

**UE22CS352B - Object Oriented Analysis & Design**

Mini Project Report

Healthcare Appointment Management System

***Submitted by:***

***SHEIK GOUSE PASHA: PES1UG22CS556  
 SHRINIVAS T G: PES1UG22CS584  
 VIKASH DUTTA: PES1UG23CS838***

|  |  |
| --- | --- |
|  |  |

Semester 6 Section B

# BHARGAVI MOKASHI

**January - May 2025**

##### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING

##### PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013) 100ft Ring Road, Bengaluru – 560 085, Karnataka, India

##### Problem Statement:

##### Managing doctor appointments manually can be inefficient, error-prone, and time-consuming for hospitals and clinics. Patients often face challenges in booking, rescheduling, or tracking appointments, while admins and doctors struggle with organizing schedules and accessing patient history. This project aims to develop a centralized Healthcare Appointment System to streamline the entire appointment lifecycle for admins, doctors, and patients

**Key Features:**

 **Admin Dashboard** to manage doctor schedules and monitor all appointments.

 **Doctor Availability Management** to define working hours and limit bookings per day.

 **Patient Portal** to book, cancel, or reschedule appointments conveniently.

 **Appointment Confirmation System** managed by doctors for better control.

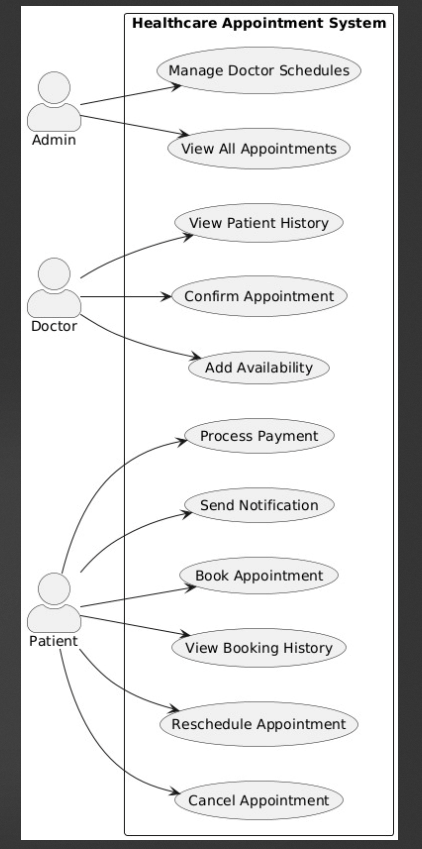
 **Booking History Tracking** for patients to view upcoming appointments.

 **Specialization-Based Doctor Filtering** to help patients choose the right doctor.

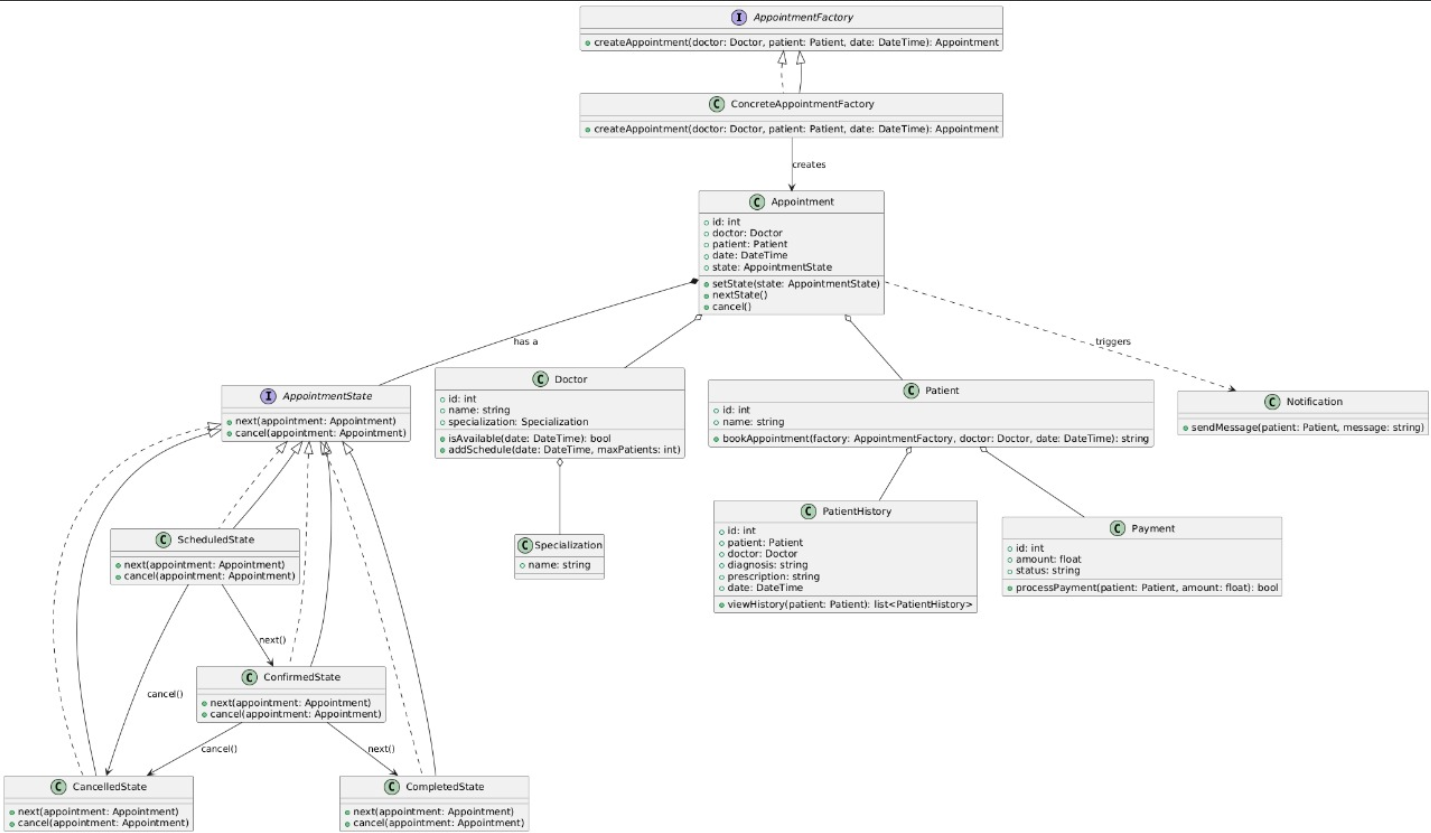
 **Date-Based Doctor Availability Search** to streamline booking.

## Models:

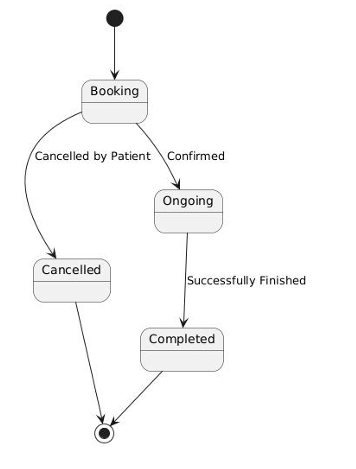
Use Case Diagram:



