A
PROJECT REPORT ON

Hospital Management System

SUBMITTED IN
PARTIAL FULFILLMENT OF

DIPLOMA IN ADVANCED COMPUTING (PG-DAC)

BY

Akansha Bacharwar
Atharva Jamdar
Dipak Ghule

UNDER THE GUIDENCE OF

Mr. Aditya Sable

AT

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY, PUNE

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY, PUNE.

CERTIFICATE

This is to certify that the project

Hospital Management System

Has been submitted by

Akansha Bacharwar
Atharva Jamdar
Dipak Ghule

In partial fulfillment of the requirement for the Course of **PG Diploma in Advanced Computing (PG-DAC AUG2024)** as prescribed by The **CDAC** ACTS, PUNE.

Place: Pune

Date: 11-Feb-2025

Mr. Aditya Sable
Project Guide

Mr. Yogesh Kolhe
Course Co-ordinator

ACKNOWLEDGEMENT

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Nitin Kudale (Center Coordinator, SIIT, Pune) and Mr. Yogesh Kolhe (Course Coordinator, SIIT ,Pune) .

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of Sunbeam Institute of Information Technology, Pune for their support.

Akansha Bacharwar

Atharva Jamdar

Dipak Ghule

0824 PG-DAC

SIIT Pune

ABSTRACT

The Hospital Management System (HMS) is a web-based application designed to streamline hospital operations by integrating various functionalities, including patient management, doctor scheduling, appointment booking, Blood Donor availability. The system aims to enhance efficiency, reduce paperwork, and provide a centralized database for quick access to patient records and hospital resources.

The HMS is developed using React.js for the frontend, Spring Boot (STS) for the backend, and MySQL as the database. The system ensures secure authentication, role-based access control, and real-time data updates. Key features include patient registration, electronic health records (EHR), doctor and staff management, appointment scheduling, billing automation, and report generation.

By implementing HMS, hospitals can improve patient care, reduce administrative workload, and ensure better resource utilization. The project follows a full-stack development approach, incorporating modern technologies to provide a scalable and user-friendly solution for hospital management.

Index

1. INTRODUCTION TO PROJECT

A patient is registered to the hospital by filling up a form. The form data is entered into the system by an operator using a graphical user interface. In the interface, there will be some mandatory fields as mentioned in the form. After filling up and submission using the interface, all data will be saved in the database to the corresponding tables. The patient is registered and then he/she will book an appointment to visit a doctor in which a form is to be filled to enter the time slot, date, symptoms and choose a doctor if a patient is not educated enough to use the online services, he/she can give a call to the hospital receptionist and register the patient as well as book the appointment in no time. The receptionist will create that patient's profile into the database by using interface and if someday the patient wants to book appointment online by someone's help it is possible as well. Then this data is transferred to the allotted doctor and receptionist where the doctor visits the patient and prescribes tests and medicines.

After the treatment is over the patient visits receptionist in the hospital and the billing is done according to the tests, doctor's fees, prescribed medicines, hospital charges and taxes and a receipt will be printed.

2. PROBLEM DOMAIN

Till now, using the traditional methods used in the hospitals , the patients had to wait in a long queue and fill up their details and then move to the doctor to have consultation, This process was repeated at every step including the prescribed tests, taking reports, buying medicines, re-consulting the doctor and getting assigned the doctor which makes it a very time-taking and exhausting process for the patients as well as the receptionist to do all the job in a smooth manner, This process exhausted the patient which sometimes lead to hard results. The billing is something which in hospitals is done in bits and pieces and is a problematic situation for families. Sometimes the patients are not so educated to fill the admission forms in the hospital which causes a delay in their admission and may sometimes lead to deprivation from the facilities provided by the government and the hospital.

3. SOLUTION DOMAIN

The main purpose of this software is that it makes the things easy for the patients as well as their receptionist and the hospital staff. The application provides an online database of the patients to the hospital as well as the patients, which helps in their further treatment. This will lead to queues getting shorter and dispersing faster as most of the patients can register themselves easily through their mobile phones. And they can be directly assigned the doctor and the doctor can view the patients' medical history just by logging in into the patient's database. Which can lead to a smoother process and the patient cannot have to wait in long queues and get exhausted as the patient will be assigned a ward directly and the doctor can directly visit the patient into his ward, eliminating the situations of waiting in queues. This will also allow the receptionist and the hospital staff to help the not so educated part of the society, as the people standing in the queues will be less the receptionist can easily help the people who are not introduced to the latest technologies. The traditional methods in the hospitals used were asking the patients to pay at every step but when the patients database is accessible to the hospital staff. Then the billing can be done at the time of the discharge of the patient not at every single point. Which makes it easier for the patients and their attendants to gather the money at one point not at every point during the treatment.

