Hospital management system

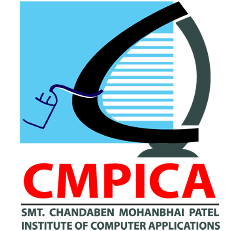
By

Under Guidance

of

**Dr. Jay Nanavati**

Submitted to



Smt. Chandaben Mohanbhai Patel Institute of Computer Applications

CHARUSAT

Changa

April 2025



[Accredited with Grade A+ by NAAC,](https://www.charusat.ac.in/)

CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

Changa

**Acknowledgement**

Knowledge in itself is a continuous process. At this moment of our substantial enhancement, we rarely find words to express our gratitude towards those who were constantly involved with us.

The completion of any inter disciplinary project depends upon coordination, cooperation and combined efforts of several resources of knowledge, creativity, skill, energy and time. The work being accomplished now, we feel our sincerest urge to recall and knowledge through these lines, trying our best to give full credit wherever it deserves.

We would like to thank our project guide **Dr. Jay Nanavati,** I/C Principal **Dr. Dharmendra Patel** and I/C Dean **Dr. Sanskruti Patel** who advised and gave us moral support through the duration of our project. Without their constant encouragement we could not have been able to achieve what we have.

It’s our good fortune that we had support and well wishes of many. We are thankful to all and those names which have been forgotten to acknowledge here but contributions have not gone unnoticed.

With Sincere Regards,

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** |  | **Subject** | **Page Number** |
| **1** |  | **Project Profile** | **5** |
| **2** |  | **Introduction to tools** | **7** |
| **3** |  | **System Study** | **10** |
|  |  | Existing System | **11** |
|  |  | Proposed System | **11** |
|  |  | Scope of the Proposed System | **11** |
|  |  | Aim and Objective of the Proposed System | **11** |
|  |  | Feasibility Study | **12** |
|  |  | Operational Feasibility | **12** |
|  |  | Technical Feasibility | **12** |
|  |  | Economic Feasibility | **12** |
|  |  | Project Overview | **13** |
| **4** |  | **System Analysis** | **14** |
|  |  | Requirements Specification (along with System Modules) | **15** |
|  |  | Use Case Diagram | **16** |
|  |  | Class Diagram | **17** |
|  |  | Activity Diagram | **18** |
| **5** |  | **System Design** | **21** |
|  |  | Data Flow / Screen Layouts | **22** |
|  |  | Data Dictionary | **34** |
| **6** |  | **System Testing** | **37** |
|  |  | Test Cases | **38** |
| **7** |  | **Future Enhancement** | **42** |
| **8** |  | **References** | **44** |

**PROJECT PROFILE**

* **Project Profile:**

**Project Name:** Doctor Appointment Booking System

**Type of Application:** Web-based Application

**Project Description:**

The Doctor Appointment Booking System is an online platform designed to

facilitate the scheduling of appointments between patients and healthcare

providers. The system allows patients to easily book and cancel

appointments while providing doctors with a streamlined way to manage their

schedules. Key features include real-time availability, a secure database for

patient records, and a Doctor Dashboard for appointment tracking and

earnings management.

**Team Size:** 3 Members

**Front End:** React.JS, Tailwind CSS

**Back End:** Node.js, Express.js, MongoDB Atlas

**Tools used:** Visual Studio Code (IDE) , GitHub, Postman

**INTRODUCTION TO TOOLS**

* **Introduction to Tools**
* **Front End Tool:**
* **Tailwind CSS:**
* A utility-first CSS framework for styling user interfaces with minimal custom CSS.
* **React.js:**
* A JavaScript library for building user interfaces, allowing for reusable UI components.
* **Back End Tool:**
* **Node.js:**
* A JavaScript runtime for developing scalable server-side applications.
* It is used in conjunction with Express.js to create RESTful APIs that handle requests from the front end, manage database interactions, and implement logic.
* **Express.js:**
* A Node.js web framework used to create RESTful APIs and handle routing.
* **MongoDB:**
* A NoSQL database that stores data in flexible. MongoDB is used

for storing user information, Appointment details, Patient Information and Doctor-related data.

**SYSTEM STUDY**

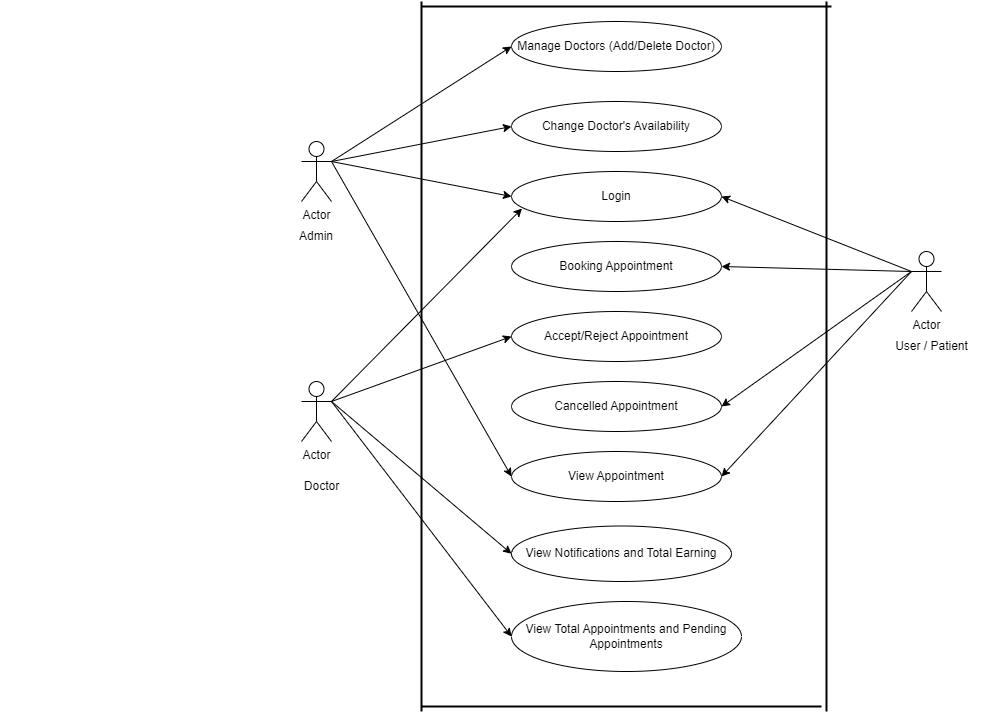
* **System Study**
* **Existing System:**
* Traditional methods of booking doctor appointments is primarily manual, involving phone calls or in-person visits. This leads to inefficiencies such as long wait times, double bookings, and miscommunication between patients and healthcare providers.
* **Proposed System:**
* The proposed system is an online application that allows patients to book appointments conveniently. Features include real-time availability, a Doctor Dashboard, an Admin Dashboard and secure patient information storage.
* **Scope of the Proposed System:**
* The system will cater to various healthcare providers, including hospitals, clinics, and individual practitioners. It will support multiple specialties and allow for integration with existing healthcare management systems.
* **Aim and Objective of the Proposed System:**

1. Provide a user-friendly interface for patients to book appointments.
2. Reduce administrative workload for healthcare providers.
3. Enhance communication between patients and doctors.
4. Maintain a secure database of patient information and appointment history.
5. Implement a Doctor Dashboard for tracking appointments and earnings.