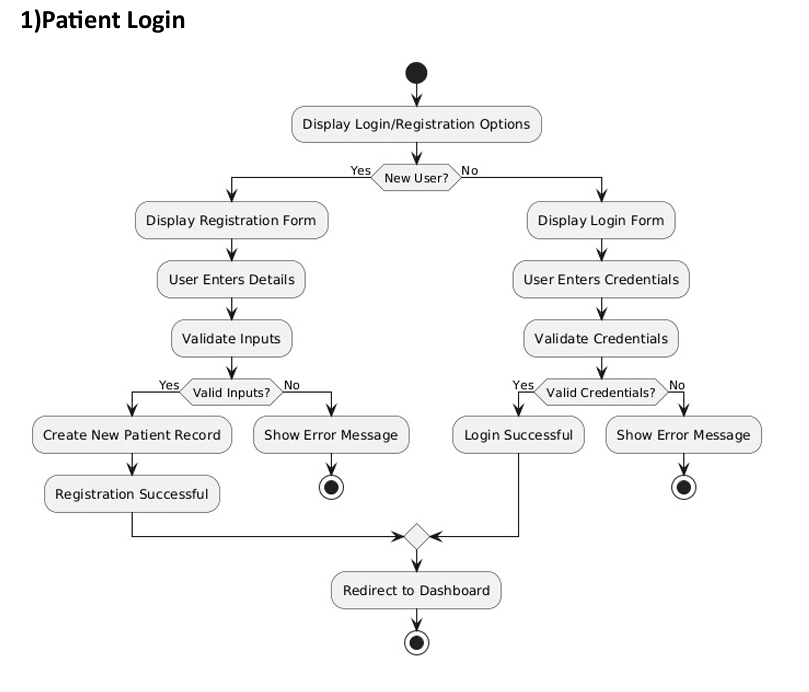
**Class Diagram:**

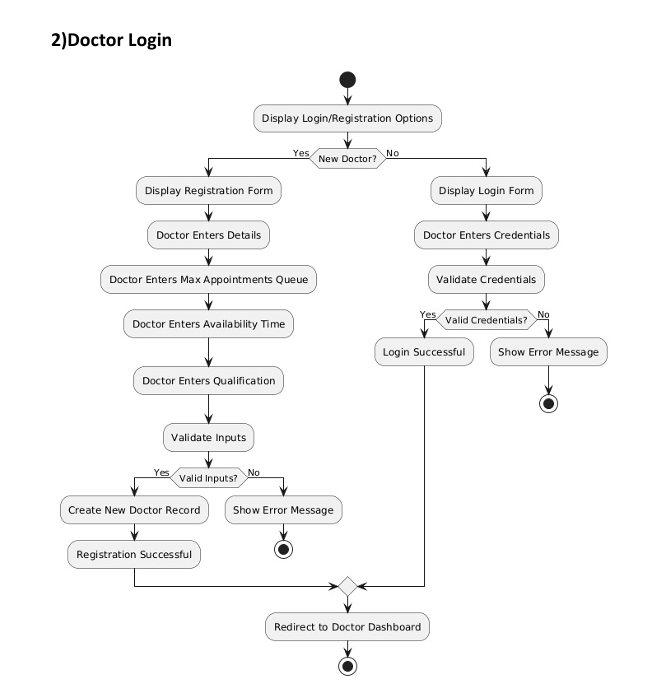
****

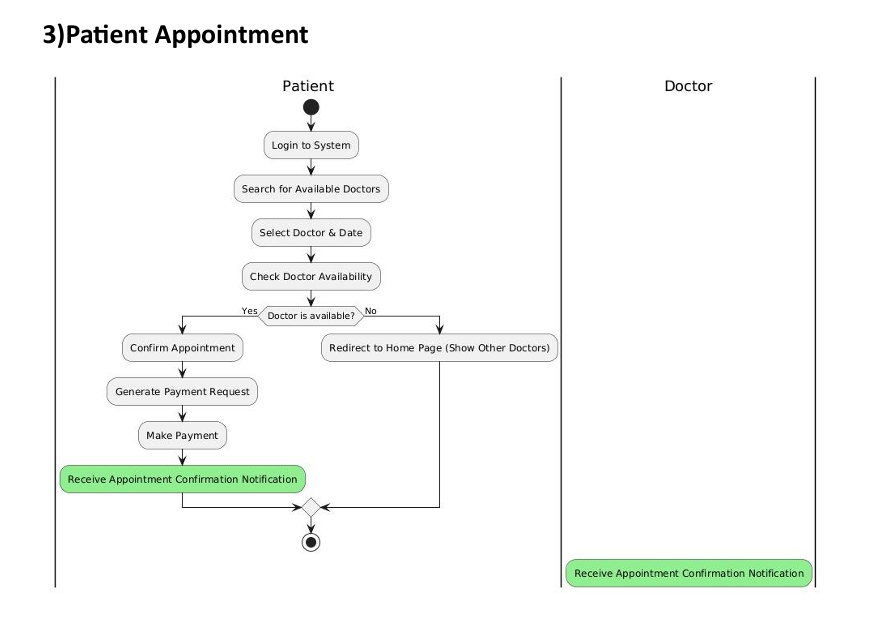
# State Diagram:

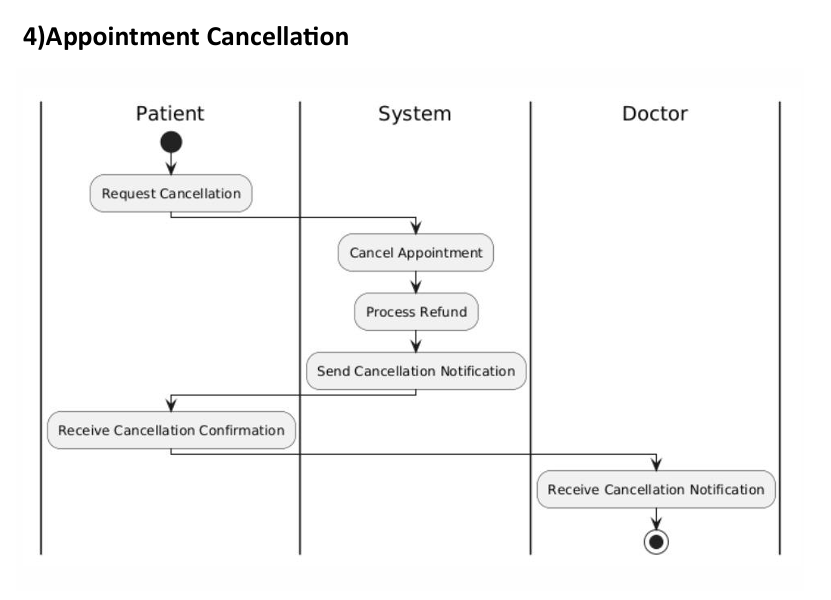
****

**Activity Diagrams:**









Architecture Patterns, Design Principles, and Design Patterns:

# Architecture Patterns

# Model – View – Controller Pattern (MVC)

# Design Principles

# Open – Close principle

# Interface Segregation principle

# Single Responsibility principle

# Design Patterns

# Strategy pattern

# Factory pattern

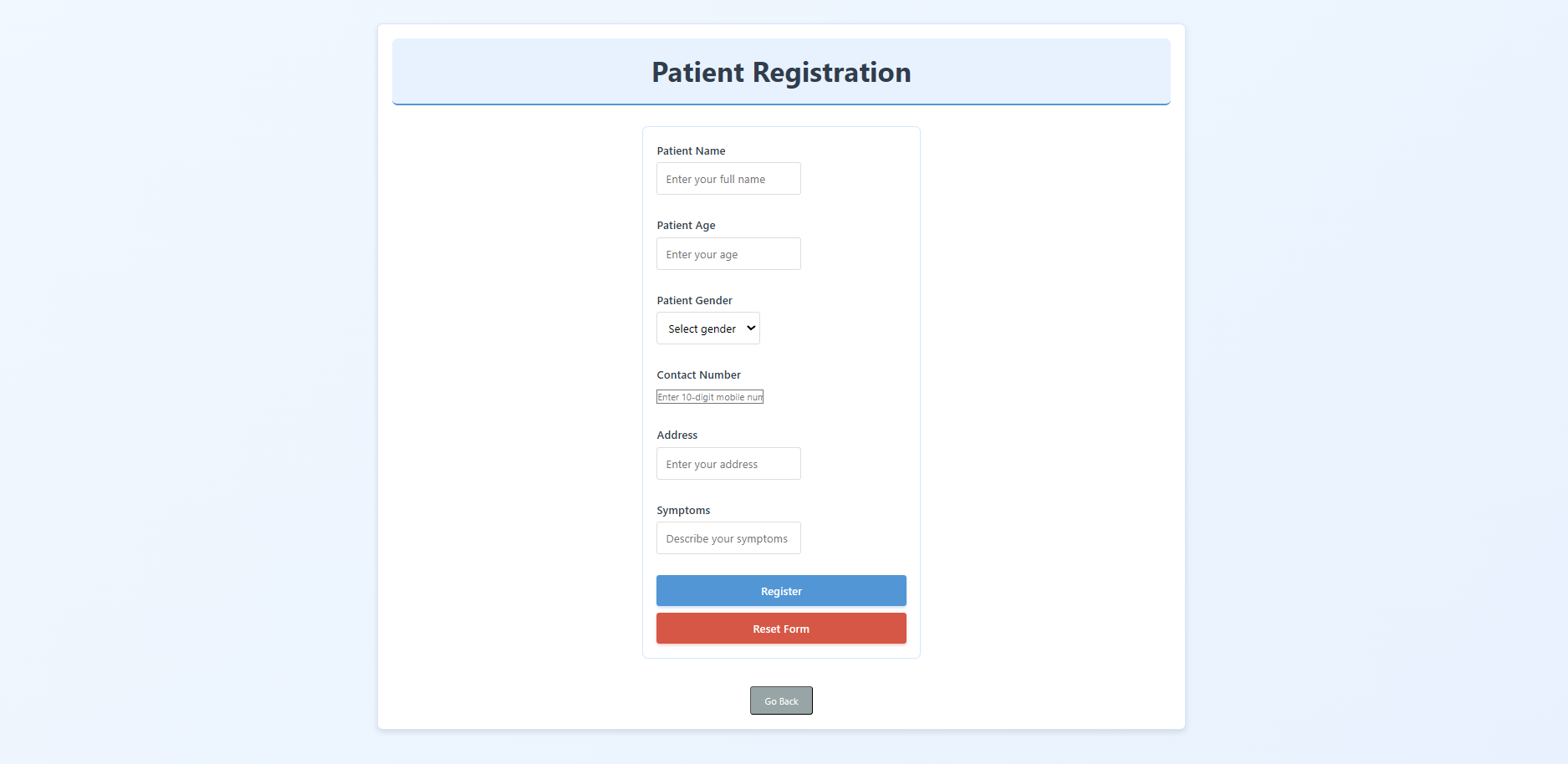
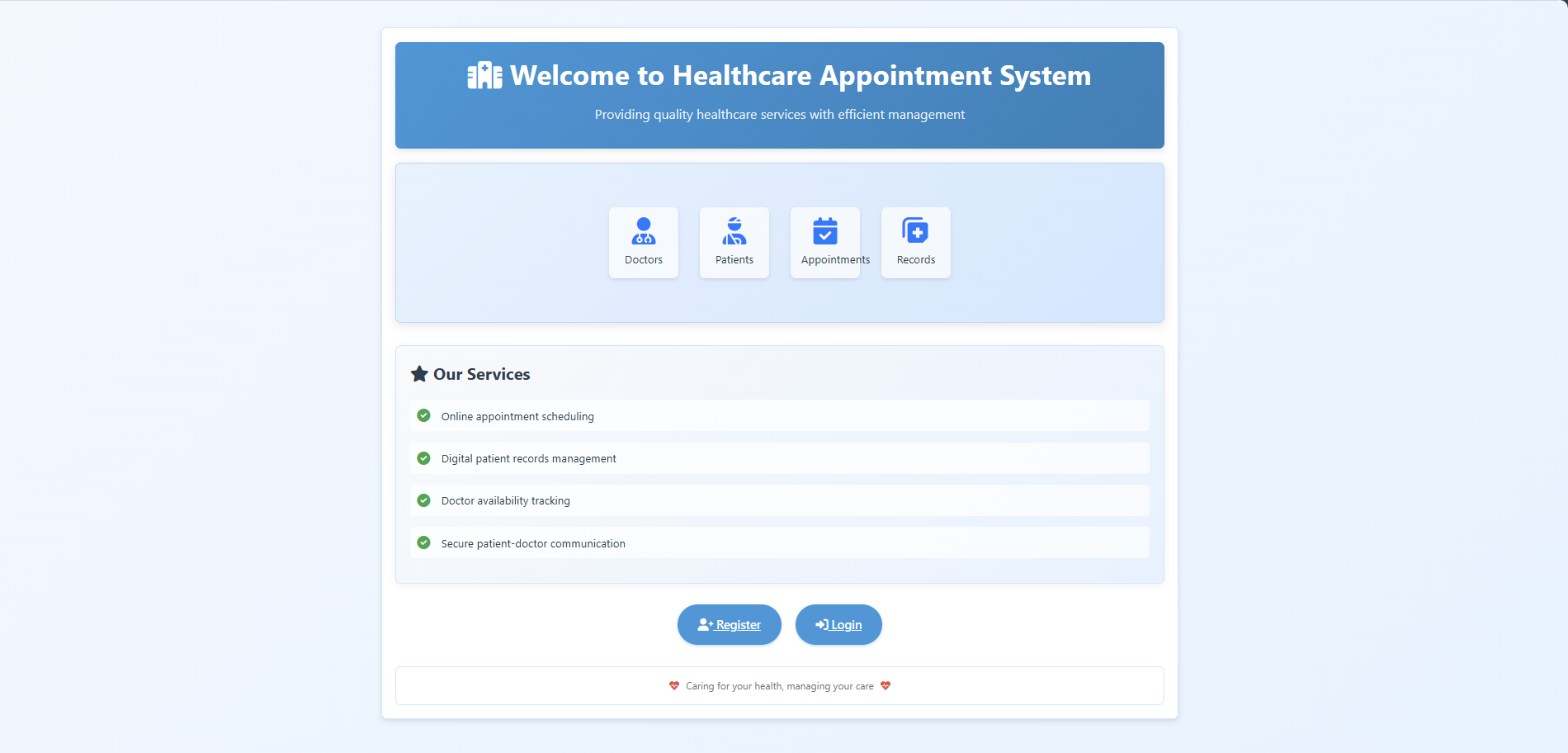
# Dependency Injection pattern

### Github link to the Codebase:

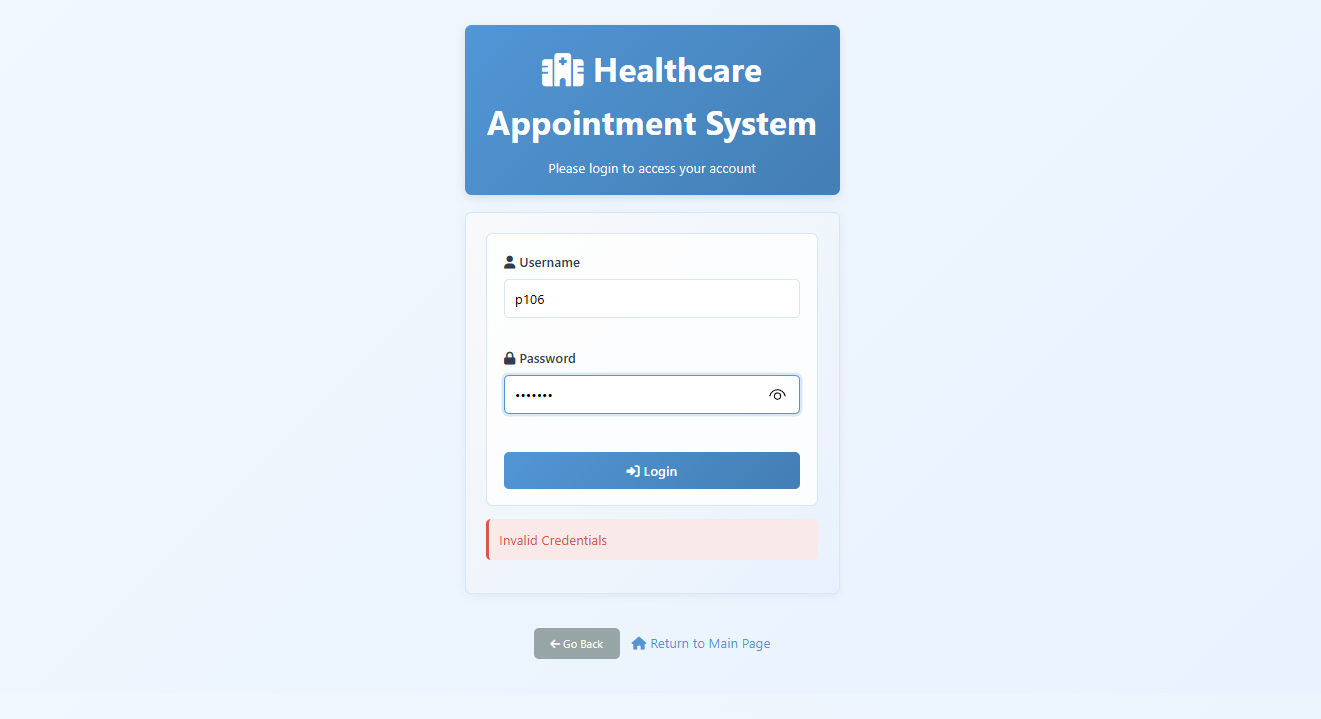
<https://github.com/Abhinandan-1008/Healthcare-Appointment-Management-System>

