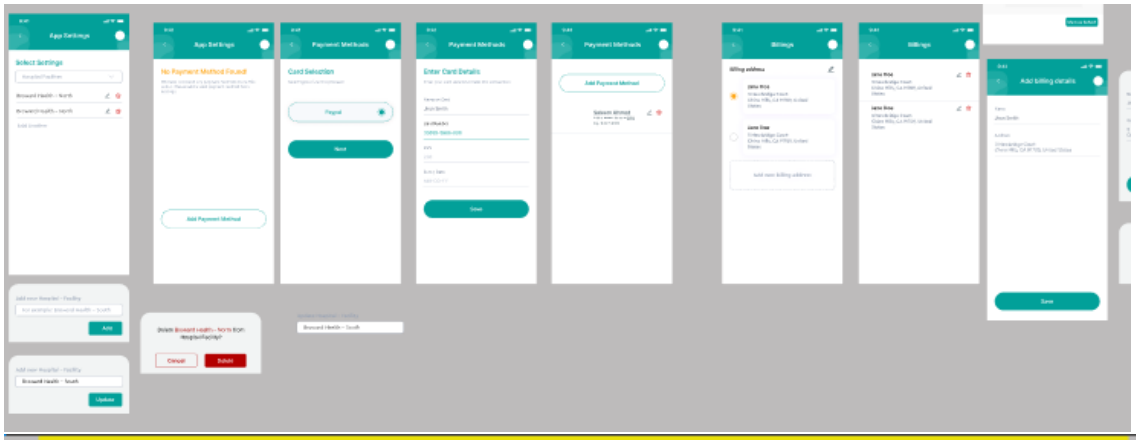

Home Page, jobs, Messages, Notifications

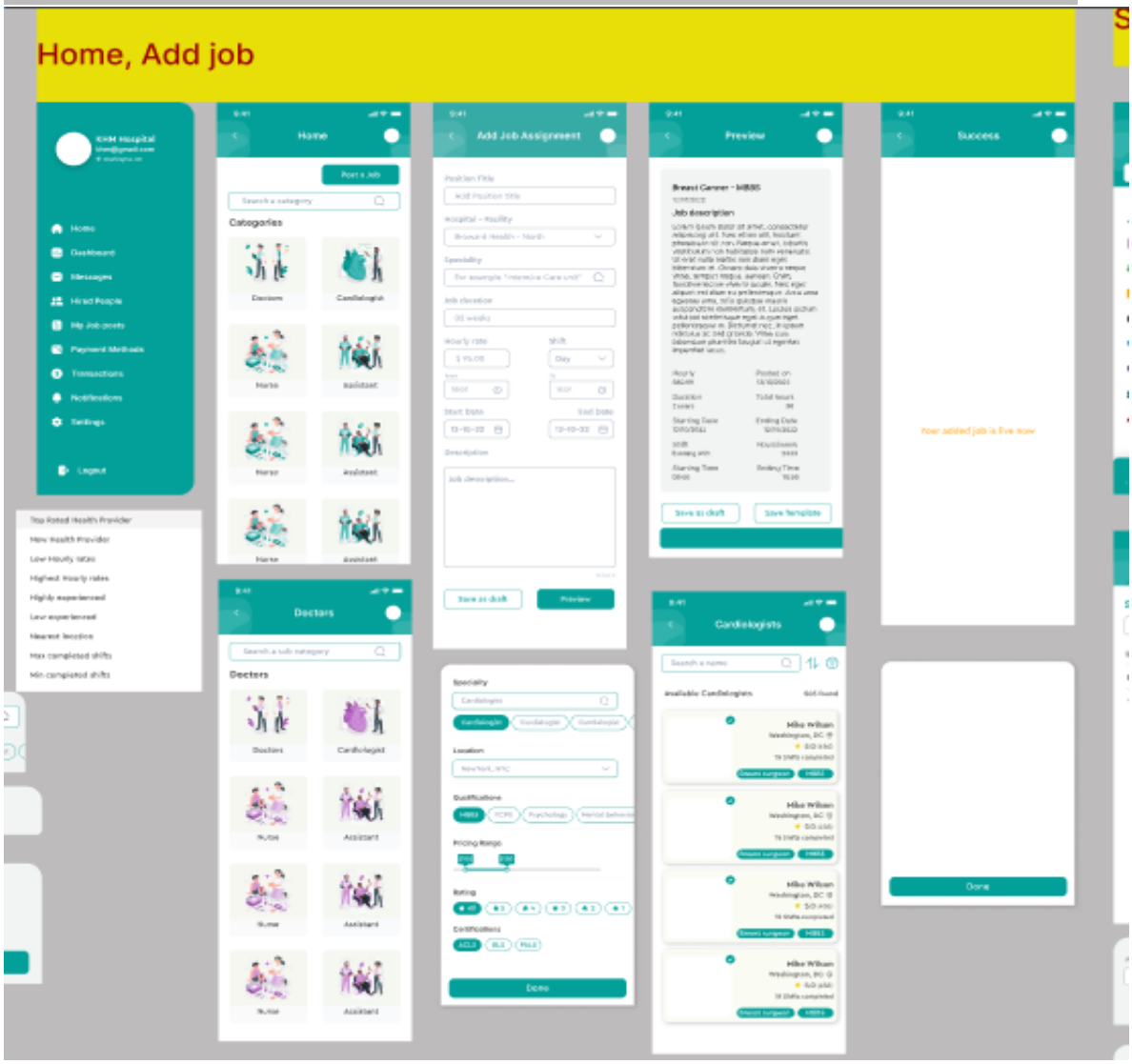

Jobs Details


9:41
Choose Payment Method
No Payment Method Found!
We have not found any payment methods to do this action. Please add a valid payment method from settings.
Add Payment Method
9:41
Payment Method
Add Paypal
Select type of card to proceed:
Username
Jhon Smith
Email
123@gmail.com
Next
9:41
Payment Method
Myself@me.com
Added on 15/10/2022
Done

