Patient Medicine and Appointment System

1, Project Overview

- **Title of the Project:** Patient Medicine and Appointment System.
- **Package Name:** MiniPorjectTwo.
- **Purpose**: This app streamlines healthcare by allowing patients to easily manage their medications and book appointments with their doctors. It aims to improve patient adherence and access to timely medical care.
- **Architecture:** MVC design pattern.
- **Core Functionality**: The system allows to perform CRUD operations on Patient, Doctor, Appointment, Patient's Appointment and Medications records.
- **Technologies Used:**
  - Front-end - React Js.
  - Back-end - Spring Boot(Java).
  - Database - MySQL.
  - API Documantation: Postman.

2, Database Structure

Appointment Manager Entity:

Column	Type	Description
appointment_id	String(PK)	Appointment unique identifier
appointment_date	Date	Doctor available date
appointment_start_time	Time	Appointment start time
appointment_end_time	Time	Appointment end time
doctor_id	String	Doctor unique identifier
doctor_name	String	Name of the doctor
doctor_education	String	Doctor education
doctor_specializedfield	String	Doctor specialize

Doctor Entity:

Column	Type	Description
doctor_id	String(PK)	Doctor unique identifier
doctor_name	String	Name of the doctor
doctor_education	String	Doctor education
doctor_specializedfield	String	Doctor specialize

Medication Manager Entity:

Column	Type	Description
doctor_id	String	Doctor unique identifier
doctor_name	String	Name of the doctor
patient_id	String	Patient unique identifier
patient_name	String	Name of the patient
medicine_id	String(PK)	Medicine unique identifier
patientappointment_id	String	Patient book unique identifier
appointment_datetime	Date/Time	Patient book date/time
medicine_morning	Boolean	Medication for morning
medicine_afternoon	Boolean	Medication for afternoon
medicine_night	Boolean	Medication for night
medicine_afterfood	Boolean	Food intake Before/After
medicine_days	Integer	Days of medicine intake

Patient Appointment Entity:

Column	Type	Description
doctor_id	String	Doctor unique identifier
doctor_name	String	Name of the doctor
patient_id	String	Patient unique identifier
patient_name	String	Name of the patient
patientappointment_id	String(PK)	Patient book unique identifier
appointment_datetime	Date/Time	Patient book date/time
doctor_education	String	Doctor education
doctor_specializedfield	String	Doctor specialize
patient_age	Integer	Patient Age
patient_illness	String	Patient illness

Patient Entity:

Column	Type	Description
patient_id	String(PK)	Patient unique identifier
patient_name	String	Name of the patient
patient_age	Integer	Patient Age
patient_illness	String	Patient illness
patient_contact	String	Patient mobile number
patient_email	String	Patient email information
patient_place	String	Patient place

Role Entity:

Column	Type	Description
id	Integer(PK)	Role unique identifier
name	String	Role name

Role Entity:

Column	Type	Description
id	Integer(PK)	User unique identifier
username	String	Username for login
password	String	Password for login

userid	String	User unique identifier

User Role Entity:

Column	Type	Description
user_id	Integer(PK)	User Id unique identifier
roles_id	Integer	Role unique identifier