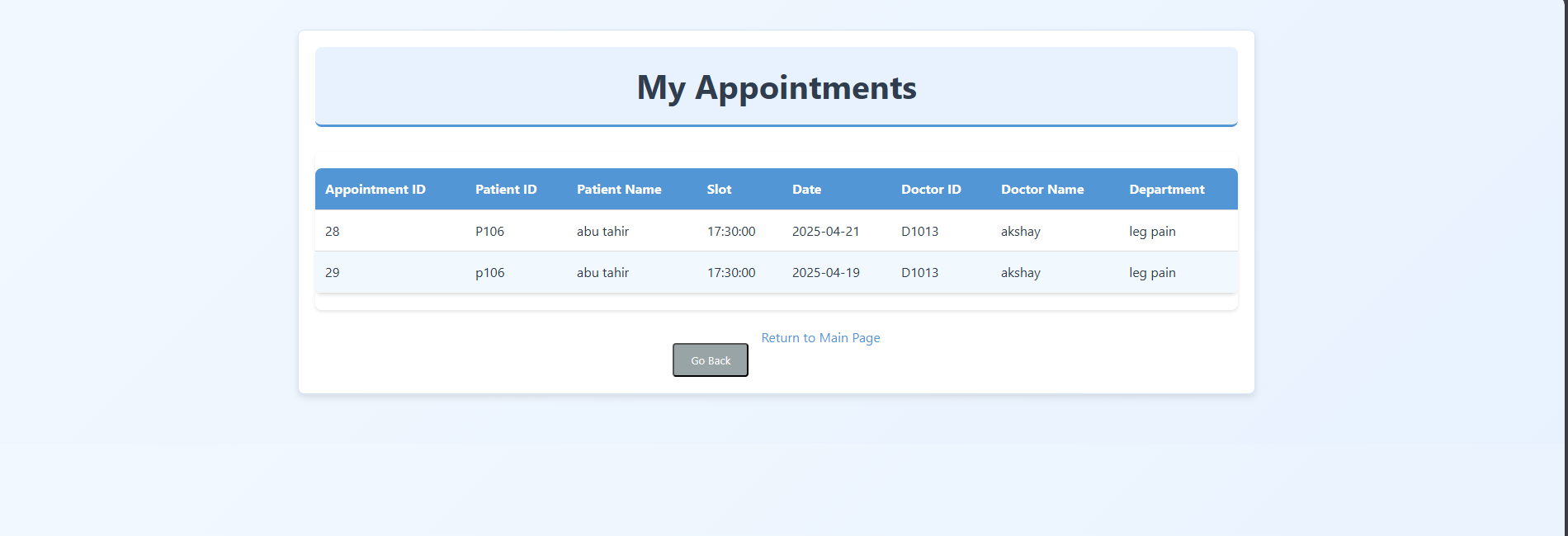
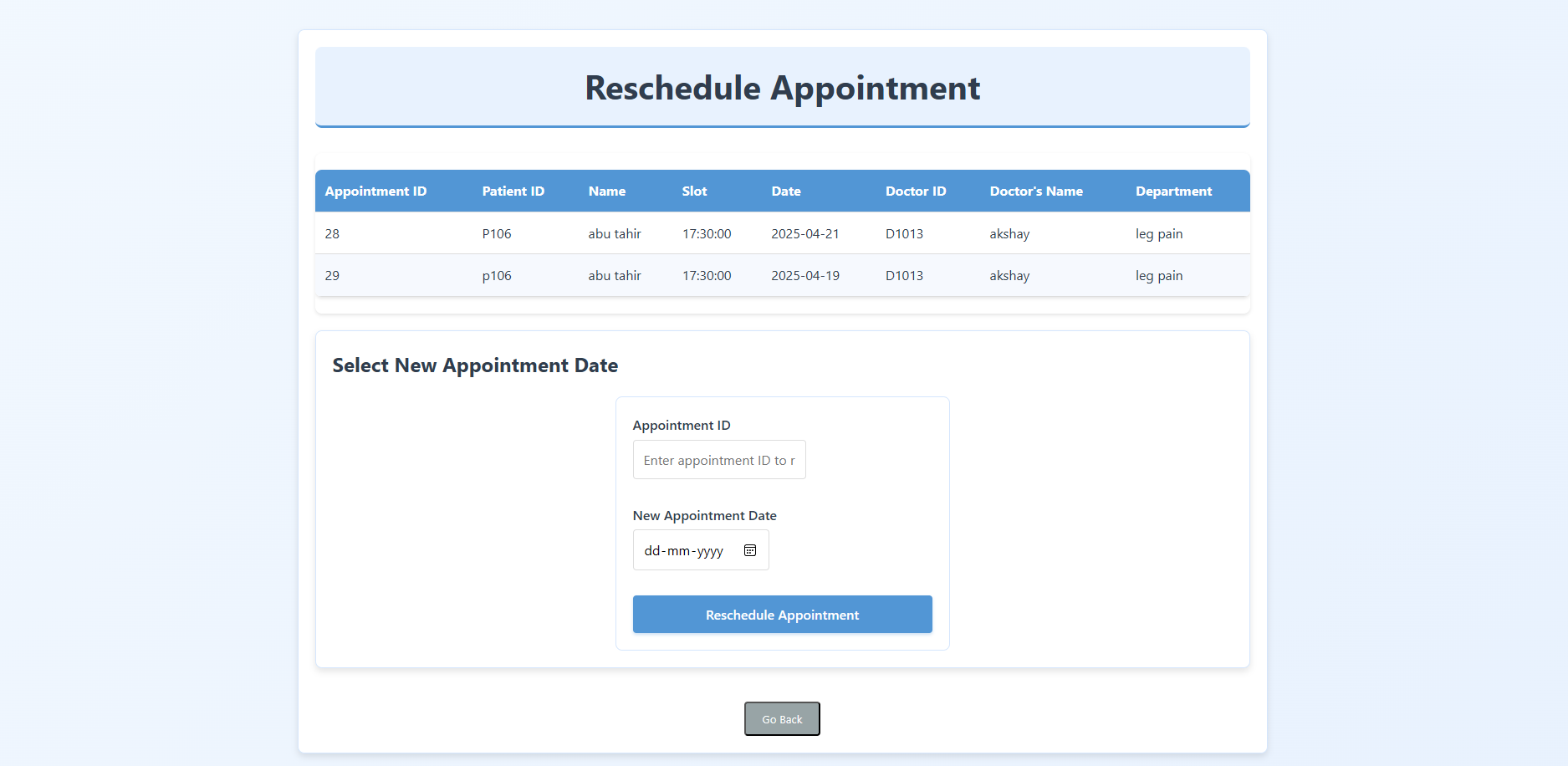
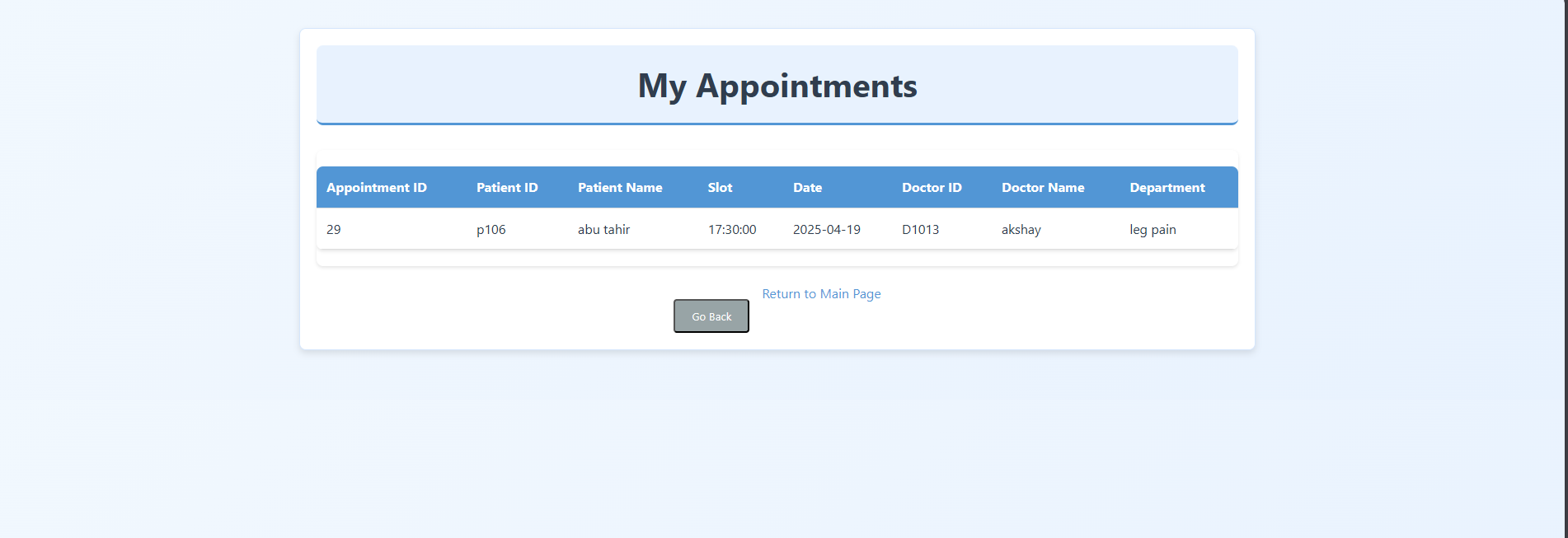
