CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

SEVAM- HEALTH LAB MANAGEMENT SYSTEM

by

Adith AV (2147206)

Yash (2147236)

J Maglin Monika(2147252)

Under the guidance of

Dr. Arul Kumar

A Project report submitted in partial fulfillment of the requirements for the award of degree of Master of ComputerApplications of CHRIST (Deemed to be University)

MAY - 2022

CERTIFICATE

This is to certify that the report titled **SEVAM HEALTH LAB MANAGEMENT SYSTEM** *is a bonafide record of work done by* ***Adith AV (2147206), Yash (2147236) and J Maglin Monika (2147252)*** *of CHRIST (Deemed to be University), Bengaluru, in partial fulfillment of the requirements of III Trimester MCA during the year 2022.*

Head of the Department
Valued-by:

Faculty In charge

Names	:	Dr. Sibu Dr. Rohini Dr. Arul Kumar
Register Number(s)	:	2147206, 2147236, 2147252
Examination Center	:	CHRIST (Deemed to be University)
Date	:	22 May 2022

ACKNOWLEDGMENT

I would like to express my profound gratitude to Mr. Ashok Emmanuel, Head of the Computer Science Department , and Mr. George Thomas (Dean) of Christ (Deemed to be University) for their contributions to the completion of my project titled SEVAM Health Lab Management System.

I would like to express my special thanks to our mentor Mr Arul Kumar for the time and efforts that he provided throughout the trimester. Your useful advice and suggestions were really helpful to us during the project's completion. In this aspect, I am eternally grateful to you.

I would like to acknowledge that this project was completed entirely by us and not by someone else.

Signature

Adith AV (2147206)

Yash (2147236)

J Maglin Monika (2147252)

Table of Contents

1 Introduction
- 1.1 Overview
- 1.2 Existing System and Disadvantages
- 1.3 Proposed System and Advantages

2 Requirement Specification
- 2.1 Requirement Description
- 2.2 Analysis Modeling (Any of, Use-case diagrams, Activity diagrams, Data-Flow Diagrams, State Diagrams, Sequence Diagrams, Class Diagrams)

3 Detailed Design
- 3.1 Architecture Diagram
- 3.2 Entity Relationship Diagram
- 3.3 Relationship Schema
- 3.4 Table Description
- 3.5 Interface Design

4 Construction
- 4.1 Software and Hardware Environments (with proper justification of choosing the software tools / frameworks)

5 Testing
- 5.1 Test Cases
- 5.2 Module Testing and Integration
- 5.3 Test Results
- 5.4 Defects

6 Screenshots
- 6.1 Page1
- 6.2 Page2
- 6.3 …

7 Conclusion

INTRODUCTION

Overview:

Project Name: Sevam Health Lab Management System
Document Title: Software Requirement Specification
Document No.: 1
Document Revision Number: 1

Scope:

This document gives the functional as well as non-functional requirements of Sevam Health Lab Management System.

Audience:

- Project Managers
- Testing team members
- Development team members
- Customers

Detail about the software system:

This Project is about Health Lab Management System. Initially, there were no online Health Care Lab management systems accessible. When COVID-19 went live, the majority of existing labs had to convert to online, and all new labs had to go online as well. Because there are numerous flaws in current systems, we decided to address many of them through this project.
SEVAM is a user-friendly online lab diagnostic platform. The system provides a complete solution from sample collection to final report update on the medical sample of users where users can book lab tests and can get their reports online . It includes agents conforming to booking with real-time updates on the sample status to the user. This system also includes an Admin side where admin can manage the processes happening from sample collection to report update.

Existing System and its disadvantage:

Some of the similar existing systems are as follows:

- Lalpathlabs is a medical lab startup which provides real-time booking of agents for sample collection with health packages. It also provides pharmaceutical facilities and sells medicines, suggests doctors.
- Healthians provide pathological laboratory solutions. It provides a variety of facilities including doctor suggestion, sample collection, report generation etc.

- Apollo 247 provides a complete healthcare solution from sample collection, to booking appointments of doctors, online consultancy, report generation and provides home delivery of medicines.
- 1mg is an online medical diagnostic system which provides facilities from online test booking to free doctors' consultancy and also sells medicines with free home delivery.
- Pharmeasy provides medical tests with free consultation of doctors and also provides a full-body-checkup package with certified labs and assured safety.

The major disadvantages of this software is as follows:

- No real-time location sharing of patients for ease of agents to collect samples.
- No Chat bot for solving patient queries.
- No reminders on bookings for patients and agents.
- No feedback on the report generated.

Proposed System and its advantage:

An online diagnostic lab reporting system that displays a variety of diagnoses while working. Patients can register on the site and log in with their email and password. After successful registration patients can now examine a variety of tests performed by the lab, as well as their fees. Users can book whatever test they require, and the system will compute prices and allow them to pay online once the booking is complete. The user is allowed to select the agent, time and date according to their convenience. The respective agents selected by the user are sent to their location to collect the samples for testing based on the time selected by the user and these samples are submitted to the laboratory. The lab generates a report for the sample . The patient or user receives an email notification of the test result after successful testing. This system also allows the administrator to upload a copy of the report and have it automatically emailed to the respective patient.

The advantages of the proposed system are as follows:

- This Real-time update on stage completion of samples.
- Setting up reminders for the booked test.
- Suggests doctors/clinics according to the report result.

SOFTWARE REQUIREMENTS SPECIFICATION

REQUIREMENT SPECIFICATION:

USERS :

Admin
Patients
Agents

MODULES :

Registers/Sign-in:
Users can register with their email, username and password and login using their username and password. Patients can register and login whereas admin and agents will be provided with username and password to login.

Manage tests :
Admin can add tests, delete, tests, update the prices of each test and add discounts if any .

Bookings:
Patients can add a booking , cancel the booking , update the timing of booking, check the cart and pay for it.

Manage bookings:
Agents have the provision to check for new bookings done . The agent can cancel , confirm booking based on his availability.

Booking status:
Check the status of each booking . If the sample is submitted by an agent the admin generates the report for the same.

Notifications / Reminders :
The patient will get status about the reports in their dashboard after submitting the samples. Agents will get a reminder about the bookings made by the patient.

Check their reports:
The patient can check their reports and status of report.

ANALYSIS MODELING:

Use Case Diagrams:

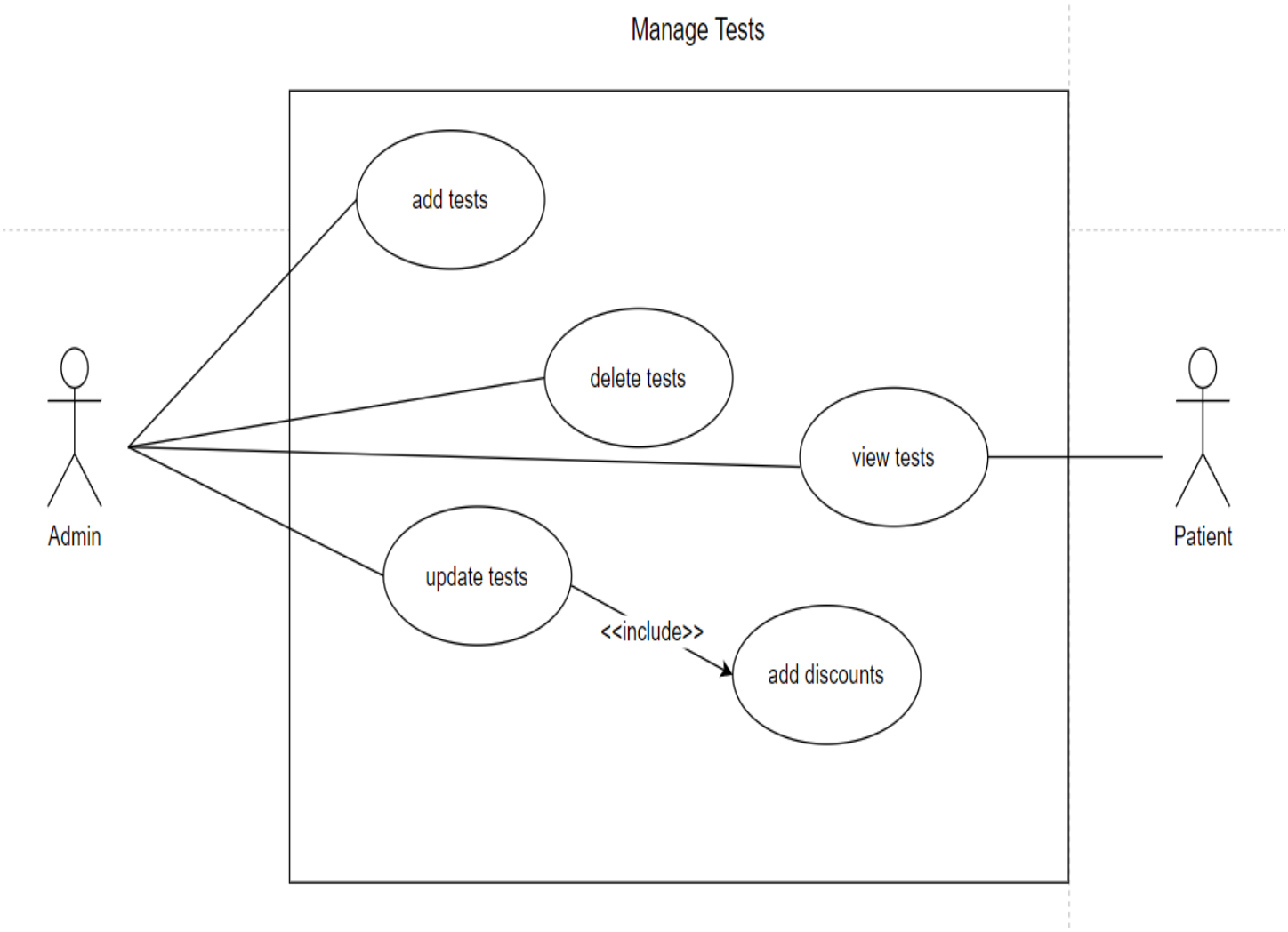


Fig 1: Use case diagram – Admin Manages Test

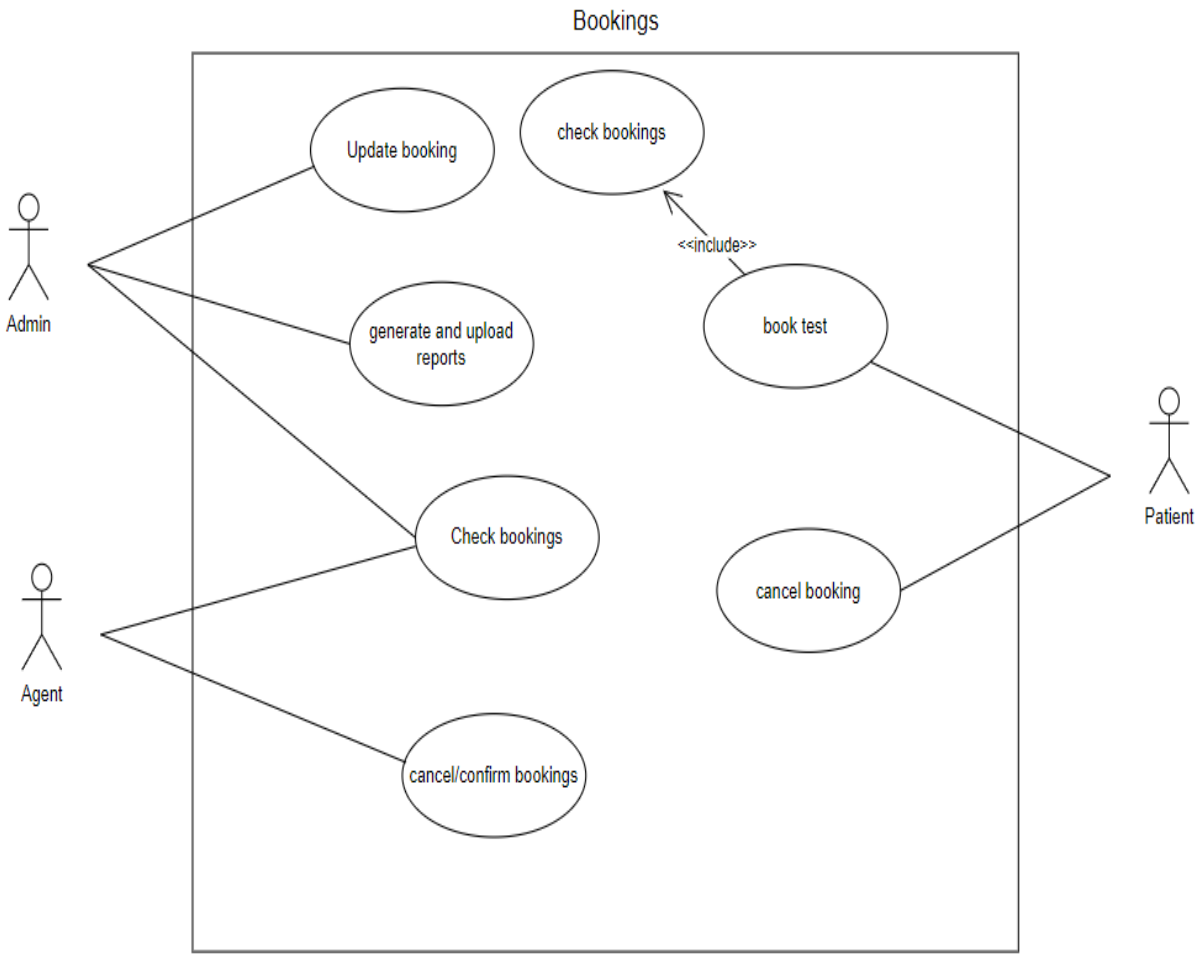


Fig 2: Use case diagram of Patient Booking Test .