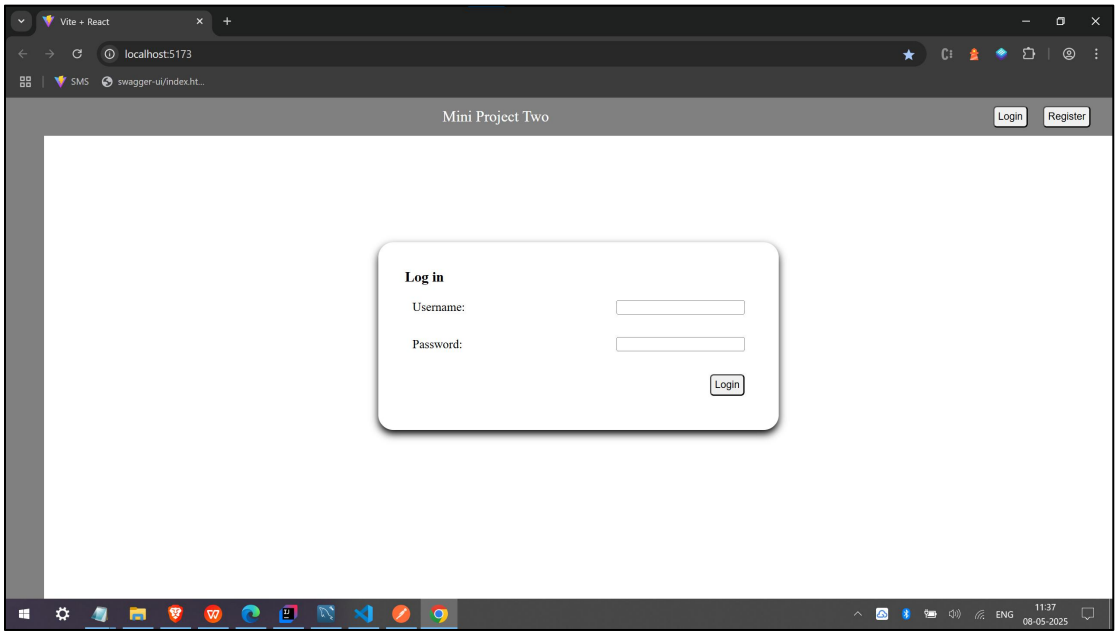
3, Frontend

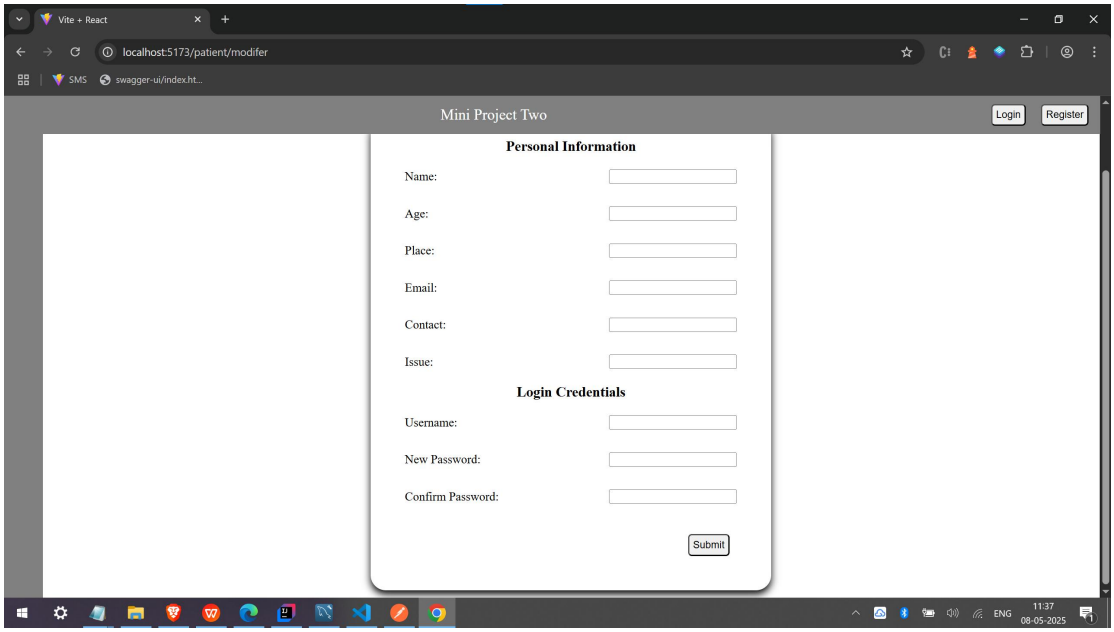
React is a popular JavaScript library for building dynamic and interactive user interfaces. Its component-based architecture allows for efficient development and code re-usability.

- React JS Package:
  - Axios.
  - Date-fns.
  - Jwt-decode.
  - React-route-dom.
  - Sonner.