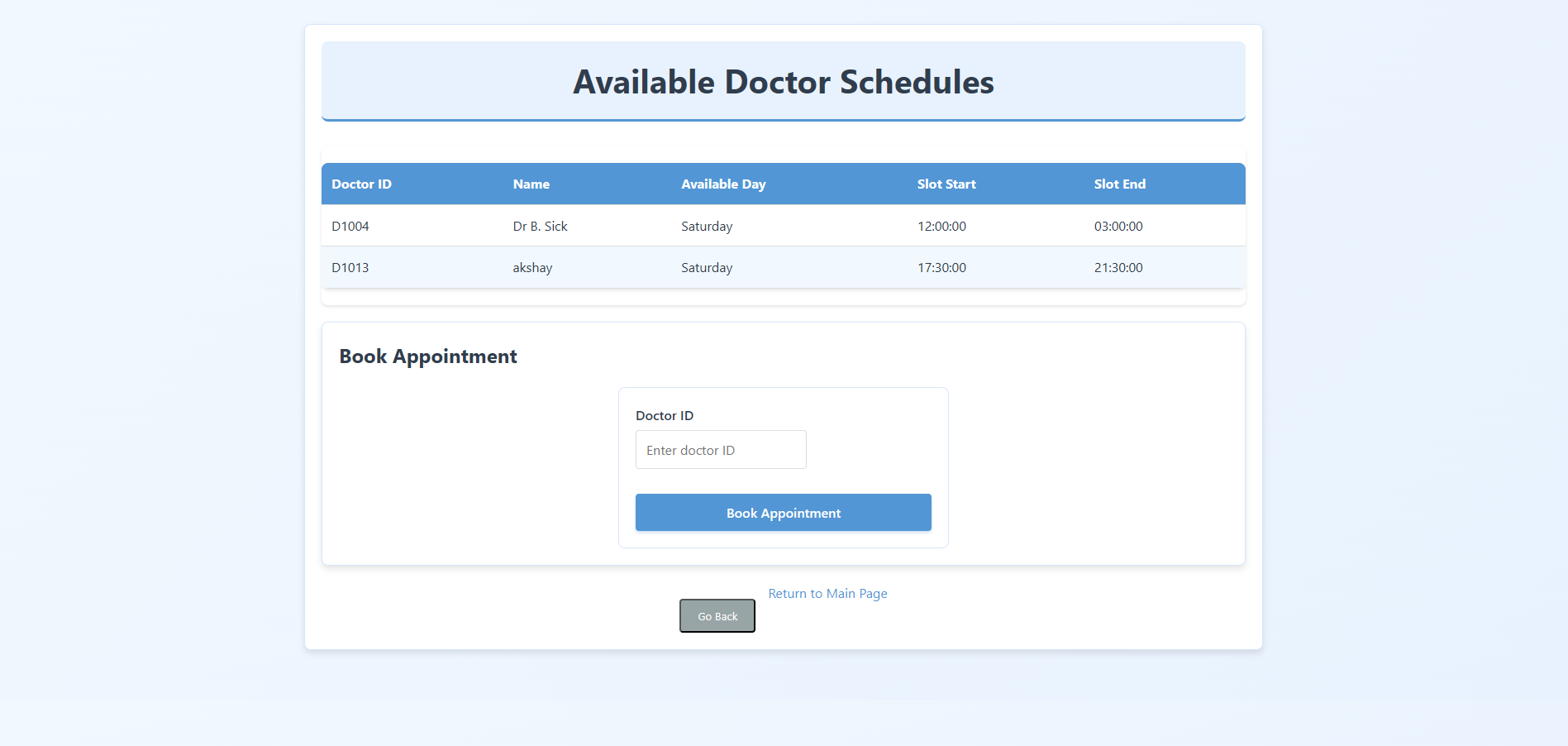
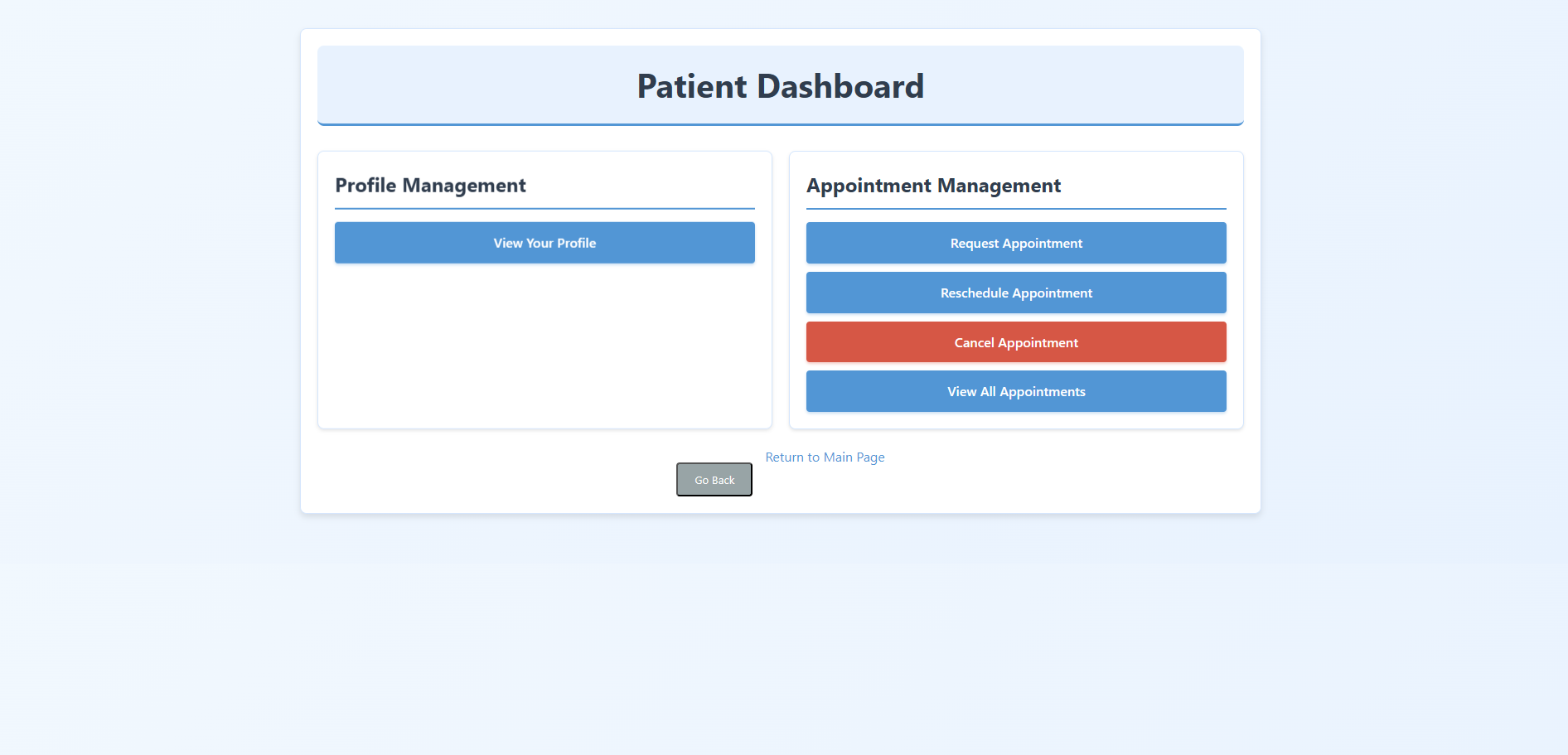
# Screenshots