4. REQUIREMENTS

4.1 FUNCTIONAL REQUIREMENTS

4.1.1 Use Case for Administrator

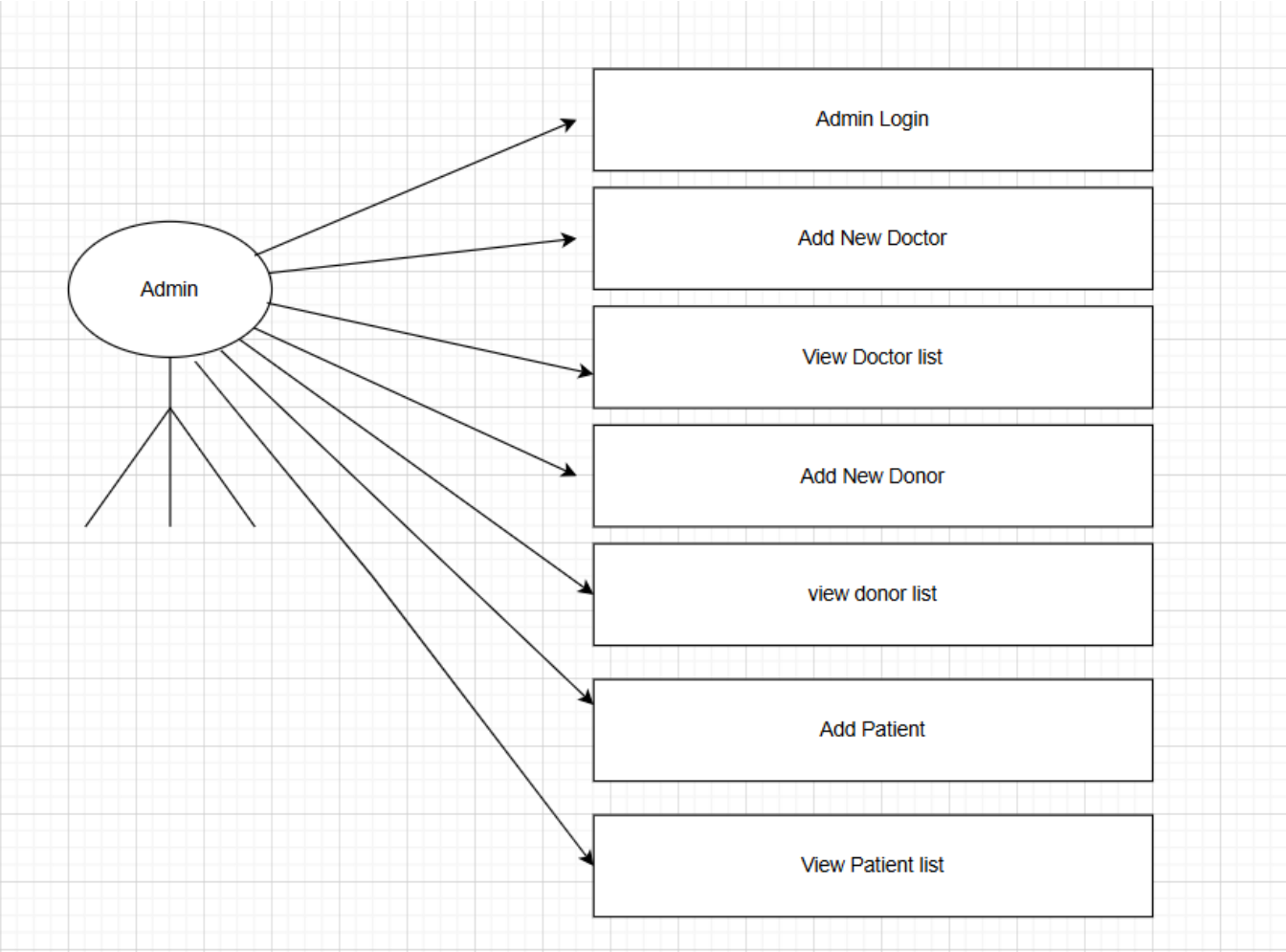

Figure 1. Use Case for Administrator

4.1.2 Use Case for Patient

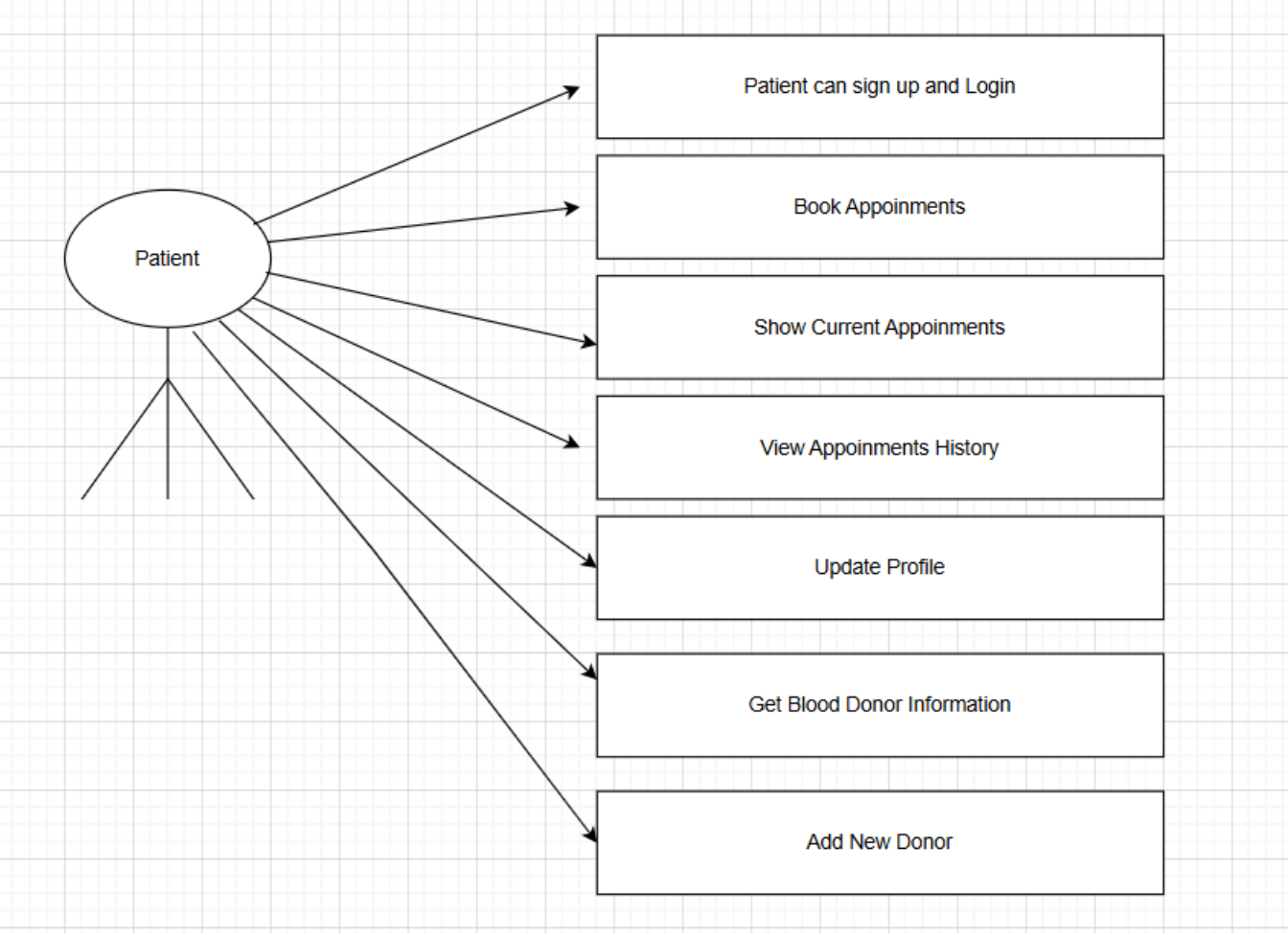

Figure 2. Use Case for Patient

4.1.3 Use Case for Doctor.

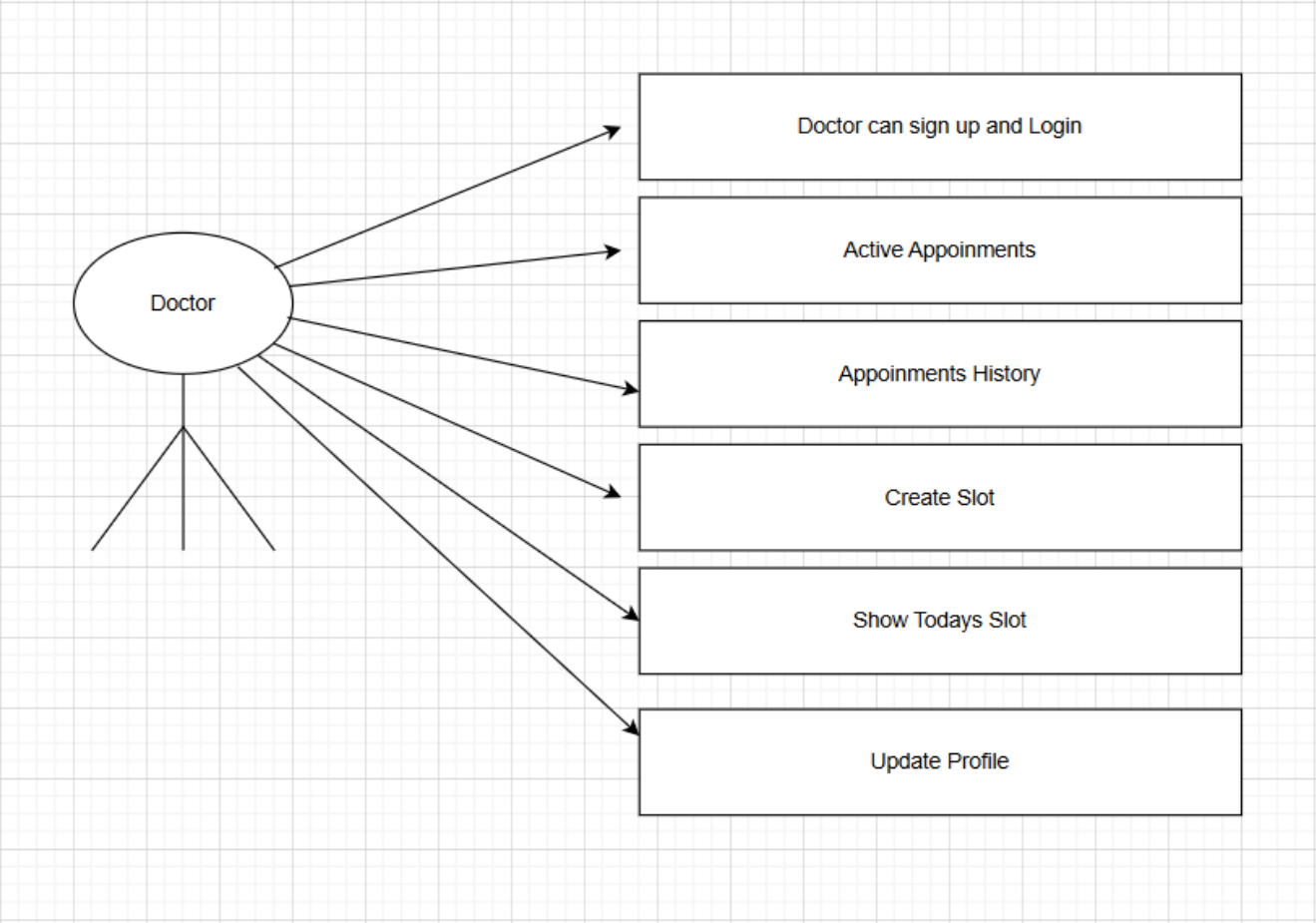

Figure 3. Use Case for Docor

4.2 NON-FUNCTIONAL REQUIREMENTS

4.2.1 Usability Requirement

Registration

Patient can register oneself by filling the signup form or Receptionist can also register Patient while Registration of Receptionist and Doctor is done by Admin only.