Architectural Diagram:

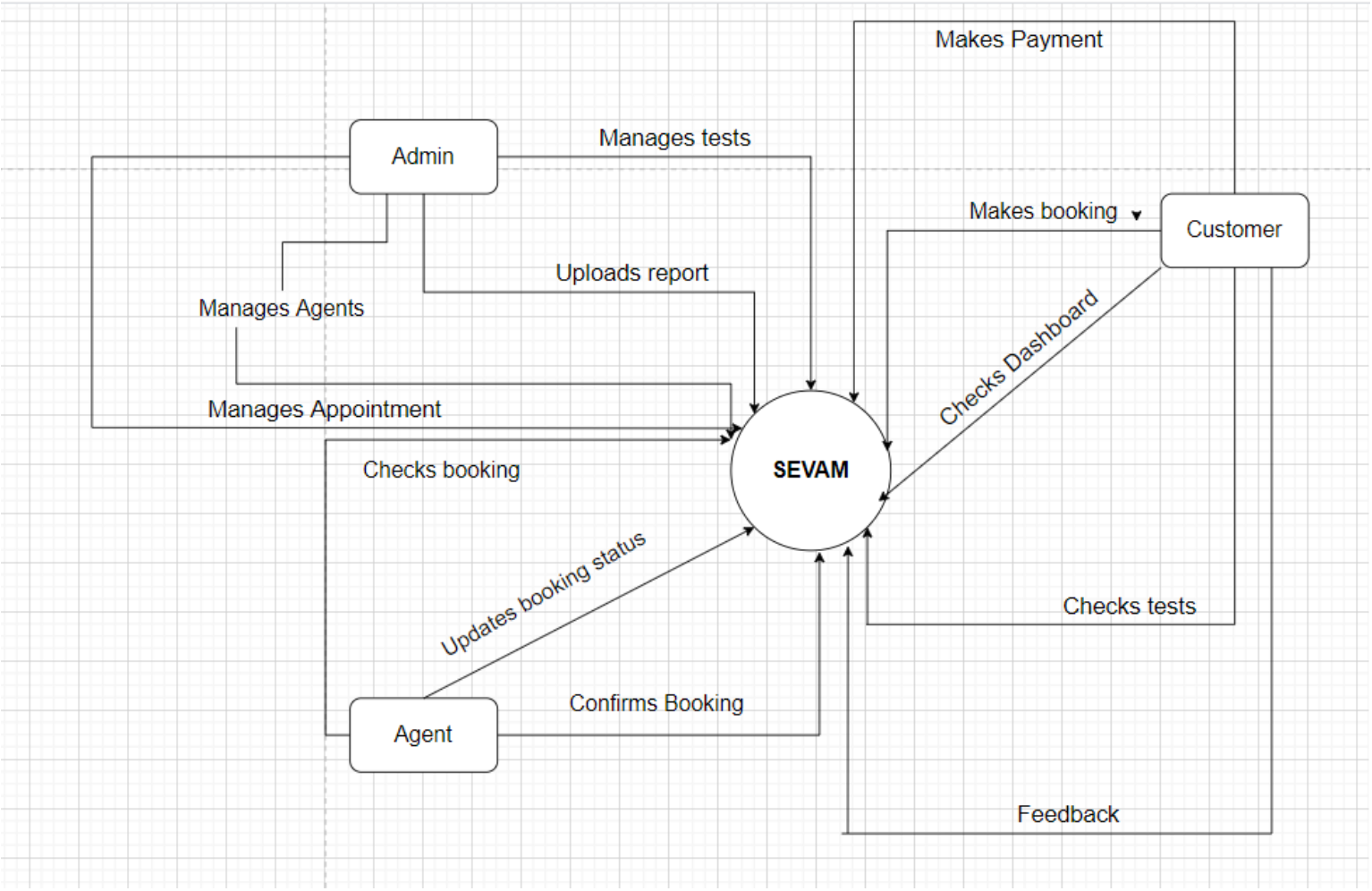


Fig3: Architectural Diagram

Entity Relationship Diagram

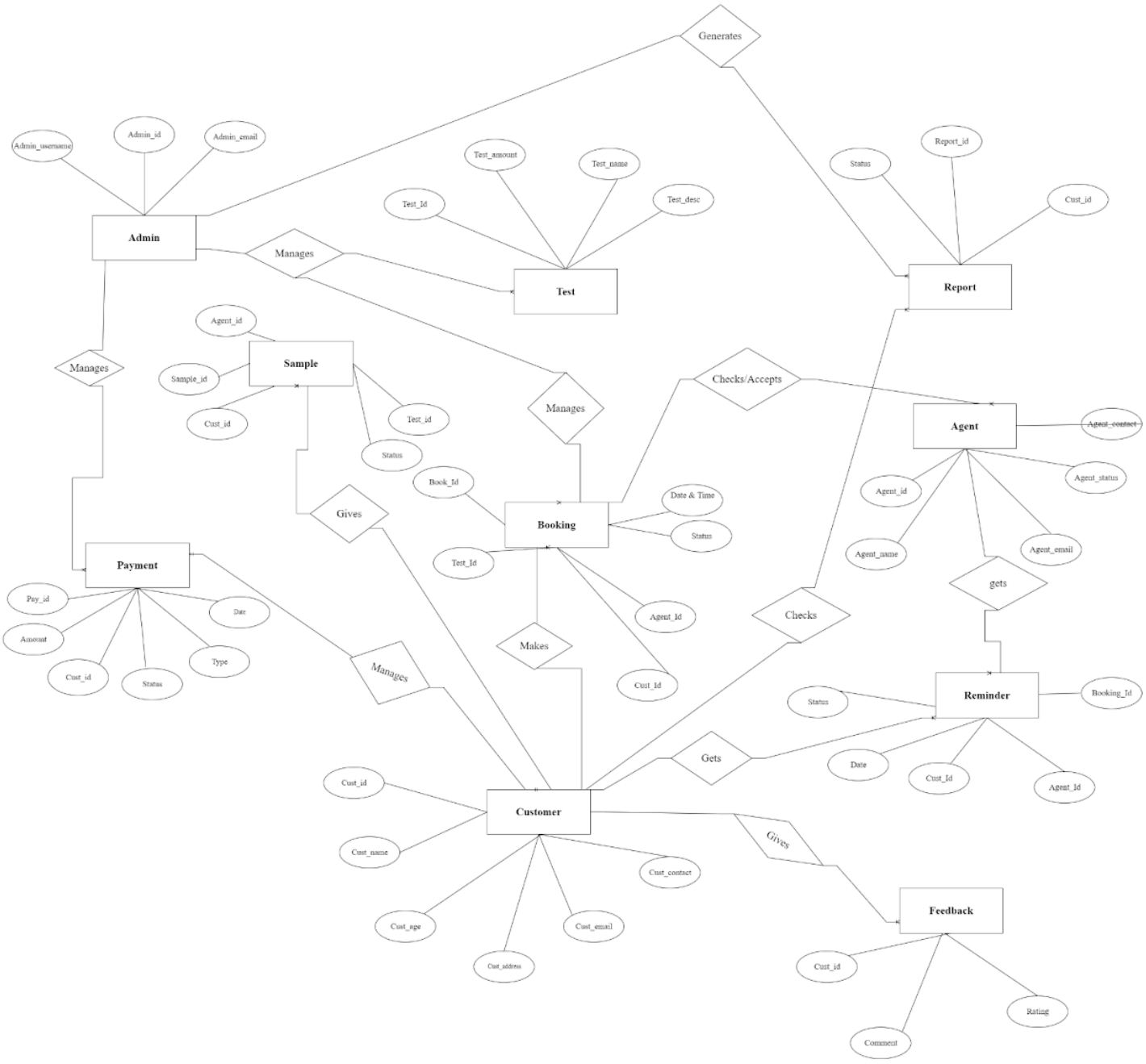


Fig4: Entity Relationship Diagram

Relationship Schema:

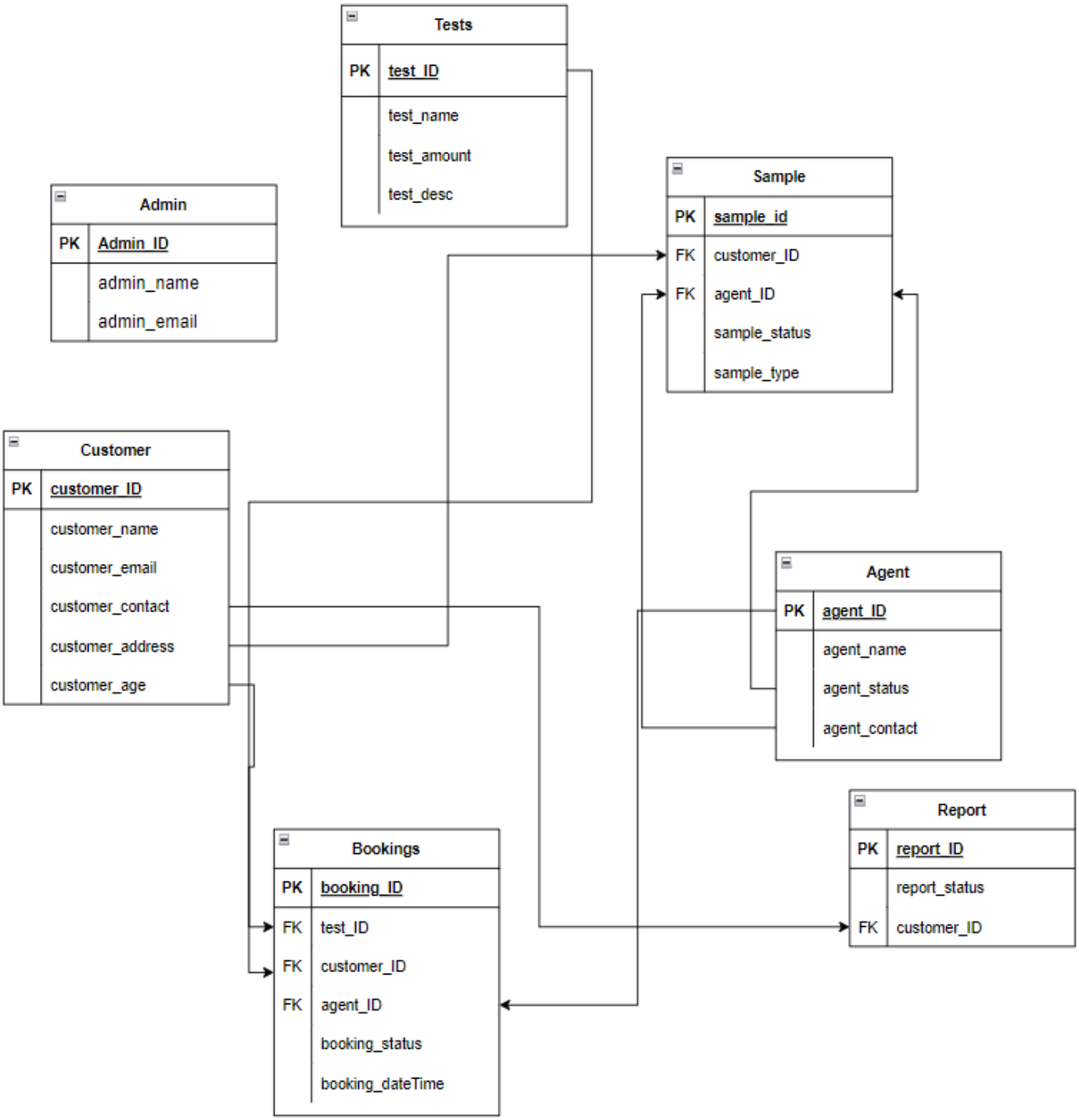


Table Description:

ADMINREG

Field_name	Data Type	Constraint	Description
name	varchar(20)	NOT NULL	name for admin
email	varchar(20)	Primary Key	Unique email of the admin
password	varchar(20)	UNIQUE	password of admin
gender	varchar(20)	NOT NULL	Gender of admin
age	Integer(4)	NOT NULL	Age of admin

USEREG

Field_name	Data_type	Constraint	Description
password	varchar(20)	NOT NULL	Password for customer
name	varchar(20)	Not Null	Name of the customer
age	integer(2)	Not Null	Age of the customer
gender	varchar(20)	Not Null	Gender of the customer
email	varchar(20)	Primary Key	Email of customer

AGENTREG

Field_name	Data_type	Constraint	Description
id	integer(4)	Primary Key	Unique ID for agent
name	varchar(20)	Not Null	Name of the agent
email	varchar(20)	Not Null	Email of the agent
phno	integer(15)	UNIQUE	Contact number of agent

APPOINTMENT

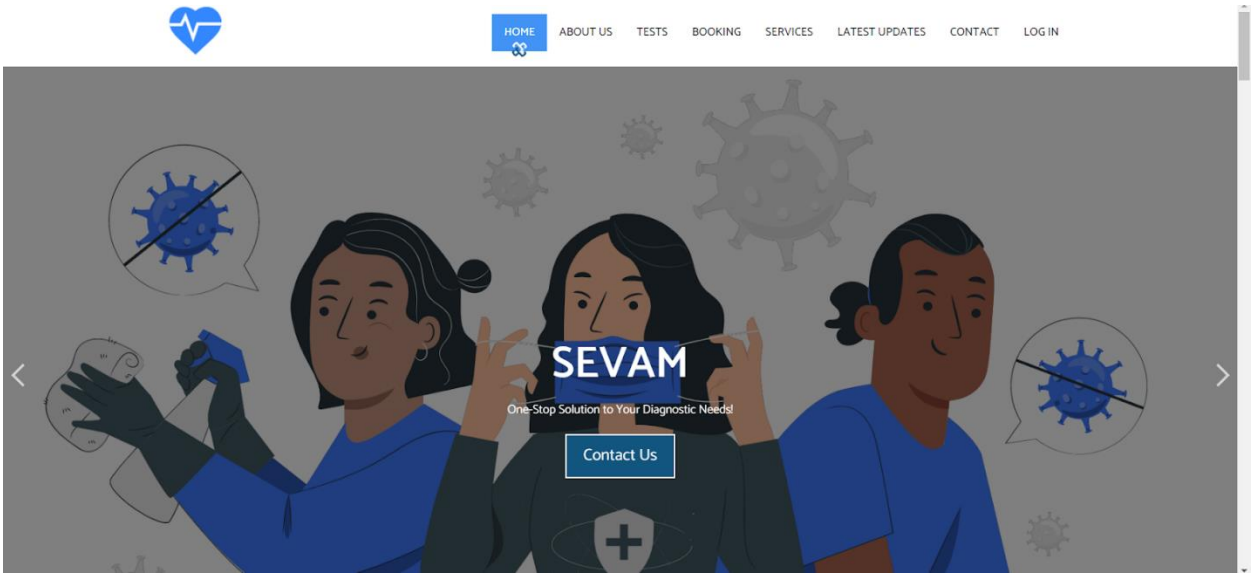
Field_name	Data_type	Constraint	Description
aptid	integer(4)	Primary Key	Unique id of each booking
name	varchar(20)	Foreign Key	Name of test
age	integer(4)	Foreign Key	Age of customer
phno	varchar(20)	Not Null	phno of agent
address	varchar(20)	Not Null	Address of agent
status	varchar(20)	Not Null	Available or unavailable
date	Date	Not Null	Date of appointment
time	time	Not Null	timing of appointment
agent	varchar(20)	Not Null	Update on report
ag_status	varchar(20)	Not Null	Submitted / collected
email	varchar(20)	Not Null	Email of customer
gender	varchar(20)	Not Null	Gender of customer
test	varchar(20)	Not Null	Test name