## UI:

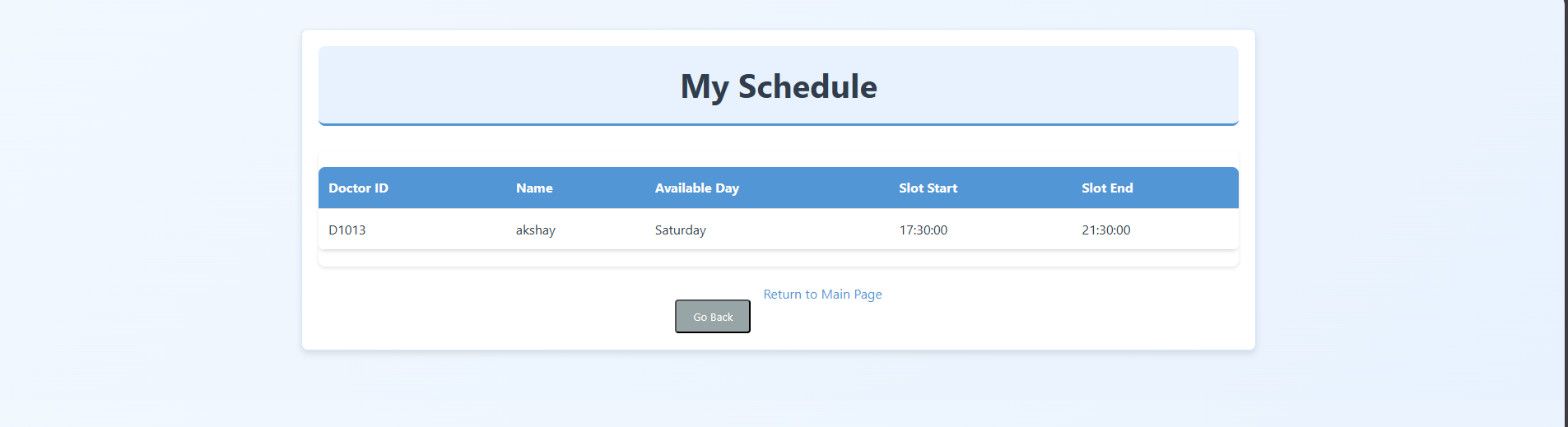
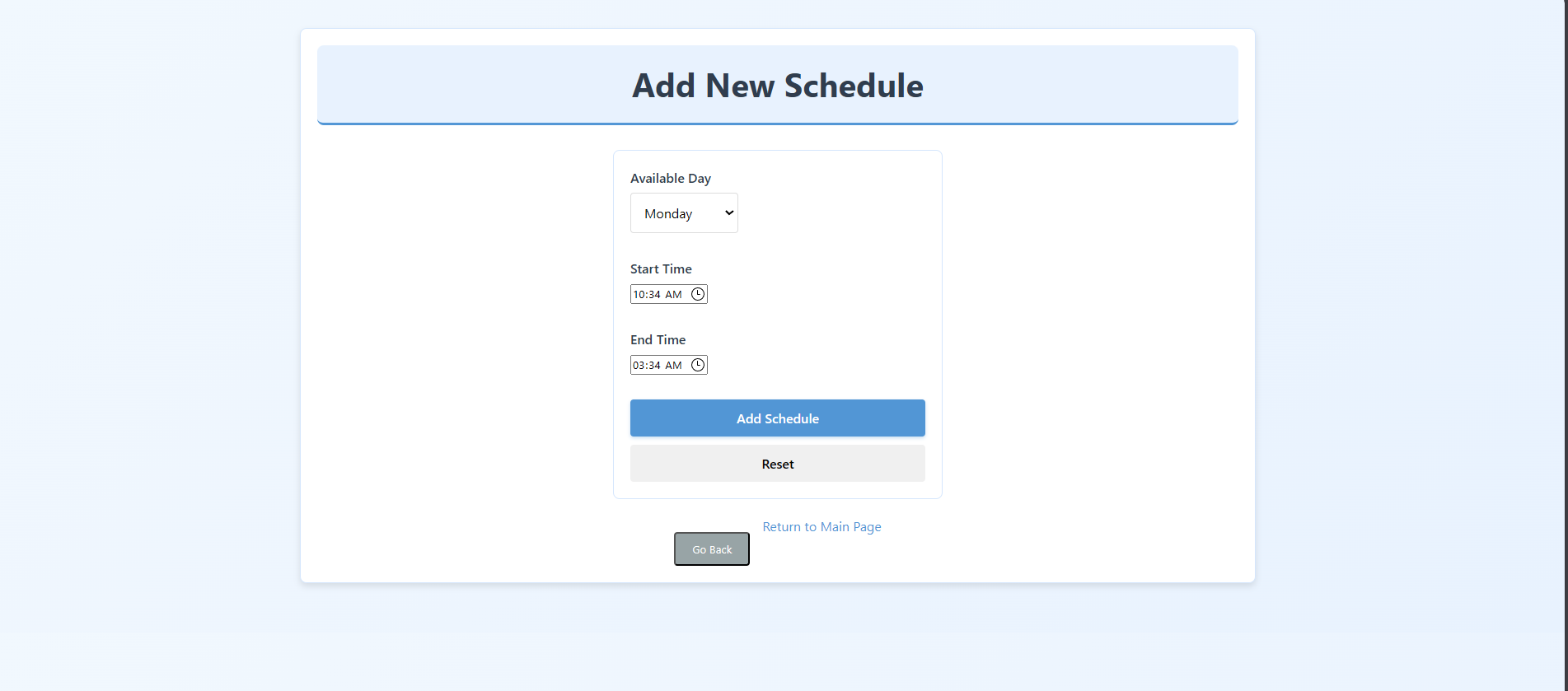