Security

The complete control of the project is under the hands of authorized person who has the password to access this project and illegal access is not supposed to deal with. Maximum control is under the administrator, then receptionist have control over patient data as well as appointment details on other hand other members including Doctor and Patient have the rights to just see the records and can edit own profile if any changes happen.

Session Management:

We can store user-related information in a session in form of key and value pairs. The HTTP Session interface defines the setAttribute (key, value) method to store a key-value entry and getAttribute(key) method to get value of a specified key.

As user login, user Id is stored in session storage and by using this user Id we can handle individual functionalities and display the data related to particular user.

As user logout from one's respective profile, the Id stored in session storage get deleted automatically.

Due to session management, if anyone tries to access the details of any user directly through then page directly renders to the homepage and by this way we achieved, security with the help of session management.

Technologies Used

React:

React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex UI's from small and isolated pieces of code called "components". React is a JavaScript library for building user interfaces. React is used to build single page applications. React allows us to create reusable UI components. All the front end was completed with the help of React.

MySQL

MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. All the User's data which is part of Hospital management system is managed with the help of MY-SQL.

Spring Tool Suite

Spring Tool Suite is an IDE to develop Spring applications. It is an Eclipse-based development environment. It provides a ready-to-use environment to implement, run, deploy, and debug the application. It validates our application and provides quick fixes for the applications. With the help of Spring tool suite, we created a Spring Boot project from Eclipse and used it for the developing the back-end part.

V S Code

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDE's, such as Visual Studio IDE. With the help of V S Code, we created a react js project and used it for the developing the front-end part.

Git Lab

GitLab is a web-based Git repository that provides free open and private repositories, issue-following capabilities, and wikis. It is a complete DevOps platform that enables professionals to perform all the tasks in a project—from project planning and source code management to monitoring and security. All the project source code and documentation version control as well as management was done using Git Lab.

5. Database Design

The following table structures depict the database design.

1) Admin_tbl

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
email	varchar(255)	NO		NULL	
name	varchar(20)	NO		NULL	
password	varchar(255)	NO		NULL	

2) Appointment_tbl

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
appointment_time	datetime(6)	YES		NULL	
appointment_type	varchar(255)	YES		NULL	
doctor_id	bigint	NO	MUL	NULL	
patient_id	bigint	NO	MUL	NULL	

3) blood_donor_tbl

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
blood_group	varchar(255)	YES		NULL	
city	varchar(30)	YES		NULL	
contact_number	varchar(10)	YES		NULL	
email	varchar(30)	YES		NULL	
name	varchar(30)	YES		NULL	
state	varchar(30)	YES		NULL	

4) **doctor_tbl**

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
area	varchar(255)	YES		NULL	
city	varchar(255)	YES		NULL	
dob	date	YES		NULL	
email	varchar(30)	YES	UNI	NULL	
first_name	varchar(30)	YES		NULL	
gender	varchar(255)	YES		NULL	
last_name	varchar(30)	YES		NULL	
mobile_number	varchar(10)	NO		NULL	
password	varchar(255)	NO		NULL	
state	varchar(255)	YES		NULL	
username	varchar(30)	YES	UNI	NULL	
fees	int	YES		NULL	
languages	varchar(30)	YES		NULL	
qualification	varchar(30)	YES		NULL	
specialization	varchar(30)	YES		NULL	
time_slot_id	bigint	YES	MUL	NULL	

5) **doctor_time_table_available_slots**

Field	Type	Null	Key	Default	Extra
doctor_time_table_id	bigint	NO	PRI	NULL	
available_slots	bit(1)	YES		NULL	
available_slots_key	datetime(6)	NO	PRI	NULL	

6) **doctor_time_table_holidays**

Field	Type	Null	Key	Default	Extra
doctor_time_table_id	bigint	NO	MUL	NULL	
holidays	varchar(255)	YES		NULL	

7) doctor_timetable_tbl

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
break_time	time	YES		NULL	
end_date	date	YES		NULL	
end_time	time	YES		NULL	
slot_duration	int	NO		NULL	
start_date	date	YES		NULL	
start_time	time	YES		NULL	

8) patient_tbl

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
area	varchar(255)	YES		NULL	
city	varchar(255)	YES		NULL	
dob	date	YES		NULL	
email	varchar(30)	YES	UNI	NULL	
first_name	varchar(30)	YES		NULL	
gender	varchar(255)	YES		NULL	
last_name	varchar(30)	YES		NULL	
mobile_number	varchar(10)	NO		NULL	
password	varchar(255)	NO		NULL	
state	varchar(255)	YES		NULL	
username	varchar(30)	YES	UNI	NULL	
blood_group	varchar(255)	YES		NULL	

ER diagram –

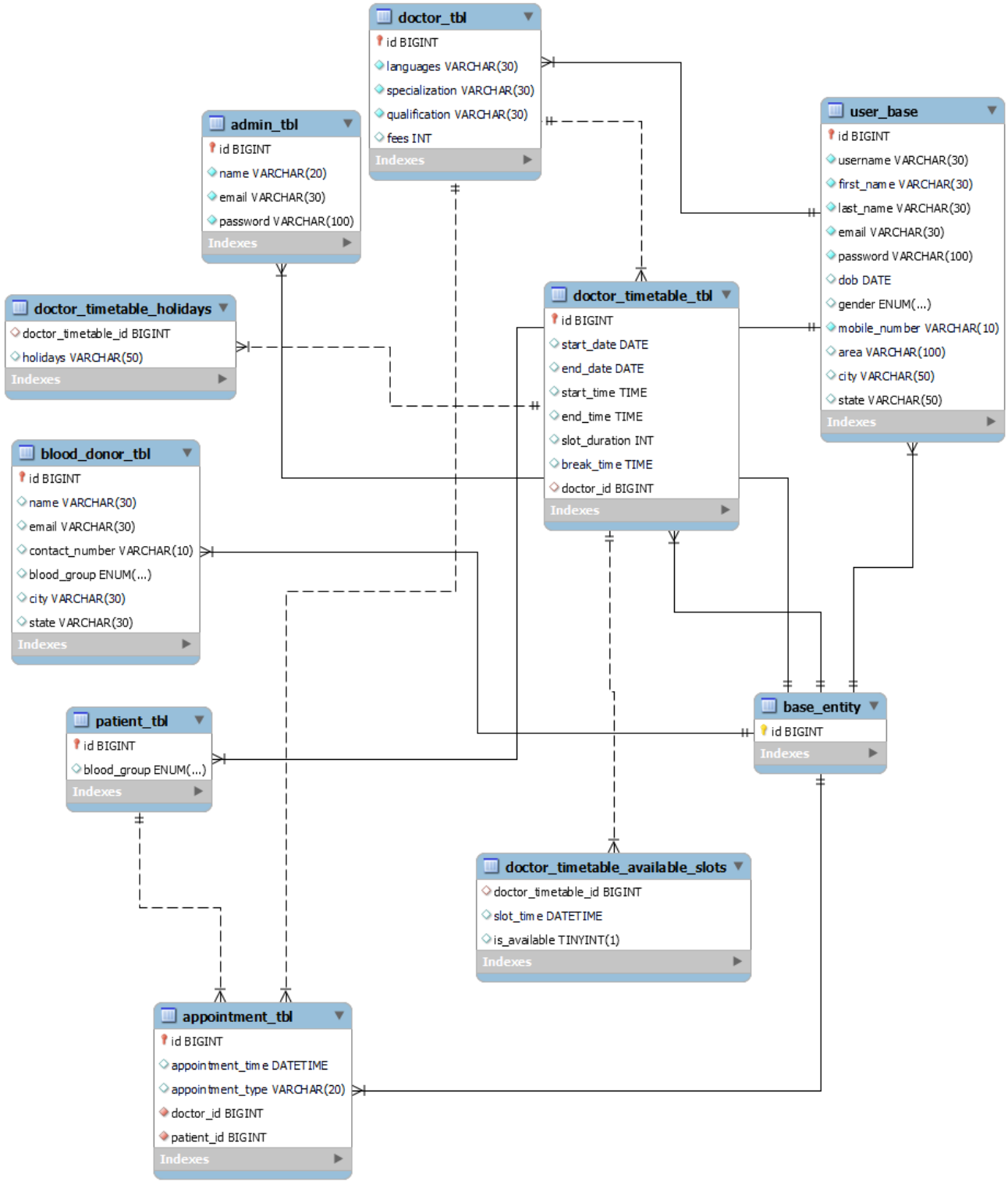

Figure 1. ER diagram.