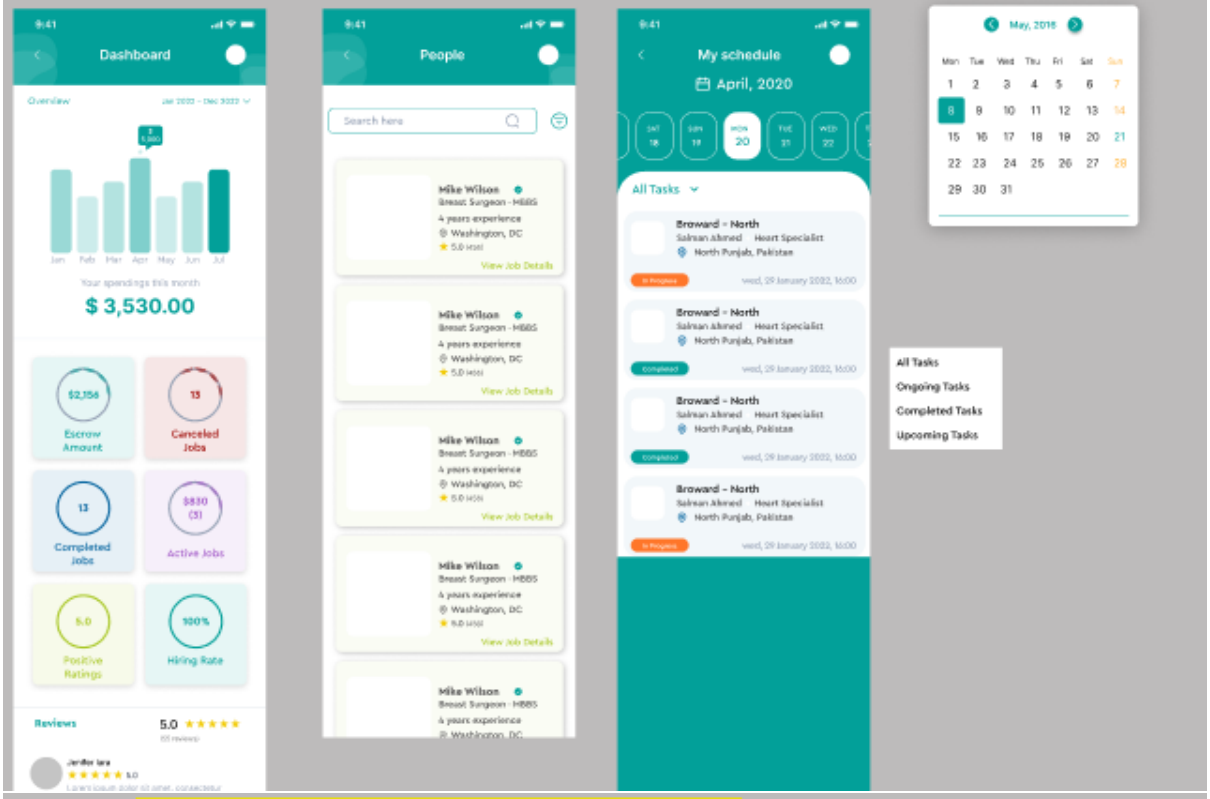
4.4 SCREENSHOT OF HOSPITALS PAGES APP



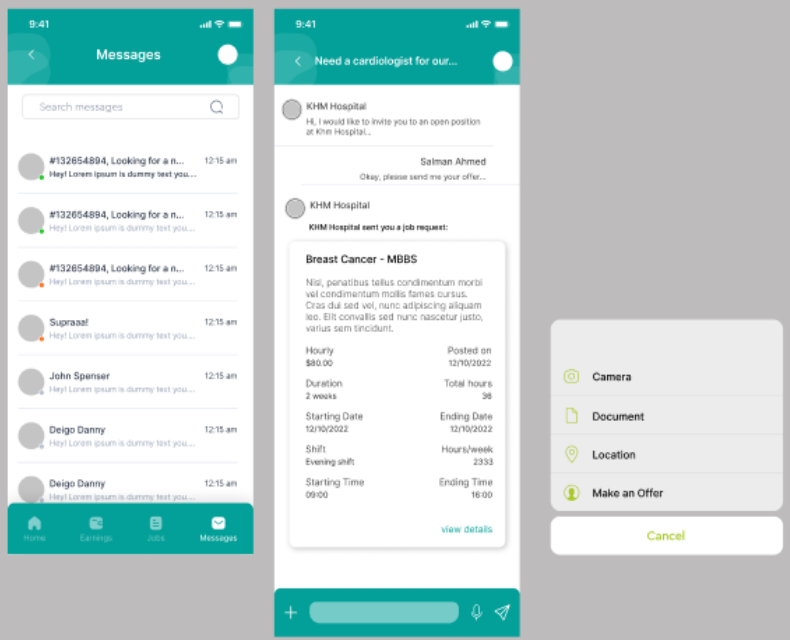


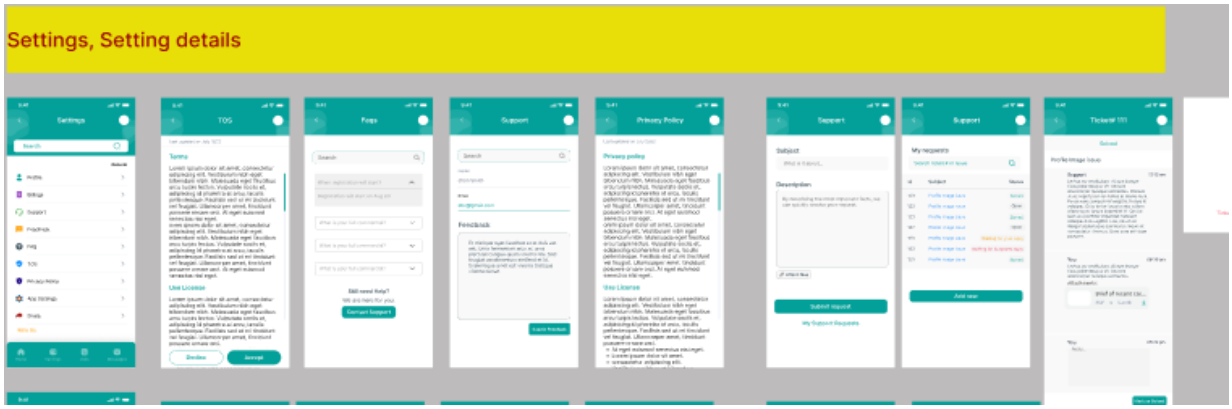

Home, Add job
Home
Add Job Assignment
Preview
Success
Doctors
Cardiologists

Dashboard, Hired people

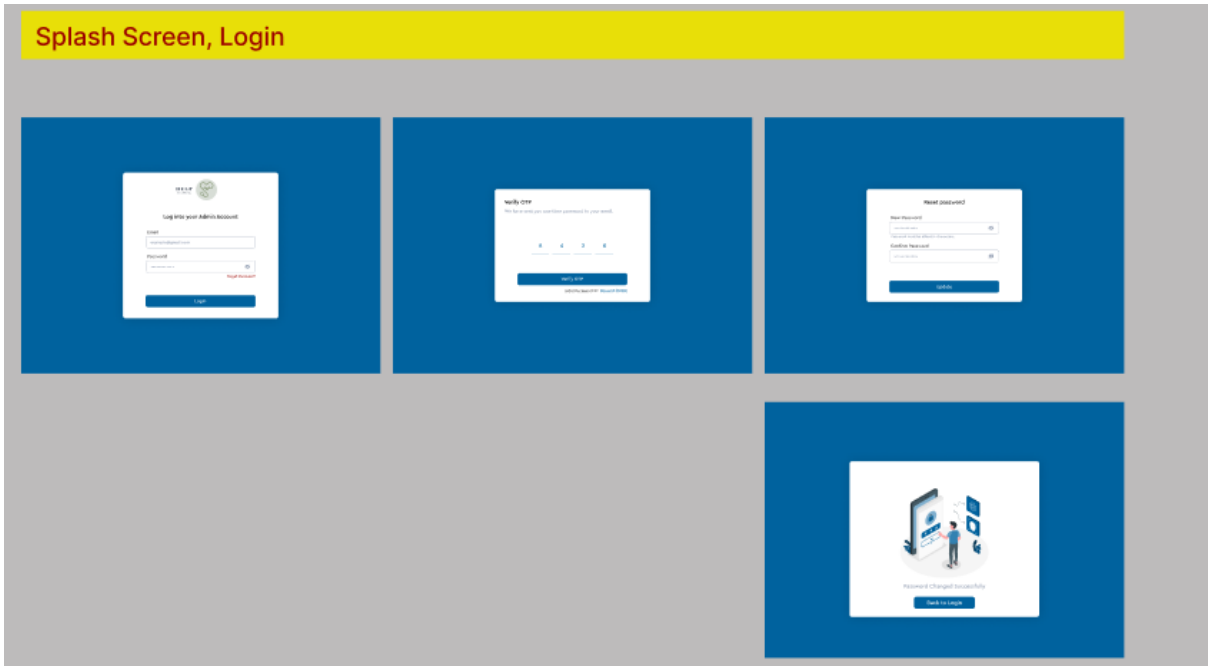


Messages (Details)