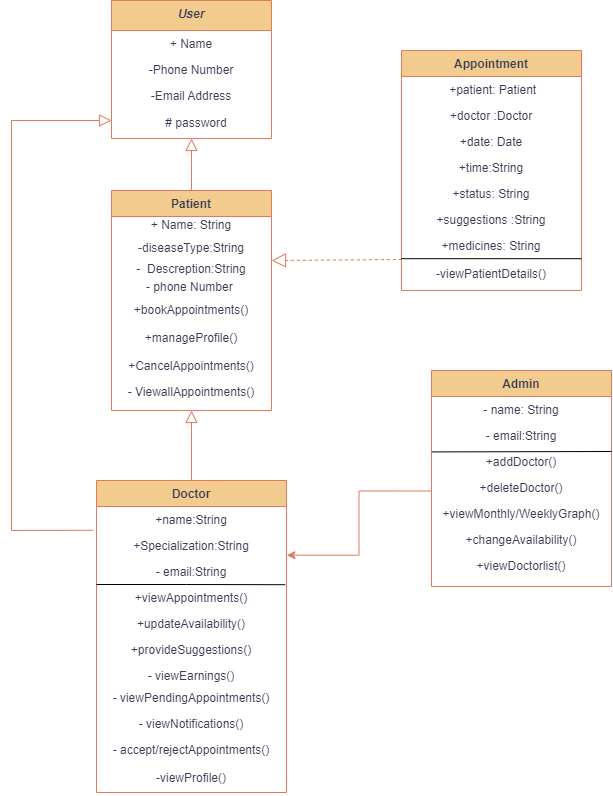
* **Feasibility Study:**
* **Operational Feasibility:** The system is operationally feasible as it simplifies the appointment booking process, making it more efficient for both patients and healthcare providers.
* **Technical Feasibility:** The technology stack chosen is widely used and supported, ensuring that the system can be developed and maintained effectively.
* **Economic Feasibility:** The cost of development and maintenance is justified by the expected increase in patient satisfaction and efficiency.
* **Project Overview:**
* This Appointment Booking System allows users to book appointments with doctors online. The system includes user authentication, appointment scheduling, and an admin/doctor dashboard for managing appointments.
* **Features:**
* User Registration & Login
* Doctor and Admin Login
* Appointment Booking System
* View & Manage Appointments
* Doctor Dashboard with Earnings & Availability Management
* Automatic Status Update for Past Appointments
* Patient Information Storage & Access
* User Profile Viewing & Editing
* Admin Panel for Managing Doctors and Appointments

**System Analysis**

* **System Analysis:**
* **Requirements Specification:**
* **User Registration/Login Module:** Patients and doctors can create accounts and log in using their email and password.
* **Appointment Scheduling Module:** Enables patients to book, reschedule, or cancel appointments.
* **Doctor Dashboard:** Doctors can view their appointments, earnings, and pending consultations.
* **Appointment Status Management:** Appointments automatically update to 'Completed' once the scheduled date has passed.
* **Patient Info Section:** Collects and stores patient details, including name, gender, email (optional), disease type, and a brief description.
* **Doctor's Notes Module:** Doctors can view detailed patient information and add suggestions or medicine-related notes, which are stored in the database.
* **Admin Module:** Administrators can manage users, doctors, and appointments.
* **Use Case Diagram:**