5.1 Process model -

5.1.1 Data flow diagram for login module

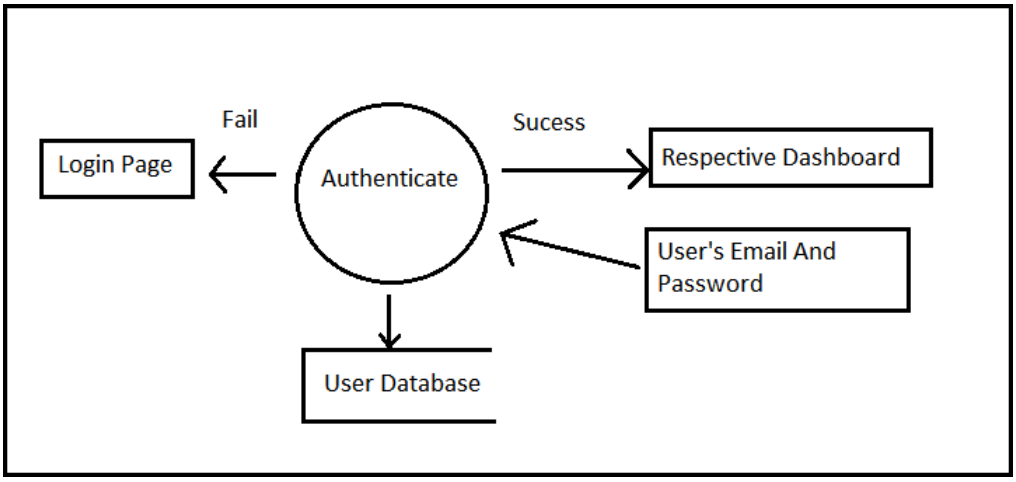

Figure 6. Data flow diagram for login module.

5.1.2 Design methodology:

What is a Methodology?

Software engineering is carried out using preferred procedure techniques to progress the quality of a software development effort. A methodology is defined as a collection of procedures, techniques, tools, and documentation aids which will help developers in their efforts (both product and process related activities) to implement a new system. For successful implementation, a well-organized and systematic approach is crucial. Therefore, several methodologies were developed to encourage the systematic approach to planning, analysis, design, testing and implementation. Methodologies offer various tools and techniques to assist in analysis, design and testing in terms of detailed design of software, data flowcharts and database design.

Why Methodology?

1. To complete a project within time and budget with the expected scope and quality we need methodologies which provide for a framework.

1. Most methodologies have a general planning, developing and managing stages in common. They suggest the development team the ways of thinking, learning and arriving at a regular feasible solution.

2. To select an ideal methodology was based on project requirements and goals.

4. Functional Decomposition**:** The methodology should have stages according to the interrelated activities which can be grouped into different functional areas.

5. Requirement Changes**:** If required, methodology provides scope to change the requirement.

3.

6. Manage Risks**:** Determined the risk is an important activity to develop a project.

Documentation: Methodology provides support for large documentation.

Analysis and Design Support: A well-defined structure of the methodology helps for analysis and designing to development process.

Implementation: The system should be implemented as per plan.

Testing Support: More testing, more reliable the product is.

Object Oriented Approach: Object oriented concepts will be used in developing the project as it supports component re-usability.

Suitable Methodologies:

Agile development

Agile software development refers to software development methodologies centered round the idea of iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams. The ultimate value in Agile development is that it enables teams to deliver value faster, with greater quality and predictability, and greater aptitude to respond to change. Scrum and Kan-ban are two of the most widely used Agile methodologies. Below are the most frequently asked questions around Agile and Scrum, answered by our experts.

What is Agile?

Figure 7. Agile Diagram.

Agile software development refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams. Agile methods or Agile processes generally promote a disciplined project management process that encourages frequent inspection and adaptation, a leadership philosophy that encourages teamwork, self-organization and accountability, a set of engineering best practices intended to allow for rapid delivery of high-quality software, and a business approach that aligns development with customer needs and company goals. Agile development refers to any development process that is aligned with the concepts of the Agile Manifesto. The Manifesto was developed by a group of fourteen leading figures in the software industry, and reflects their experience of what approaches do and do not work for software development. Read more about the Agile Manifesto. Did you know that Agile can also be applied to hardware projects? Learn about C prime's revolutionary Agile for Hardware framework.

6. MODULES

MODULES USED

The system comprises of 2 major modules with their sub-modules as follows:

1. **New User Registration:**
    a. **User Login:** User email of the User logging in. here we have provided 3 login UI for admin, doctor and patient.
    b. **Password:** The Login Credentials entered by the user to log in.
        c. **Home page:** every staff member from (Admin, Doctor, Staff) can visit his/her home page after login.
        d. **Admin login:** After login go to admin home page and can view particular Admin details and functions provided for admin such as he can view/update/delete staff and doctor he also can view patient details.
        e. **Doctor login:** After login go to doctor home page and can view particular Doctor details and functions provided for doctor such as he can view own appointments and regarding patient details.
        f. **Patient login:** After login go to staff home page and can view particular staff details and functions provided for staff such as he can view/update/delete patients appointments and also can view doctor appointment details. Receptionist can add patient and book appointment for patients by selecting doctor. After check-up creating invoice done by receptionist via doctor.

2. **New Patient Registration:**
    a. **Registration:** Patient can register his detail.
    b. **Login:** Patient Login his account.
    c. **Home page:** Patient can visit his home page after login.
d. **Patient Detail:** Can view Patient details by selecting a Patient and view its details such as name, disease, doctor allotted, bill etc…
e. **View/Edit/Delete:** Can view/update/delete the added Patient from the database.
f. **Book Appointment:** Patient can book his/her appointment by selecting a doctor.

g. **View Appointment History:** Can track the Past Appointments by the doctor.

h. **Payment:** The payments will be only on COD basis after invoice generation.

i. **Change Password:** User can change his current password and make new password.

7. TESTING

7.1 TESTING PHASE

One of the purposes of the testing is to validate and verify the system. Verification means checking the system to ensure that it is doing what the function is supposed to do and Validation means checking to ensure that system is doing what the user wants it to do.

No program or system design is perfect; communication between the user and the designer is not always complete or clear, and time is usually short. The result is errors and more errors. Theoretically, a newly designed system should have all the pieces in working order, but in reality, each piece works independently. Now is the time to put all the pieces into one system and test it to determine whether it meets the user's requirements. The purpose of system testing is to consider all the likely variations to which it will be subjected and then push the system to its limits. If we implement the system without proper testing then it might cause the problems.

7.1.1. LEVELS OF TESTING:

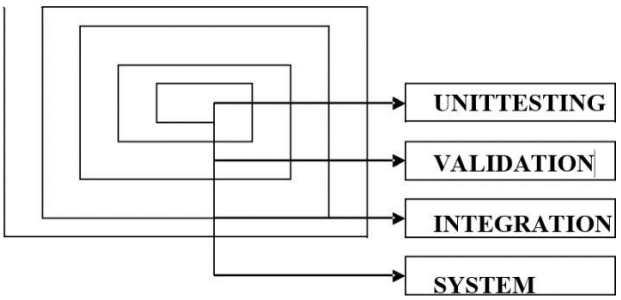

Figure 8. Levels Of Testing

7.2 TEST REPORT

The report of the testing is given here under.

