PES UNIVERSITY

OOAD

6th Sem 'I' Section

Mini Project

HEALTHCARE WEBSITE

GIRISH G GONDA – PES1UG20CS552

D ADARSHA BHAT – PES1UG20CS544

GAGAN GOUTHAM – PES1UG20CS550

PRATEEK KESHRI – PES1UG20CS526

GIRISH G GONDA
D ADARSHA BHAT
GAGAN GOUTHAM
PRATEEK KESHRI

Project GitHub Repository Link

https://github.com/Girishg17/Health_website

Abstract

This report outlines the Object-Oriented Analysis and Design (OOAD) project for a healthcare web application. The project aims to develop a web application that allows patients to easily access healthcare services and providers. The app allows users to search for healthcare providers, book appointments, access medical records, and receive reminders for upcoming appointments. The project follows the OOAD principles of abstraction, encapsulation, inheritance, and polymorphism. We have used UML diagrams to represent the system's architecture and design and Entity Relationship diagrams for database design. The report provides a detailed explanation of the project's requirements, design, implementation, and testing, along with the challenges faced during the project.

Introduction

Motivation

A healthcare website that allows users to book appointments is an online platform where individuals can easily schedule a doctor's visit without having to physically visit the clinic or hospital. These websites usually provide a list of healthcare providers, their availability, and the types of services they offer. Users can select a suitable time slot and book an appointment based on their convenience. These websites also offer other features such as reminders, cancellation options, and rescheduling options to

ensure a hassle-free experience for the users. The ultimate goal of these websites is to provide convenient and accessible healthcare services to the users.

Why this project?

1. Convenience: By providing a platform that allows users to book appointments online, the process of scheduling a doctor's visit becomes much more convenient for patients. They can avoid the hassle of physically visiting the clinic or hospital and can schedule appointments from the comfort of their own homes.
2. Efficiency: With an online booking system, healthcare providers can manage their schedules more efficiently, reducing the chances of double-bookings or missed appointments.
3. Accessibility: An online booking system can make healthcare services more accessible to individuals who might have difficulty accessing care due to geographic or mobility constraints.
4. Improved patient experience: By making it easier for patients to book appointments, healthcare providers can improve the overall patient experience and build stronger relationships with their patients.

Overall, creating a healthcare website that allows users to book appointments can improve the efficiency, accessibility, and overall experience of healthcare services for both patients and healthcare providers.

.

Objectives and Goals:

The objective of a healthcare website that allows users to book appointments can be summarized as follows:

To provide an online platform for patients to easily and conveniently book appointments with healthcare providers, with the ultimate goal of improving the accessibility, efficiency, and overall experience of healthcare services.

In order to achieve this objective, the following goals can be identified:

1. Develop a user-friendly website interface that allows patients to easily search for healthcare providers and book appointments online.

2. Provide accurate and up-to-date information about healthcare providers, their specialties, availability, and services.

3. Ensure the security and privacy of patient information by implementing appropriate security measures, such as encryption and data backups.

4. Enable patients to manage their appointments, including rescheduling or cancelling appointments as necessary.

5. Develop a system to track and manage appointments, ensuring that healthcare providers can efficiently manage their schedules and avoid double-bookings or missed appointments.

By achieving these goals, the healthcare website can effectively meet its objective of improving the accessibility, efficiency, and overall experience of healthcare services for both patients and healthcare providers.

Background:

The background of the project titled HealthCare website was to design and develop software that would help users to book appointments easily.

Tools and Platforms:

Hardware Requirements:

- Server: The server should have sufficient processing power, memory, and storage capacity to handle the expected traffic and data volume.
- Database Server: Separate database server to store and manage the data generated. The database server should have enough storage capacity and processing power to handle the expected data volume.

- Networking Equipment: Networking equipment such as routers, switches, and firewalls connect the server and database server to the internet and enable communication between them.
  .

Software Requirements:

- Java Development Kit (JDK): The JDK is required to develop Java applications. It includes the
- Java Virtual Machine (JVM), compiler, and other tools needed to develop and run Java applications.
- Integrated Development Environment (IDE): Visual Studio provides code editing, debugging, and other development features that make it easier to develop Java applications.

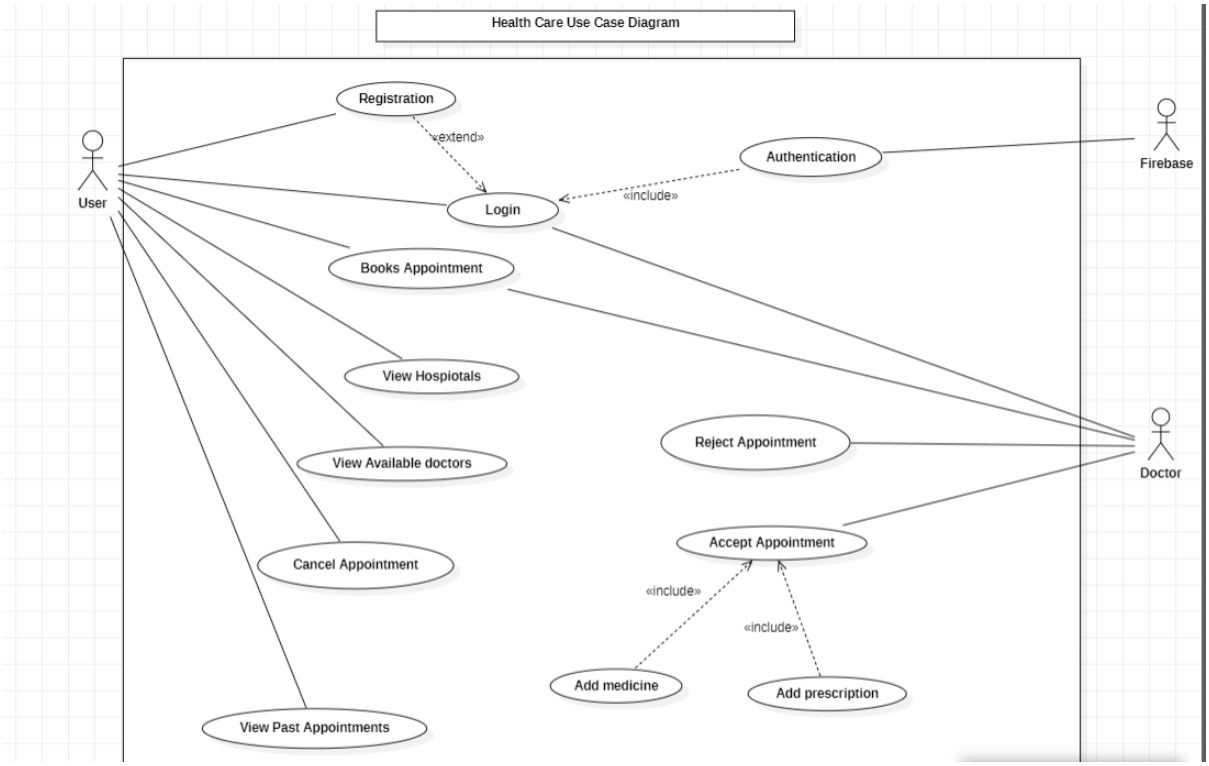
● Database Management System (DBMS): A DBMS such as MySQL or Firebase will be used to store and manage the data generated by the online Doctor Appointment System.