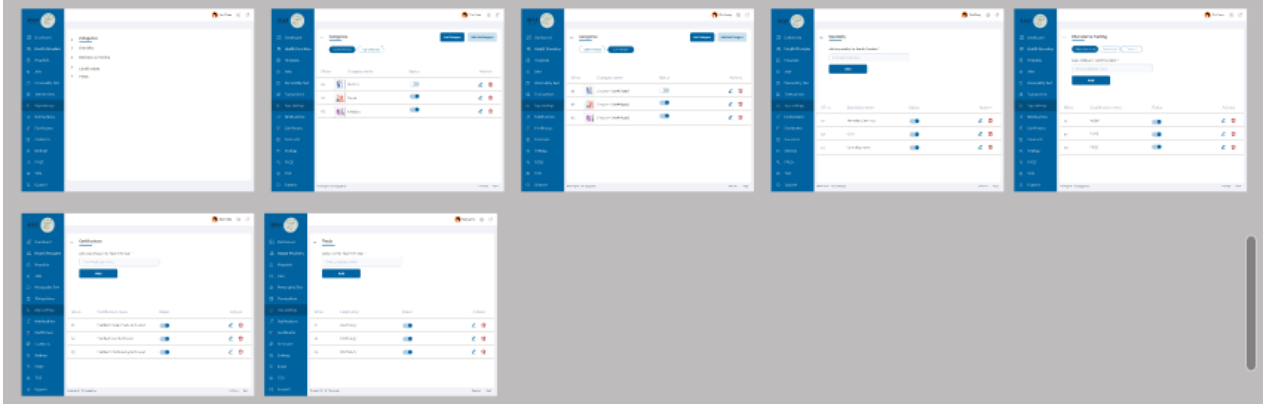




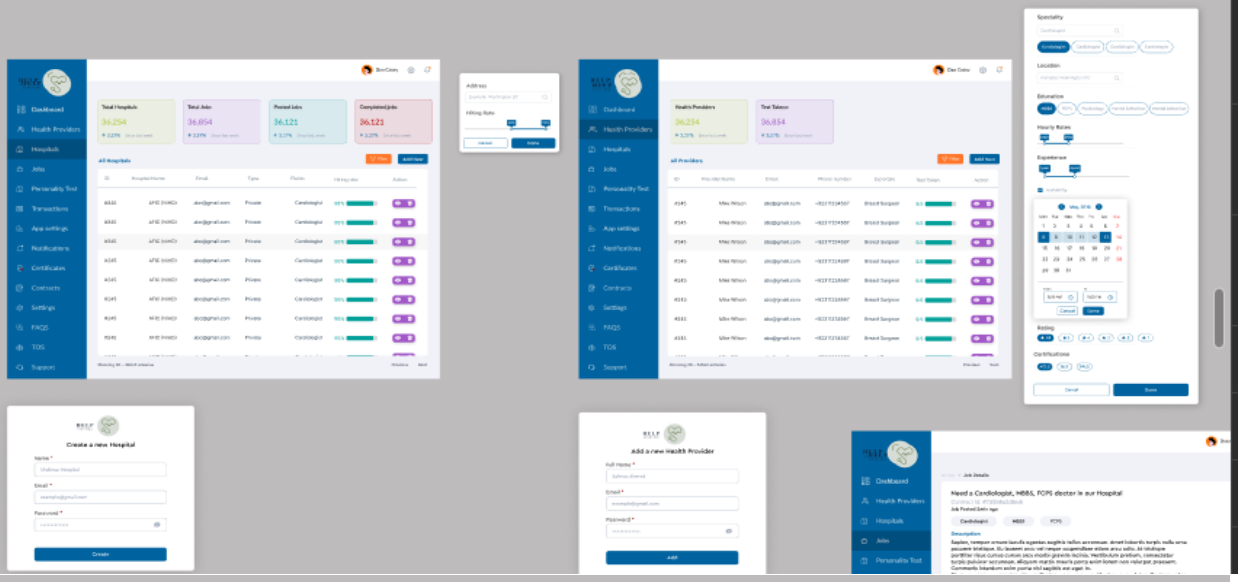
4.5 SCREENSHOT OF ADMIN



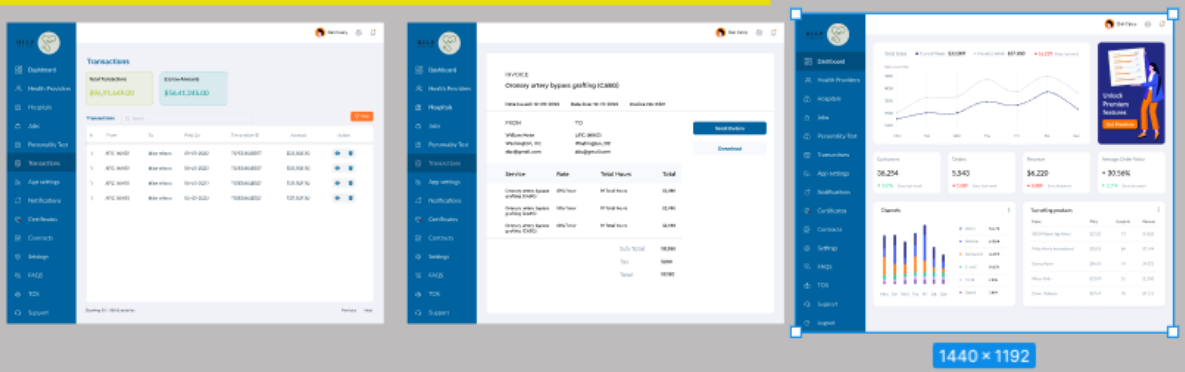
App settings



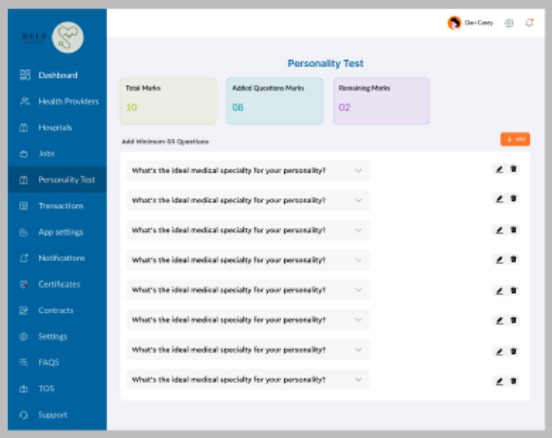
Hospitals, Providers details

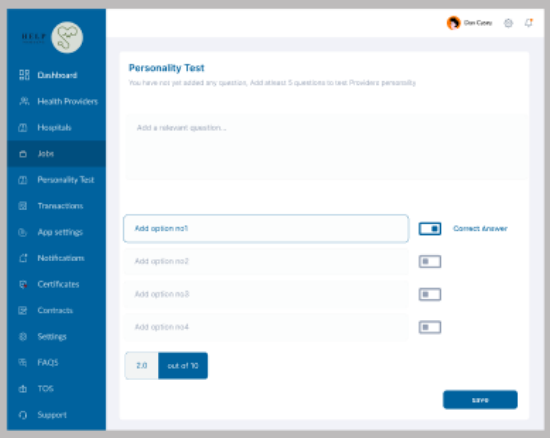


Transactions, Invoice

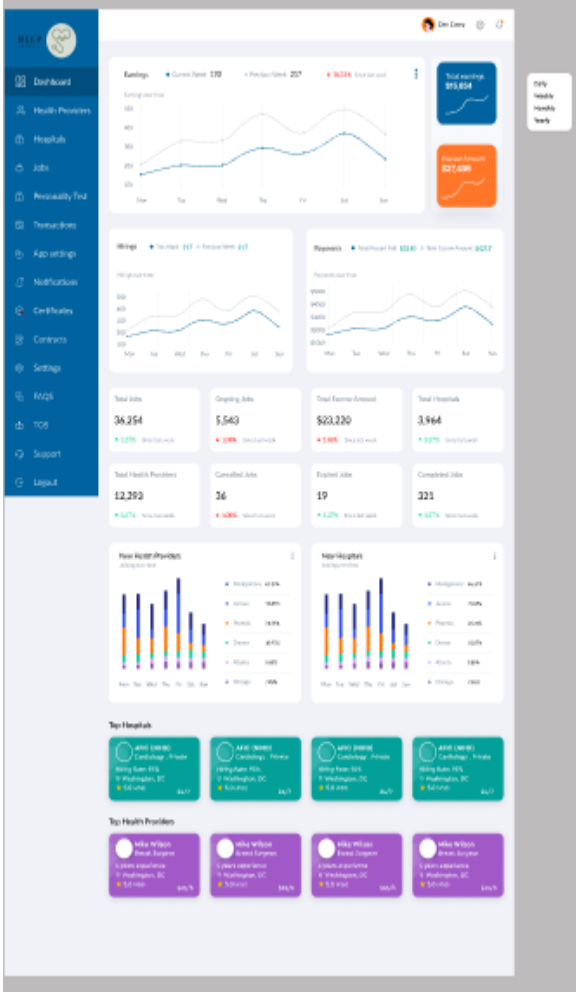


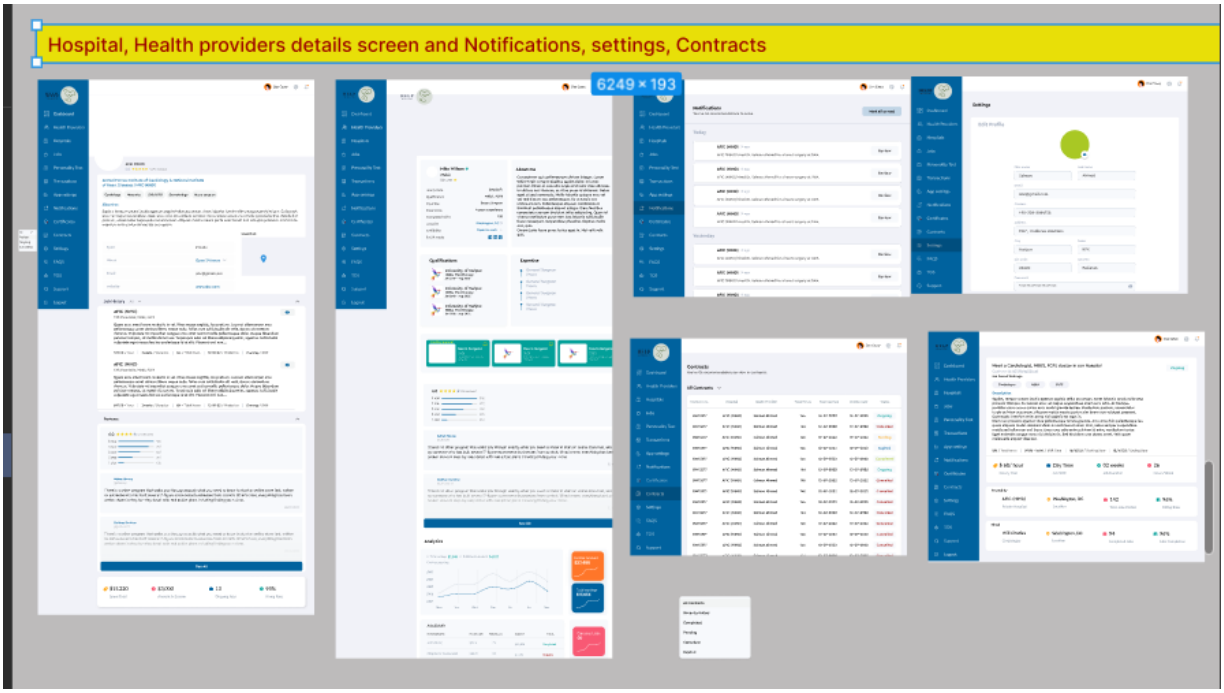
Personality Test