GENERAL TESTING

SR-NO	TEST CASE	EXPECTED RESULT	ACTUAL RESULT	ERROR MESSAGE
1	Register Page	Redirected to login page	OK	Nothing
2	Login Page	Pop-up will come	OK	Invalid credentials
3	Reset password	Only users password will be reset	OK	Enter valid email
4	Add appointment	Appointment is added	OK	Nothing
5	View appointment	Appointments are shown of that patient	OK	Nothing
6	Edit profile	Edit own profile	OK	Nothing
7	View Donor	View donor list	Ok	Noting
8	View patient	All patient list is shown	OK	Nothing
9	Doctor login	Patients appointed are shown	OK	Nothing
10	View appointments / delete appointments	Receptionist can add / delete appointment	OK	Nothing
11	Add Donor	Add new donor	Ok	Nothing
12	View Donor	View donor list	Ok	Noting

8. USER INTERFACE

1. Home Page

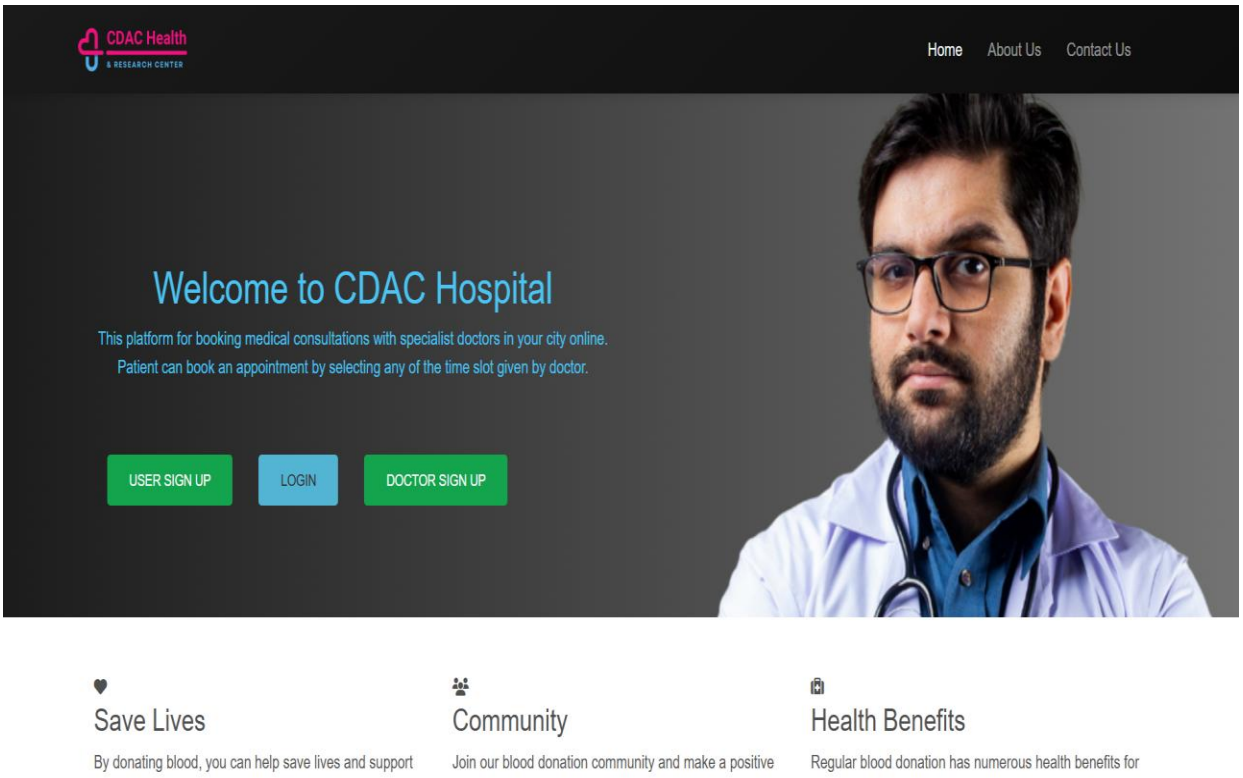