Future Scope:

- AI and ML technologies can be integrated to automate repetitive tasks, analyze user data, and improve the user experience.
- With the increasing use of mobile devices,healthcare can be optimized for mobile devices to make the process more convenient and accessible for all sections of society.

Use Case Diagram:

GIRISH G GONDA
D ADARSHA BHAT
GAGAN GOUTHAM
PRATEEK KESHRI

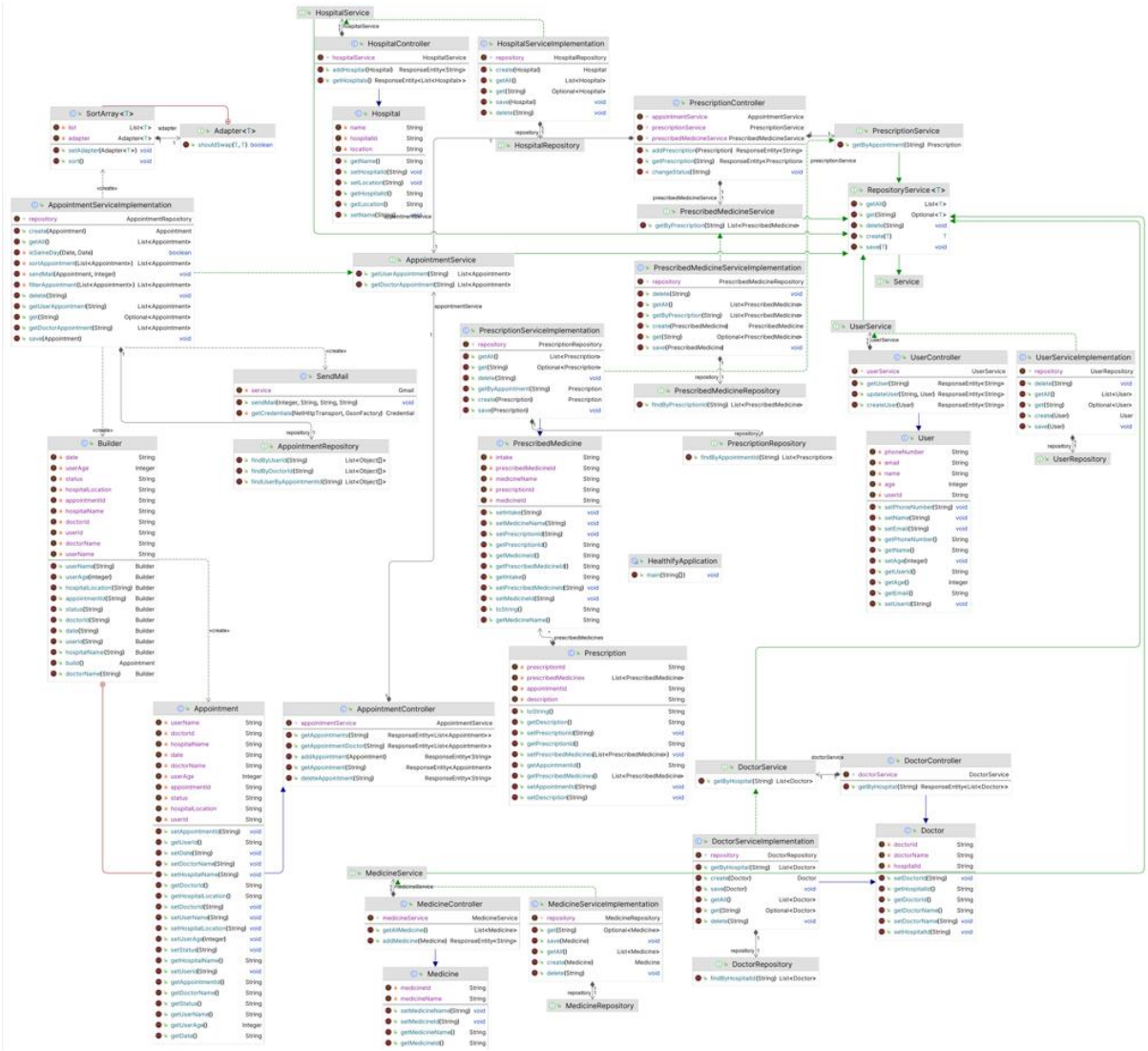


Class Diagram:

OOAD Mini-Project
HEALTHCARE WEBSITE

GIRISH G GONDA
D ADARSHA BHAT

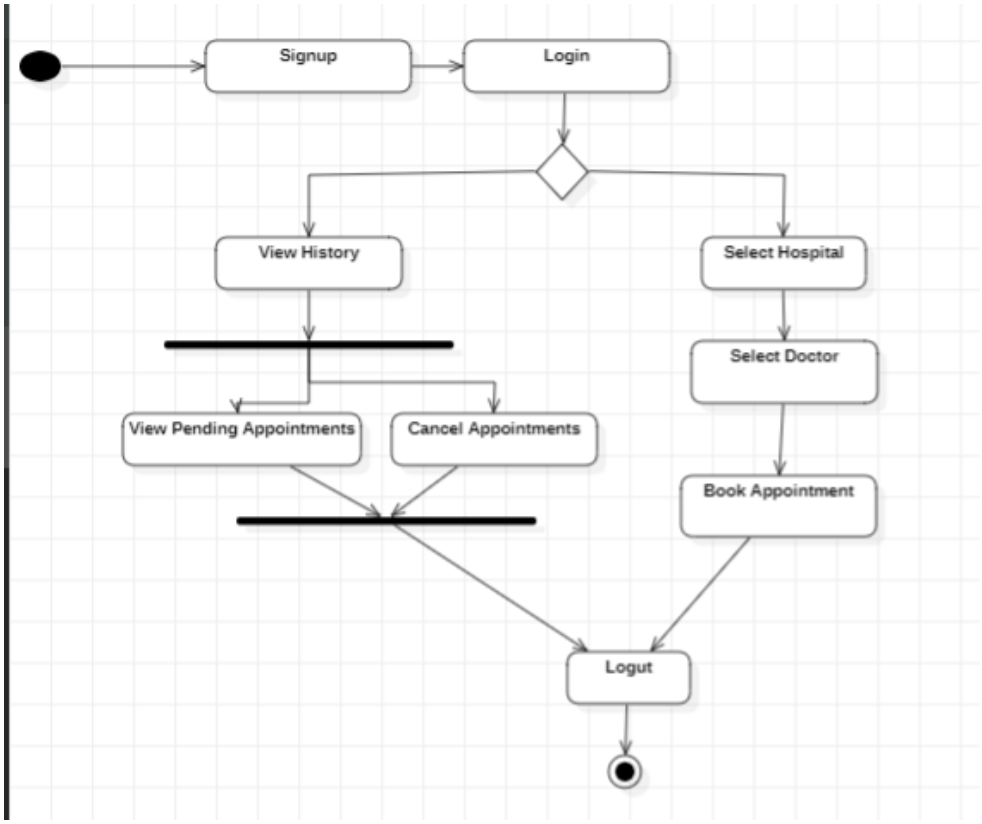
GAGAN GOUTHAM
PRATEEK KESHRI



GIRISH G GONDA
D ADARSHA BHAT
GAGAN GOUTHAM
PRATEEK KESHRI

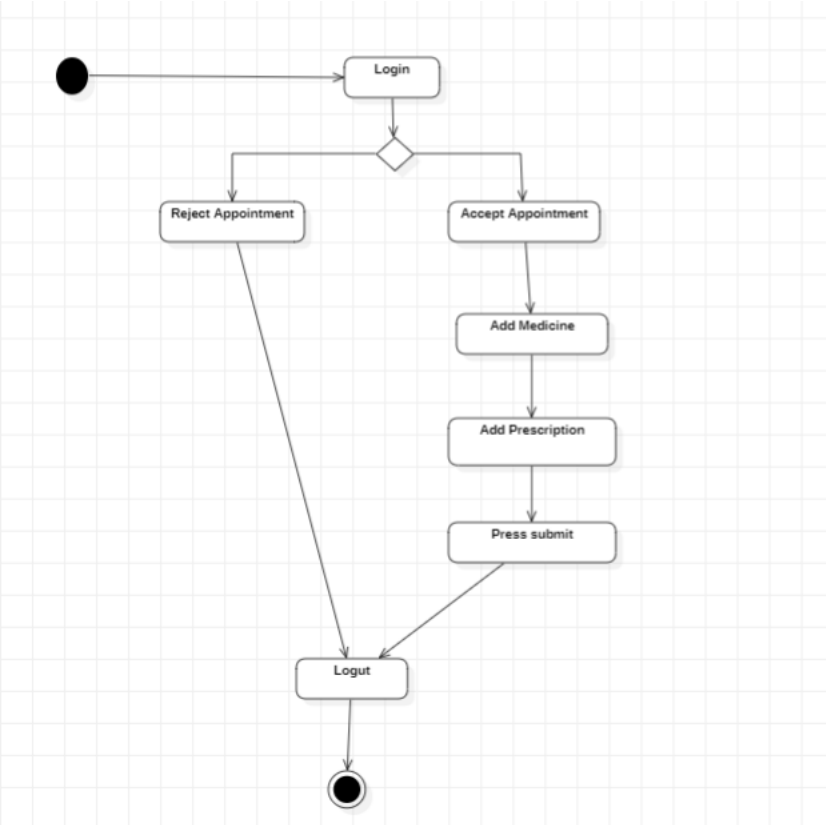
Activity Diagram:

Patient:



GIRISH G GONDA
D ADARSHA BHAT
GAGAN GOUTHAM
PRATEEK KESHRI

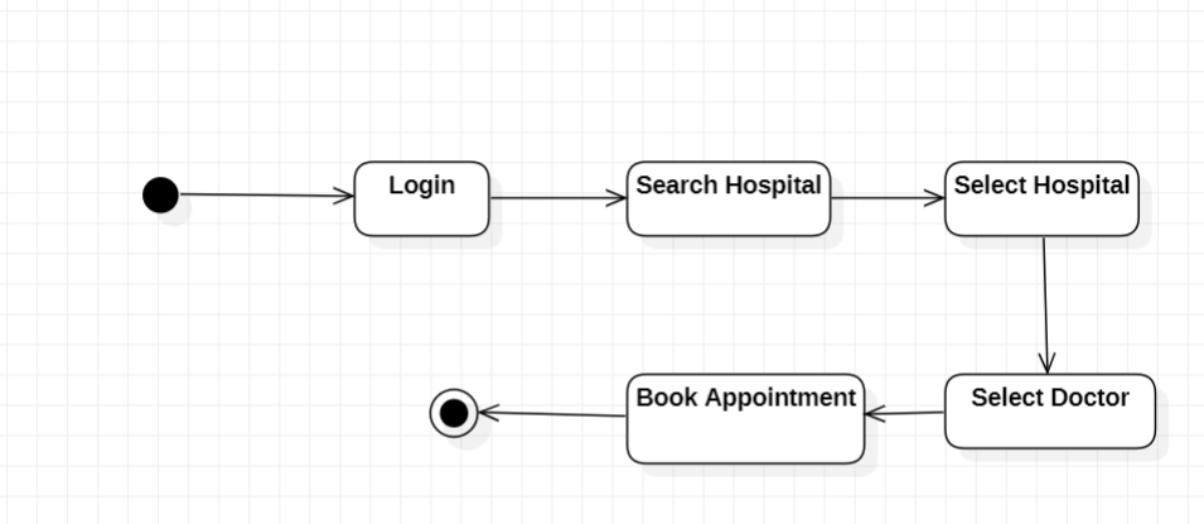
Doctor:



GIRISH G GONDA
D ADARSHA BHAT
GAGAN GOUTHAM
PRATEEK KESHRI

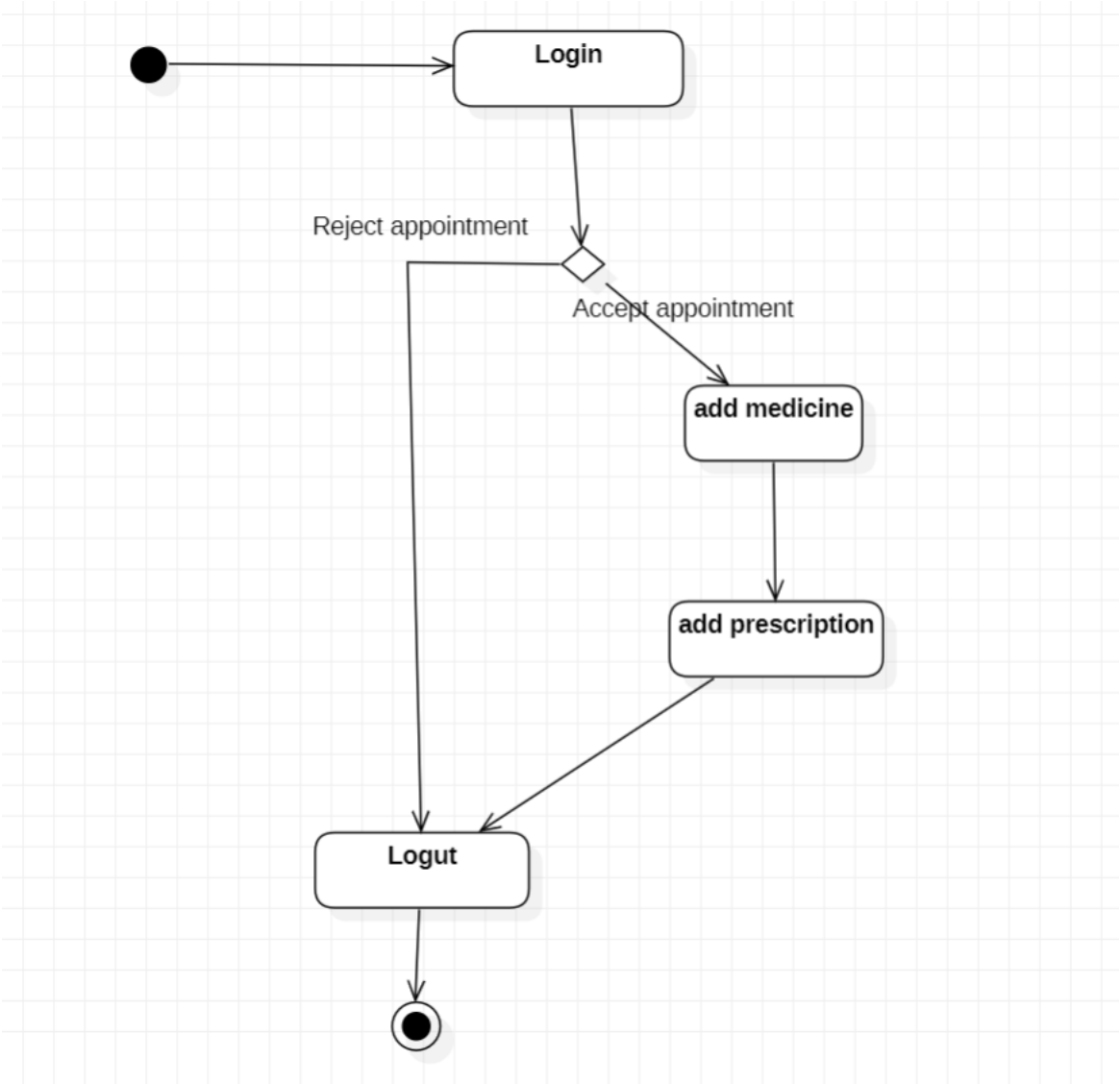
State Diagram

User:-



GIRISH G GONDA
D ADARSHA BHAT
GAGAN GOUTHAM
PRATEEK KESHRI

Doctor:



GIRISH G GONDA
D ADARSHA BHAT
GAGAN GOUTHAM
PRATEEK KESHRI

GRASP Patterns:

- **Information expert**: Each controller performs a specific job and has entire information about its particular job. So it follows an information expert design pattern
- **Creator**: Services are the creators of the models(multiple services).
- **Controller** - map the request from the view to the service and hence follow the controller design pattern.
- **Cohesion and coupling**: There is low coupling and high cohesion between controller classes. The operations that require access are passed through the controller.
- **Polymorphism:** The controller handles the incoming request from User,Doctor.

SOLID PRINCIPLES:

- SINGLE RESPONSIBILITY PRINCIPLE:
- OPEN CLOSE PRINCIPLE:
- INTERFACE SEGREGATION PRINCIPLE:

DESIGN PATTERNS:

These are the design patterns we will be implementing or using to some capacity .

Creational Design Pattern

- **Builder Pattern :**

- **FAÇADE**: This we haven't implemented for now. We can implement this pattern in between controllers and services to make them loosely couple with each other

Behavioural Design Pattern

- **Information Expert :** Here in our project we have implemented controllers in such a way that each controllers are responsible for mapping specific type of job, so that it will have more information about specific job. Here all the controllers are Information experts.