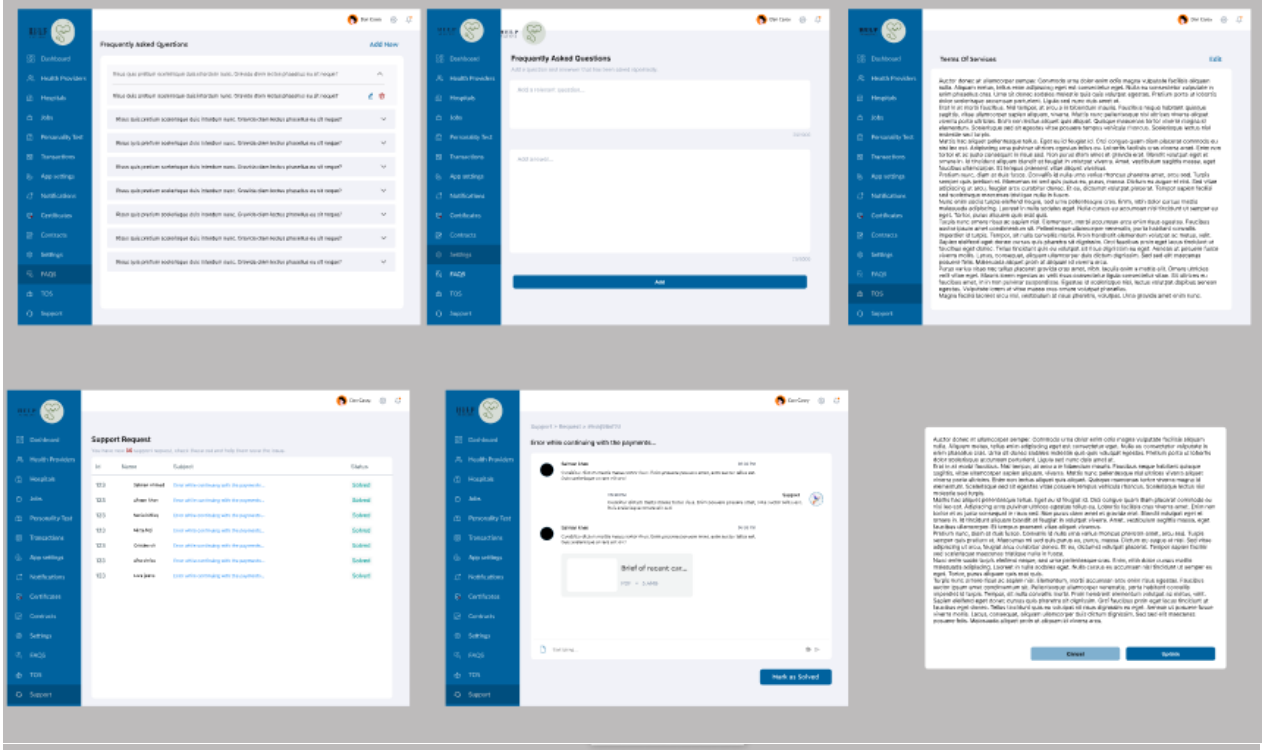


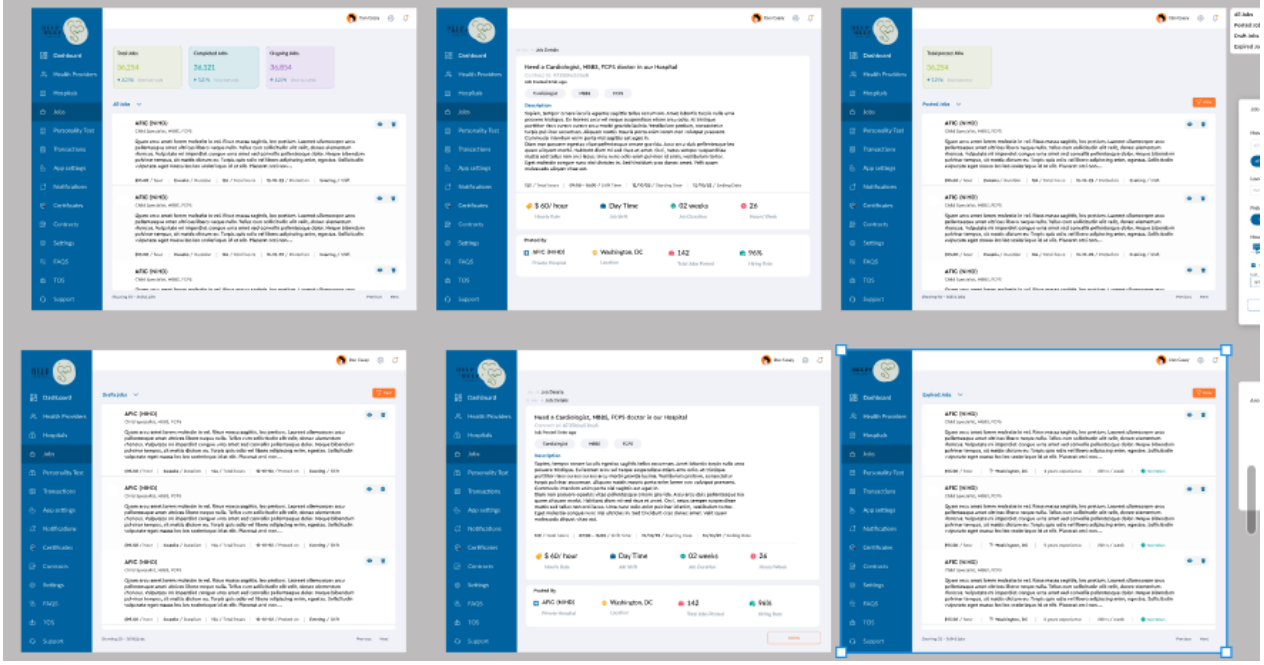
DashBoard

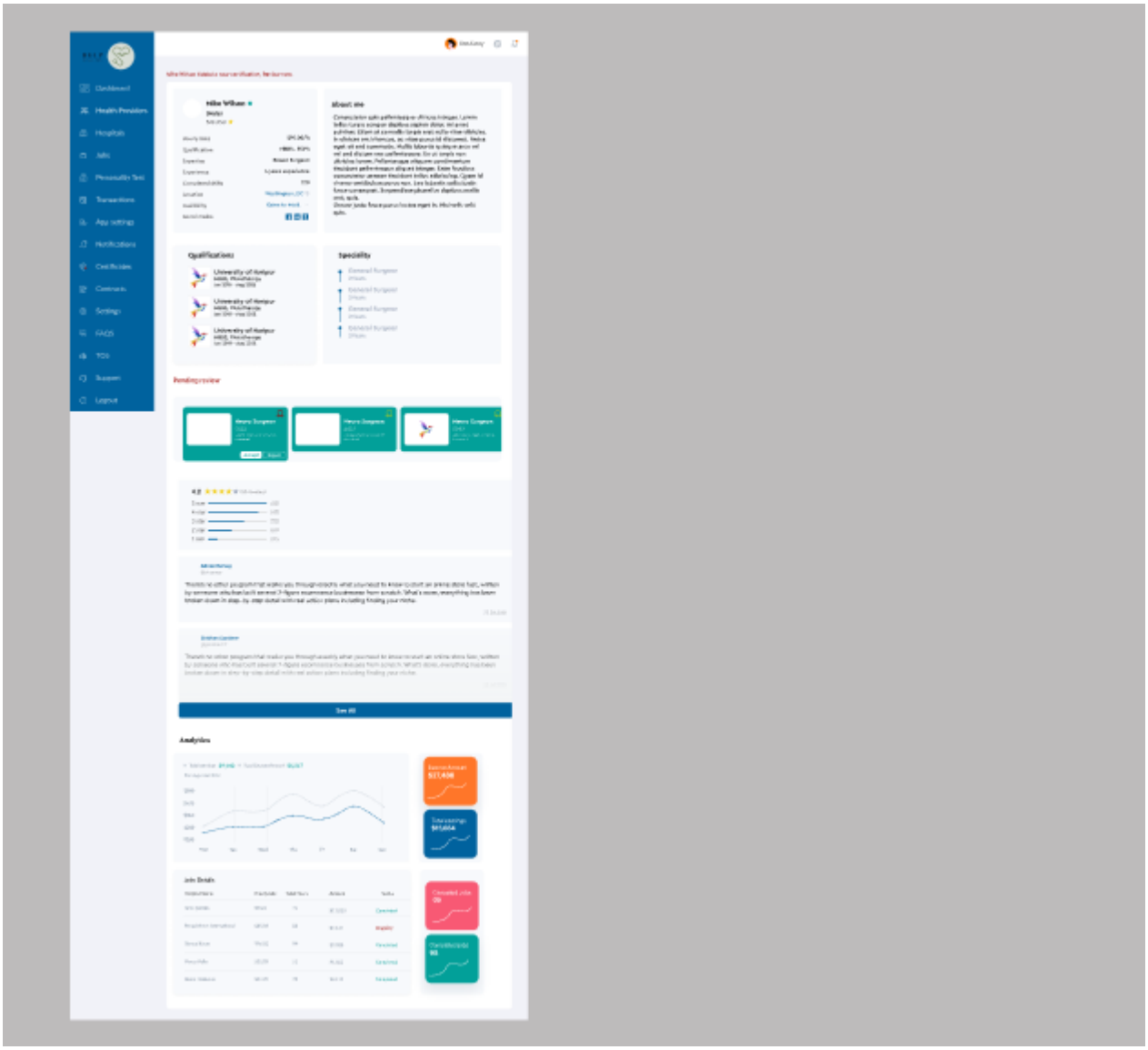



Hospital, Health providers details screen and Notifications, settings, Contracts
6249 × 193

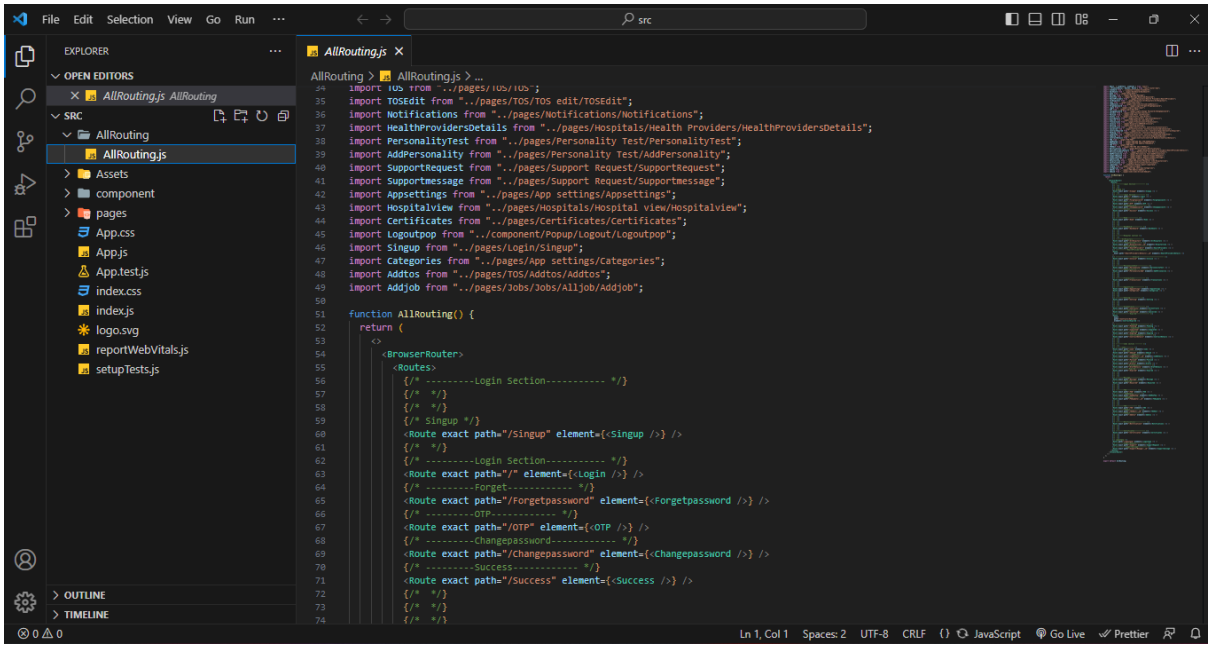
Frequently Asked Questions, TOS

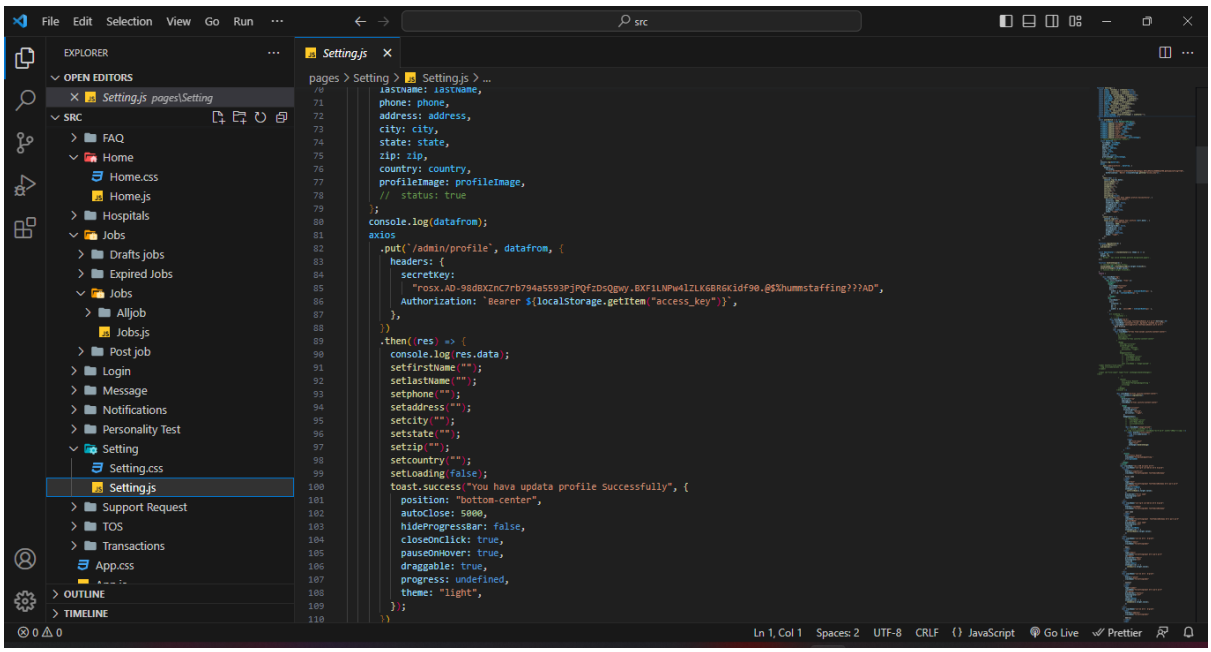


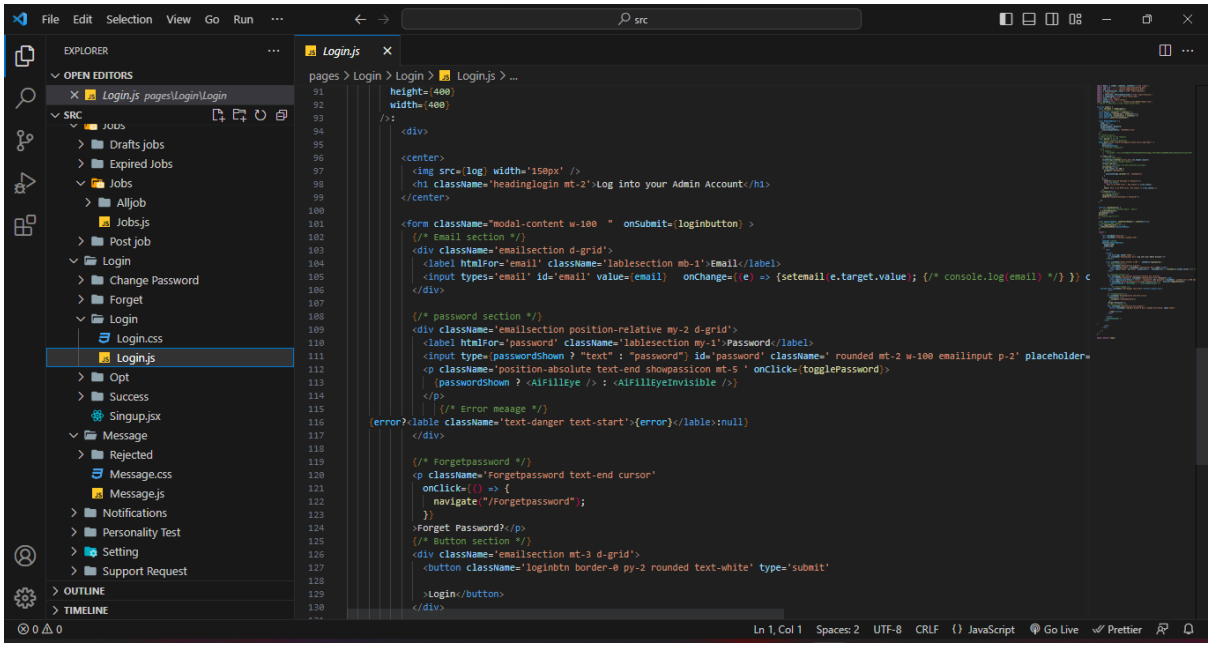




4.6 CODING SCREENSHOT






File Edit Selection View Go Run
src
EXPLORER
OPEN EDITORS
Login.js pages\Login\Login
SRC
Drafts jobs
Expired Jobs
Jobs
Alljob
Jobs.js
Post job
Login
Change Password
Forget
Login
Login.css
Login.js
Opt
Success
Singup.jsx
Message
Rejected
Message.css
Message.js
Notifications
Personality Test
Setting
Support Request
OUTLINE
TIMELINE
Login.js
pages > Login > Login > Login.js > ...
height={400}
width={400}
/>:

<h1 className='headinglogin mt-2'>Log into your Admin Account</h1>

<form className="modal-content w-100 " onSubmit={loginbutton} >
{/* Email section */}

<label htmlFor='email' className='lablesection mb-1'>Email</label>
<input types='email' id='email' value={email} onChange={(e) => {setemail(e.target.value); {/* console.log(email) */} }} c

{/* password section */}

<label htmlFor='password' className='lablesection my-1'>Password</label>
<input type={passwordShown ? "text" : "password"} id='password' className=' rounded mt-2 w-100 emailinput p-2' placeholder=

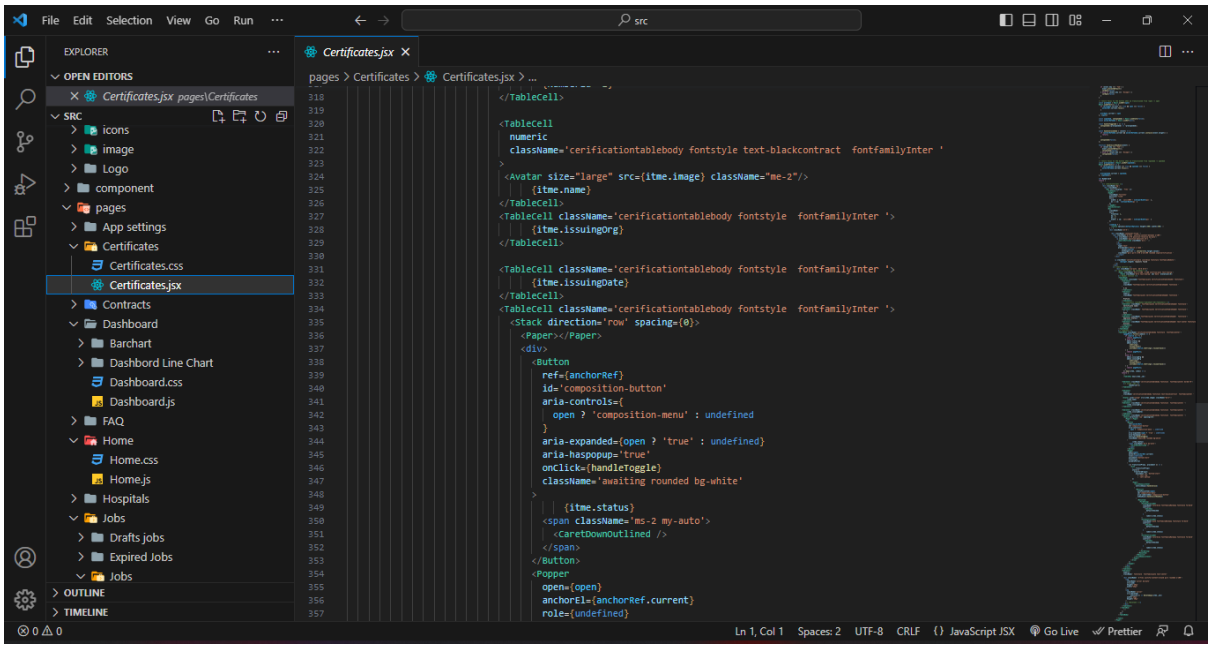
{passwordShown ? <AiFillEye /> : <AiFillEyeInvisible />}

{/* Error meaage */}
{error?<lable className='text-danger text-start'>{error}</lable>:null}

{/* Forgetpassword */}
<p className='Forgetpassword text-end cursor'
onClick={() => {
navigate("/Forgetpassword");
}}
>Forget Password?
{/* Button section */}

<button className='loginbtn border-0 py-2 rounded text-white' type='submit'
>Login</button>

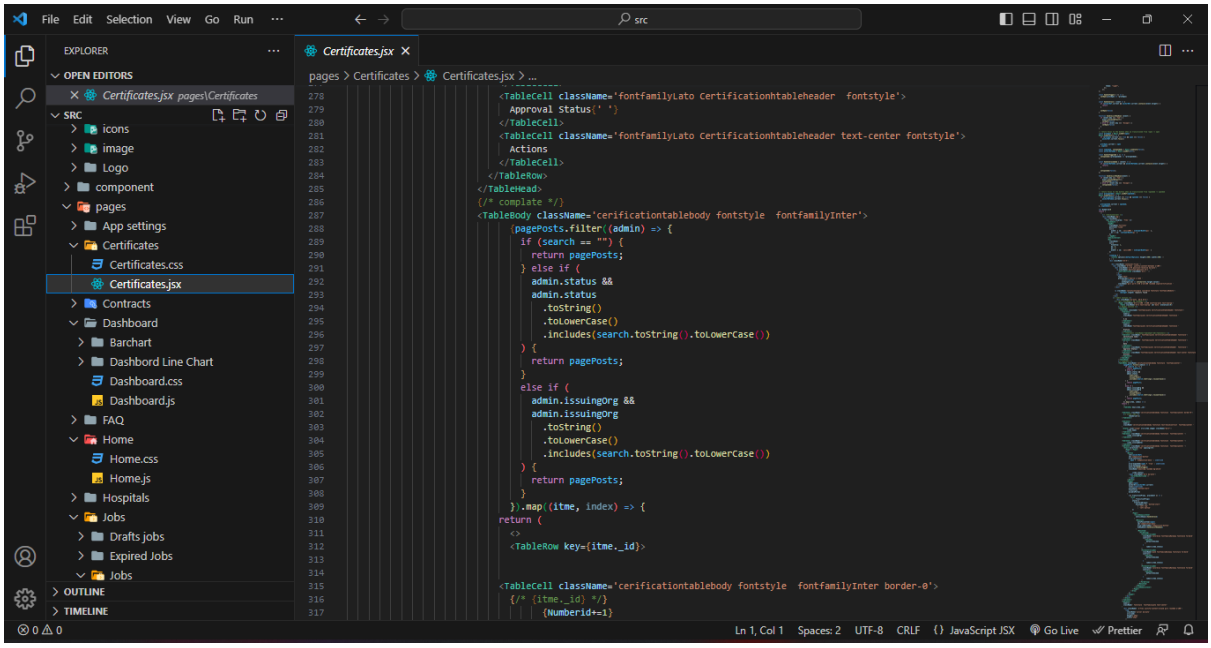
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF JavaScript Go Live Prettier