Figure 9. Home Page

2. Login Page

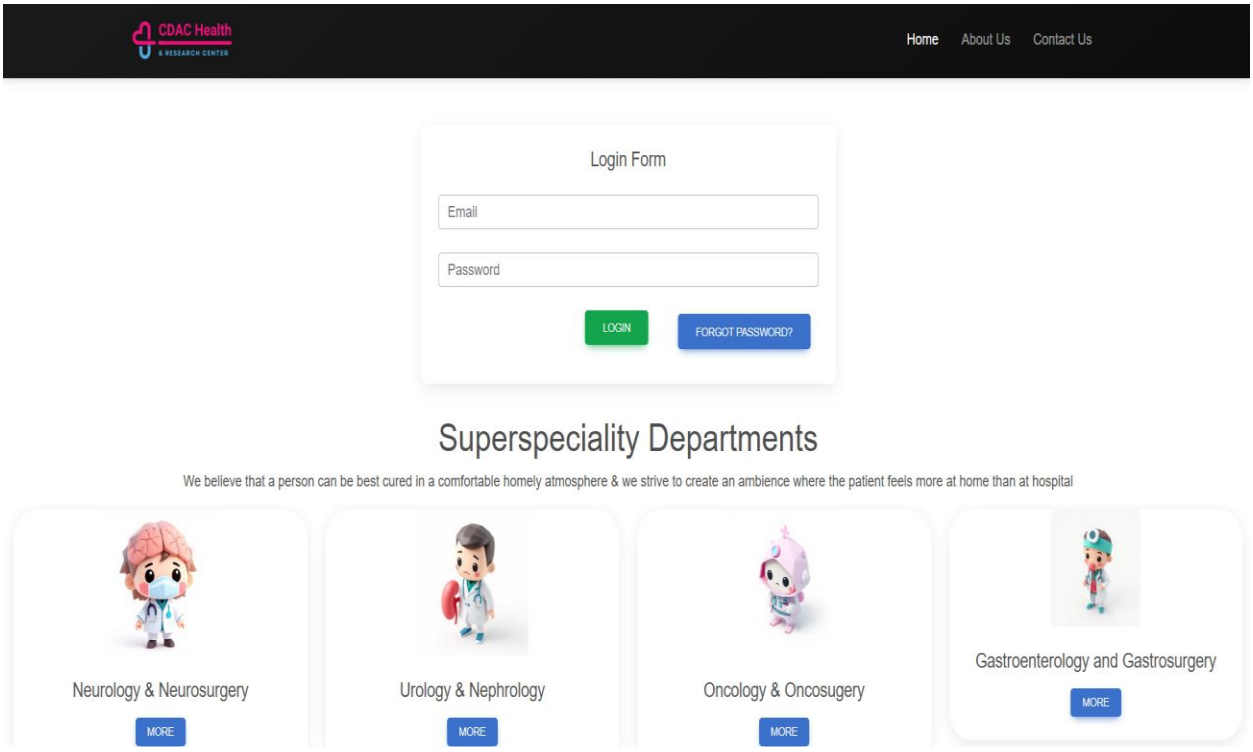

Figure 10. Login Page

3. Reset Password

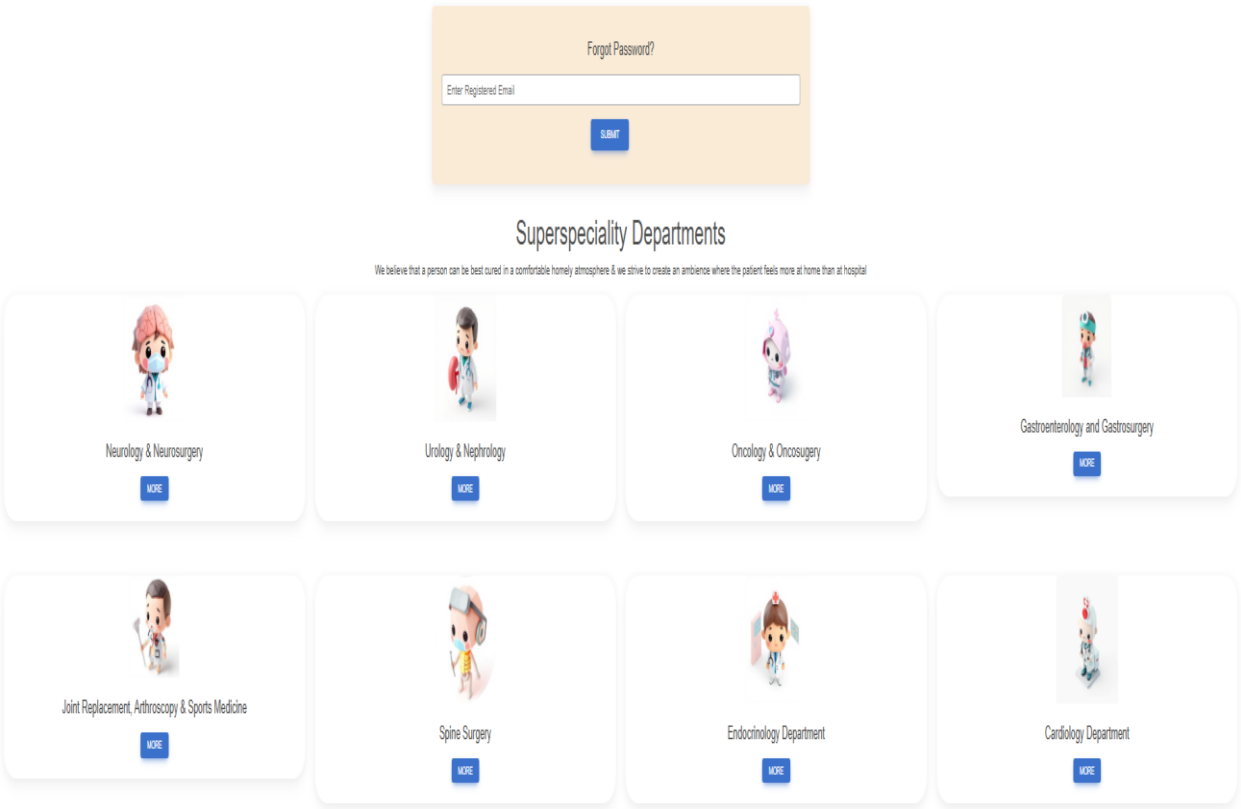

Figure 12. Reset password

4. Admin Homepage

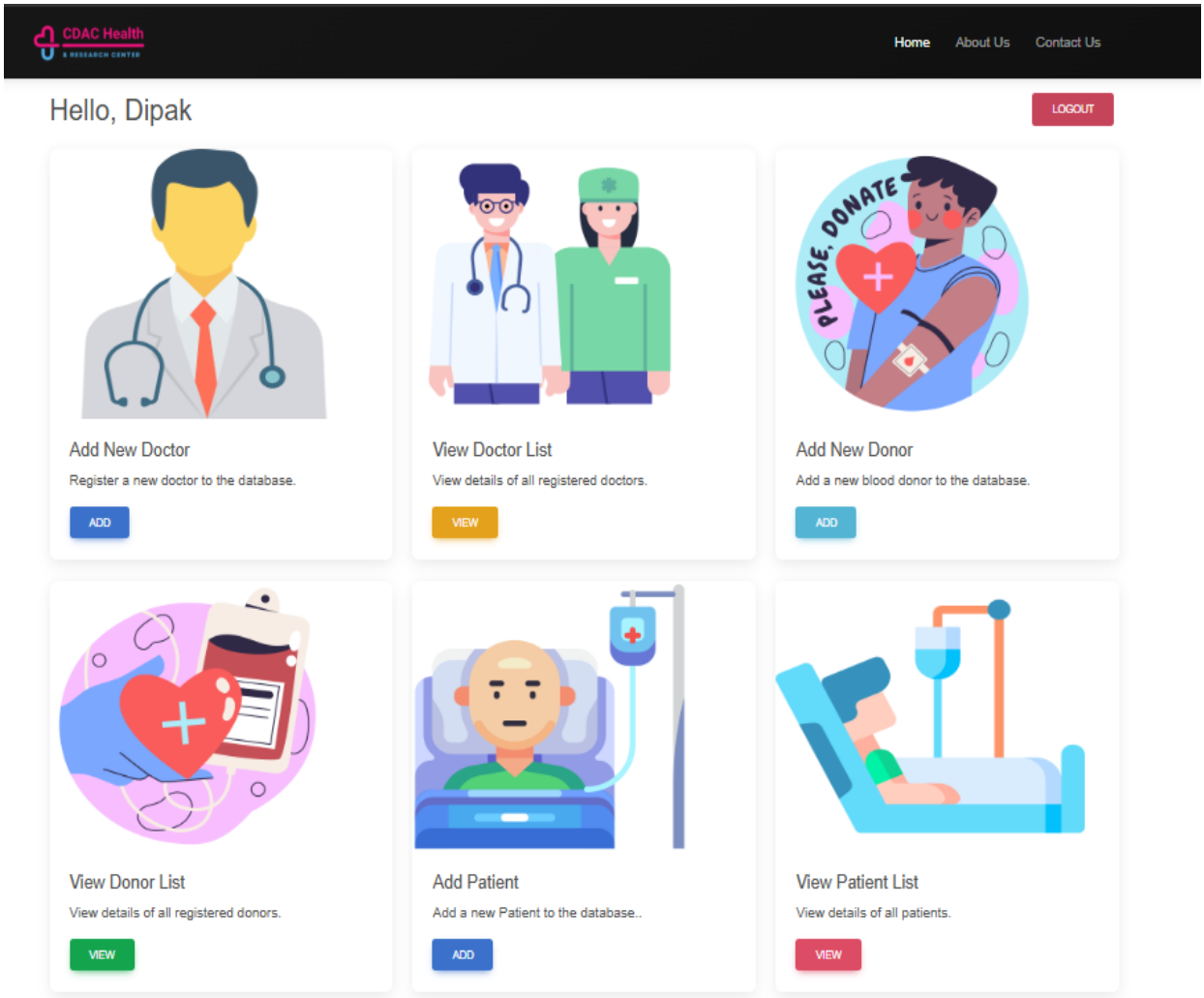

Figure 15.Admin Homepage

5. Patient sign up page

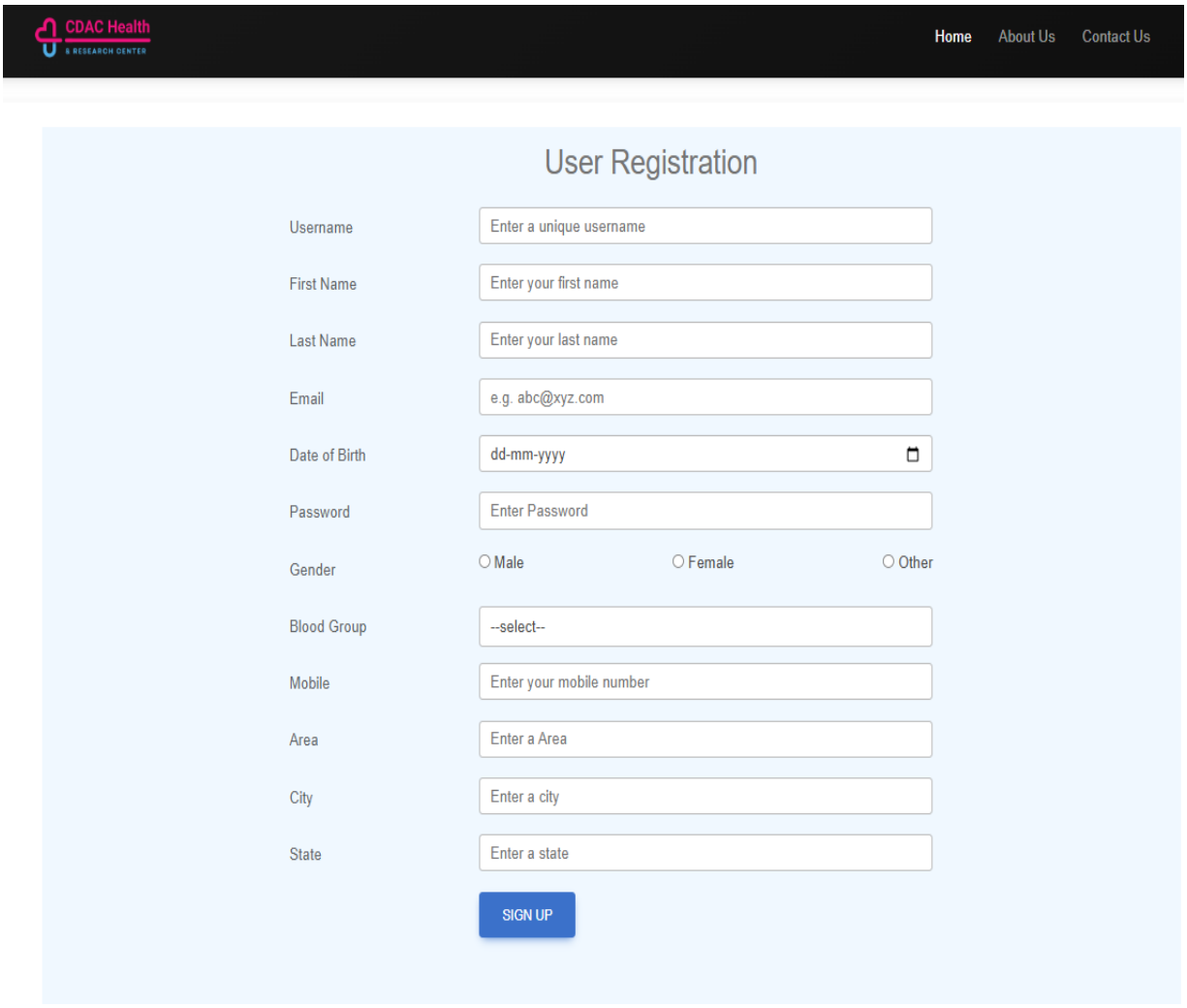

Figure 11. Patient sign up page

6. Patient homepage

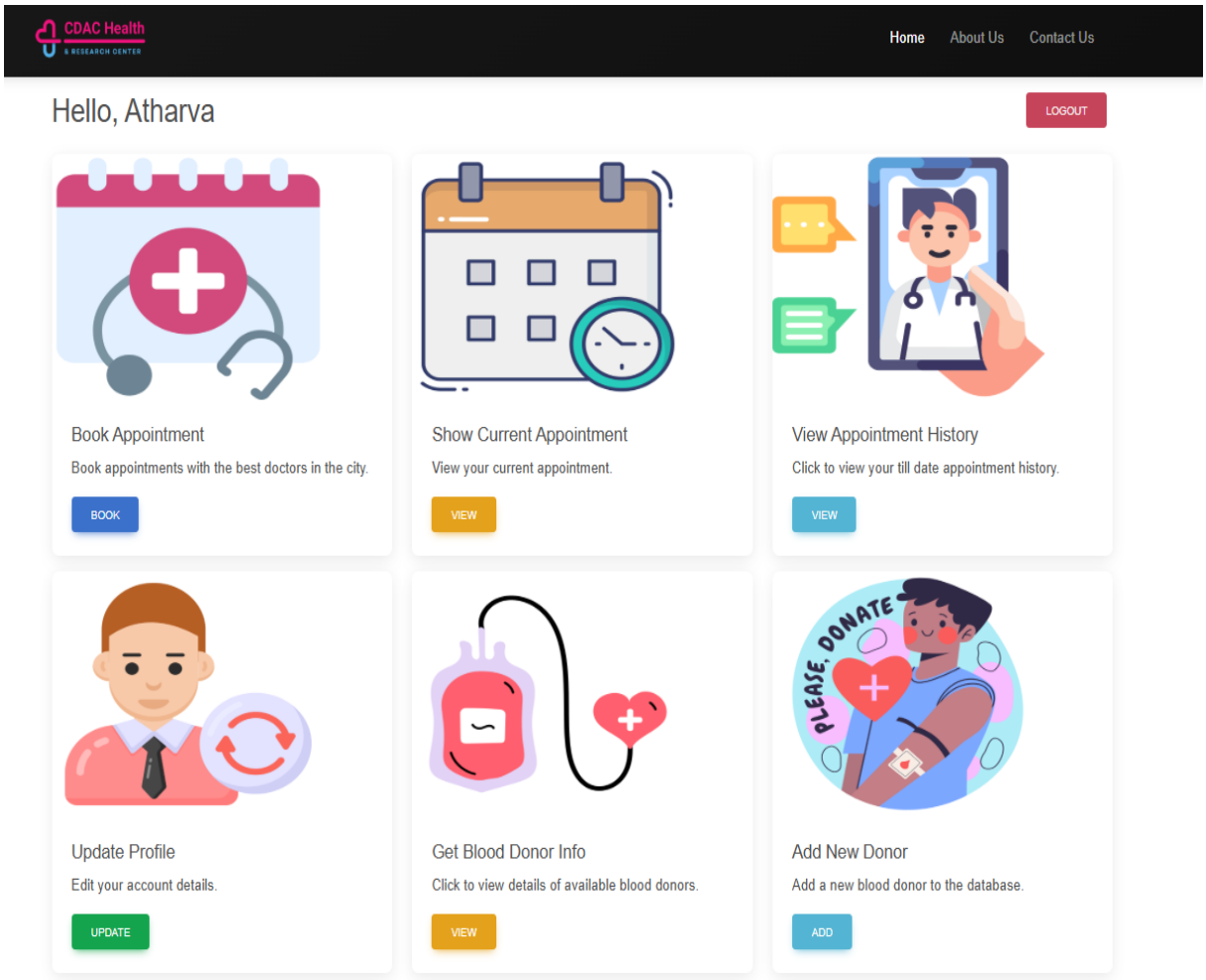

Figure 13. Patient home page

7. Doctor homepage

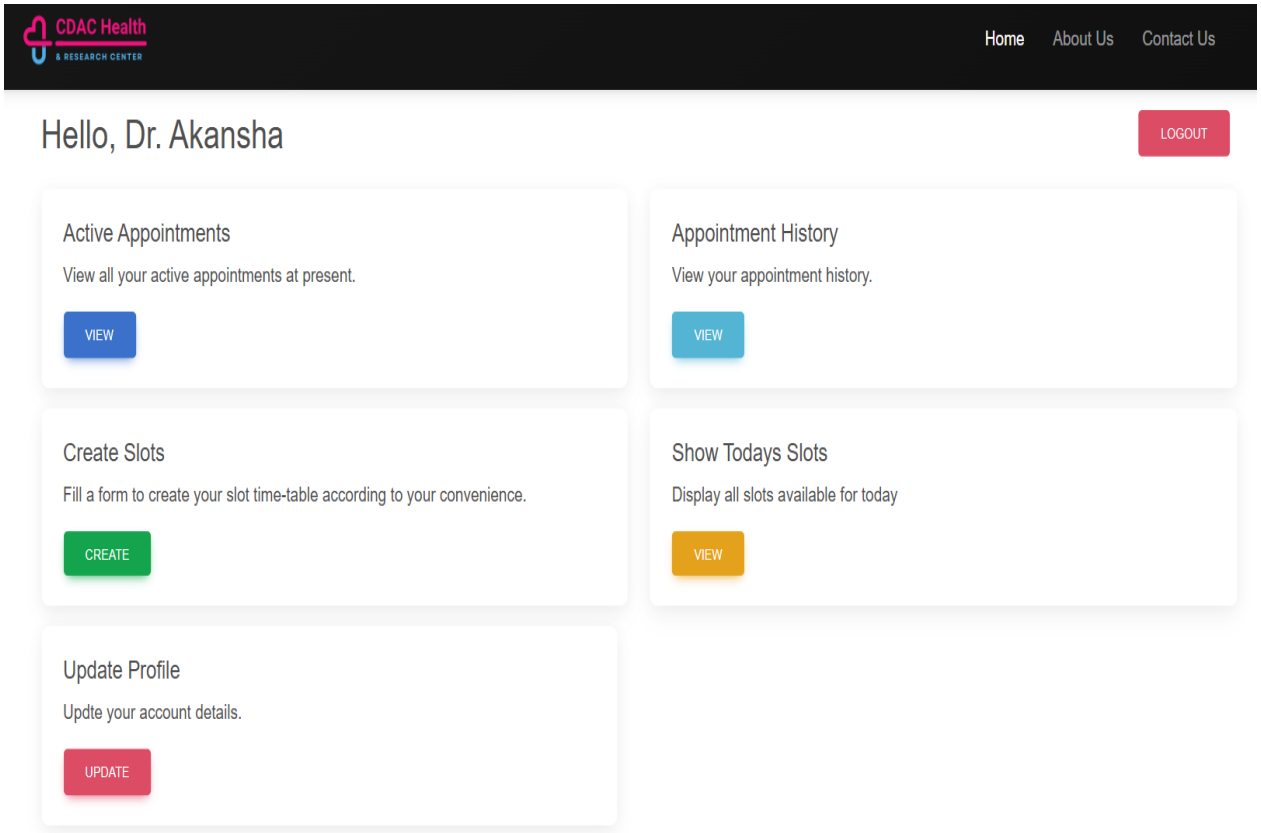

Figure 13. Patient home p

8. Add Patient

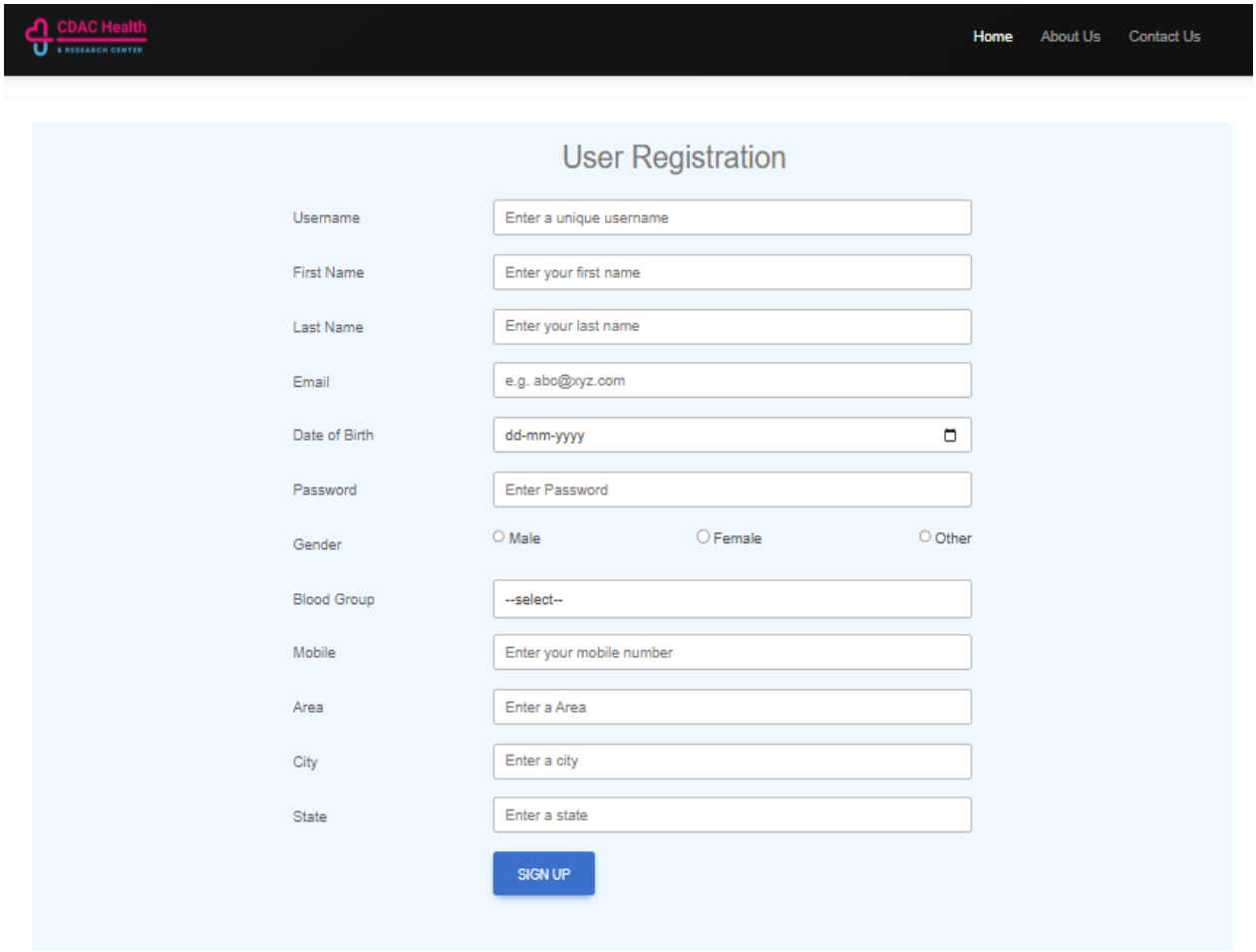

Figure 17. Add patien

9. Add Doctor

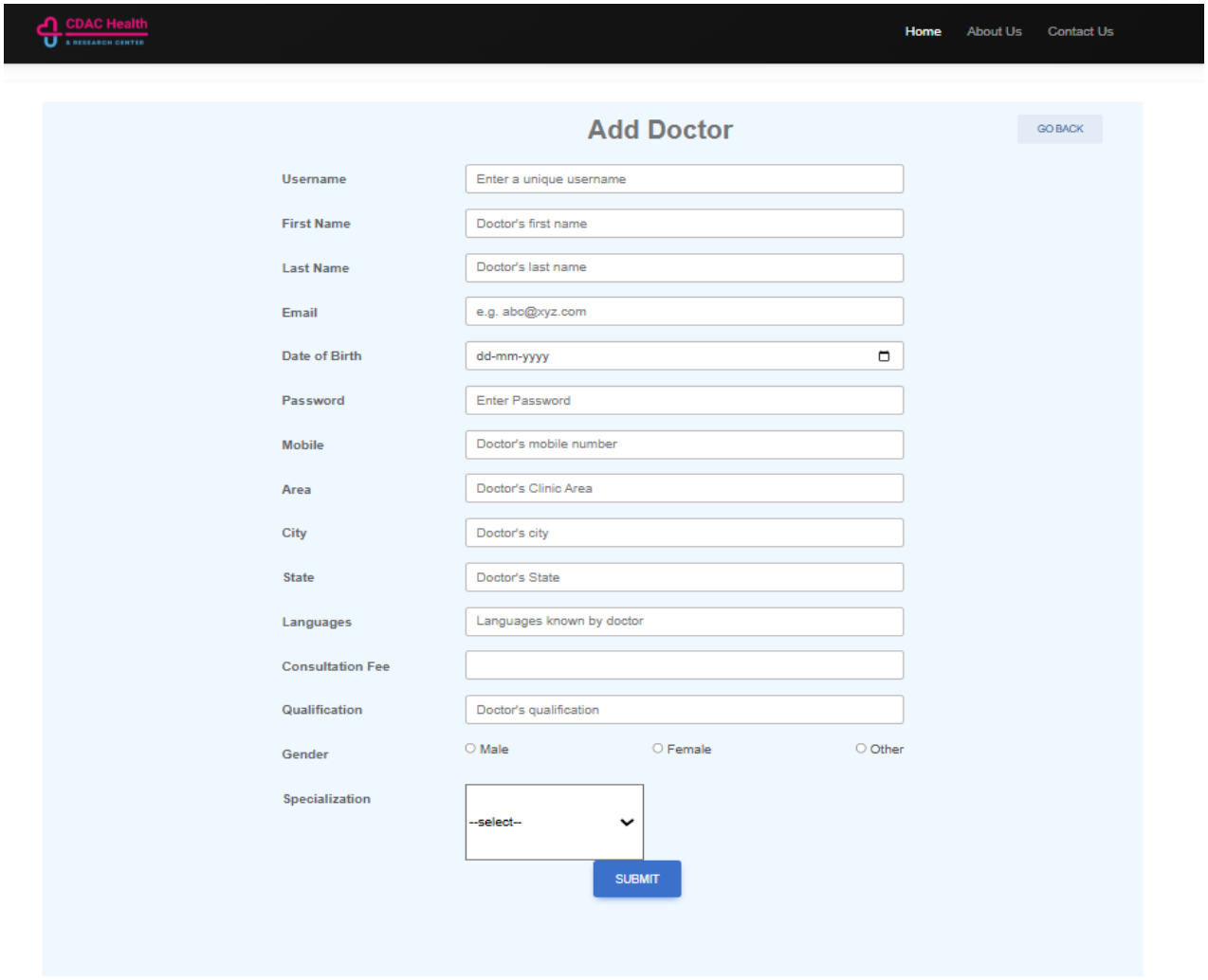

Figure 23 Add Doctor

9. CONCLUSION