TEAM MEMBER ROLES:

GIRISH G GONDA

- **Worked on the Signup controller for user validation and user classification.**
- **Worked on building user page.**

D ADARSHA BHAT

- **Worked on building doctor page.**
- **Worked on Database Util for database connection.**

GAGAN GOUTHAM

- **Worked on the interaction between the doctor and the user.**
- **Worked on Firebase Authentication**

PRATEEK KESHRI

- **Worked on interface which integrates all the parts of project.**

JOINT RESPONSIBILITY

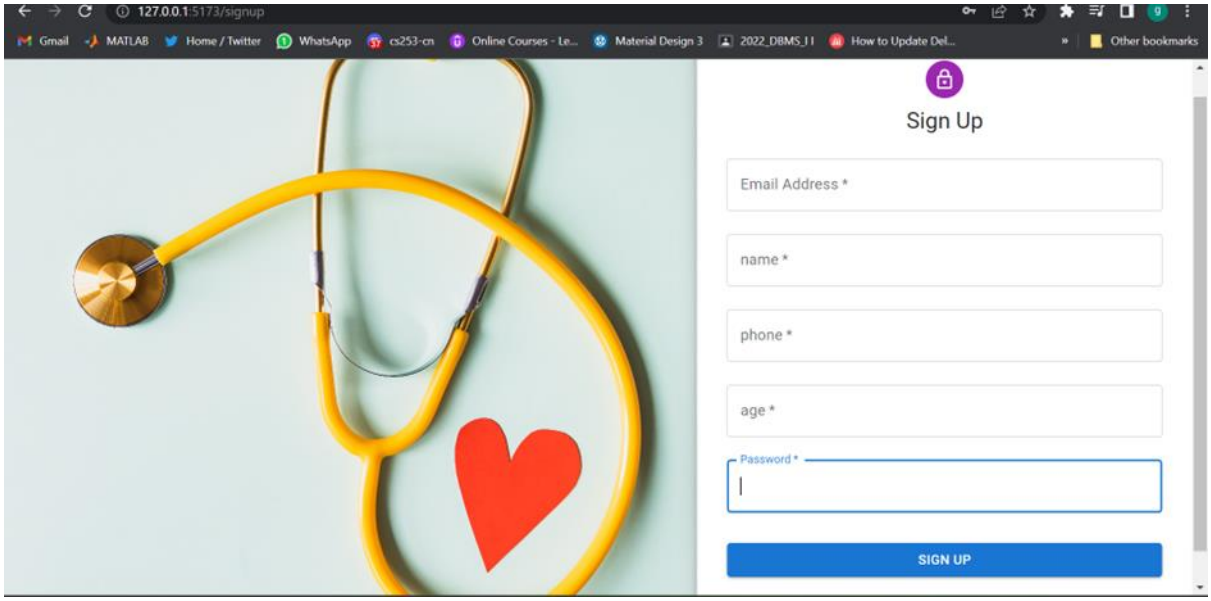
Worked on multiple controllers like

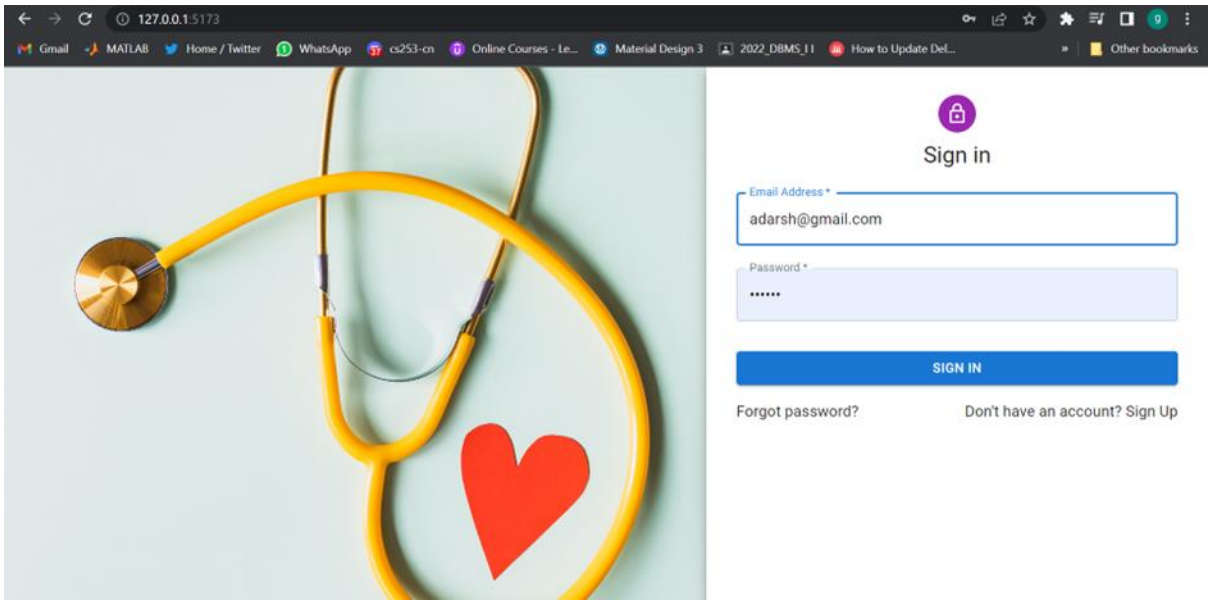
- **General controller**

GIRISH G GONDA
D ADARSHA BHAT
GAGAN GOUTHAM
PRATEEK KESHRI

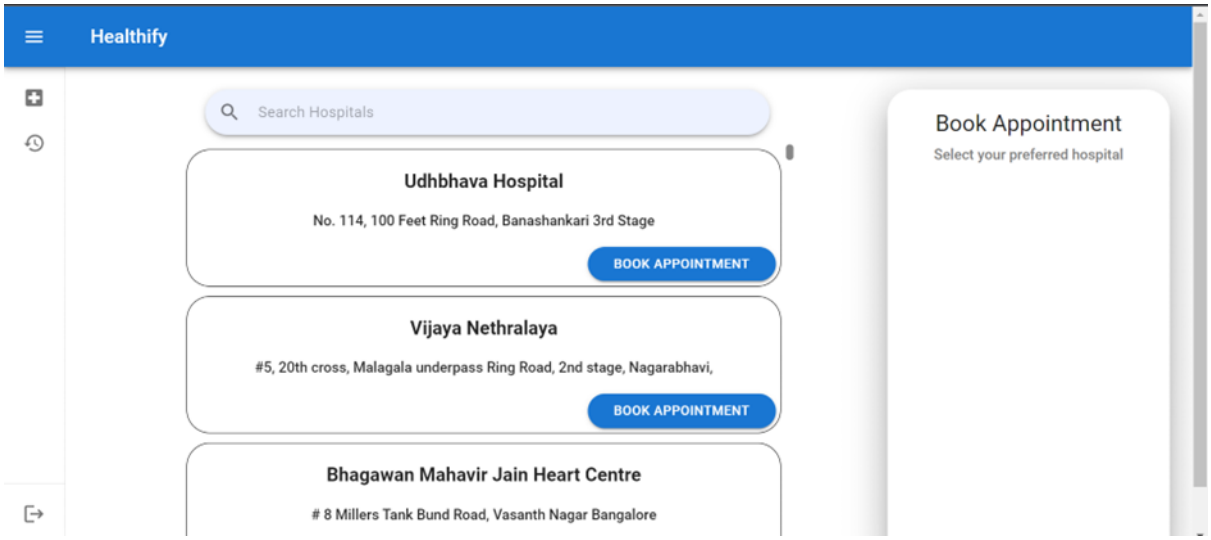
DEMONSTRATION

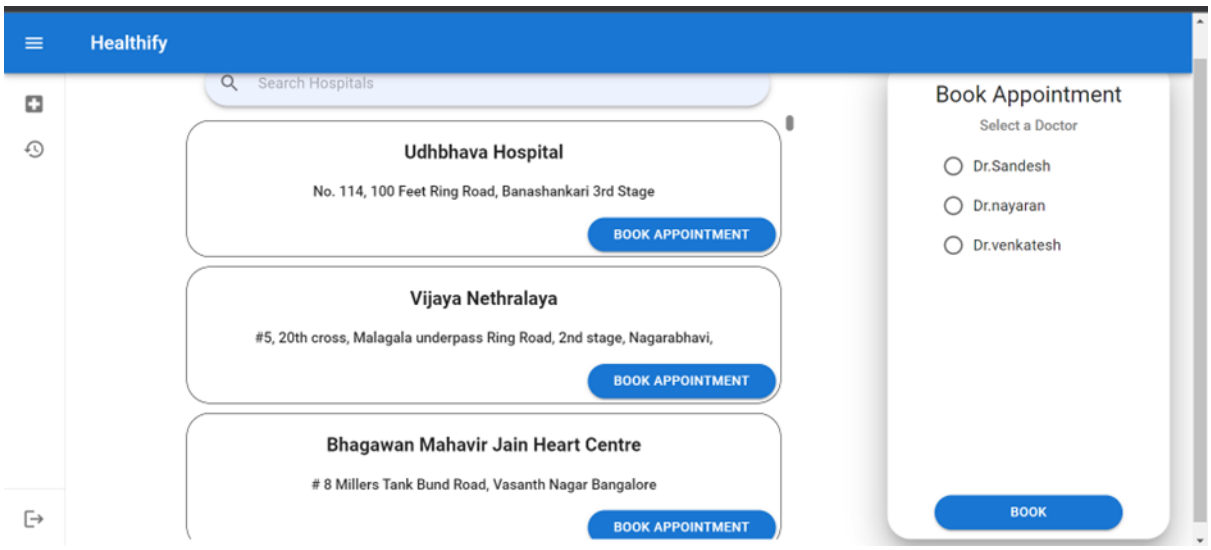
User



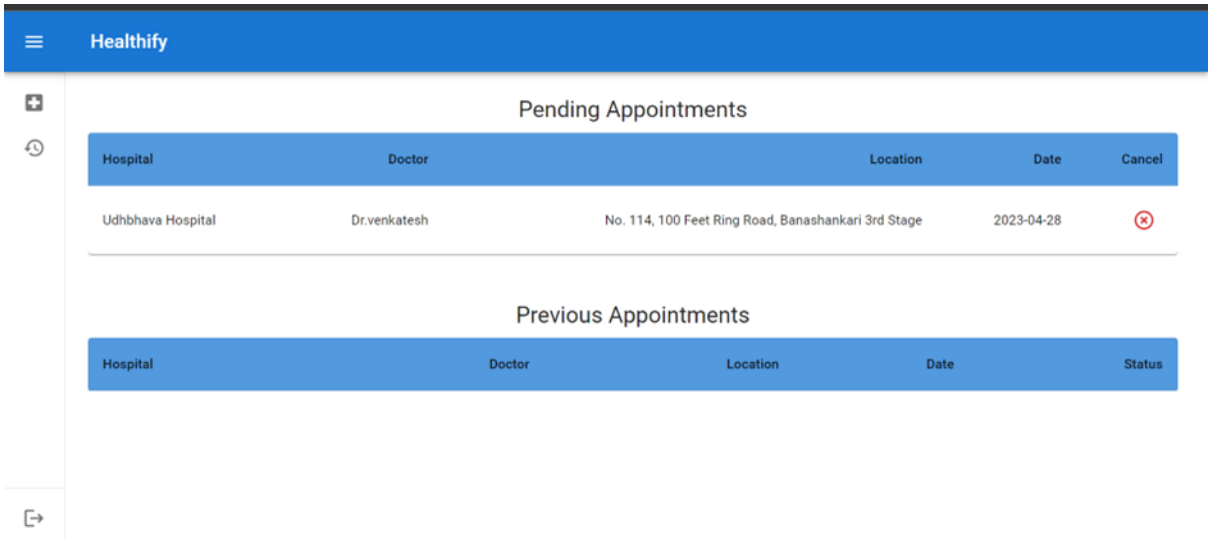