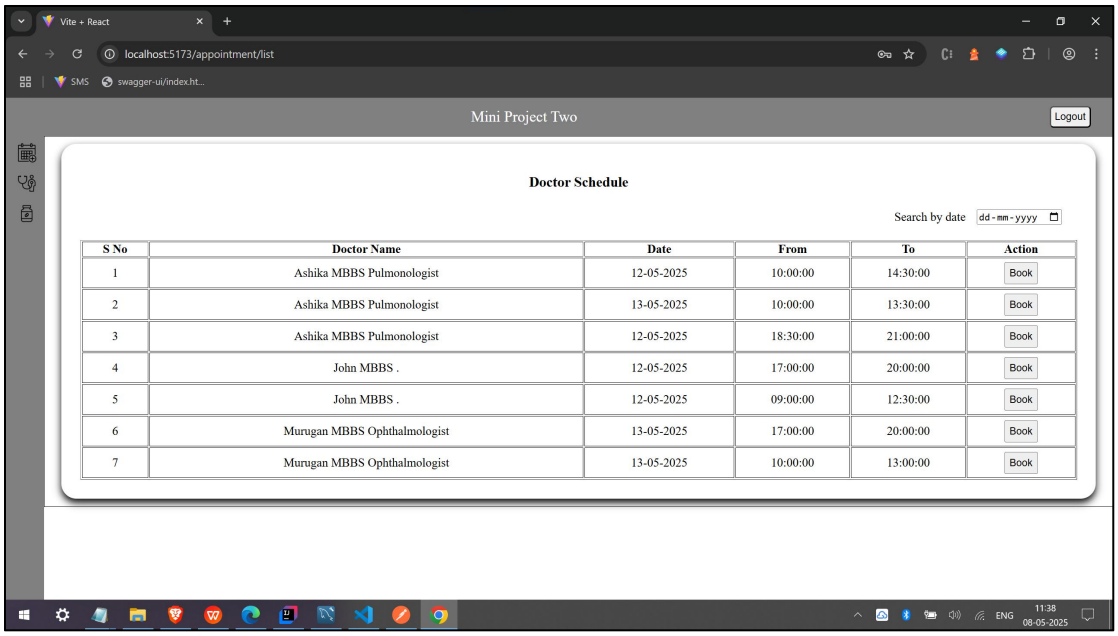
Frontend Screenshots:

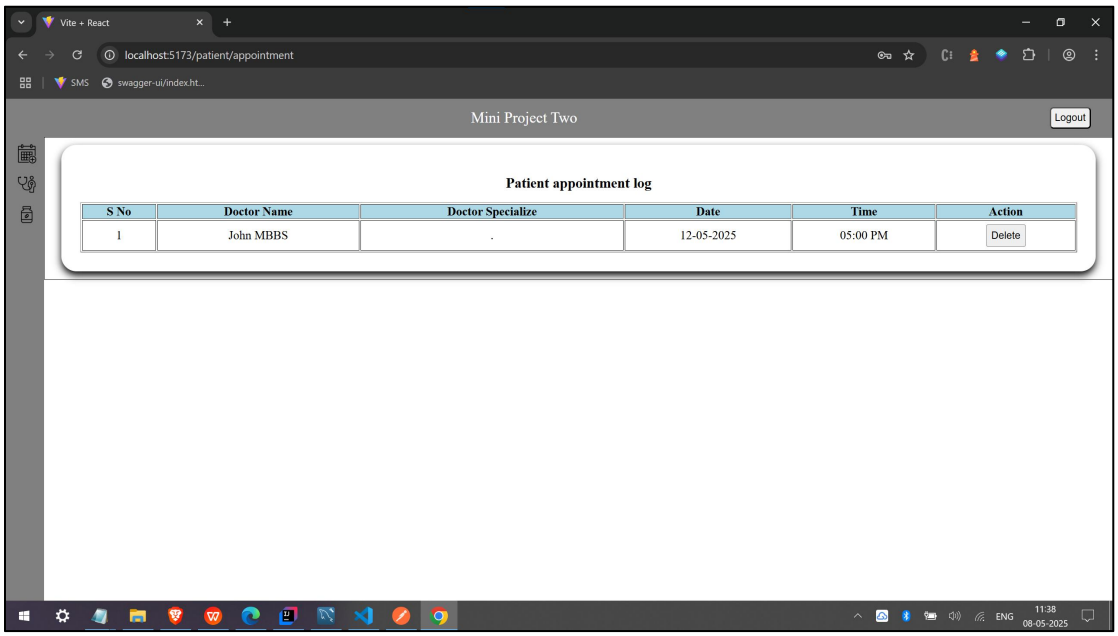
NON-AUTH:

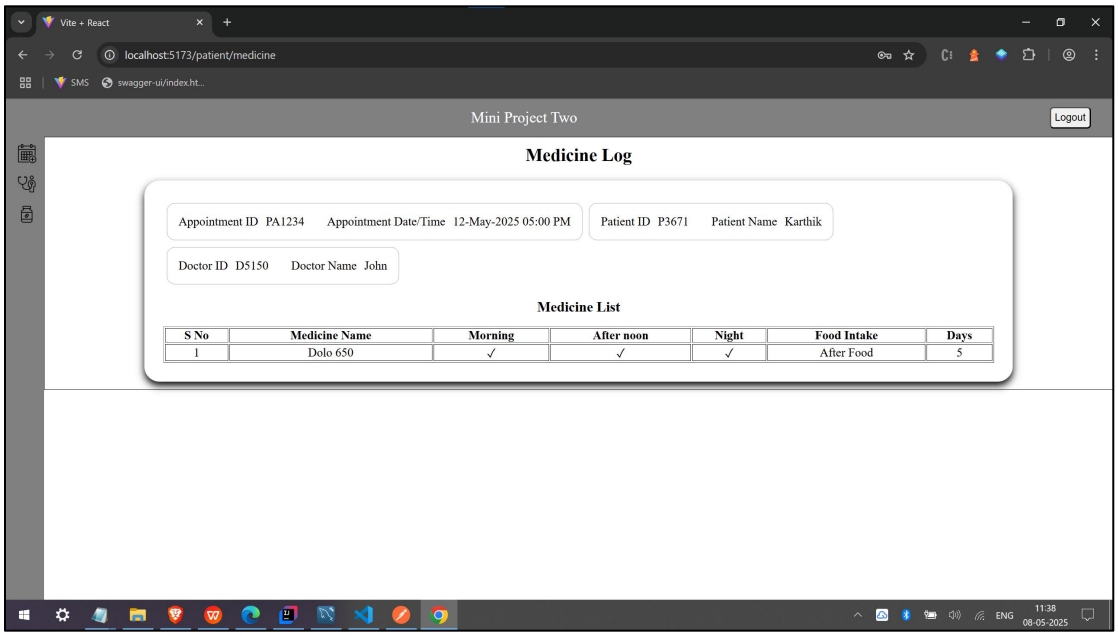



Vite + React
localhost:5173/patient/modifer
Mini Project Two
Login
Register
Personal Information
Name:
Age:
Place:
Email:
Contact:
Issue:
Login Credentials
Username:
New Password:
Confirm Password:
Submit

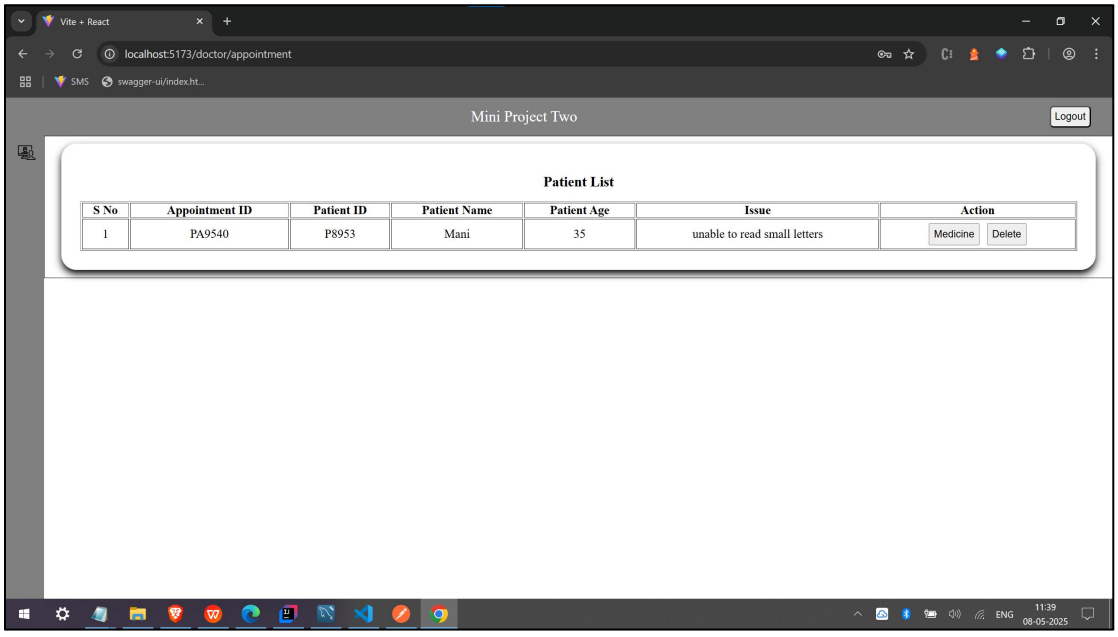
PATIENT:

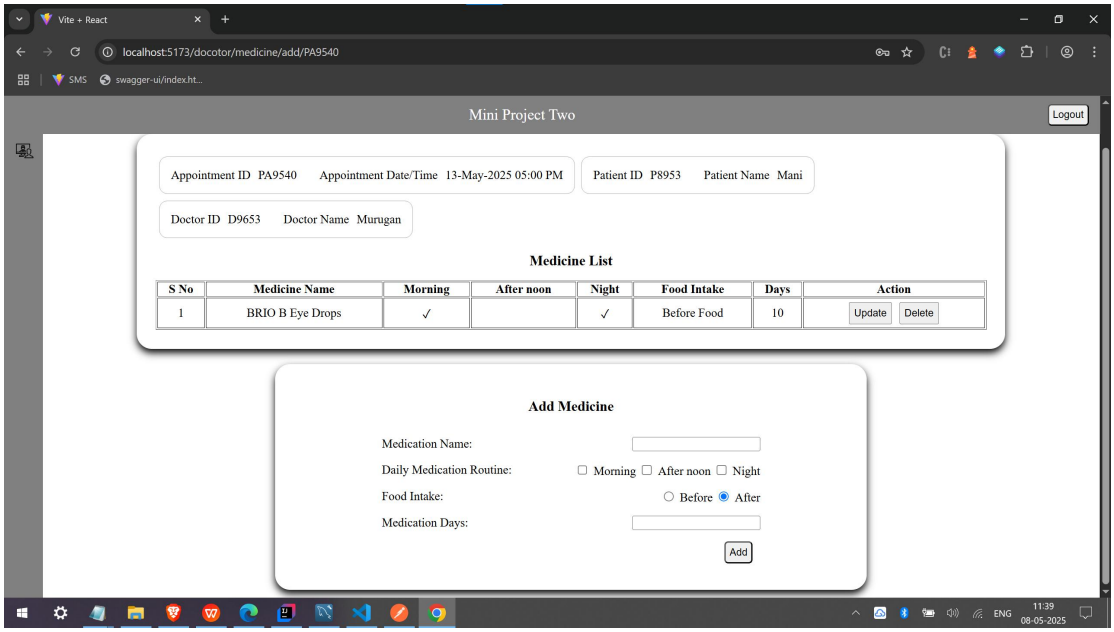

Vite + React
localhost:5173/appointment/list
Mini Project Two
Logout
Doctor Schedule
Search by date
dd-mm-yyyy
S No | Doctor Name | Date | From | To | Action
1 | Ashika MBBS Pulmonologist | 12-05-2025 | 10:00:00 | 14:30:00 | Book
2 | Ashika MBBS Pulmonologist | 13-05-2025 | 10:00:00 | 13:30:00 | Book
3 | Ashika MBBS Pulmonologist | 12-05-2025 | 18:30:00 | 21:00:00 | Book
4 | John MBBS . | 12-05-2025 | 17:00:00 | 20:00:00 | Book
5 | John MBBS . | 12-05-2025 | 09:00:00 | 12:30:00 | Book
6 | Murugan MBBS Ophthalmologist | 13-05-2025 | 17:00:00 | 20:00:00 | Book
7 | Murugan MBBS Ophthalmologist | 13-05-2025 | 10:00:00 | 13:00:00 | Book


Vite + React
localhost:5173/patient/appointment
Mini Project Two
Logout
Patient appointment log
S No | Doctor Name | Doctor Specialize | Date | Time | Action
1 | John MBBS | . | 12-05-2025 | 05:00 PM | Delete