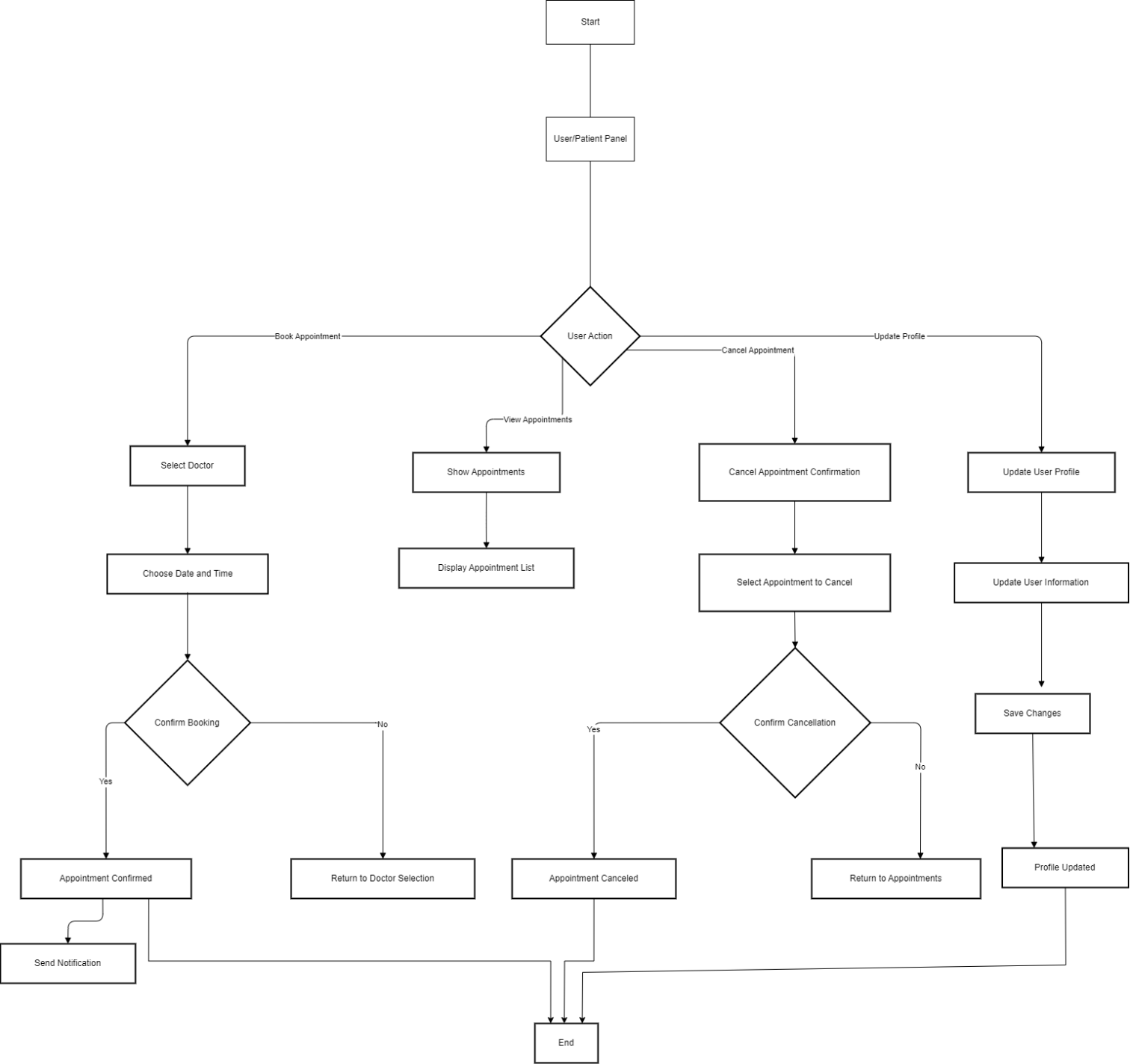


* **Class Diagram:**

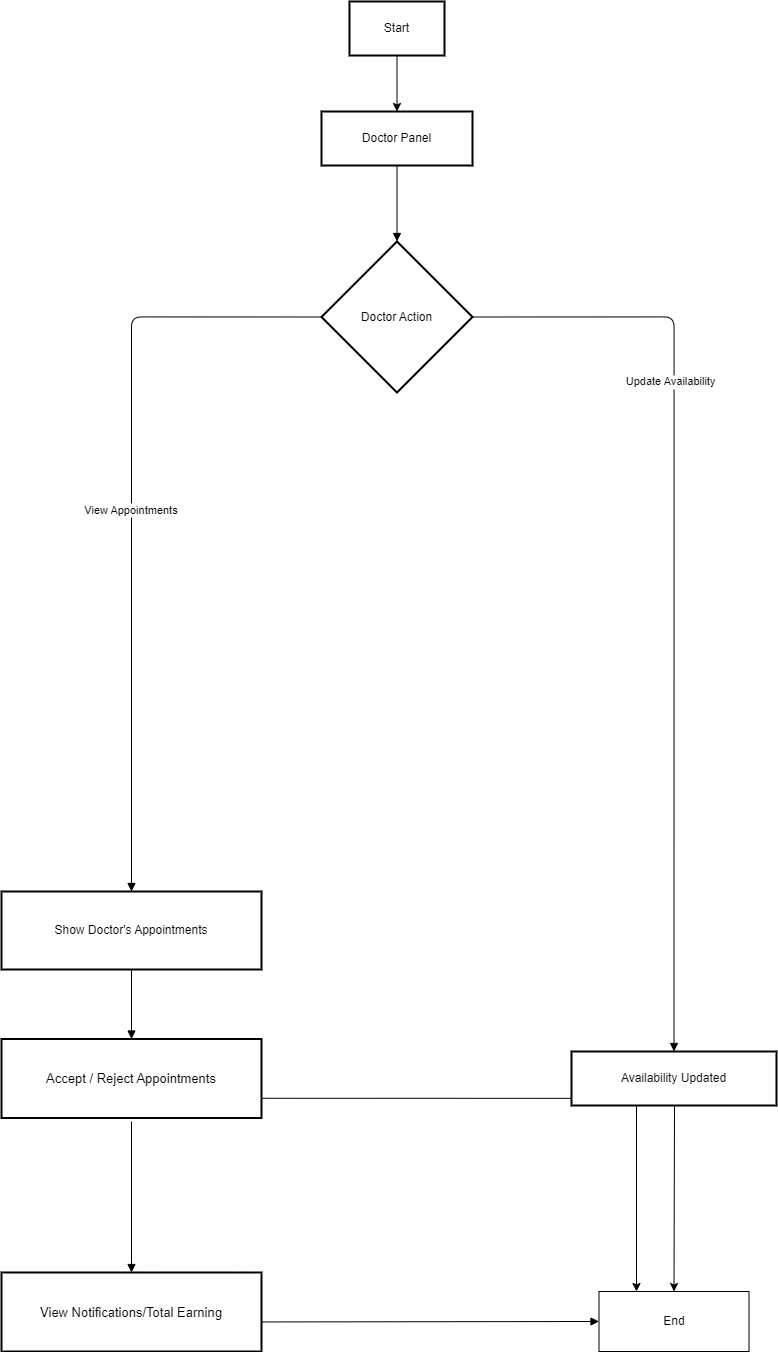


* **Activity Diagram:**

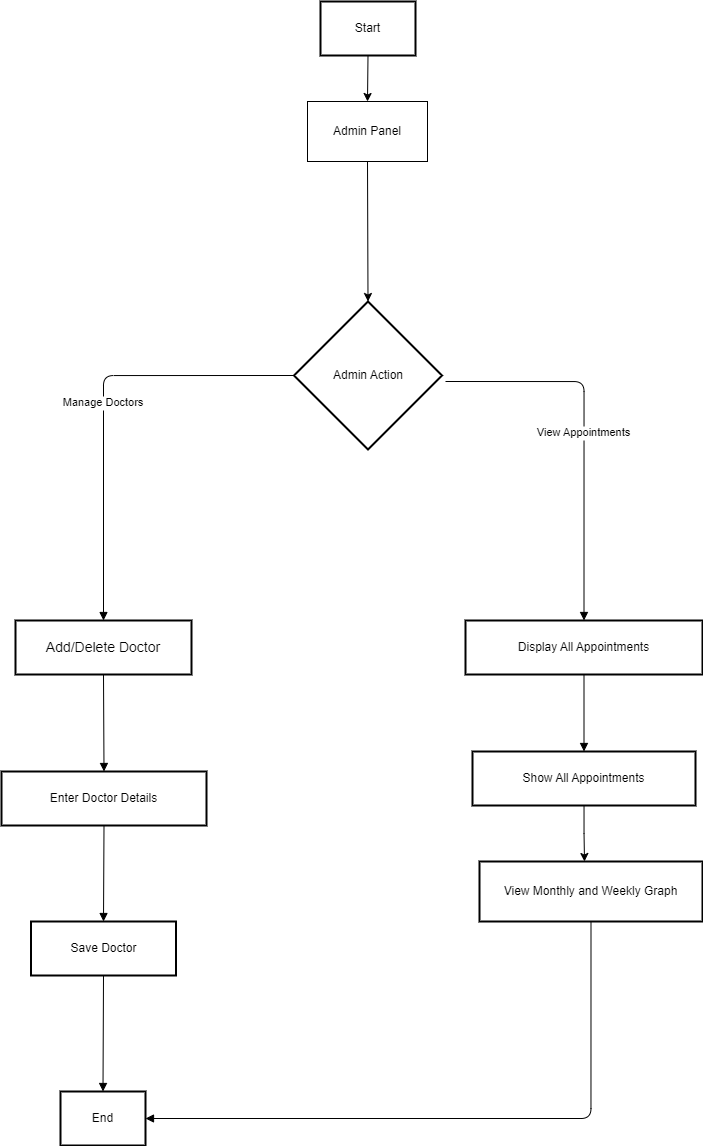