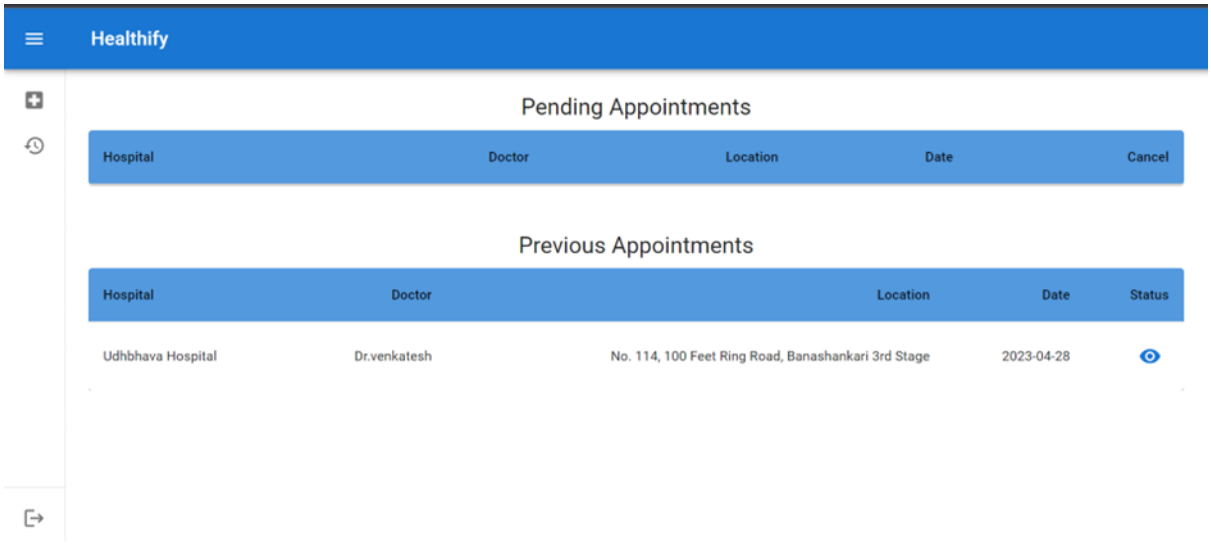
GIRISH G GONDA
D ADARSHA BHAT
GAGAN GOUTHAM
PRATEEK KESHRI



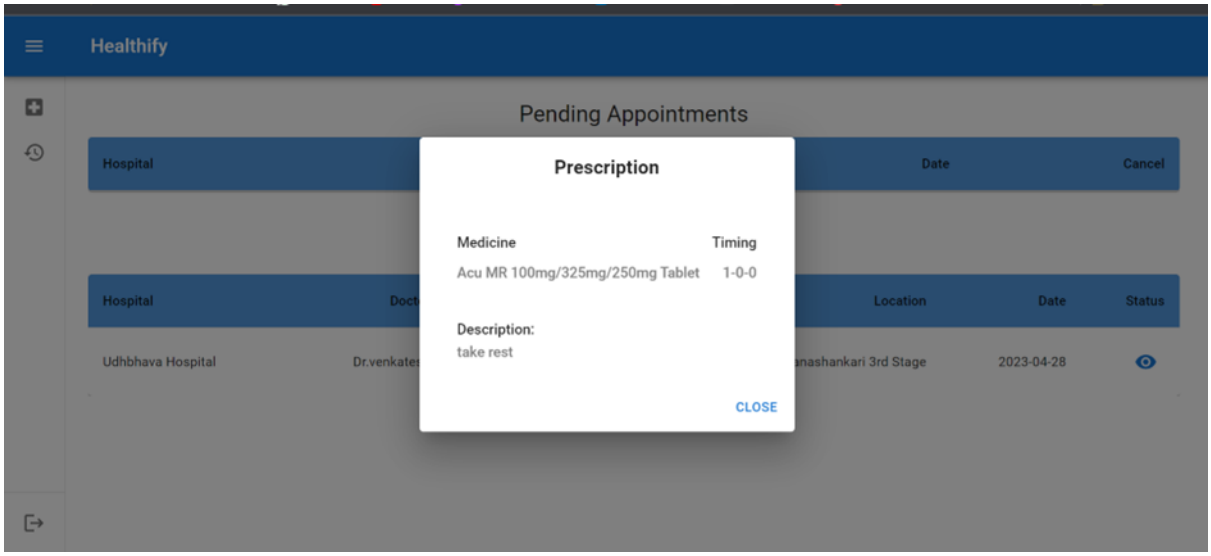


GIRISH G GONDA
D ADARSHA BHAT
GAGAN GOUTHAM
PRATEEK KESHRI

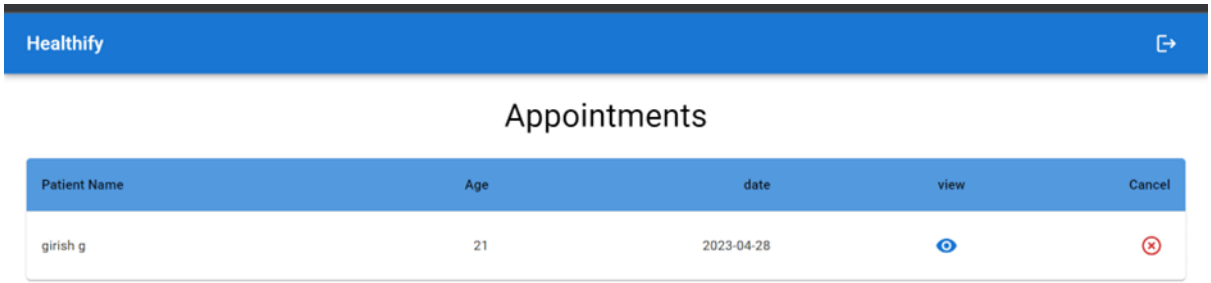




GIRISH G GONDA
D ADARSHA BHAT
GAGAN GOUTHAM
PRATEEK KESHRI

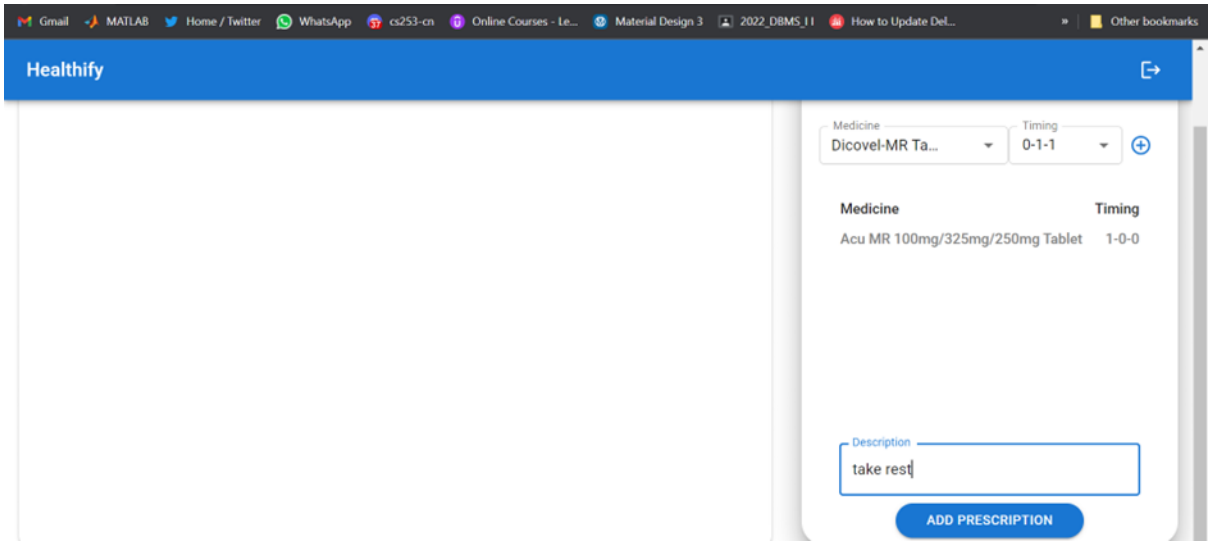


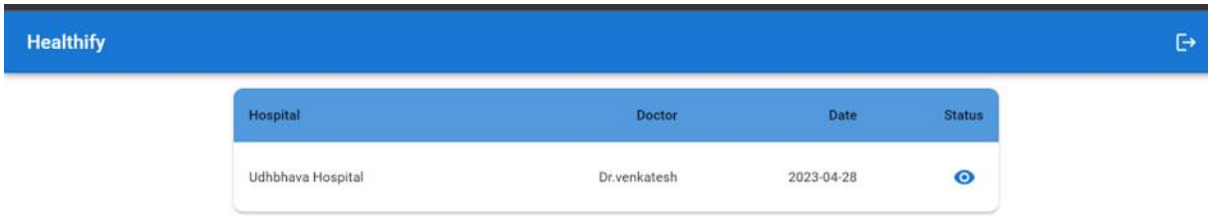
Doctor

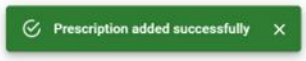