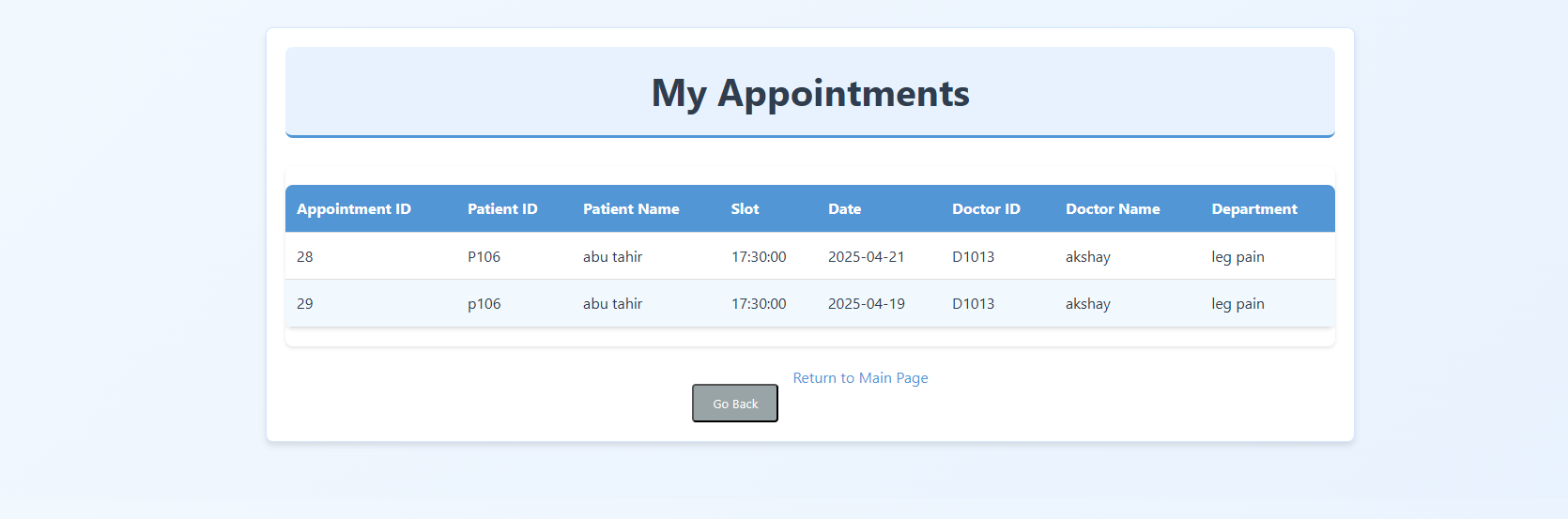
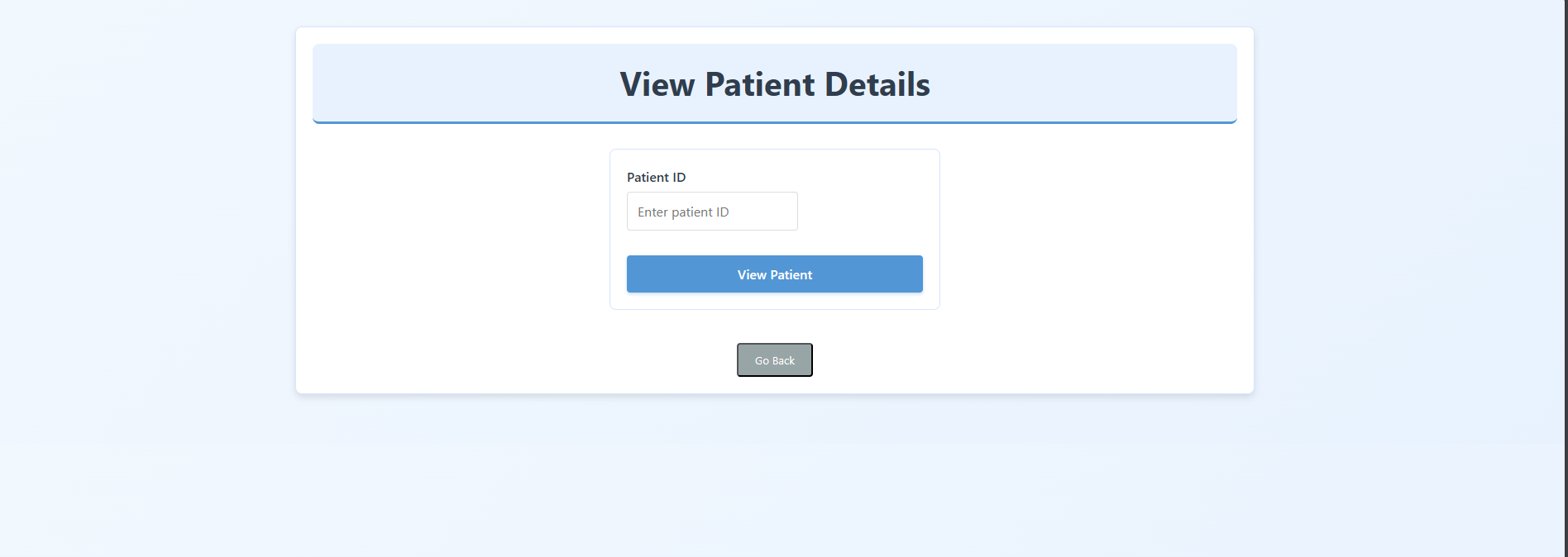
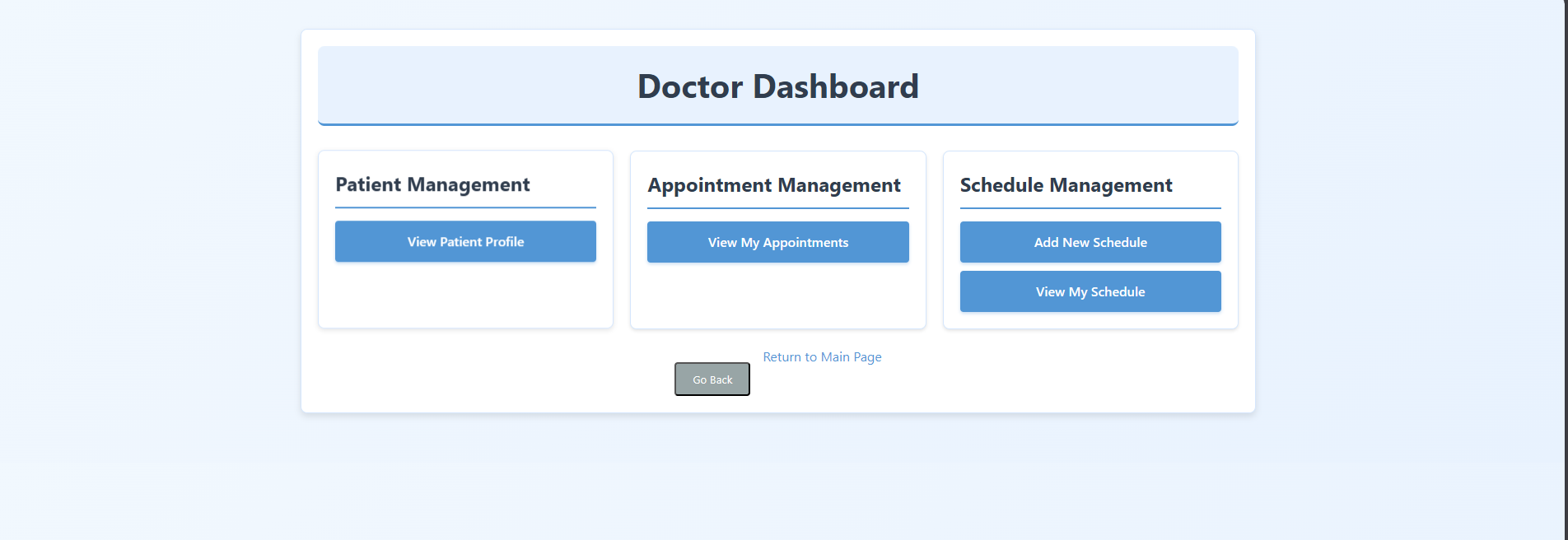