TESTDB

Field_name	Data_type	Constraint	Description
Test_Id	integer(4)	Primary Key	Unique test id of each test
Test_amount	integer(6)	Not Null	Amount of each test
Test_name	varchar(20)	UNIQUE	Unique name of each test
Test_desc	varchar(100)	Not Null	Description of each test

Interface Design:

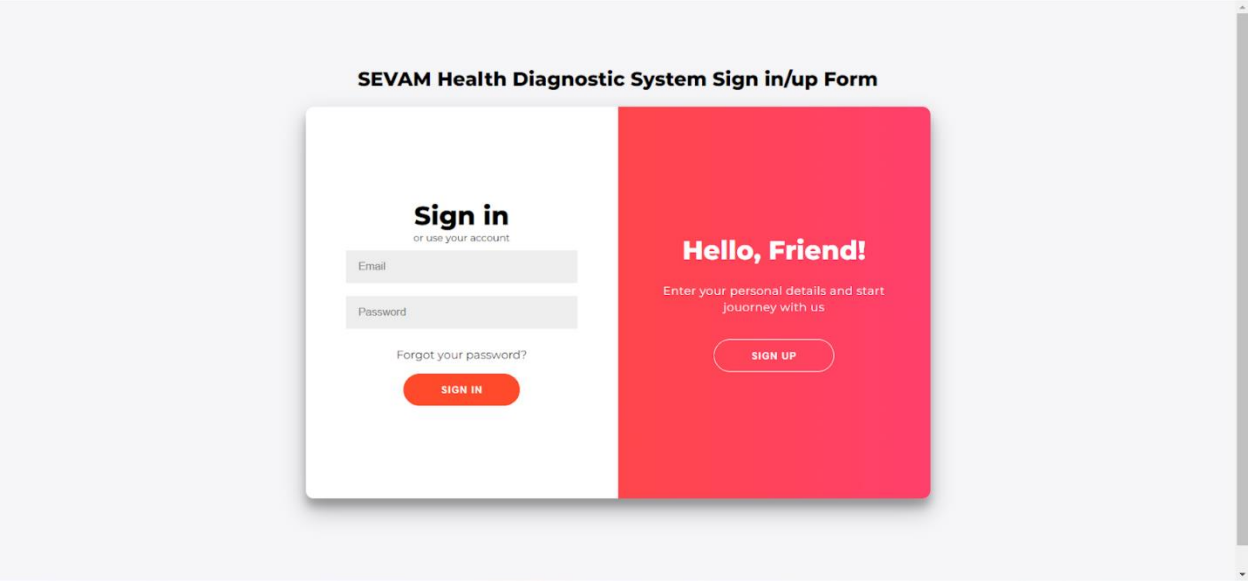
User Interface:



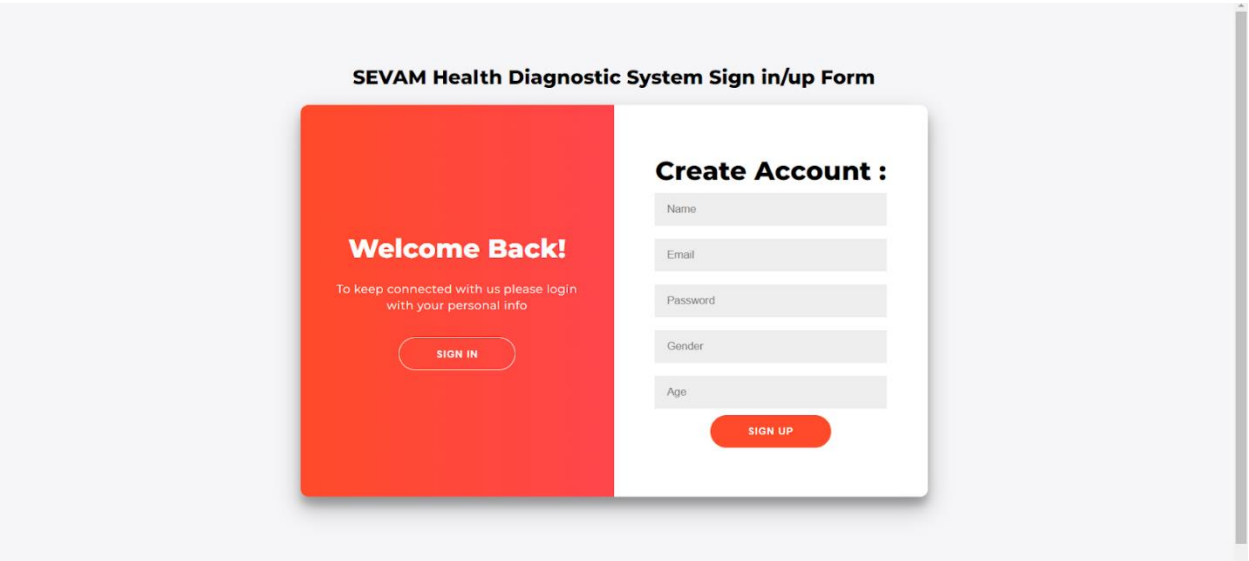
Common Login Page:

WELCOME TO SEVAM

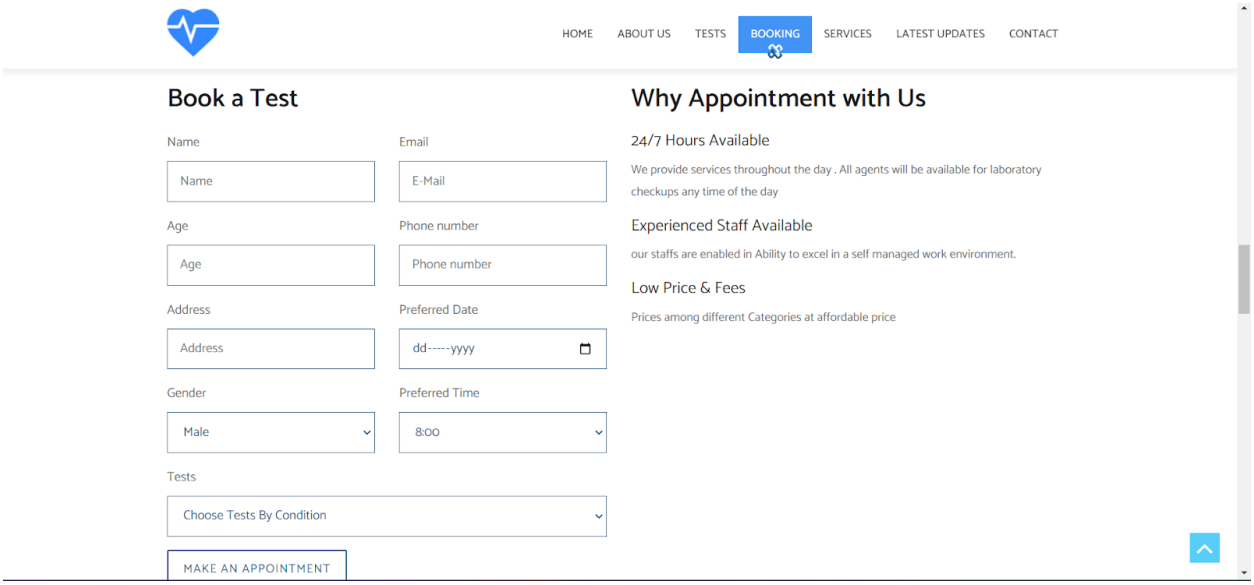
User Login Page:



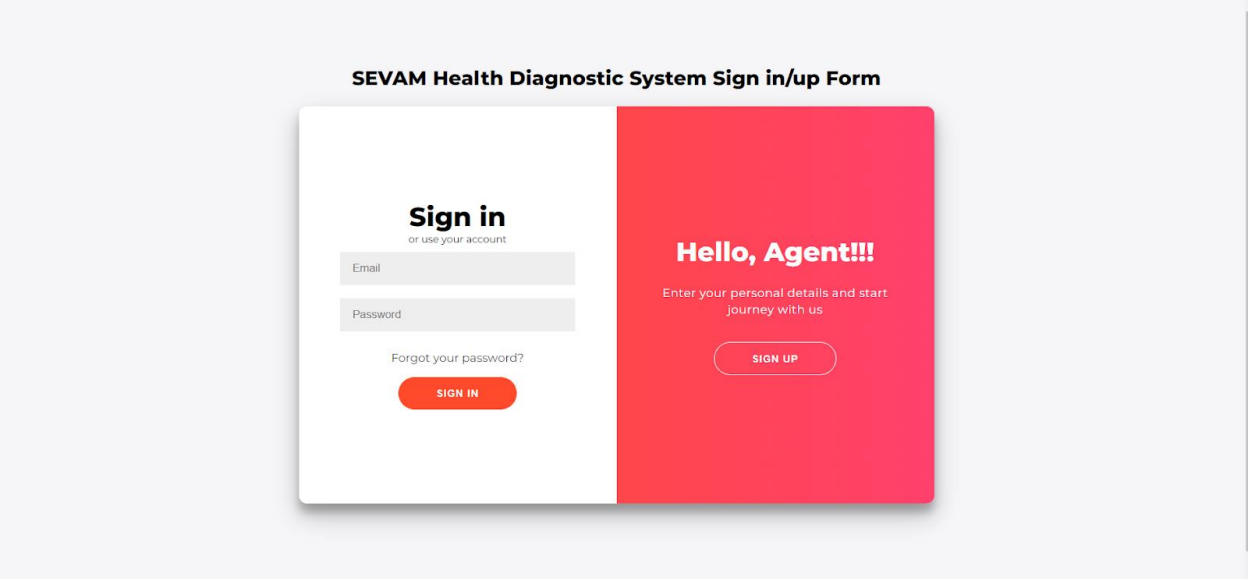
User registration Page:



User Booking Page:

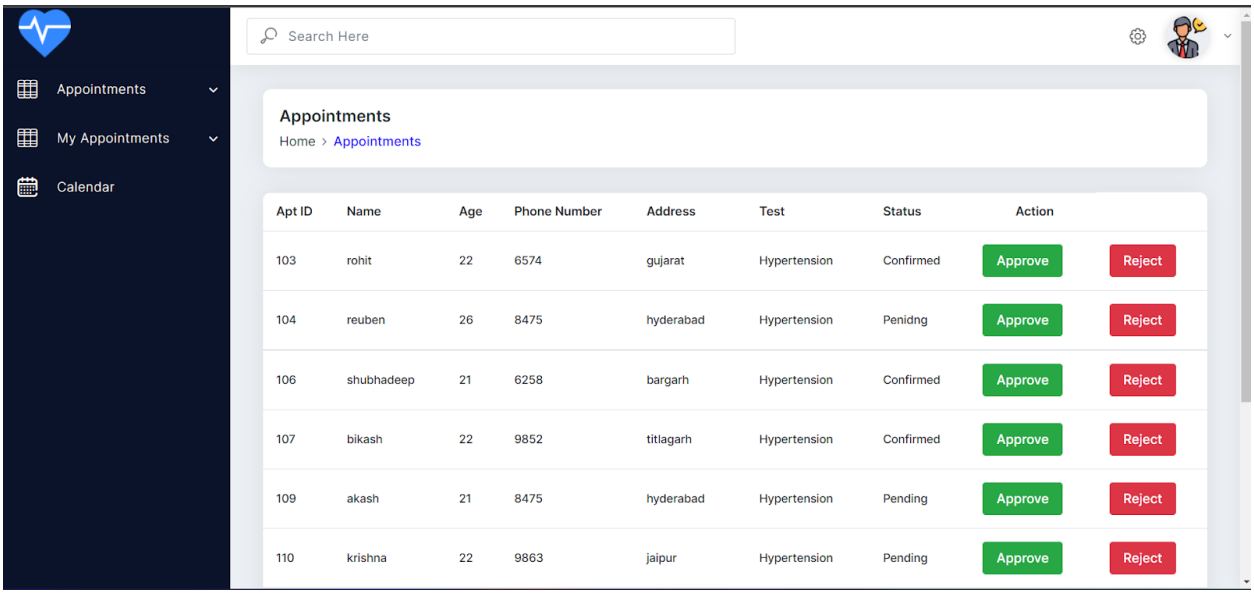


Agent Sign in Sign Up page:

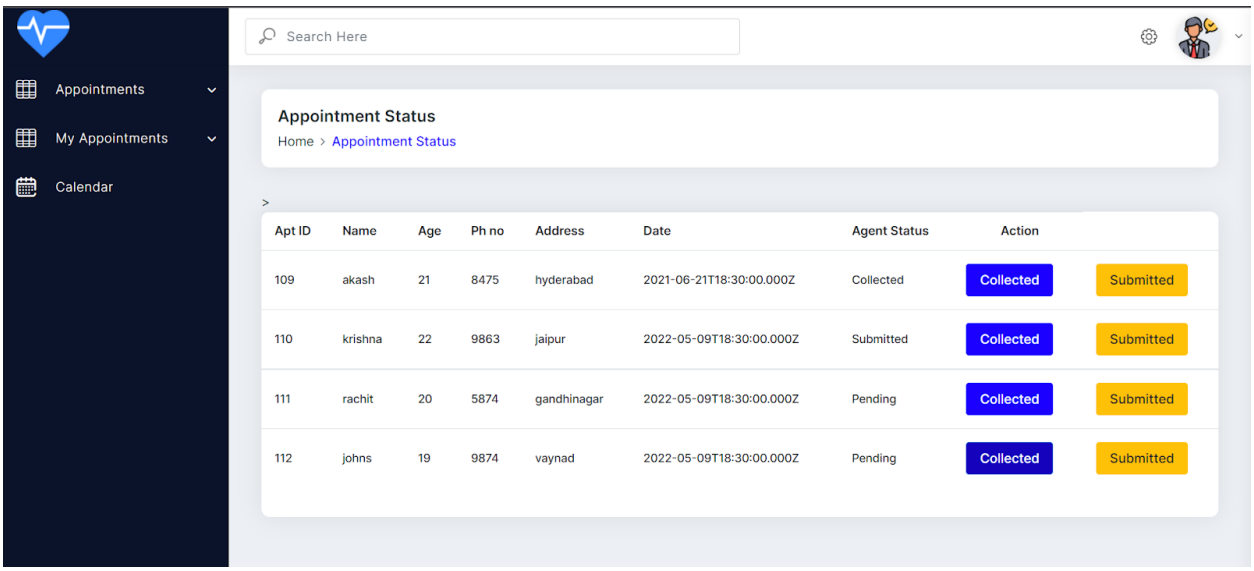


Agent Interface:

Appointments:

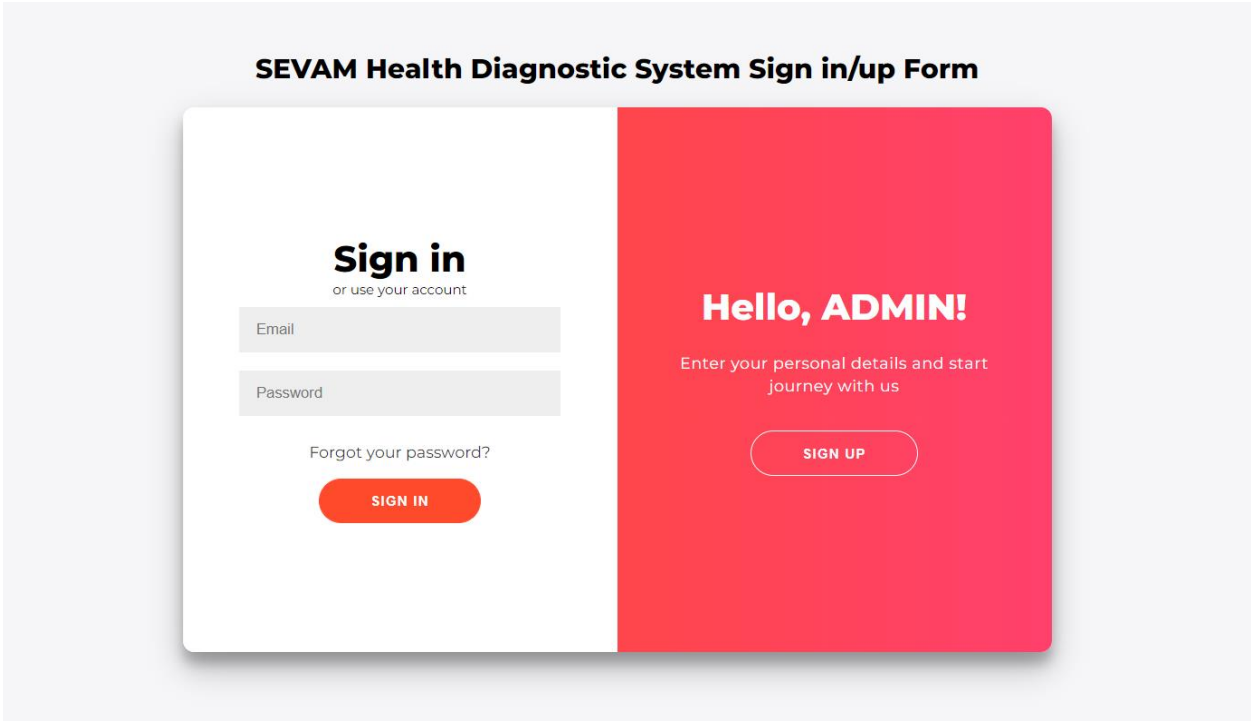


Booking Status:



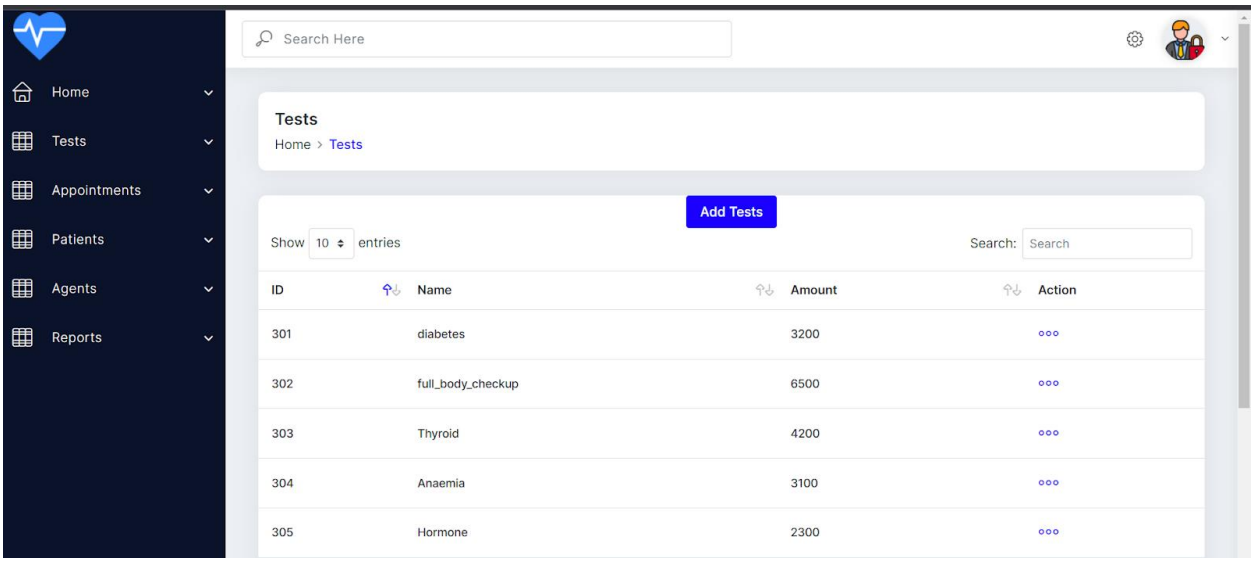
Logout:

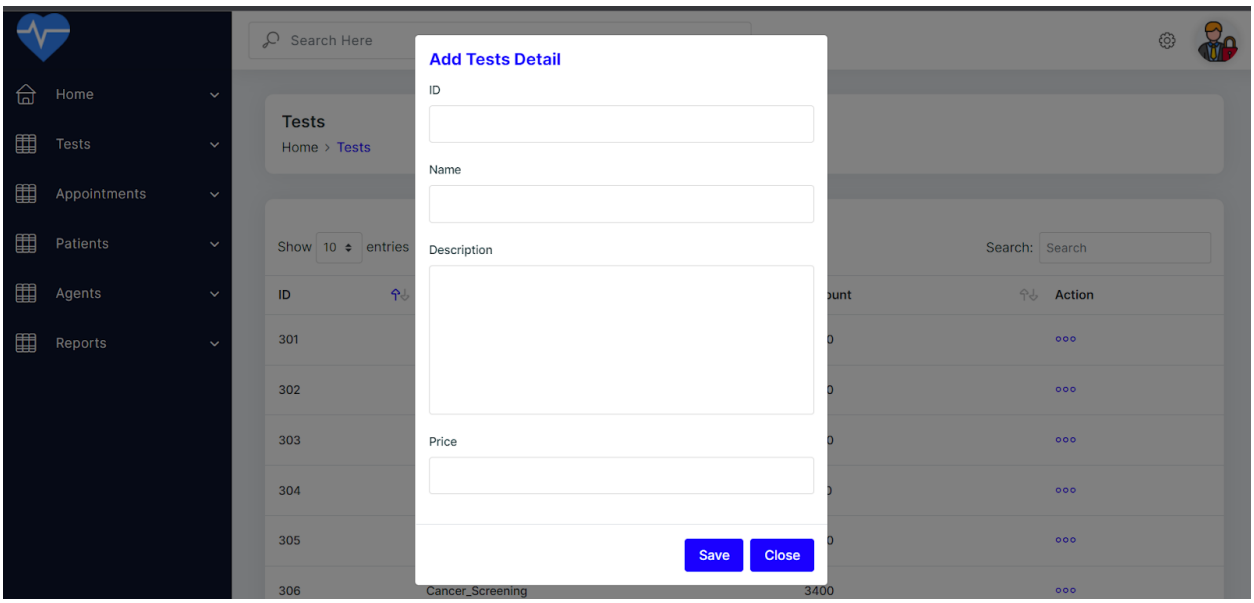
Admin Sign in Sign up page:



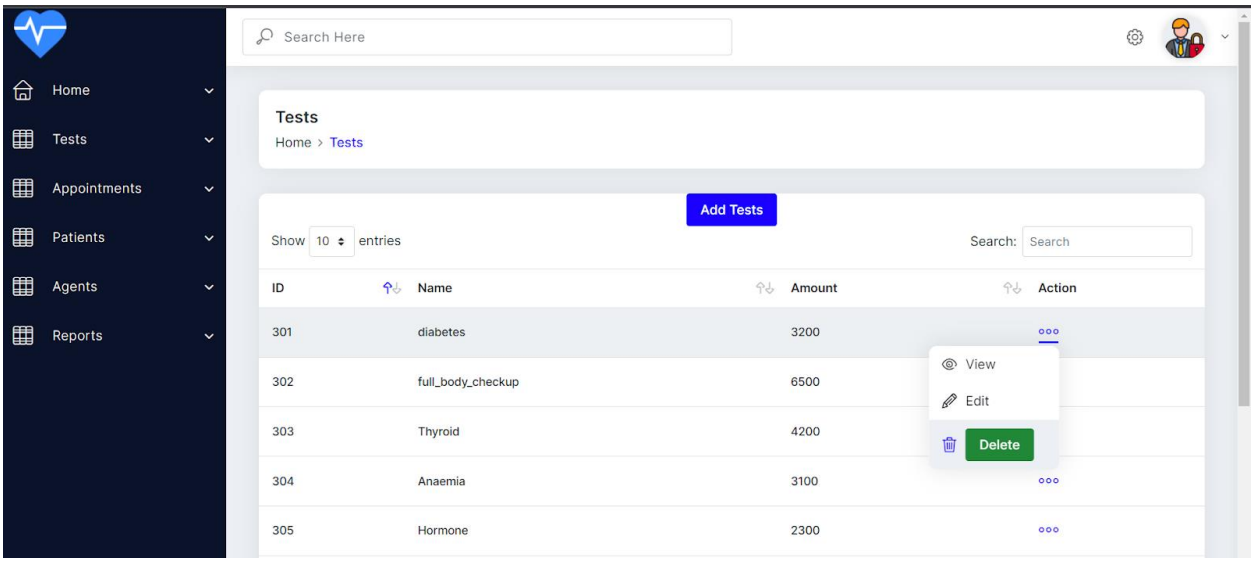
Admin Interface:

Add Test:

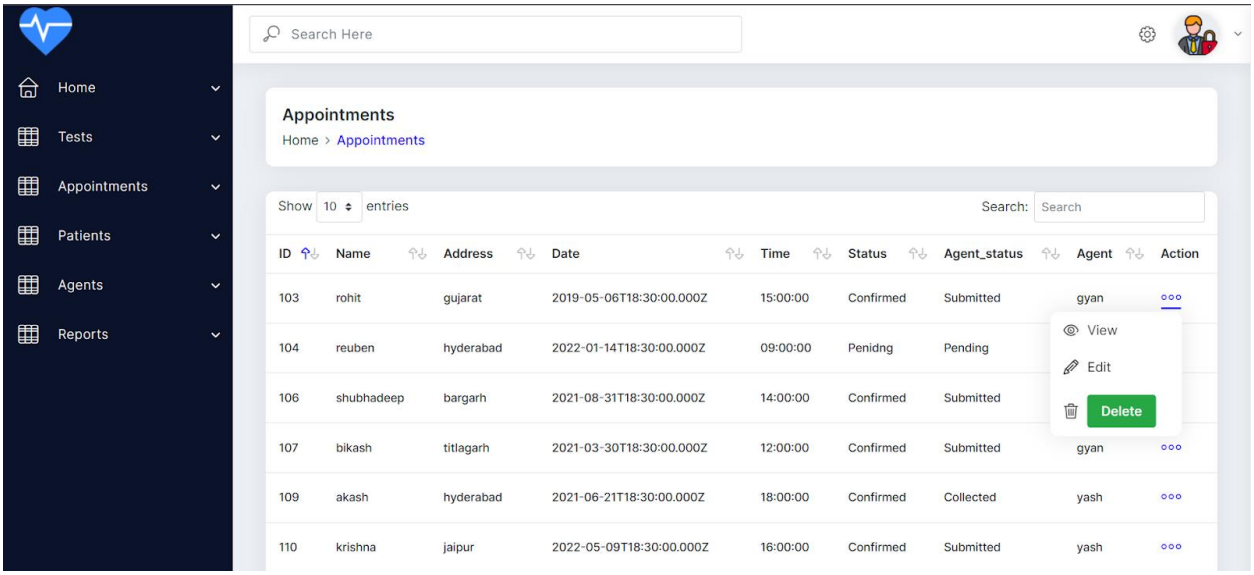




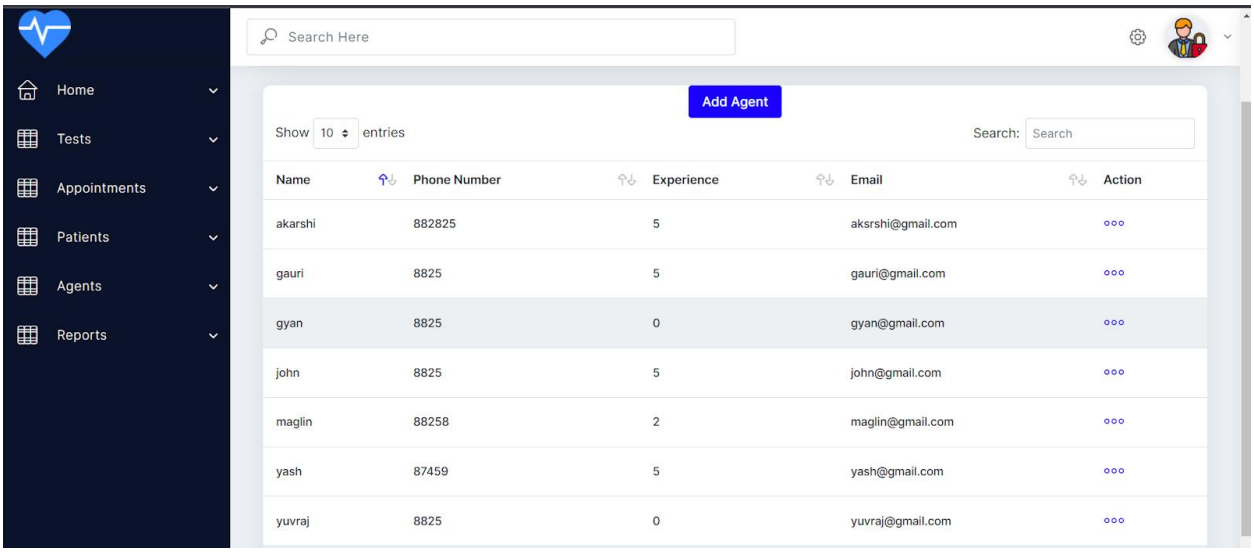
Delete Test:


Search Here
Home
Tests
Appointments
Patients
Agents
Reports
Tests
Home > Tests
Add Tests
Show 10 entries
Search: Search
ID
Name
Amount
Action
301 diabetes 3200
302 full_body_checkup 6500
303 Thyroid 4200
304 Anaemia 3100
305 Hormone 2300
View
Edit
Delete

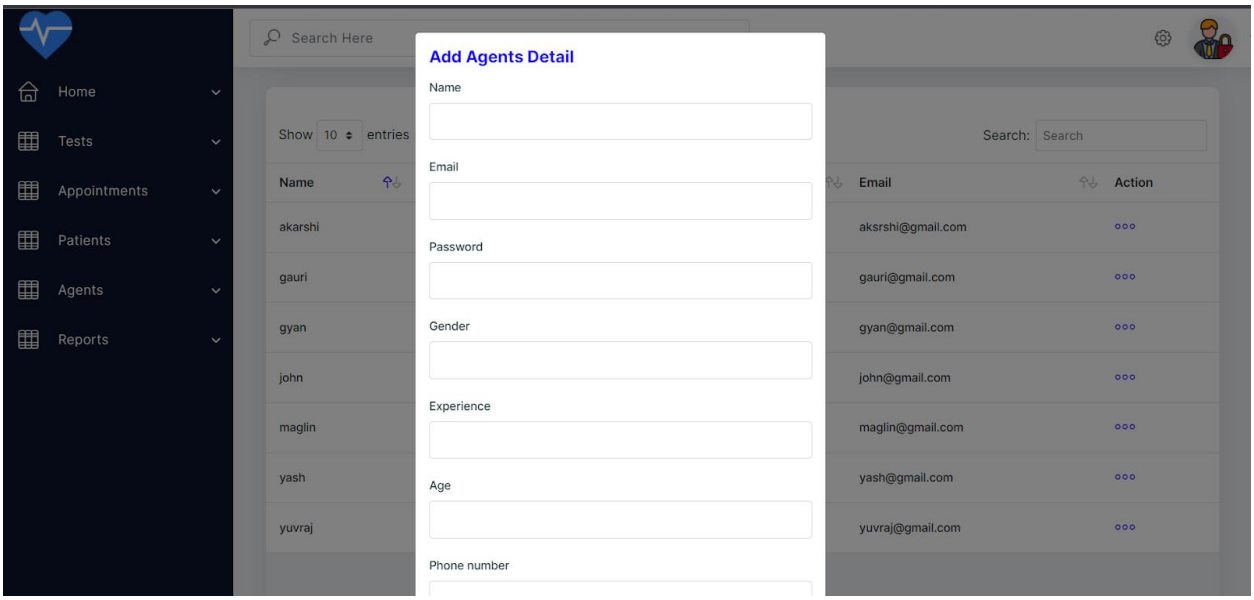
Delete Patient Entry:



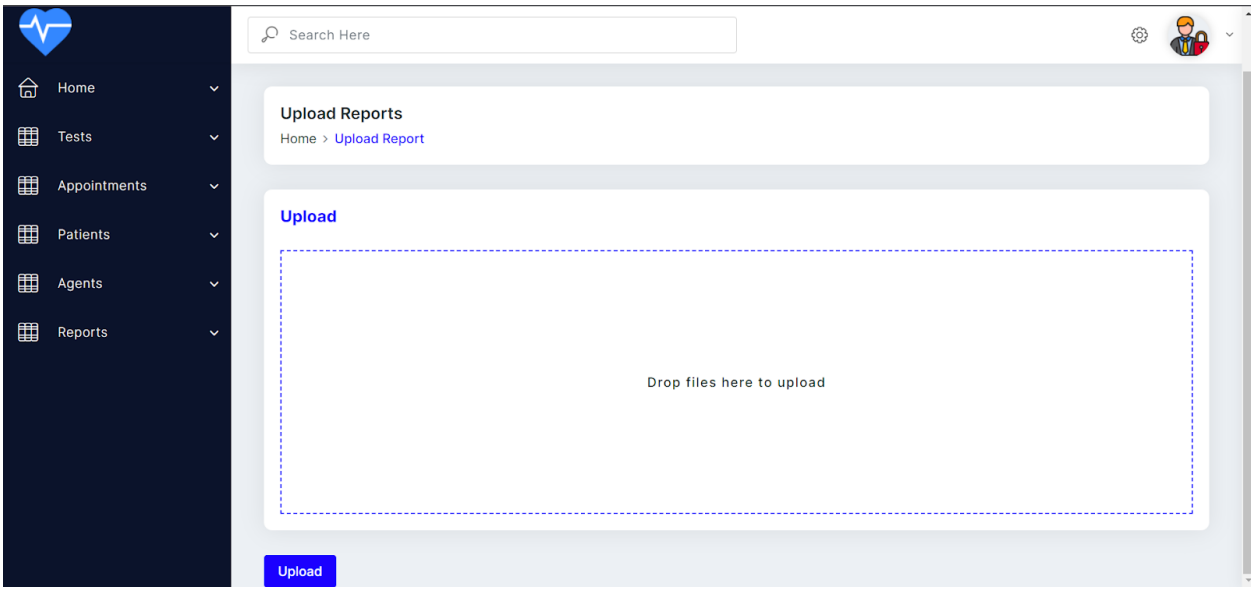
Add Agent:



Add Agent Details:



Upload Report:



Software and Hardware Environments:

Software Environment:

mySQL, Express JS, Angular JS, Node JS, Cookie-parser, is used for the full-stack development.

mySQL is a relational database system
Express JS is a framework used to build web applications in NodeAngularJS is a JavaScript framework developed by Google
Node.js is a server-side JavaScript execution environment
cookie-parser is a middleware which parses cookies attached to the client request object.

MySQL is often regarded as the safest and most dependable database management technology for popular online applications. It has unrivaled scalability, allowing it to manage deeply embedded apps with a lower footprint,even in enormous data warehouses with terabytes of data. MySQL has its own storage-engine structure, which makes it easier for system administratorsto set up the MySQL database server for optimal performance.
Express JS develops Node.js web applications quickly and easily.
Configuration and customization are simple. It allows you to define applicationroutes using HTTP methods and URLs. MySQL databases are simple to connect to with express JS.
Angular JS is easy to learn for people with knowledge of HTML, JavaScriptand CSS. It has a two way binding feature. The view and the model can be synchronized immediately with AngularJS.
Node JS offers easy scalability and it is easy to learn as well. It is used as a Single Programming Language for writing server-side applications as well asfront-end in JavaScript using a runtime environment.
Cookie Parser Cookies are simple, small files/data that are sent to client with a server request and stored on the client side. Every time the user loads the website back, this cookie is sent with the request. This helps in keep track of the user's actions.

Hardware Environment:

OS name	Microsoft windows 11 Single Language
Version	11.0.19043
RAM	8GB
Disk Storage	1 TB HDD and 256 SSD

System Type	X64 Based PC
Processor	i5 8th Gen

Testing

Test Cases:

Test ID	Test Name	Description	Req ID	Procedure	Expected Output
1	To verify whether the site logged in.	Check the website script is for localhost.	1	1. Login to the machine 2. Login into the website	1. login should be successful 2. Should be able to see the landing page

- **User Registration**

Test ID	Test Name	Description	Req ID	Procedure	Expected Output
2.1	To register successfully.	Customers successfully register.	2	1) Customer loads sevam. 2) Customers fill in all the details in the sign up form and click the submit button.	1) Customers are successfully registered.
2.2	Unsuccessful register.	Customer fails to register.	2	1) Customer loads sevam. 2)Customers don't fill in all the details and click the submit button.	1) Error can be seen to fill all the details. 2)Customers don't get registered.

- **Login**

Test ID	Test Name	Description	Req ID	Procedure	Expected Output
3.1	Check the response when a valid username,email and password is entered.	To check the response when a user enters valid user-details(username,email,password) for login.	3	1. Open user login page. 2. Enter valid username,email and password.	Login successful. Users should be able to see the front page.

3.2	Check the response when a invalid username,email or password is entered.	To check the response when a user enters invalid user-details(username,email,password) for login.	3	1. Open user login page. 2. Enter invalid username,email and password.	The user will not be able to login.

- **Forgot Password**

Test ID	Test Name	Description	Req ID	Procedure	Expected Output
4.1	Check the "Forgot Password" functionality.	To check if the "Forgot password" link redirects the user to the user verification page.	4	1. Open user login page. 2. Click on the "Forgot Password" link.	The user will be redirected to the user verification page before the user can reset the password.
4.2	Verify the email entered is registered or not.	To check only registered users updates password	4	1. Enters wrong email	The user will not be directed to the next page.
4.3	Check the response of the user verification page if the user enters a registered email and phone number.	Check the response of the user verification page if the user enters a registered email and phone number.	4	1. Click on the "Forgot Password" link. 2. Enter valid email and phone number.	A message will be displayed if the user authentication is successful and the user will receive a mail which contains a link to login and a link to reset the password.
4.4	Check the response of the user verification page if the user enters an unregistered email or phone number.	Check the response of the user verification page if the user enters an unregistered email or phone number.	4	1. Click on the "Forgot Password" link. 2. Enter invalid email and phone number.	An error message will be displayed if the user authentication wasn't successful and it'll ask the user to enter a valid email or phone number.

- **Reset Password**

Test ID	Test Name	Description	Req ID	Procedure	Expected Output
5.1	Check the functionality of password reset link.	To check the functionality of the link to reset the password.	5	1. Click on the link provided in the email.	The user will be redirected to a “Reset Password” page.
5.2	Check the response if the user has entered a valid password and provided the same for the confirm password field to reset password.	To check the response if the user has entered a valid password following all the conditions provided for a password to be valid and provided the same for confirming password.	5	1. Enter valid password. 2. Fill the confirm password field with the same.	Users will be notified if the password reset was successful.
5.3	Check the response if the user has entered an invalid password and provided the same for the confirm password field to reset password.	To check the response if the user has entered a valid password following all the conditions provided for a password to be valid and provided the same for confirming password.	5	1. Enter invalid password. 2. Fill the confirm password field with the same.	All the conditions for a valid password will be displayed to the user.
5.4	Check the response if the passwords in the reset password field and confirm password field don't match.	To check the response if the passwords in the reset password field and confirm password field don't match.	5	1. Enter valid password. 2. Fill the confirm password field with a different password.	An error message “Passwords do not match” will be displayed below the “Confirm Password” field.

- **Bookings**

Test ID	Test Name	Description	Req ID	Procedure	Expected Output
6.1	To verify that the user is able to book a test.	User books an appointment successfully.	6	1) User logins in. 2)User finds a test. 3)User selects the date and time and books the appointment.	1) User successfully books an appointment. 2) User successfully pays the amount.
6.2	To verify that the user was unable to book a test.	User couldn't book an appointment.	6	1) User logins in. 2)User finds a test. 3)User selects dates and time and proceeds further. 4) While paying, the transaction fails.	1) Message will be visible that payment was unsuccessful. 2) User wasn't able to book the appointment.
6.3	To verify that the user was unable to book a test.	User couldn't book an appointment.	6	1) User logins in. 2)User finds a product. 3)User doesn't select dates and time and clicks next.	1) Error will be visible to select date and time. 2)User couldn't book the appointment.

- **Logout**

Test ID	Test Name	Description	Req ID	Procedure	Expected Output
7.1	To verify that the user successfully logs out.	User was successfully logged out.	7	1) User logins in. 2)User clicks the logout button .	1)User was successfully logged out from the site.
7.2	To verify that the user was unable to log out.	User wasn't successfully logged out.	7	1)User doesn't login in.	1.User was not able to logout.

7.3	To verify that the user was unable to log out.	User wasn't successfully logged out.	7	1)User logins in. 2)User doesn't click the logout button.	1.User was not able to logout.

- **Admin**

Test ID	Test Name	Description	Req ID	Procedure	Expected Output
8.1	Check if the Admin is able to add tests	Admin adds test in admin dashboard	8	1)Admin logs in 2)Admin Clicks dd test	1.Admin successfully Adds test .
8.1.1	Check if the Admin is unable to add tests	Admin cannot add test in admin dashboard	8	1)Admin logs in 2)Admin Clicks dd test	1.Admin cannot Add test .
8.2	Check if Admin is able to add test details	Admin adds test details in dashboard	8	1) Admin logs in 2)Admin clicks add test 3)Admin Adds Test Details	1.test details get added and are reflected in the admin dashboard.
8.2.1	Check if Admin is unable to add test details	Admin cannot add test details in dashboard	8	1) Admin logs in 2)Admin clicks add test 3)Admin Adds Test Details	1.test details do not get added and are not reflected in the admin dashboard.
8.3	Check if Admin is able to delete test	Admin deletes test details in dashboard	8	1) Admin logs in 2)Admin deletes test .	1.Admin successfully deletes test .
8.3.1	Check if Admin is unable to delete test	Admin deletes test details in dashboard	8	1) Admin logs in 2)Admin deletes test .	1.Admin cannot delete test .
8.4	Check if Admin is able to delete patient data	Admin Deletes patient data	8	1) Admin logs in 2)Admin deletes patient data by clicking the delete button.	Admin Successfully deletes patient data.
8.4.1	Check if Admin is unable to delete patient data	Admin Deletes patient data	8	1) Admin logs in 2)Admin deletes patient data by clicking the delete button.	Admin cannot delete patient data.
8.5	Check if Admin is able to add agent	Admin adds agent	8	1)Admin logs in	Admin Successfully adds an agent.