**1.User:**

****

**2.Doctor:**



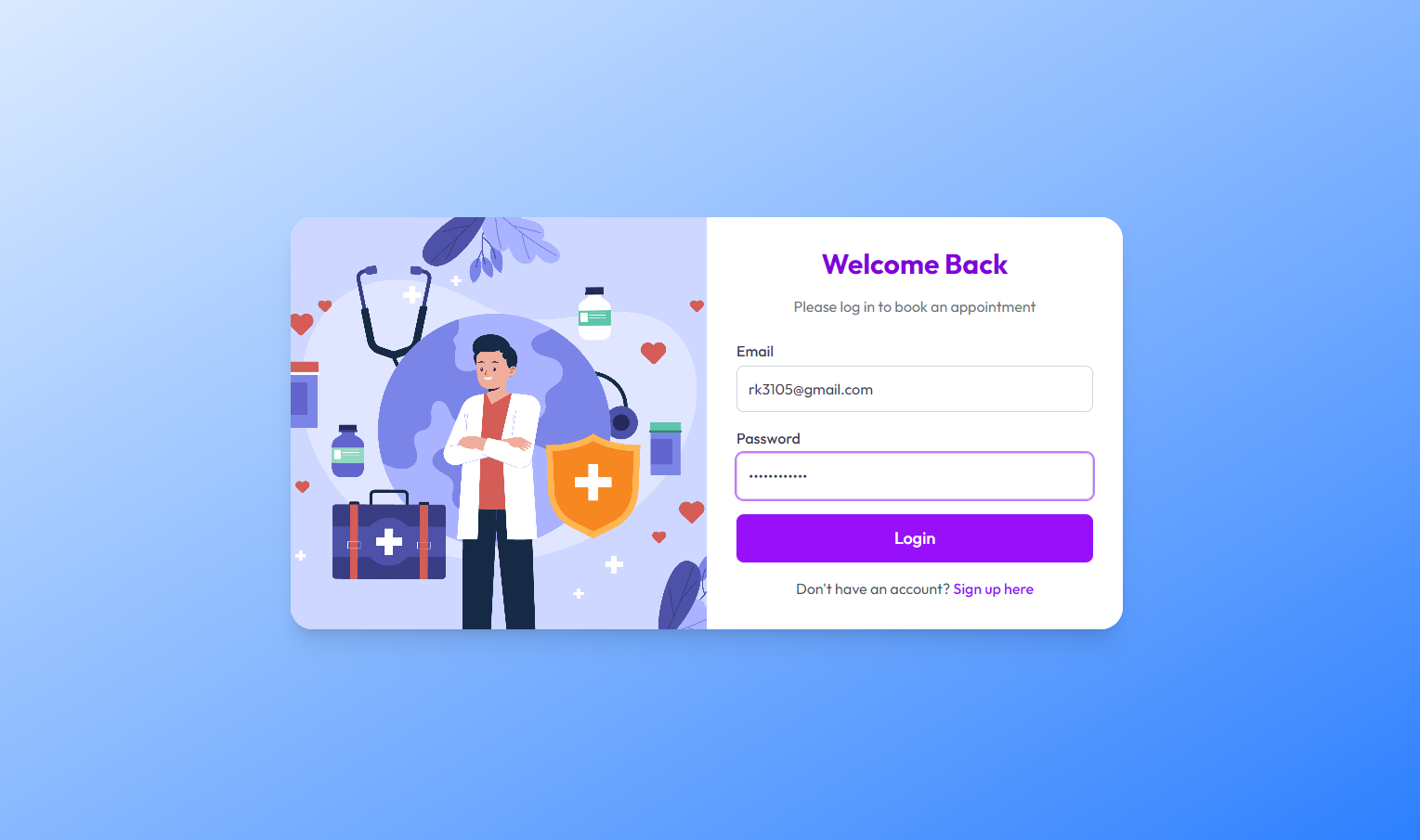
**3.Admin:**

****

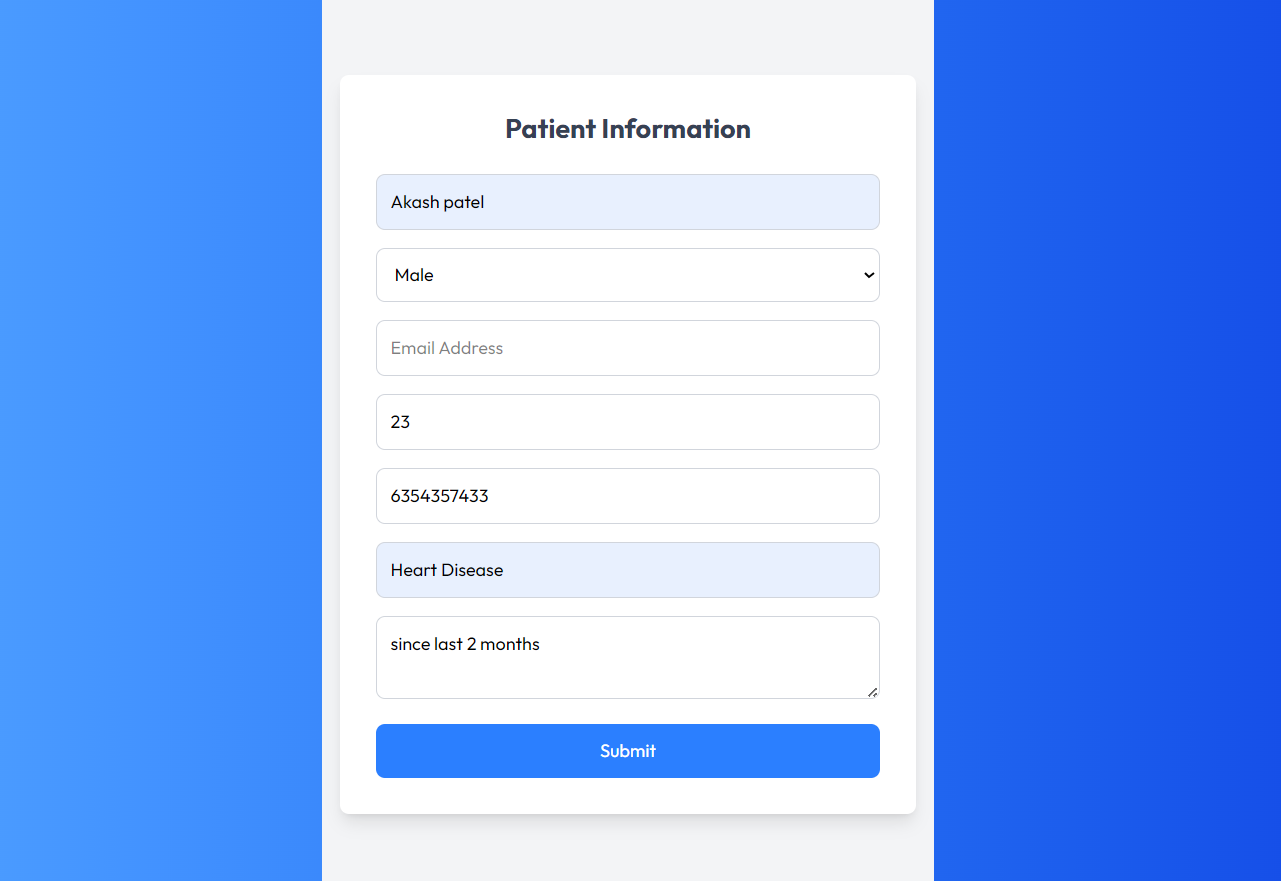
**System Design**

* **User Flow:**
* **Patients / Users:**

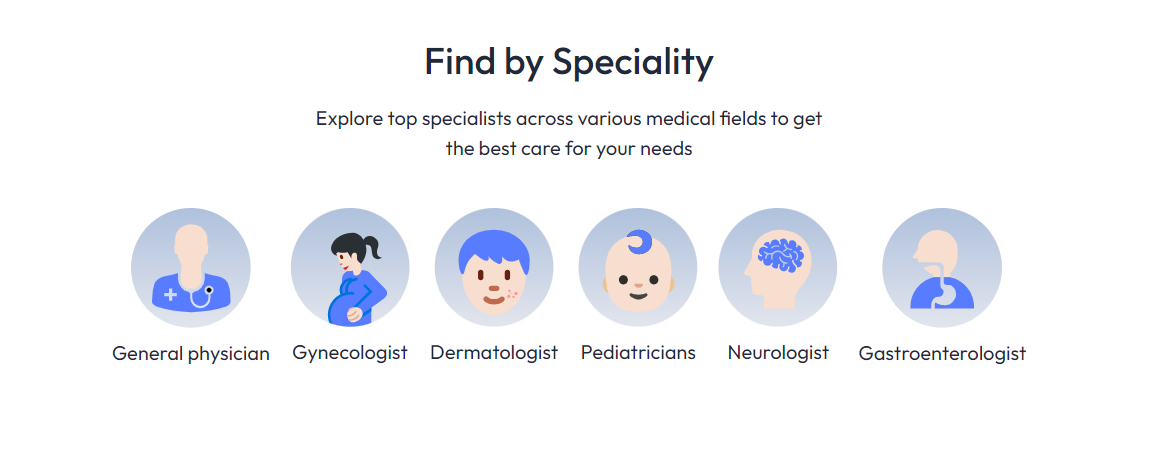