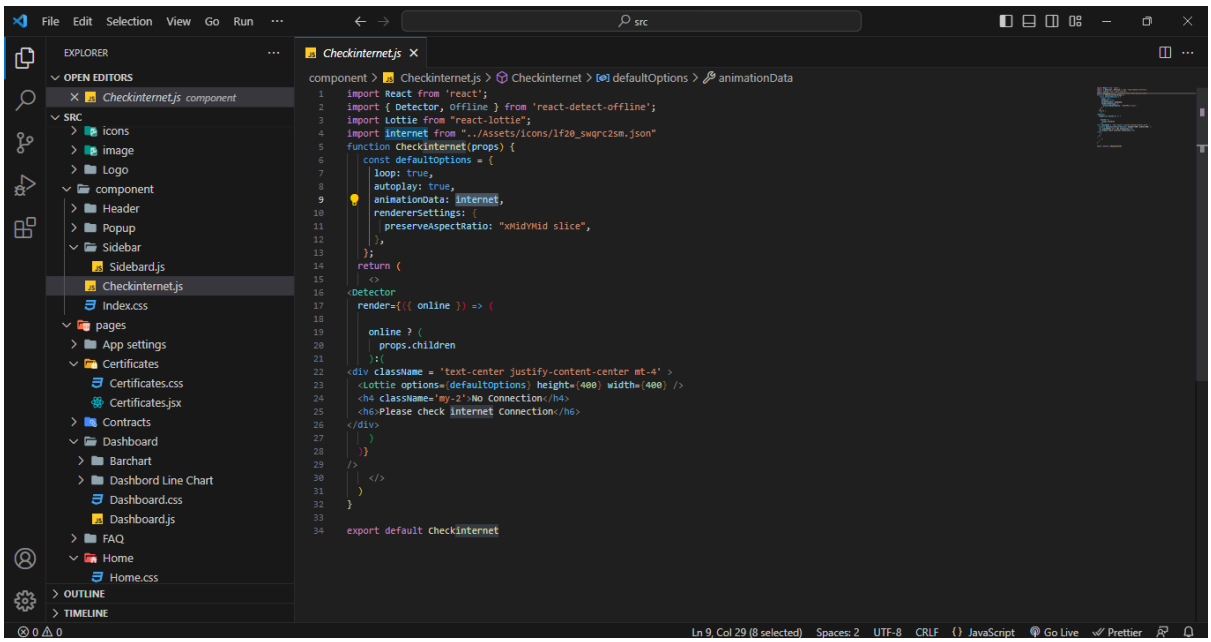

File Edit Selection View Go Run
src
EXPLORER
OPEN EDITORS
Certificates.jsx pages\Certificates
SRC
icons
image
Logo
component
pages
App settings
Certificates
Certificates.css
Certificates.jsx
Contracts
Dashboard
Barchart
Dashbord Line Chart
Dashboard.css
Dashboard.js
FAQ
Home
Home.css
Home.js
Hospitals
Jobs
Drafts jobs
Expired Jobs
Jobs
OUTLINE
TIMELINE
Certificates.jsx
pages > Certificates > Certificates.jsx > ...
</TableCell>
<TableCell
numeric
className='cerificationtablebody fontstyle text-blackcontract fontfamilyInter '
>
<Avatar size="large" src={itme.image} className="me-2"/>
{itme.name}
</TableCell>
<TableCell className='cerificationtablebody fontstyle fontfamilyInter '>
{itme.issuingOrg}
</TableCell>
<TableCell className='cerificationtablebody fontstyle fontfamilyInter '>
{itme.issuingDate}
</TableCell>
<TableCell className='cerificationtablebody fontstyle fontfamilyInter '>
<Stack direction='row' spacing={0}>
<Paper></Paper>

<Button
ref={anchorRef}
id='composition-button'
aria-controls={
open ? 'composition-menu' : undefined
}
aria-expanded={open ? 'true' : undefined}
aria-haspopup='true'
onClick={handleToggle}
className='awaiting rounded bg-white'
>
{itme.status}

<CaretDownOutlined />

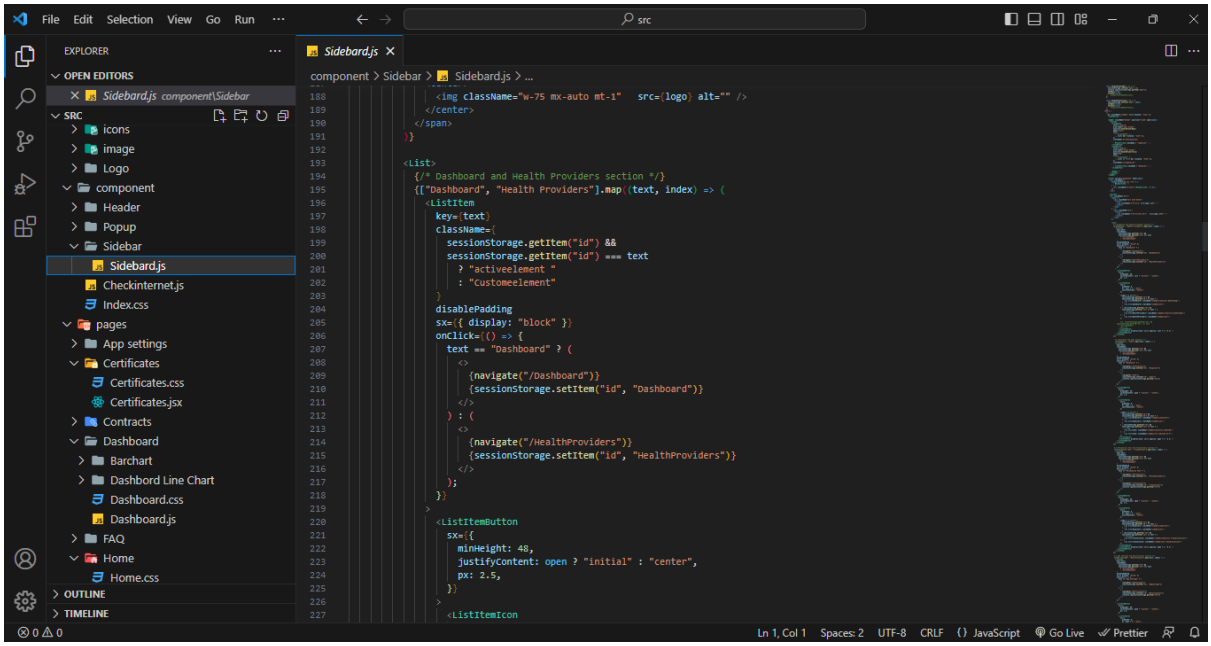
</Button>
<Popper
open={open}
anchorEl={anchorRef.current}
role={undefined}
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF JavaScript JSX Go Live Prettier

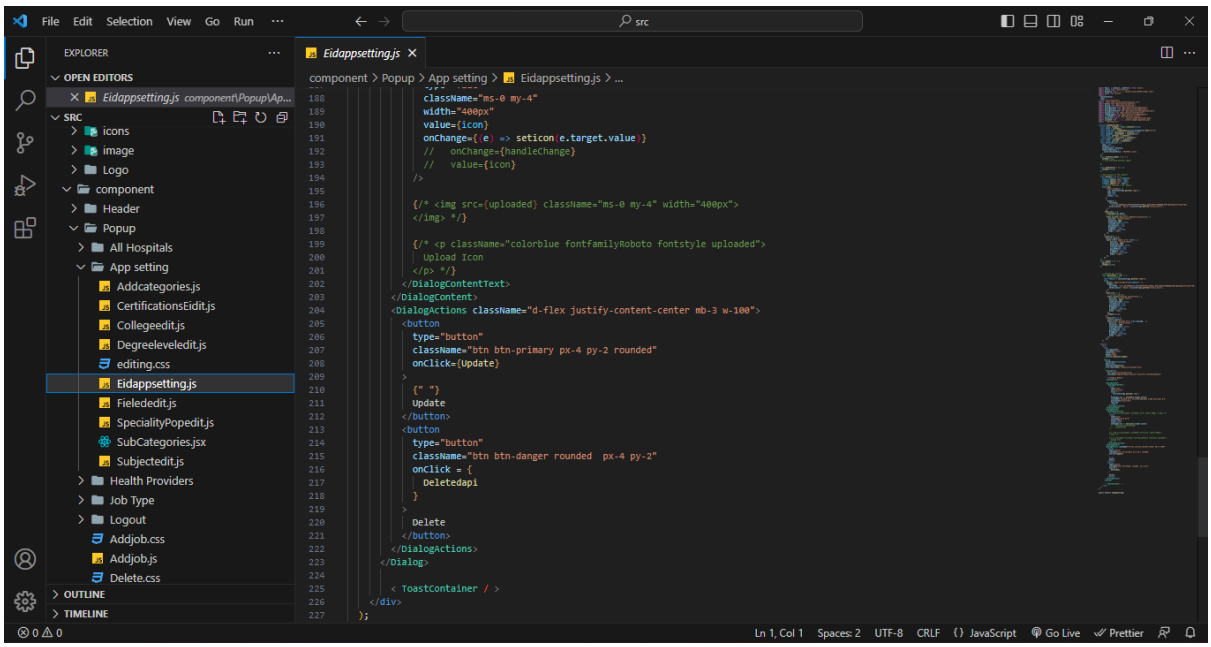

File Edit Selection View Go Run
src
EXPLORER
OPEN EDITORS
Certificates.jsx pages\Certificates
SRC
icons
image
Logo
component
pages
App settings
Certificates
Certificates.css
Certificates.jsx
Contracts
Dashboard
Barchart
Dashbord Line Chart
Dashboard.css
Dashboard.js
FAQ
Home
Home.css
Home.js
Hospitals
Jobs
Drafts jobs
Expired Jobs
Jobs
OUTLINE
TIMELINE
Certificates.jsx
pages > Certificates > Certificates.jsx > ...
<TableCell className='fontfamilyLato CertificationhtableheaderfontstyleApproval Status</TableCell>
<TableCell className='fontfamilyLato CertificationhtableheaderfontstyleActions</TableCell>
</TableRow>
</TableHead>
{/* complate */}
<TableBody className='cerificationtablebody fontstyle fontfamilyInter'>
{pagePosts.filter((admin) => {
if (search == "") {
return pagePosts;
} else if (
admin.status &&
admin.status
.toString()
.toLowerCase()
.includes(search.toString().toLowerCase())
) {
return pagePosts;
}
else if (
admin.issuingOrg &&
admin.issuingOrg
.toString()
.toLowerCase()
.includes(search.toString().toLowerCase())
) {
return pagePosts;
}
}).map((itme, index) => {
return (
<>
<TableRow key={itme._id}>
<TableCell className='cerificationtablebody fontstyle fontfamilyInter border-0'>
{/* {itme._id} */}
{Numberid+=1}
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF JavaScript JSX Go Live Prettier