GIRISH G GONDA
D ADARSHA BHAT

GAGAN GOUTHAM
PRATEEK KESHRI

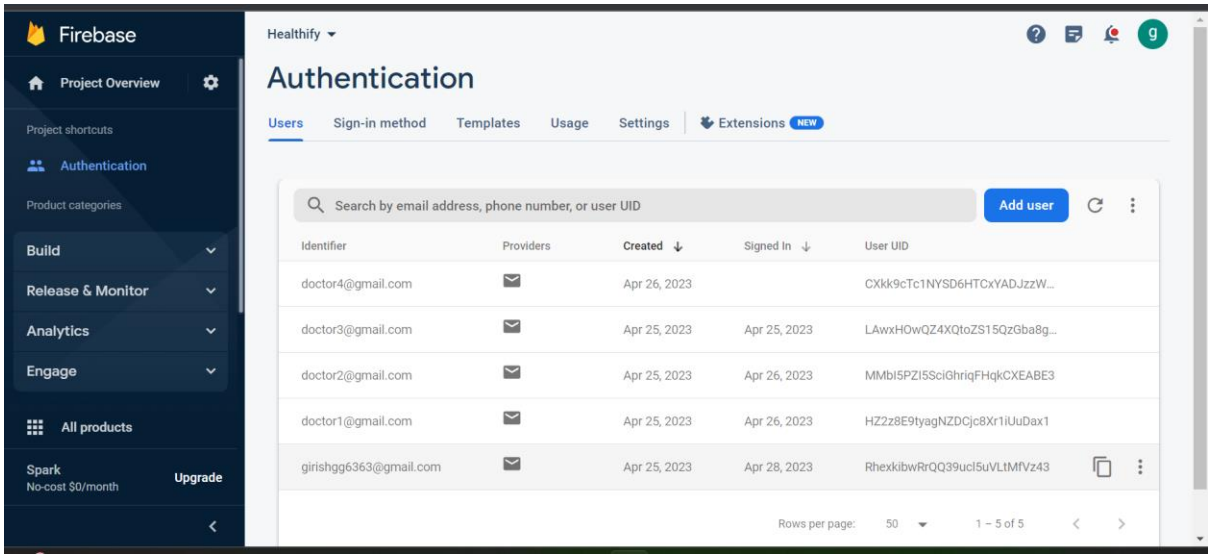




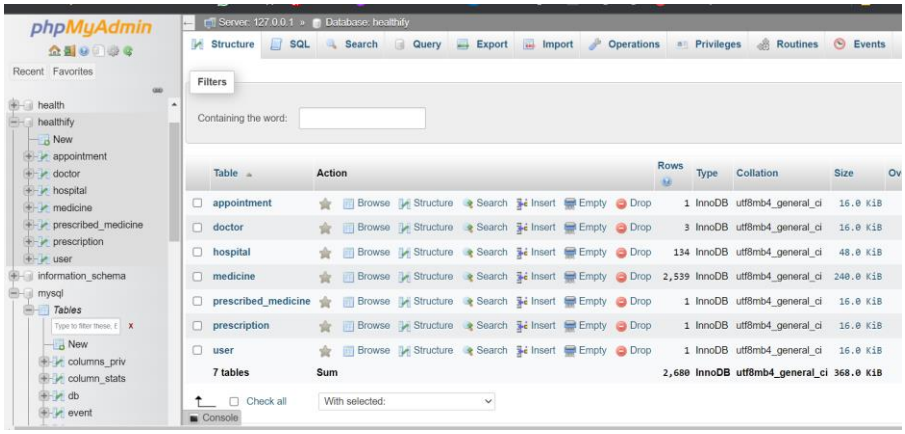


GIRISH G GONDA
D ADARSHA BHAT
GAGAN GOUTHAM
PRATEEK KESHRI

Firebase



MySQL



GIRISH G GONDA
D ADARSHA BHAT
GAGAN GOUTHAM
PRATEEK KESHRI

APPENDIX

Spring Boot Documentation:

https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/

Chat GPT:

https://chat.openai.com/