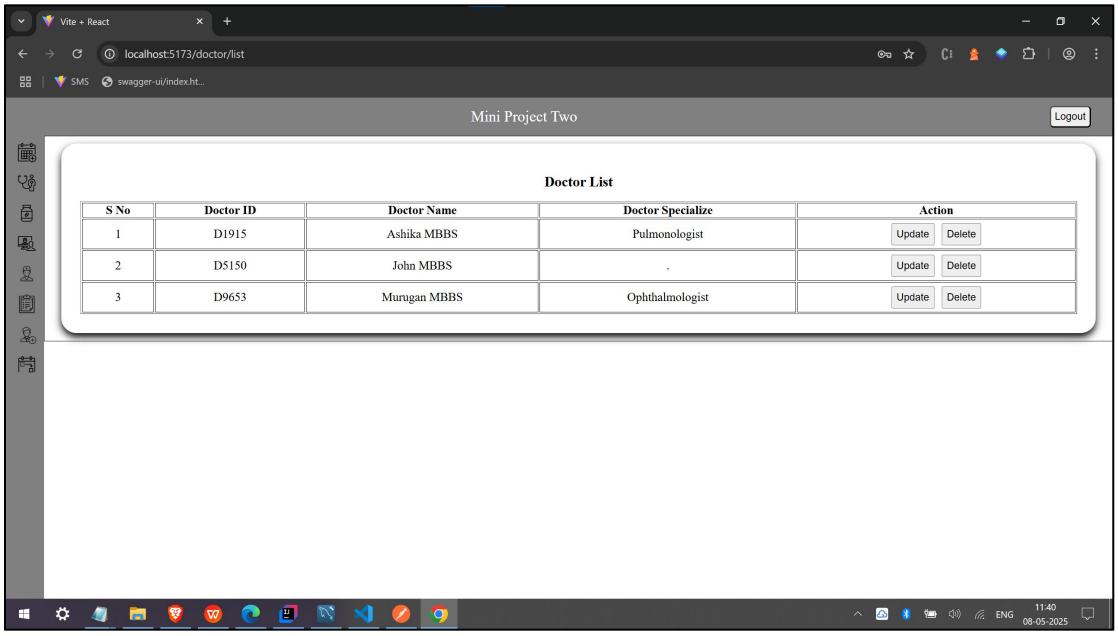

Vite + React
localhost:5173/patient/medicine
Mini Project Two
Logout
Medicine Log
Appointment ID PA1234 Appointment Date/Time 12-May-2025 05:00 PM Patient ID P3671 Patient Name Karthik
Doctor ID D5150 Doctor Name John
Medicine List
S No | Medicine Name | Morning | After noon | Night | Food Intake | Days
1 | Dolo 650 | ✓ | ✓ | ✓ | After Food | 5

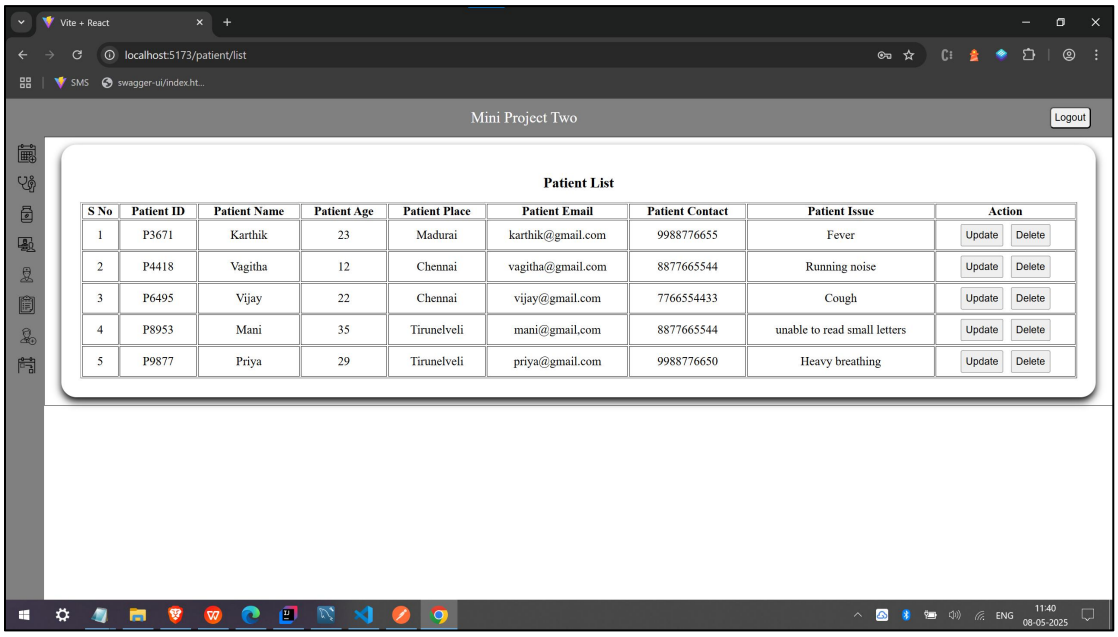
Doctor:

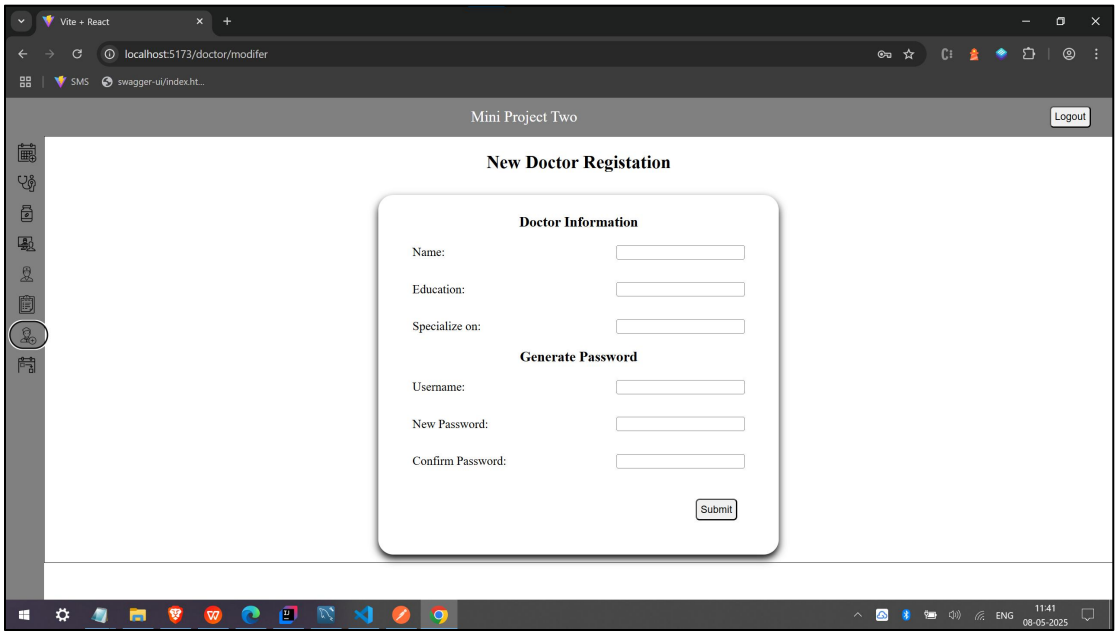

Vite + React
localhost:5173/doctor/appointment
Mini Project Two
Logout
Patient List
S No | Appointment ID | Patient ID | Patient Name | Patient Age | Issue | Action
1 | PA9540 | P8953 | Mani | 35 | unable to read small letters | Medicine Delete