File Edit Selection View Go Run
src
EXPLORER
OPEN EDITORS
Checkinternet.js component
SRC
icons
image
Logo
component
Header
Popup
Sidebar
Sidebard.js
Checkinternet.js
Index.css
pages
App settings
Certificates
Certificates.css
Certificates.jsx
Contracts
Dashboard
Barchart
Dashbord Line Chart
Dashboard.css
Dashboard.js
FAQ
Home
Home.css
OUTLINE
TIMELINE
Checkinternet.js
component > Checkinternet.js > Checkinternet > defaultOptions > animationData
import React from 'react';
import { Detector, Offline } from 'react-detect-offline';
import Lottie from "react-lottie";
import internet from "../Assets/icons/lf20_swqrc2sm.json"
function Checkinternet(props) {
const defaultoptions = {
loop: true,
autoplay: true,
animationData: internet,
rendererSettings: {
preserveAspectRatio: "xMidYMid slice",
},
};
return (
<>
<Detector
render={({ online }) => (
online ? (
props.children
):(

<Lottie options={defaultoptions} height={400} width={400} />
<h4 className='my-2'>No Connection</h4>
<h6>Please check internet Connection</h6>

)}
/>
</>
)
}
export default Checkinternet
Ln 9, Col 29 (8 selected) Spaces: 2 UTF-8 CRLF JavaScript Go Live Prettier





4.7 BRANCH TESTING

Branch coverage is a testing method, which aims to ensure that each one of the possible branches from each decision point is executed at least once and thereby ensuring that all reachable code is executed. That is, every branch is taken each way, true and false. It helps in validating all the branches in the code making sure that no branch leads to abnormal behavior of the application. We test the whole branches of the project one by one. Where we find some kind of errors then we

will fix these errors. Some of the errors we will solve during the user testing module also.

4.8 BOUNDARY VALUE TESTING

This testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery. We have seen that there are high chances of finding the defects at the boundaries of a partition (E.g., A developer using >10 instead of >= 10 for a condition). Equivalence partitioning alone was not sufficient to catch such defects.

Hospital Doctor analysis can perform at all test levels, and it's primarily used for a range of expertise.

4.9 RESULT VALIDATIONS

The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements.

Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfills its intended use when deployed in an appropriate environment.

Result Validations

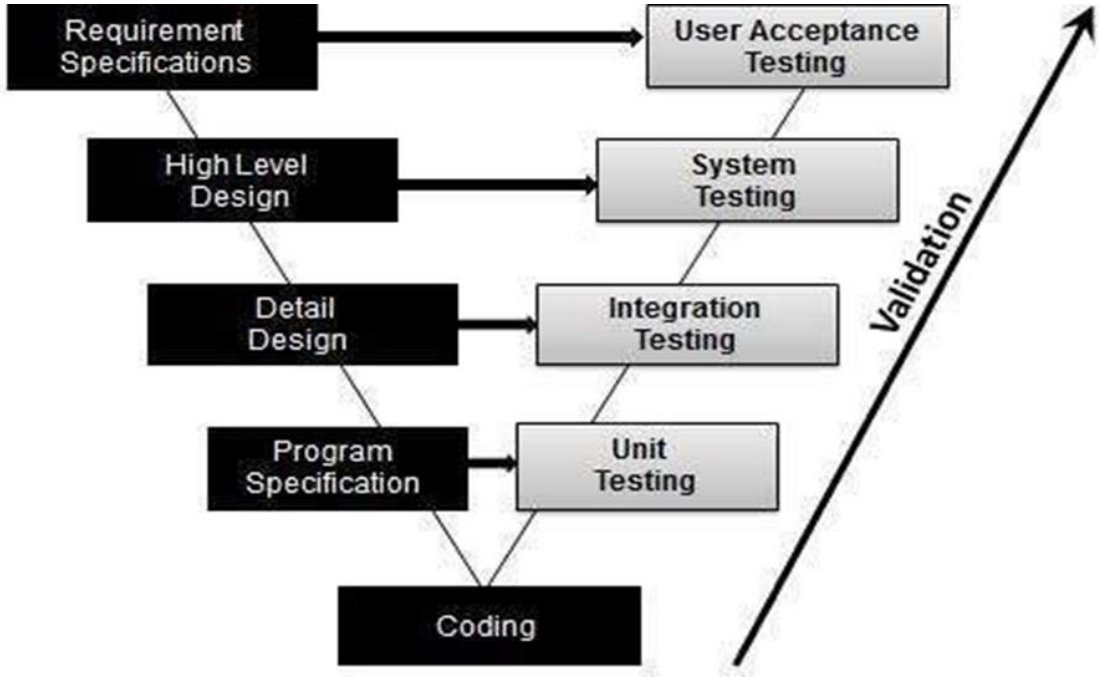


FIGURE V: RESULT AND VALIDATION

4.9 CONCLUSIONS

In conclusion, effective hospital management is crucial for the provision of high-quality healthcare services to patients. Effective hospital management involves a range of activities, including strategic planning, financial management, human resource management, quality assurance, and risk management. Hospital managers must possess strong leadership skills, be knowledgeable about health care policies and regulations, and be able to adapt to changes in the healthcare industry. Effective hospital management requires collaboration and communication among healthcare professionals, including physicians, nurses, and administrators, to ensure the delivery of safe and efficient care to patients. By implementing best practices in hospital management, healthcare organizations can improve patient outcomes, increase efficiency, and achieve financial sustainability.

CHAPTER 5

REFERENCE MANUAL OR TRAINING MANUAL

5.1 TRAINING AND GUIDELINE

Training in hospital management is critical to ensure that healthcare managers possess the skills and knowledge required to effectively manage healthcare organizations. The following are some of the areas that are typically covered in hospital management training:

1. **Leadership and communication skills**: Hospital managers need to possess strong leadership skills to inspire and motivate staff, and effective communication skills to effectively communicate with patients, staff, and other stakeholders.
2. **Financial management:** Hospital managers should have a good understanding of financial management principles to manage budgets, track financial performance, and make informed financial decisions.
3. **Human resource management:** Hospital managers need to be knowledgeable about human resource management practices, including recruitment, retention, training, and development of staff.

4. **Quality improvement:** Hospital managers should be familiar with quality improvement principles, such as continuous improvement, risk management, and patient safety.
5. **Healthcare policies and regulations:** Hospital managers need to keep up to date with healthcare policies and regulations at the local, state, and federal levels.
6. **Information technology:** Hospital managers should have a good understanding of information technology systems and how they can be used to improve healthcare delivery.
7. **Strategic planning:** Hospital managers need to have the skills to develop strategic plans and implement them effectively.

Training in hospital management can be obtained through various channels, including academic programs, conferences, workshops, and online courses. It is essential that hospital managers continue to develop their skills and knowledge through ongoing training to keep up with changes in the healthcare industry and to improve the delivery of healthcare services.

5.2 JOBS DESCRIPTION

A hospital manager is responsible for overseeing the day-to-day operations of a healthcare facility, with the aim of ensuring the delivery of high-quality healthcare services to patients. The following are some of the key responsibilities of a hospital manager:

1. **Strategic planning:** Develop and implement strategic plans to achieve the hospital's mission and objectives.
2. **Financial management:** Manage the hospital's budget, financial performance, and resources.
3. **Human resource management:** Recruit, train, manage, and evaluate staff, and ensure compliance with employment laws and regulations.
4. **Quality improvement:** Develop and implement quality improvement programs to ensure patient safety, satisfaction, and outcomes.
5. **Facility management:** Ensure that the hospital's physical facilities are properly maintained, clean, and safe for patients and staff.
6. **Information management:** Oversee the collection, storage, and retrieval of patient and hospital data, and ensure compliance with data privacy regulations.

A hospital manager should possess strong leadership, communication, and problem-solving skills, and have a good understanding of healthcare policies and regulations. They should also have a background in healthcare administration or management and hold a relevant degree or certification. The specific job description and requirements may vary depending on the size, complexity, and type of healthcare organization.

5.3 QUALIFICATION

Successful healthcare administrators of hospitals manage and supervise all areas of healthcare delivery in the hospital. Administrators may manage a single department or oversee an entire facility; they are responsible for budgeting, health information systems, facility maintenance, maintaining the facility's records and creating work schedules for hospital administrative staff. The administration of a hospital requires the ability to manage a business, deliver quality health care and follow government regulations.