1. Register/Login to the system (Using Email and Password):



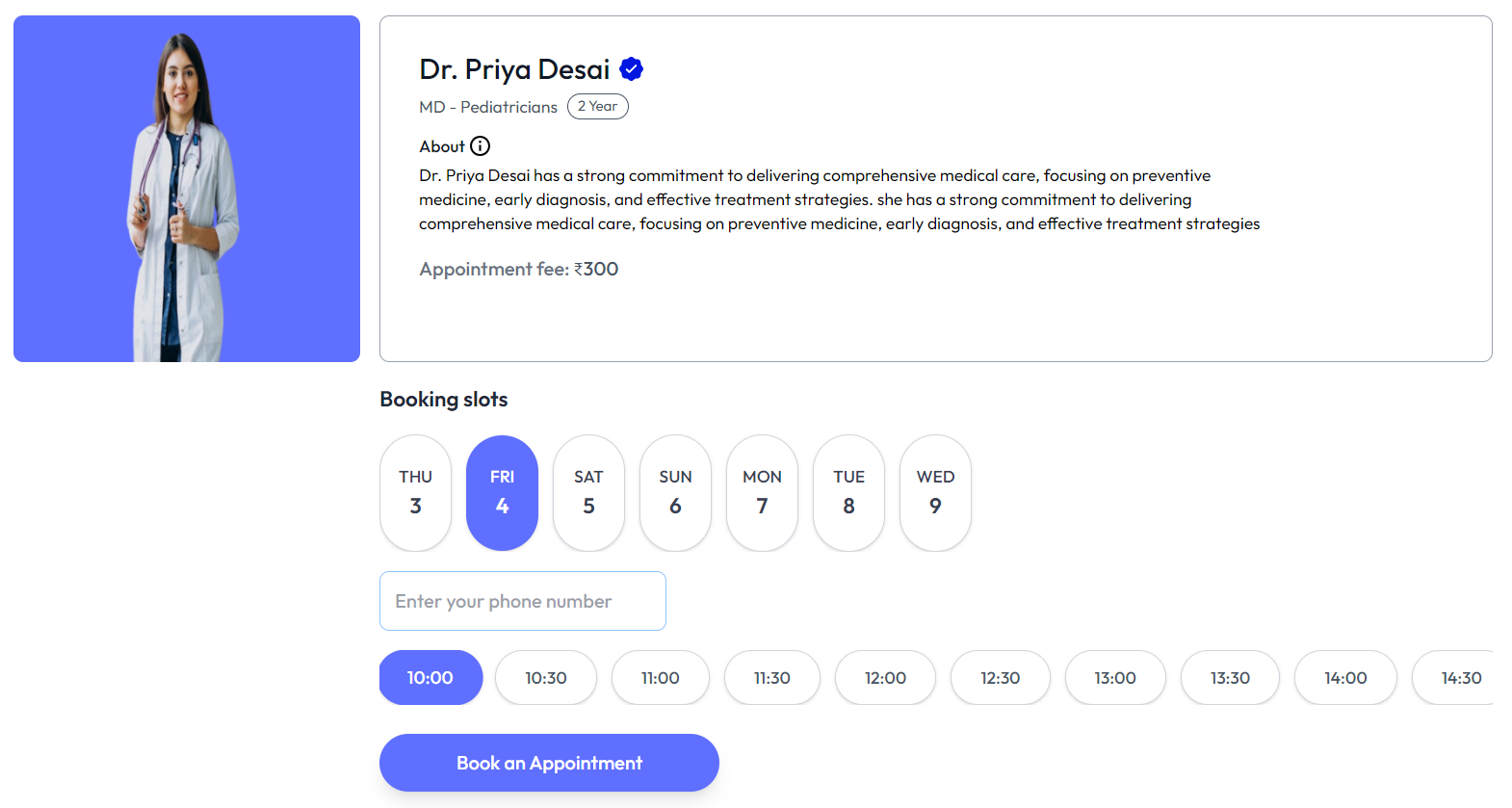
1. Fill out the patient information form (fill all details except email):