After reviewing our work, the conclusion is that we minimize the efforts of the patients to book an appointment. As we know the current pandemic situation the crowd gathering at the hospitals will be reduced as hospitals are the most critical places to spread the virus.

Our project will help to reduce the crowd contact, the patients are given appointment time slots to visit the hospital and meet the doctor. The doctor can also see the patients visiting and their health profile or symptoms.

We enable the hospital staff and management to securely save the data and fetch the data whenever necessary. The website is user friendly and secure to use. If any patient is not aware of the technology, he/she can register their data securely via receptionist.

We give access to 4 kinds of users namely admin, receptionist, doctor and patient using email and password which will be verified by the system, in case if any user happens to forget the password, we have given the service to change the password after validating the user data.

We have provided static as well as dynamic pages to display data. Every specified user has our functionalities and, in this manner, we have tried to achieve contact less appointments according to time slots. Managing staff information and to view information about users and make changes accordingly if necessary.

10. REFERENCES

Links:

- https://getbootstrap.com/docs/5.1/customize
- https://www.w3schools.com/css/
- https://docs.oracle.com/javase/8/docs/api/
- https://stackoverflow.com/
- https://javaee.github.io/javaee-spec/javadocs/
- https://developer.mozilla.org/en-US/docs/Web/JavaScript
- https://hibernate.org/orm/
- https://reactjs.org/