**Patient:**

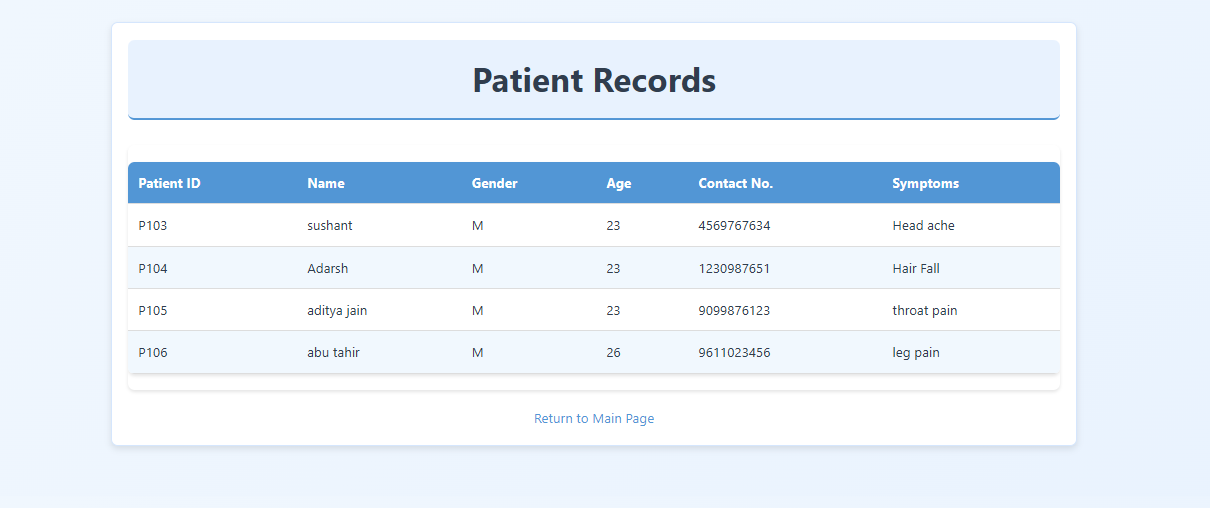
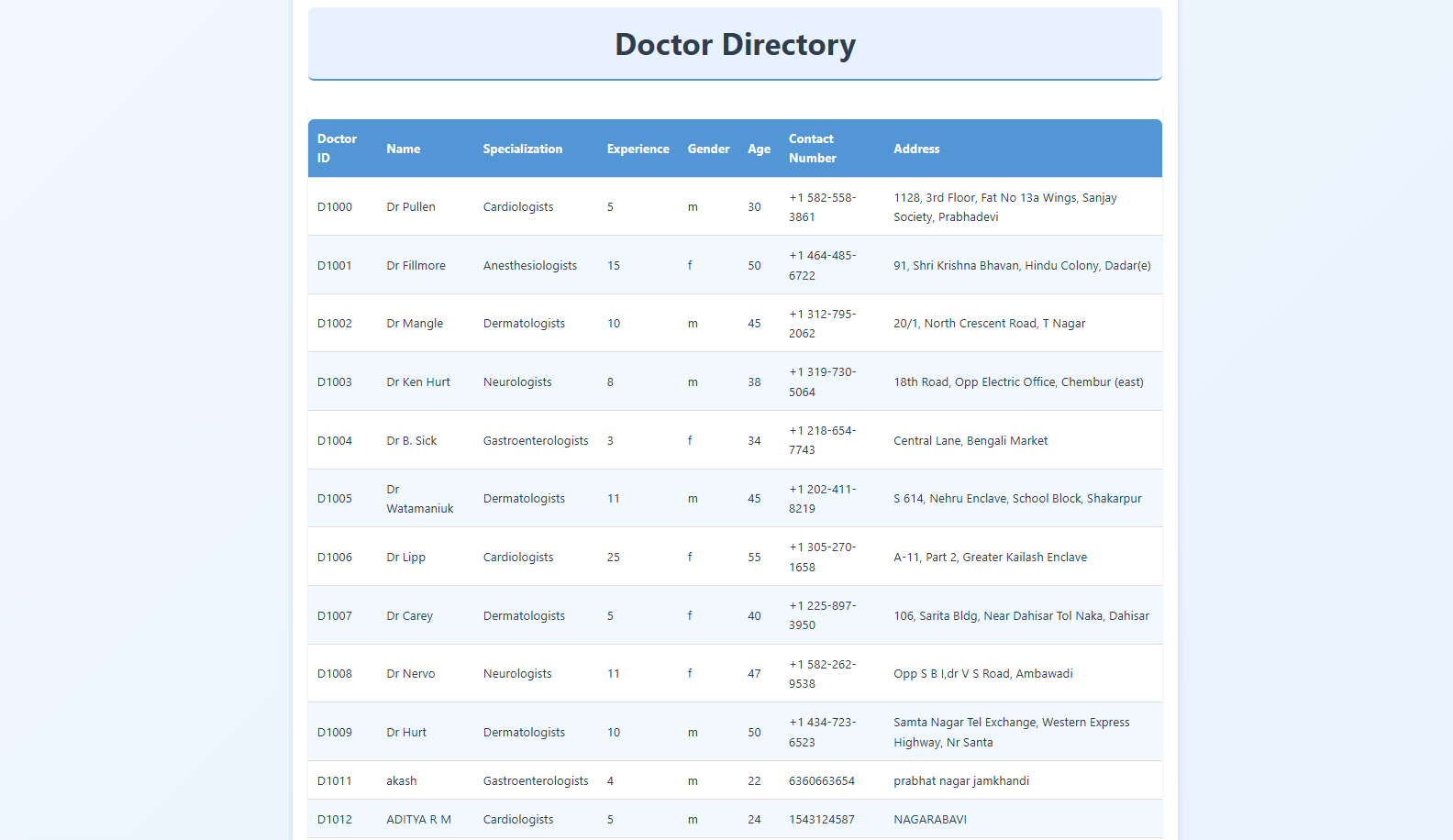
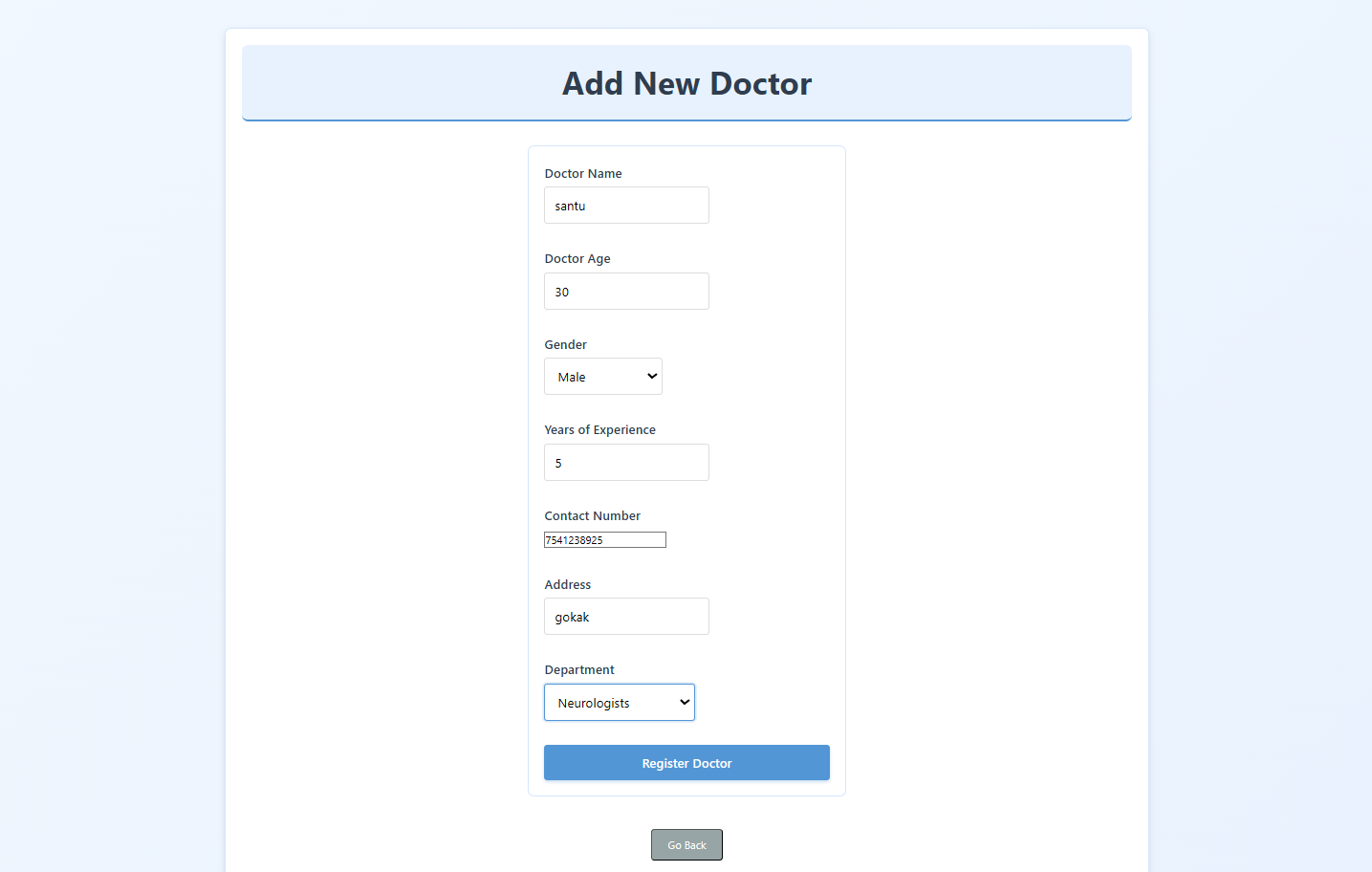
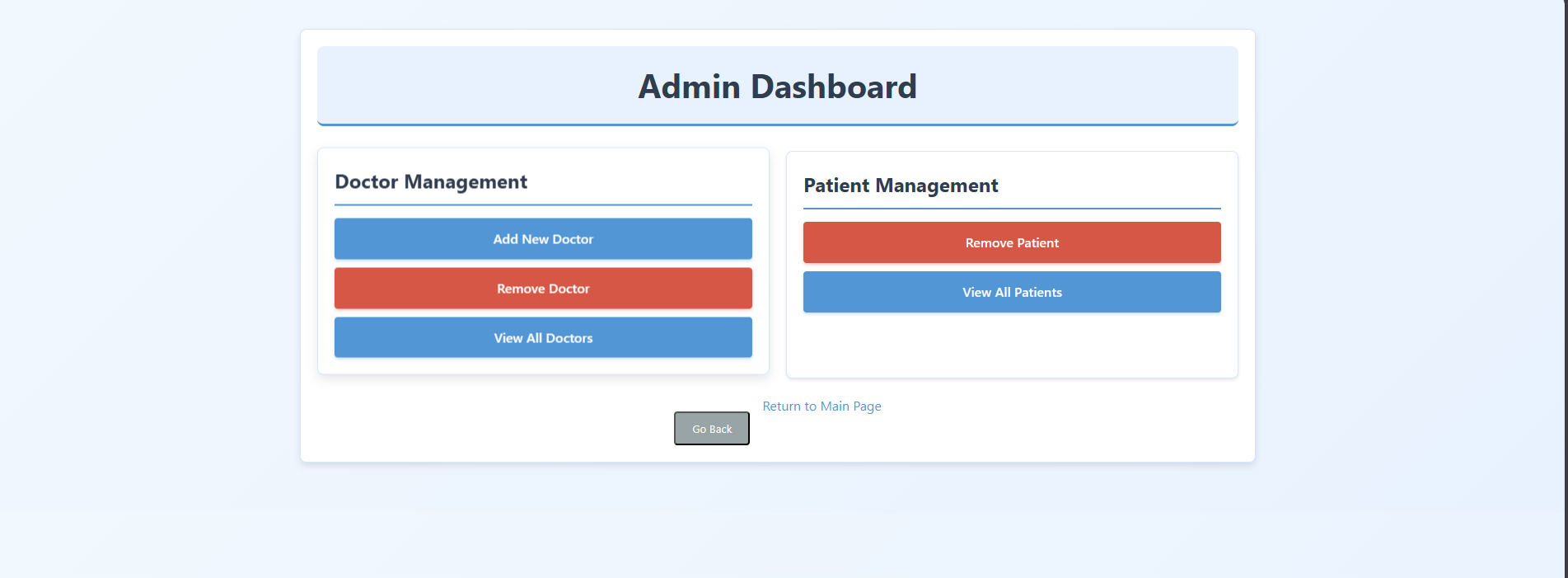




**Doctor:**



**Admin:**



Individual contributions of the team members:

|  |  |
| --- | --- |
| **Name** | **Module worked on** |
| **Abhinandan S** | Integrated SPRING beans and worked on Services to implement interfaces |
| **Aditya R Mirji** | Model part of MVC and implementation of design patterns |
| **Aditya Shenoy** | Controller part of MVC and implementation of design principles |
| **Akash CN** | Managed setup and added dependencies and worked on View part of MVC and UI of project along with error handling |