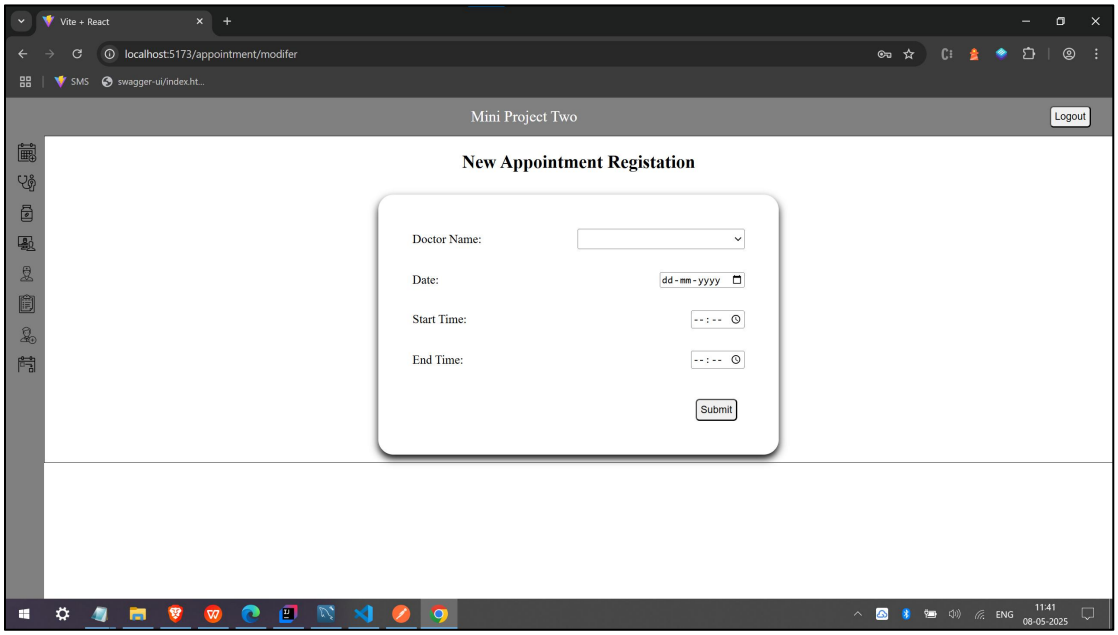

Vite + React
localhost:5173/docotor/medicine/add/PA9540
Mini Project Two
Logout
Appointment ID PA9540 Appointment Date/Time 13-May-2025 05:00 PM
Patient ID P8953 Patient Name Mani
Doctor ID D9653 Doctor Name Murugan
Medicine List
S No | Medicine Name | Morning | After noon | Night | Food Intake | Days | Action
1 | BRIO B Eye Drops | ✓ | | ✓ | Before Food | 10 | Update Delete
Add Medicine
Medication Name:
Daily Medication Routine: Morning After noon Night
Food Intake: Before After
Medication Days:
Add

ADMIN:


Vite + React
localhost:5173/doctor/list
Mini Project Two
Logout
Doctor List
S No | Doctor ID | Doctor Name | Doctor Specialize | Action
1 | D1915 | Ashika MBBS | Pulmonologist | Update Delete
2 | D5150 | John MBBS | . | Update Delete
3 | D9653 | Murugan MBBS | Ophthalmologist | Update Delete


Vite + React
localhost:5173/patient/list
Mini Project Two
Logout
Patient List
S No | Patient ID | Patient Name | Patient Age | Patient Place | Patient Email | Patient Contact | Patient Issue | Action
1 | P3671 | Karthik | 23 | Madurai | karthik@gmail.com | 9988776655 | Fever | Update Delete
2 | P4418 | Vagitha | 12 | Chennai | vagitha@gmail.com | 8877665544 | Running noise | Update Delete
3 | P6495 | Vijay | 22 | Chennai | vijay@gmail.com | 7766554433 | Cough | Update Delete
4 | P8953 | Mani | 35 | Tirunelveli | mani@gmail.com | 8877665544 | unable to read small letters | Update Delete
5 | P9877 | Priya | 29 | Tirunelveli | priya@gmail.com | 9988776650 | Heavy breathing | Update Delete





4, Backend (Spring Boot)

- **Spring Dependencies(Backend/Java):**
  - Spring Web.
  - Spring Data JPA.
  - MySQL Driver.
  - Spring Boot DevTools.
  - Lombok.
  - Validation.
  - Spring Security.
  - Swagger.
  - Jjwt-impl.
  - Jjwt-jackson.
  - Jjwt-api.

- **Controller Layer:**
  - AppointmentManagerController
  - DoctorController
  - MedicationManagerController
  - PatientAppointmentController
  - PatientController
  - UserController

- **Database Interaction:**
  Used JPA (Java Persistence API) for connect spring into database.

- **Service Layer:**
  Encapsulated business logic in service classes

  - AppointmentManagerService
  - DoctorService
  - MedicationManagerService
  - PatientAppointmentService
  - PatientService

- **Global Exception Handling:**
  Used @ControllerAdvice to handle exceptions like ResourceNotFoundException.

- **Application.properties**

```
#Custom port
server.port = 3000

#db config
spring.datasource.url = jdbc:mysql://localhost:3310/miniprojecttwo
spring.datasource.username = root
spring.datasource.password = root
spring.datasource.driver.class-name = com.mysql.cj.jdbc.Driver

#Hiberate config
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.hibernate.format_sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect

logging.level.org.springframework.security=DEBUG
```