				2)Admin Adds agent through dashboard	
8.5.1	Check if Admin is unable to add agent	Admin adds agent	8	1)Admin logs in 2)Admin Adds agent through dashboard	Admin cannot add an agent.
8.6	Check if Admin is able to delete agent data	Admin deletes agent data	8	1) Admin logs in 2)Admin deletes data by clicking the delete button.	Admin Successfully deletes an agent data.
8.6.1	Check if Admin is unable to delete agent data	Admin deletes agent data	8	1) Admin logs in 2)Admin deletes data by clicking the delete button.	Admin cannot delete an agent data.
8.7	Check if admin is able to upload report	Admin uploads report	8	1)Admin logs in 2)Admin clicks upload button, selects files.	Admin Adds report via upload
8.7.1	Check if admin is unable to upload report	Admin uploads report	8	1)Admin logs in 2)Admin clicks upload button, selects files.	Admin Fails to add report via upload

- **Agent**

Test ID	Test Name	Description	Req ID	Procedure	Expected Output
9	To check if agent can login	Agent logs in	9	1)Agent enters credentials 2)Agent clicks login	Agent logs in successfully
9.1	To check if agent cant login	Agent cannot login	9	1)Agent enters credentials 2)Agent clicks login	Agent cannot login
9.2	To check if agent can logout	Agent can logout	9	Agent clicks logout	Agent log outs successfully
9.2.1	To check if agent cannot logout	Agent cannot log out	9	Agent doesn't log out	Agent doesn't logout.
9.3	To Verify if agent can check appointments	Agents checks appointments	9	1)Agent logs in 2)Agent checks appointments in	Agent sees appointment list.

				dashboard	
9.3.1	To Verify if agent cannot check appointments	Agents checks appointments	9	1)Agent logs in 2)Agent checks appointments in dashboard	Agent cannot see appointment list.
9.4	To verify if agent can access Booking status	Agent access booking status.	9	1)Agent logs in 2)Agents checks Booking status	Agent updates status
9.4.1	To verify if agent cannot access Booking status	Agent access booking status.	9	1)Agent logs in 2)Agents checks Booking status	Agent updates status

Module Testing and Integration:

The different modules in this project were developed separately and later integratedinto the main project. We used the Git Version Control System to track the changesin the source code. Each module was developed as parallel branches which were later merged with the main branch. While integrating some of the existing features of the module did not show theexpected results, which were later resolved.

Test Execution

a. Cycle 1

Test Case ID	Test Case Name	Test Case Description	Requirement ID	Execution Steps	Expected Output	Result
3.1	User Authenticatio n	Checks whether theuser is valid.	3	1.Check the validity of theuser id and password.	Successful login and redirection tohome page.	Success
7.1	User Logout	This should destroy thesession cookie while logging out.	7	1.Clicking onthe logout button	Successfully logsout the user and destroys all the session cookies.	Failure

4.1	Check the "Forgot Password" functionality.	To check if the "Forgot password" link redirects the user to the user verification page.	4	1. Open user login page. Click on the "Forgot Password" link.	The user will be redirected to the user verification page before the user can reset the password	Successs
4.2	Verify the email entered is registered or not.	To check only registered users updates password	4	1. Enters wrong email	The user will not be directed to the next page.	Success
5.2	Check the functionality of password reset link.	To check the functionality of the link to reset the password.	5	Click on the link provided in the email.	The user will be redirected to a "Reset Password" page.	Success
7.1	To verify that the user successfully logs out.	User was successfully logged out.	7	1) User logins in. 2)User clicks the logout button .	1)User was successfully logged out from the site.	Success

b. Cycle 2

Test Case ID	**Test Case Name**	**Test Case Description**	**Requirement ID**	**Execution Steps**	**Expected Output**	**Result**
7.1	User Logout	This should destroy thesession cookie while logging out.	7	1.Clicking onthe logout button	Successfully logsout the user.	Succcess

Defects:

Some of the defects of this project are as follows:

- Payment method is not specified.
- At times, CSS gets slow loaded on Button click.

Screenshots:

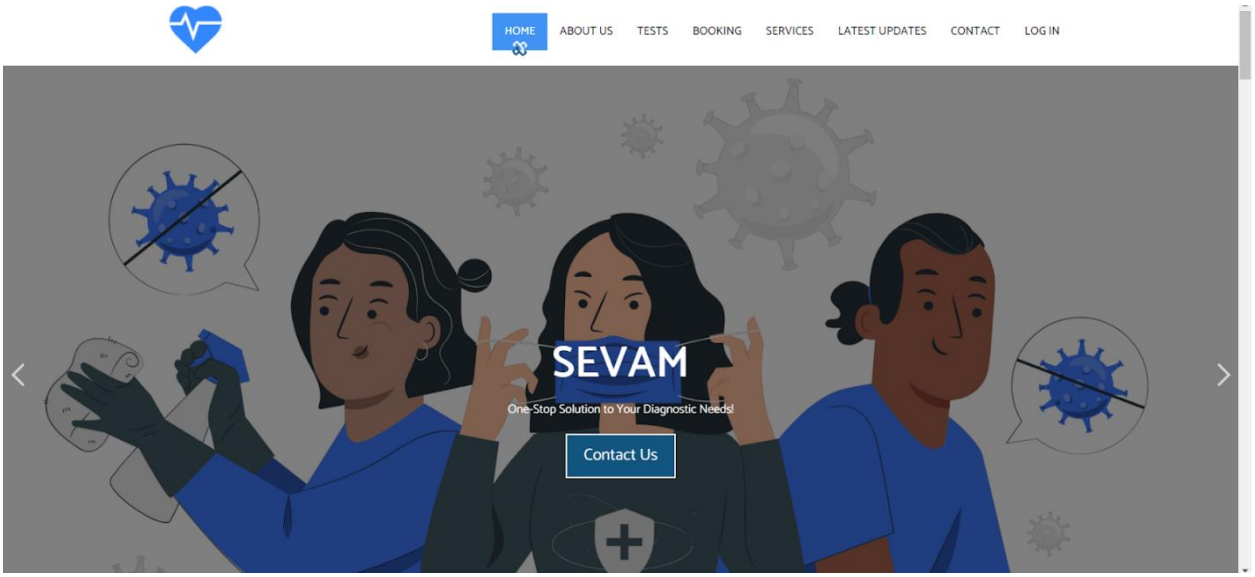
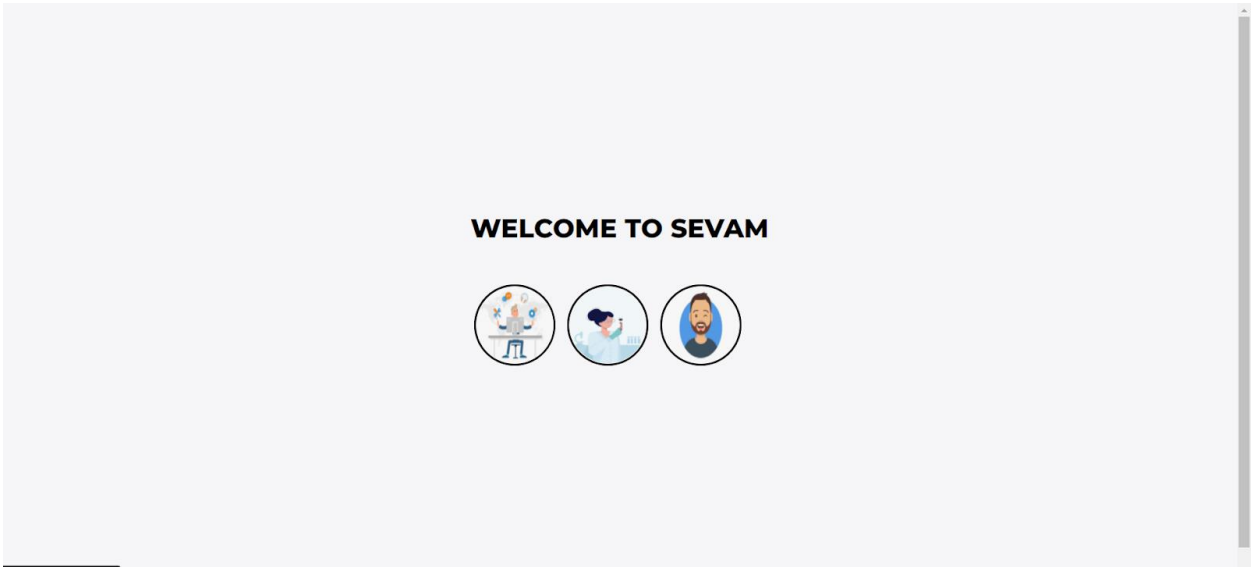
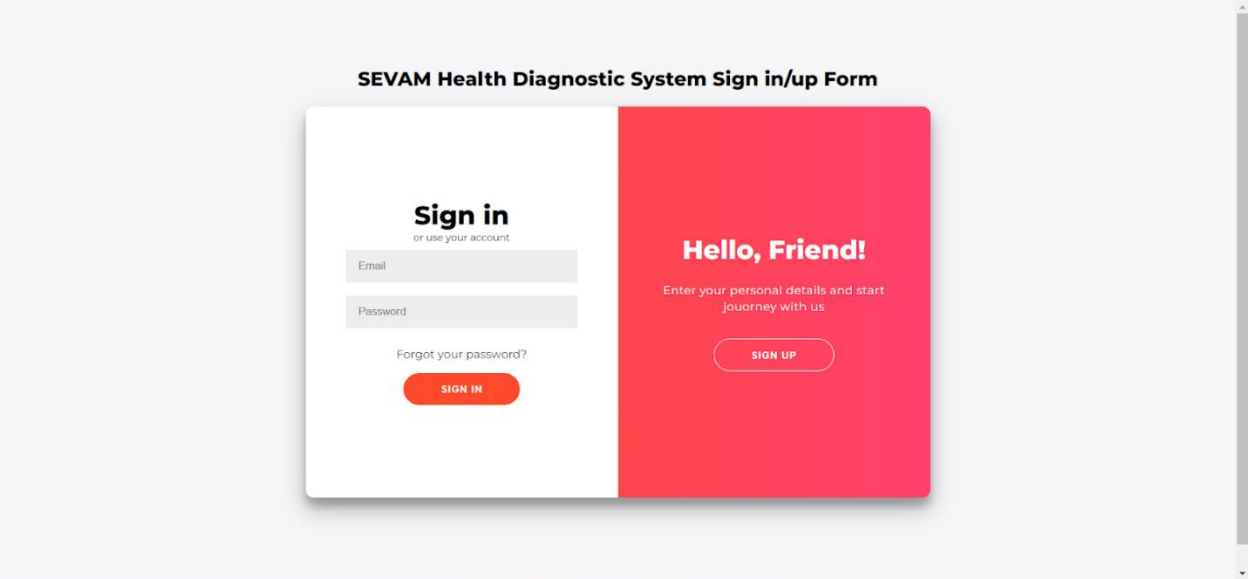


Fig: Main Page

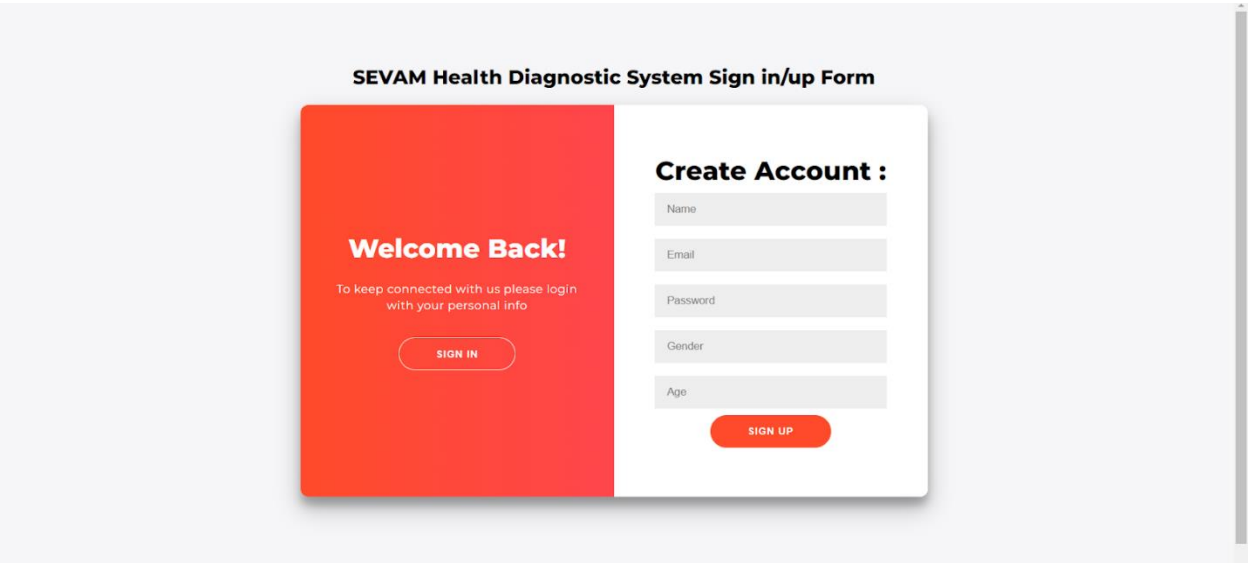
Common Login Page:



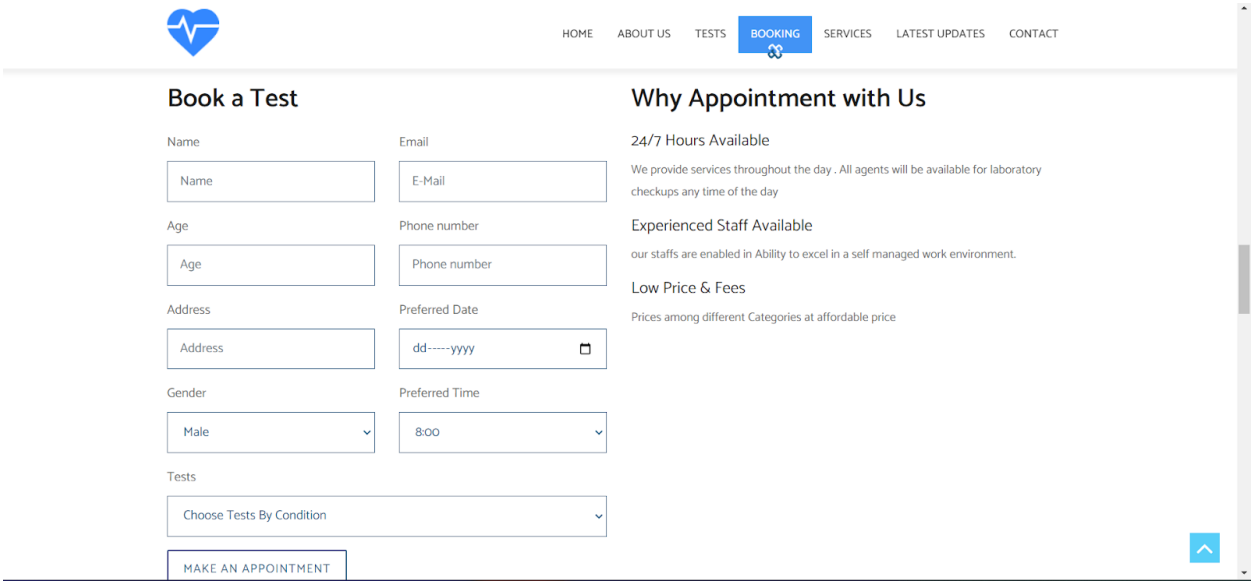
User Login Page:



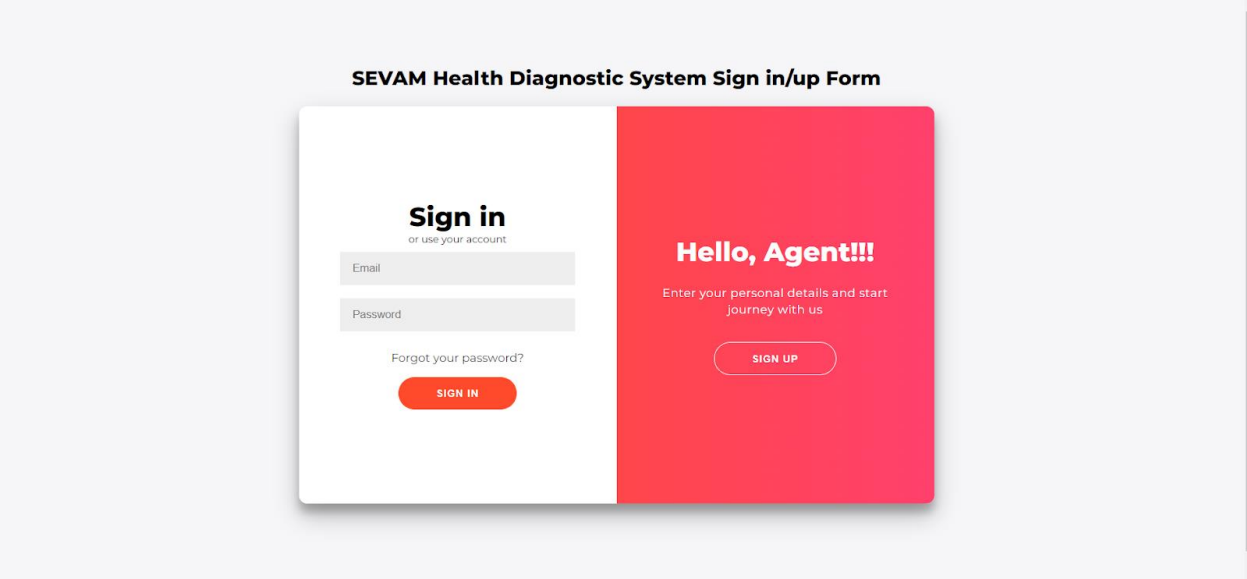
User registration Page:



User Booking Page:

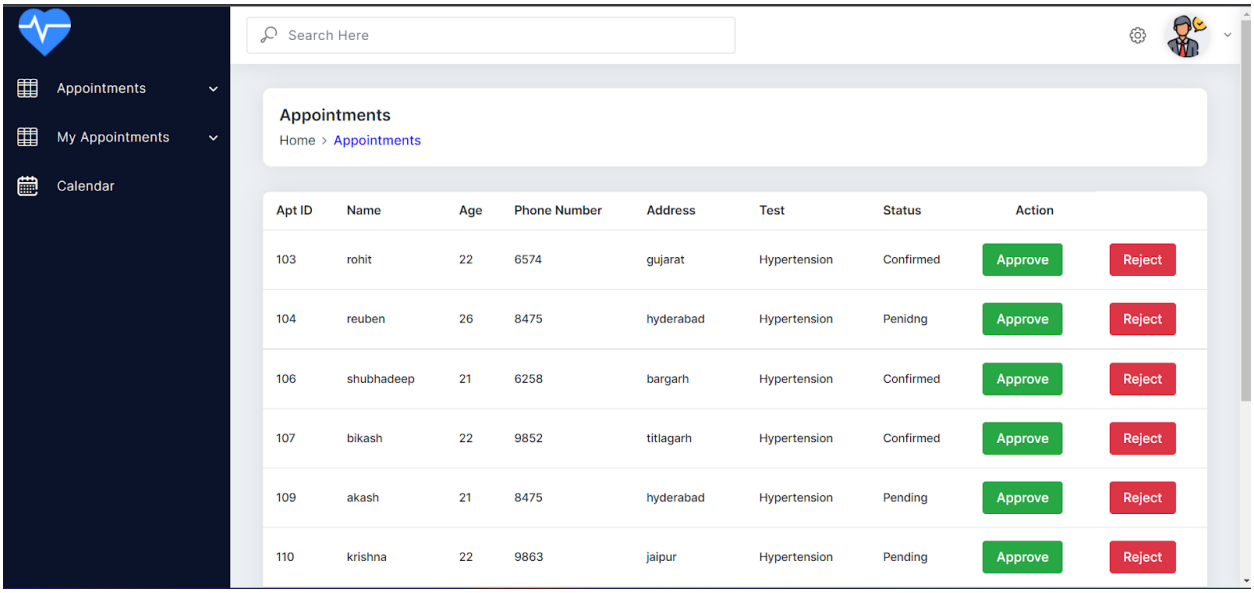


Agent Sign in Sign Up page:

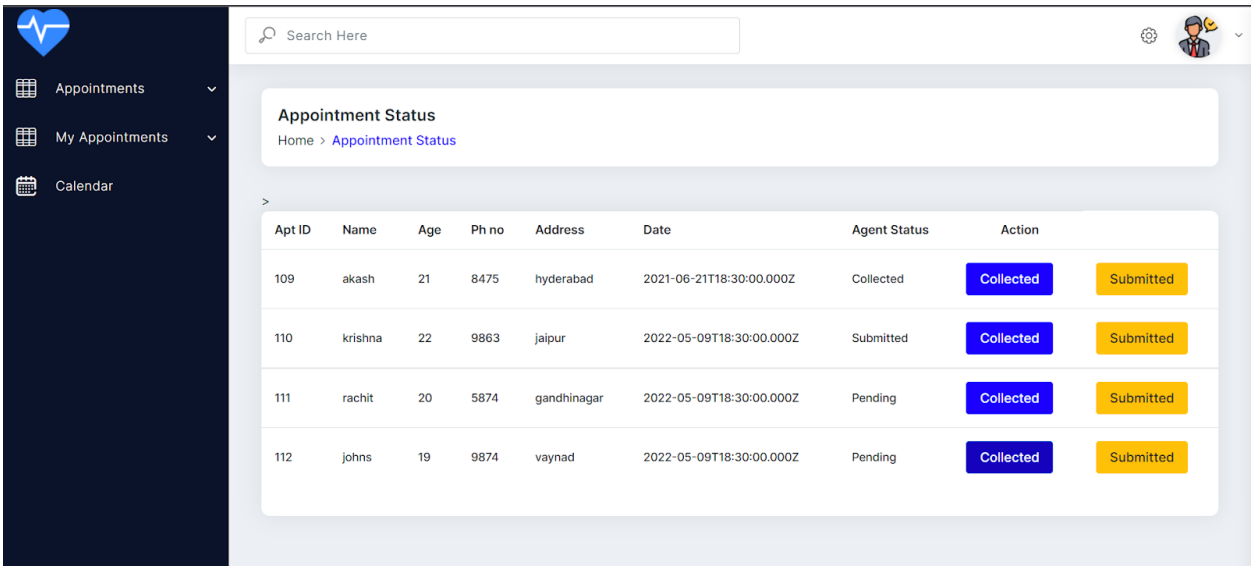


Agent Interface:

Appointments:

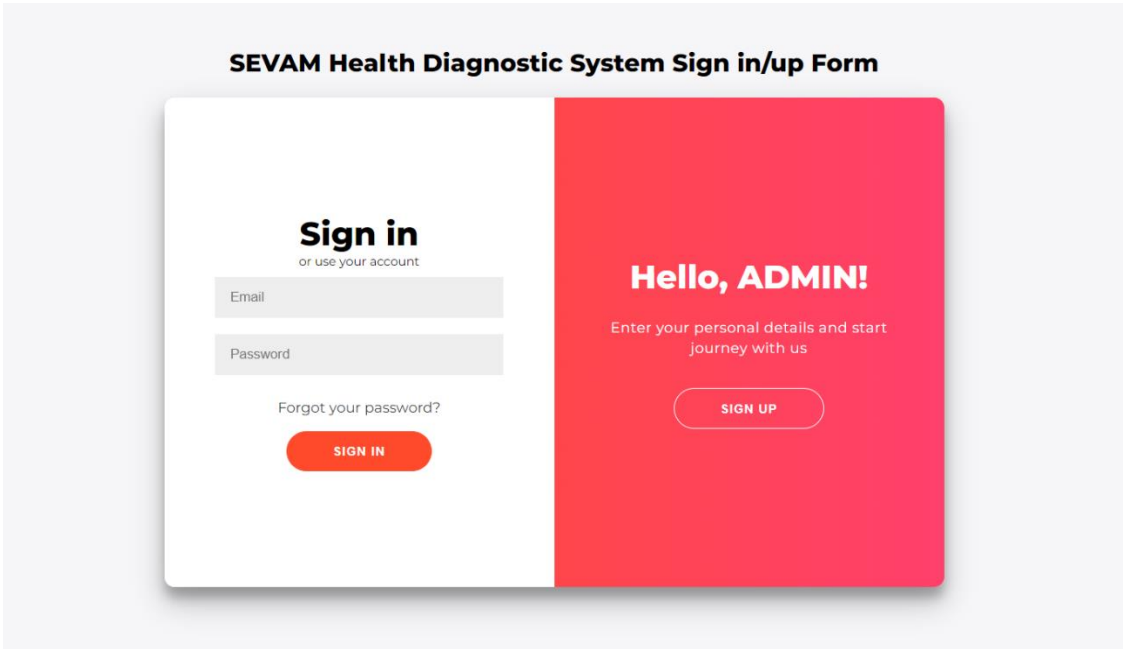


Booking Status:



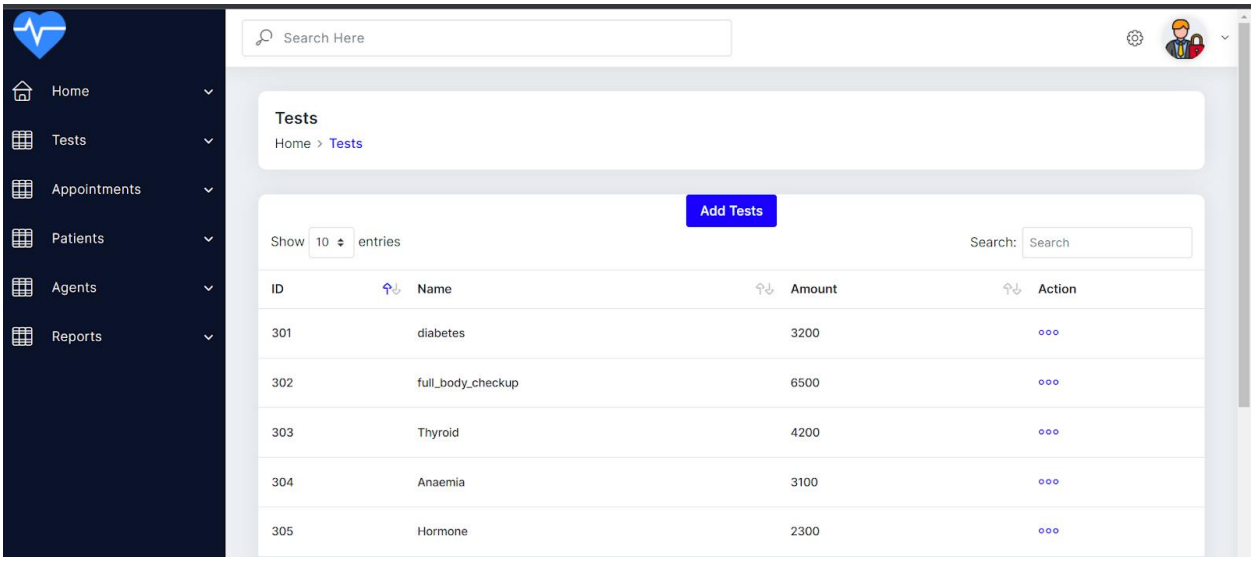
Logout:

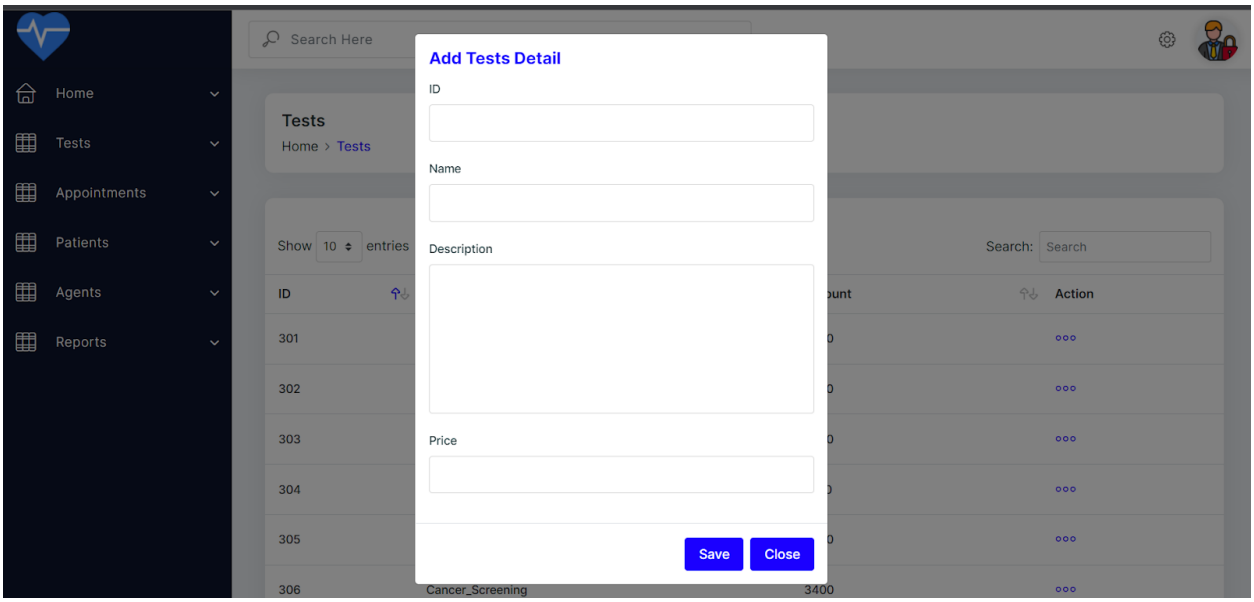
Admin Sign in Sign up page:



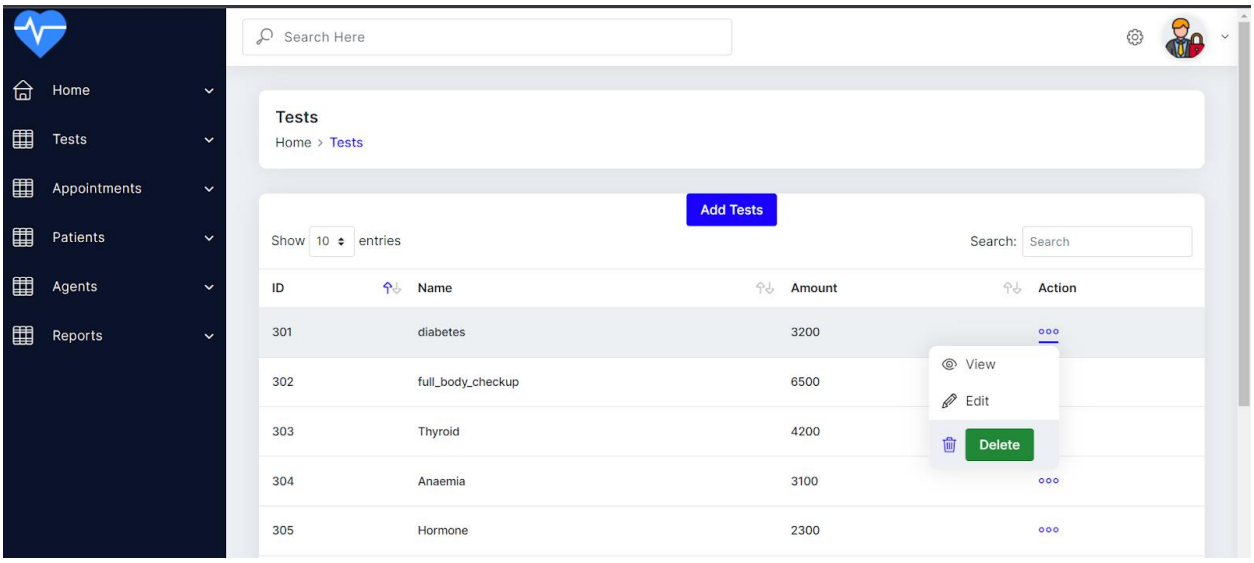
Admin Interface:

Add Test:

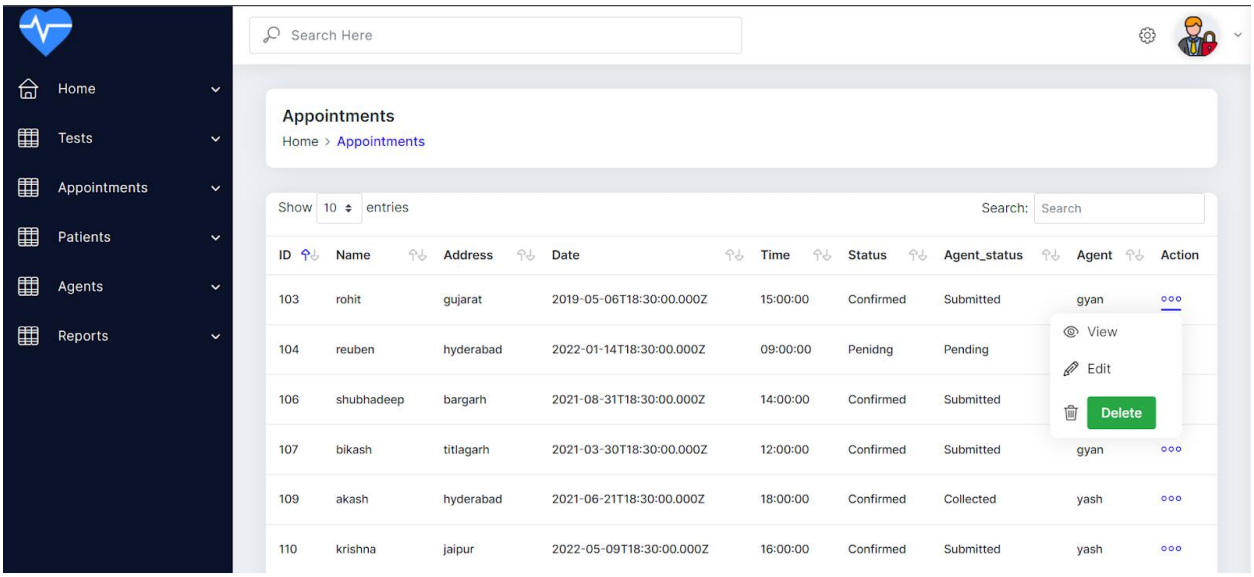




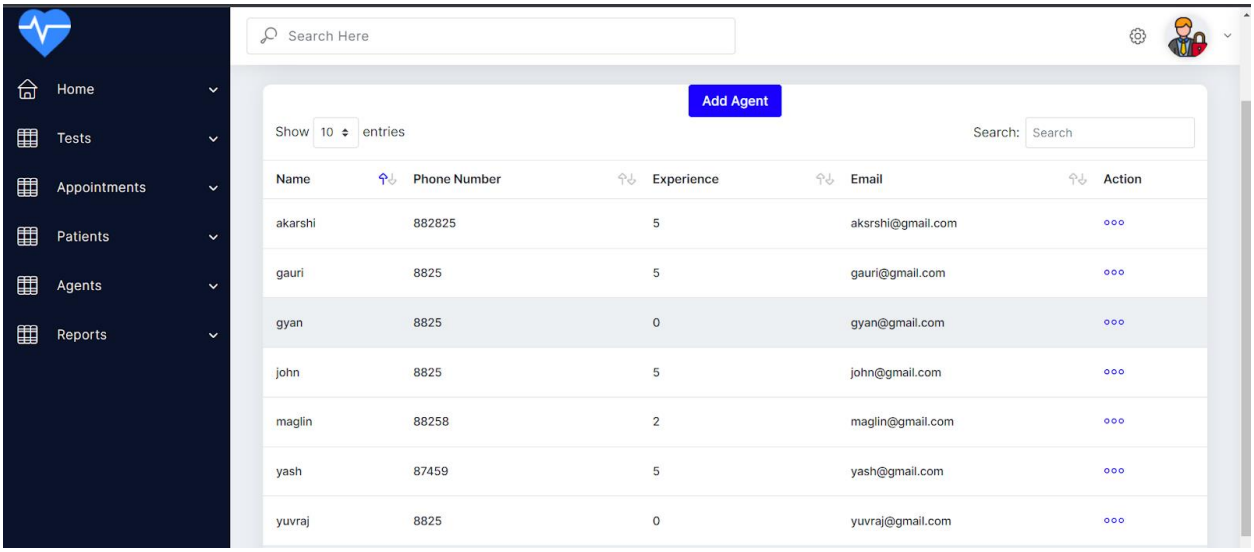
Delete Test:



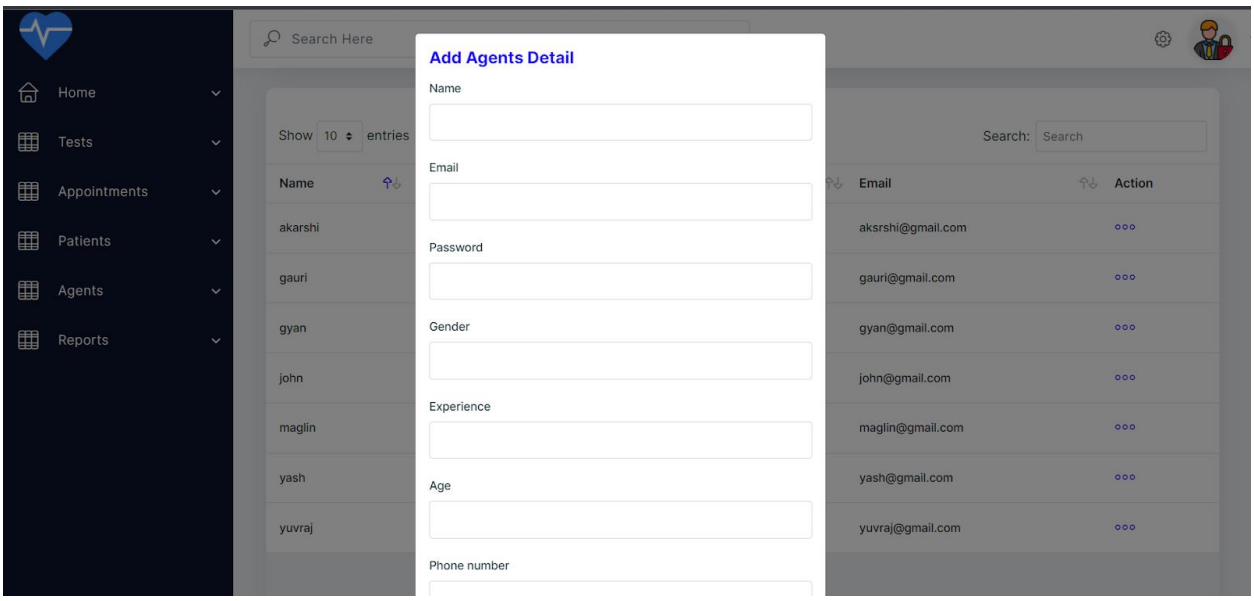
Delete Patient Entry:



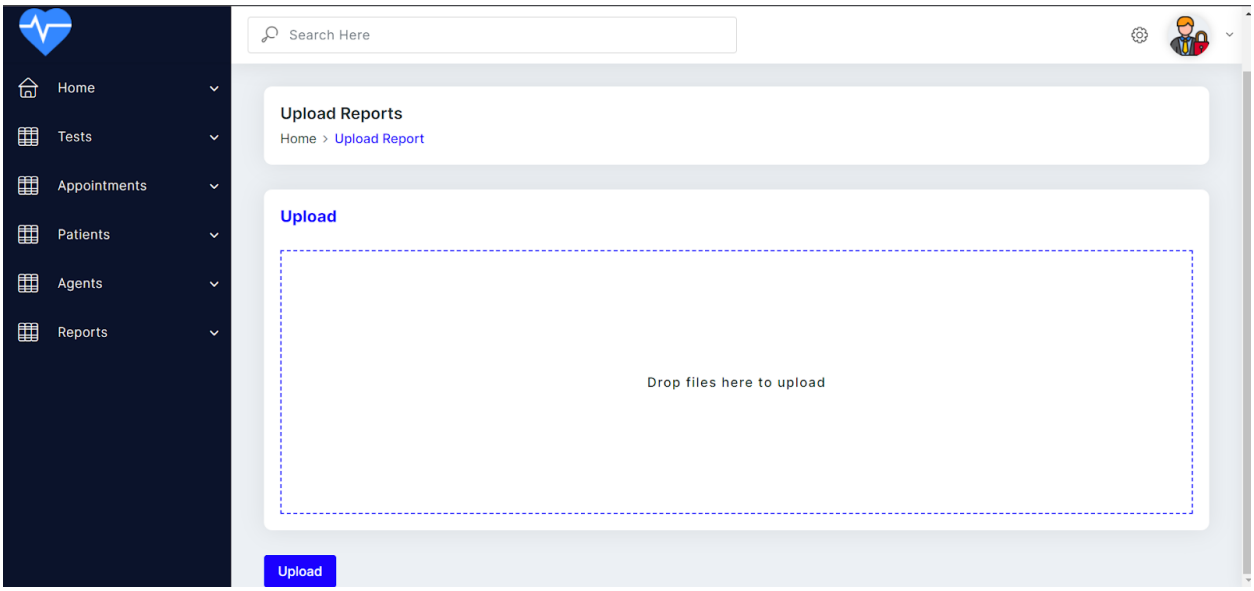
Add Agent:



Add Agent Details:



Upload Report:



Conlcusion And Learnings:

Sevam Health Labs assisted in the enhancement of current online lab systems.

We could learn about the different programming languages, software tools, frameworks, and databases during the beginning stages of the project, and after a successful completion of the planning stage, we could select the best resources and platforms for our project. Following this step, we learned how to create the website's structure using ER diagrams, architectural diagrams, and use case diagrams.

After determining the framework of our website, we began coding and creating a database for the project, which assisted us in learning HTML and CSS, , MySQL, JS, Node JS , Angular JS, Express JS, Cookie-Parser.

We began testing our project after the coding phase, which was a novel notion for us because it had previously been only theoretical, but now we could work with it literally. We learned about various software and tools and how they work, and we were able to use most of them in our project. Apart from technical skills, we could work on team building and communication skills, which helped us boost our confidence in taking on any task at any time. Finally, with all of these skills, we were able to understand and learn about the timely completion of the project, and we were able to complete most of the modules in the time allotted.

References:

- https://www.geeksforgeeks.org/nodejs/
- https://www.geeksforgeeks.org/angularjs/
- https://youtu.be/i9MHigUZKEM - Angular JS
- https://www.youtube.com/watch?v=TlB_eWDSMt4 – Node JS