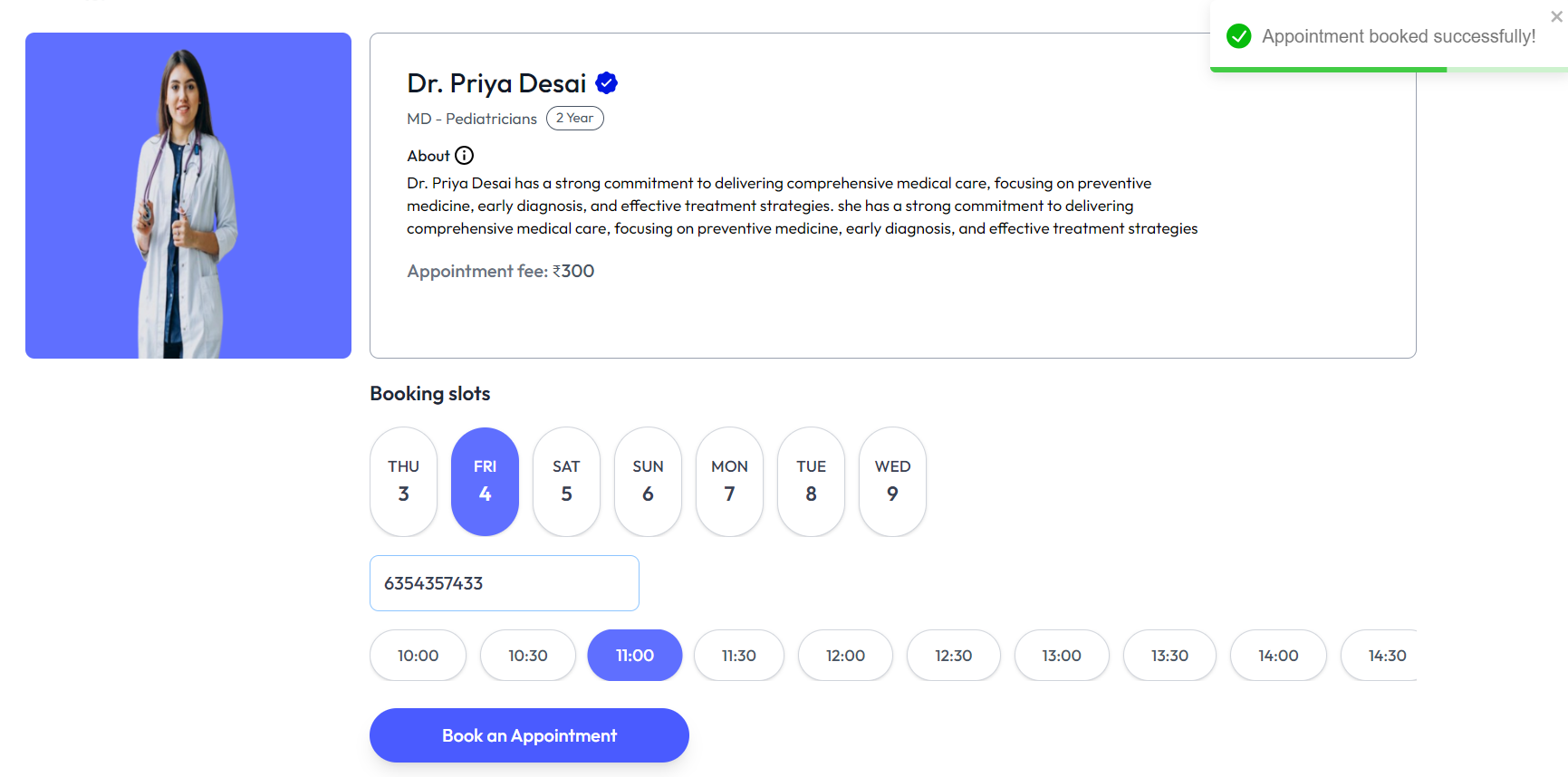
3.Find your doctor by Speciality:



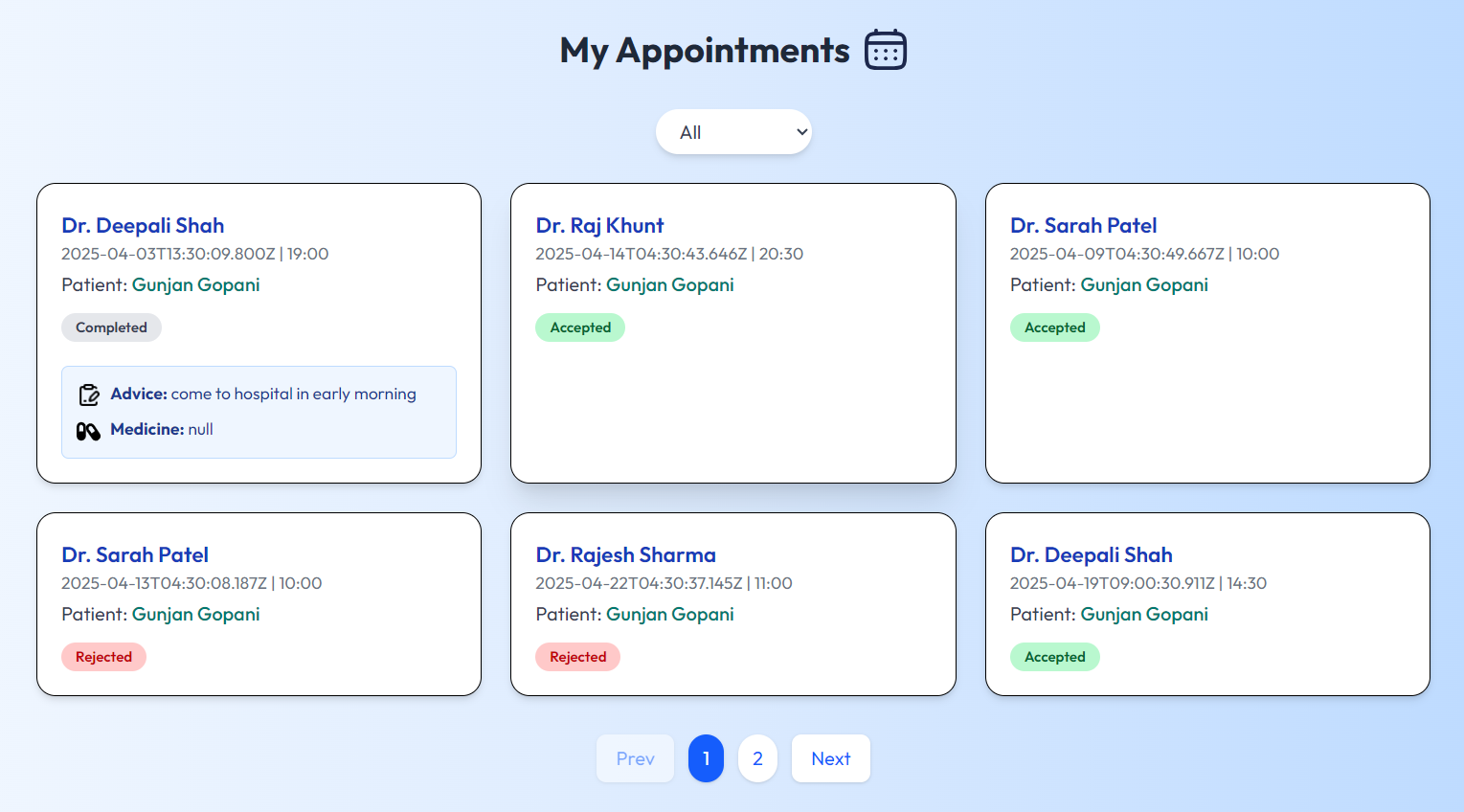
4.Enter Phone Number to proceed

5.Select a doctor & choose an available slot:

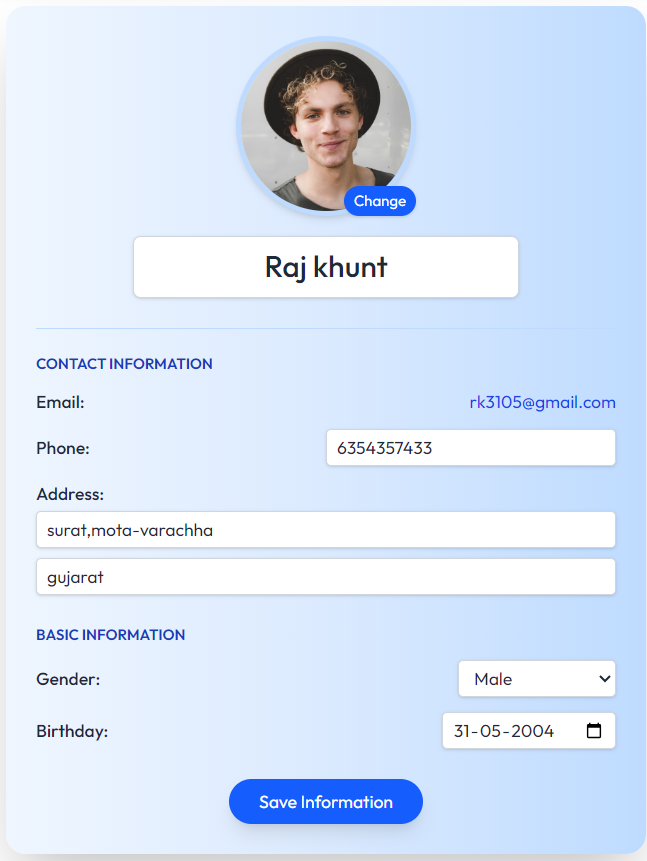
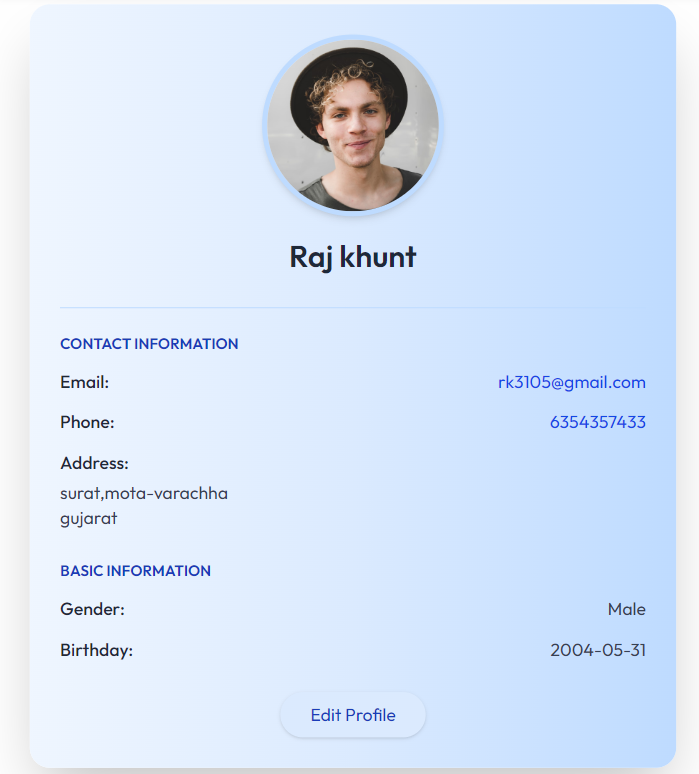
6.Provide Details & Confirm Booking:



7.View appointment on 'My Appointments' page(Through phone number):

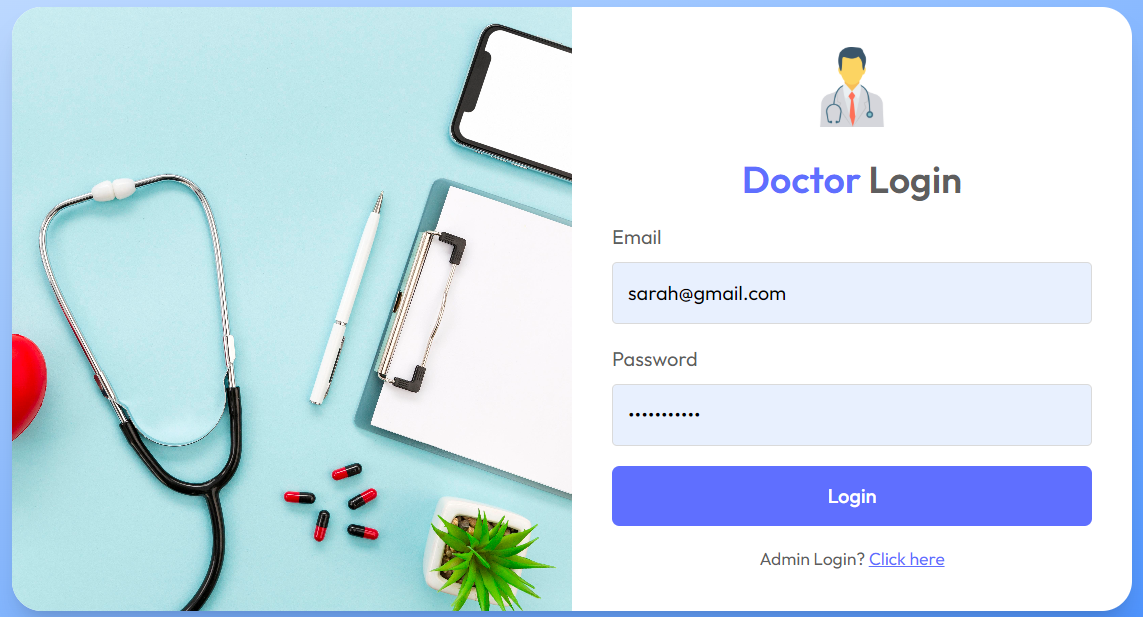


8.View and update personal profile:

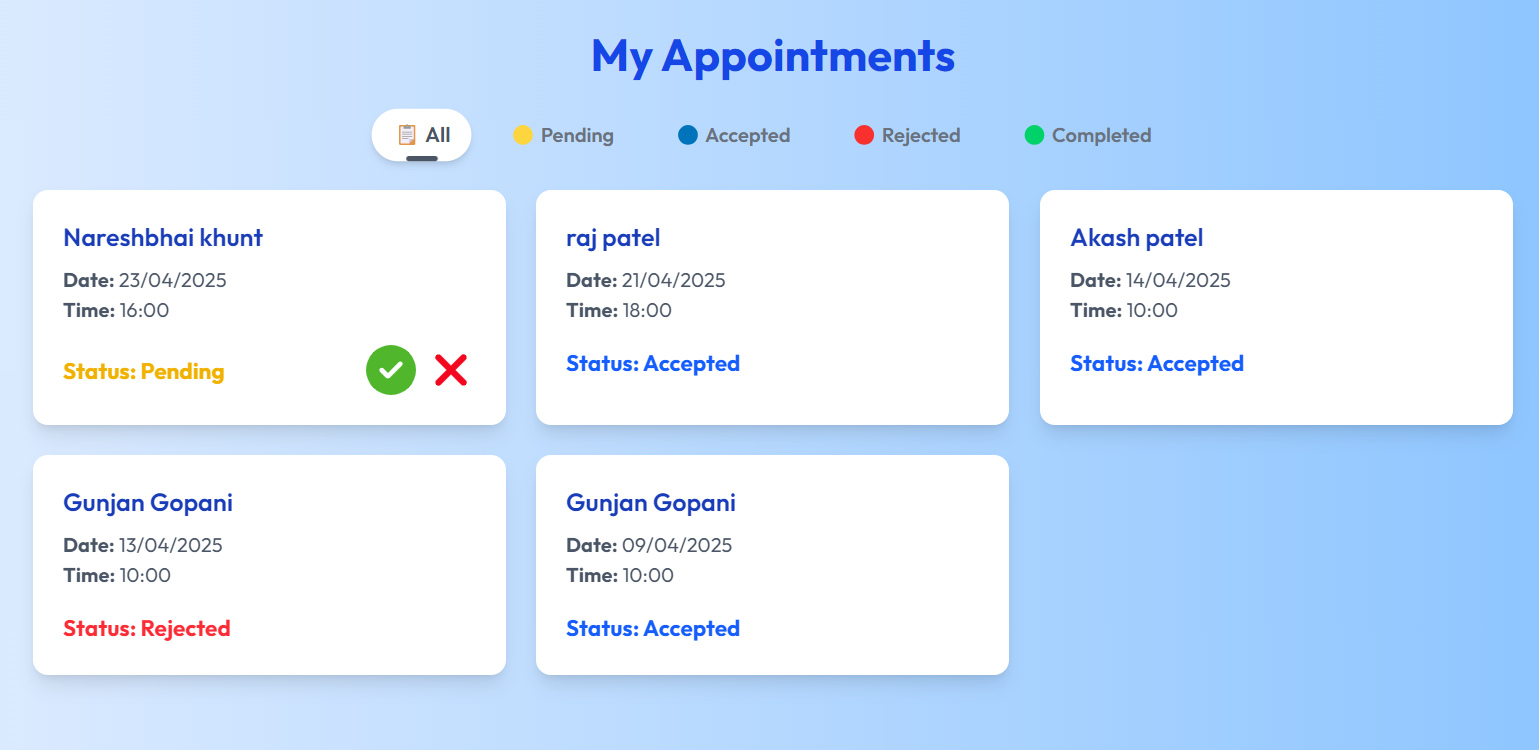


* **Doctors:**

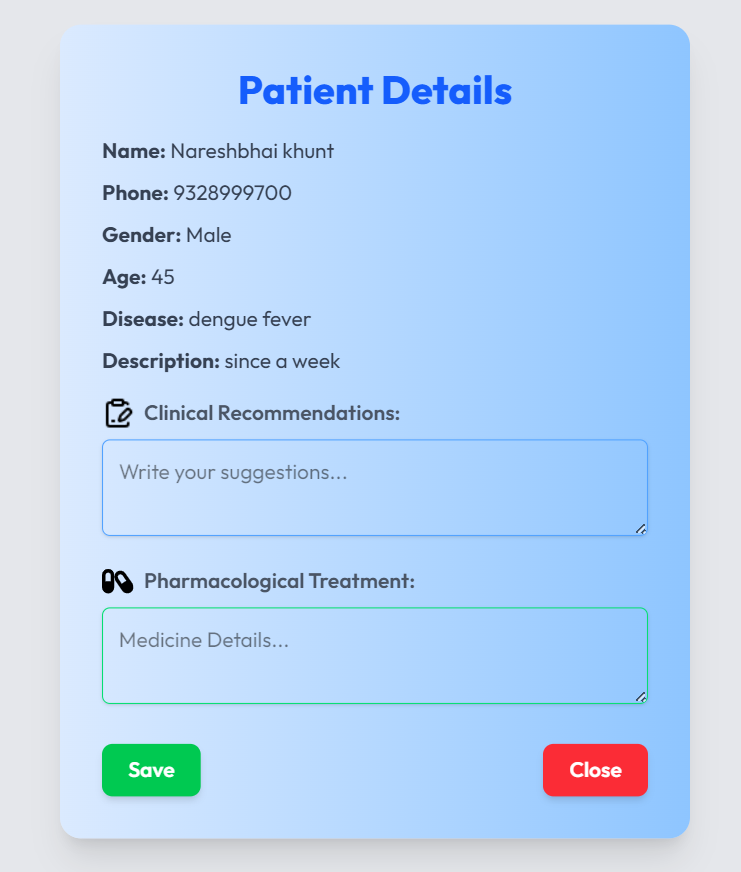
1. Login to Doctor Dashboard:



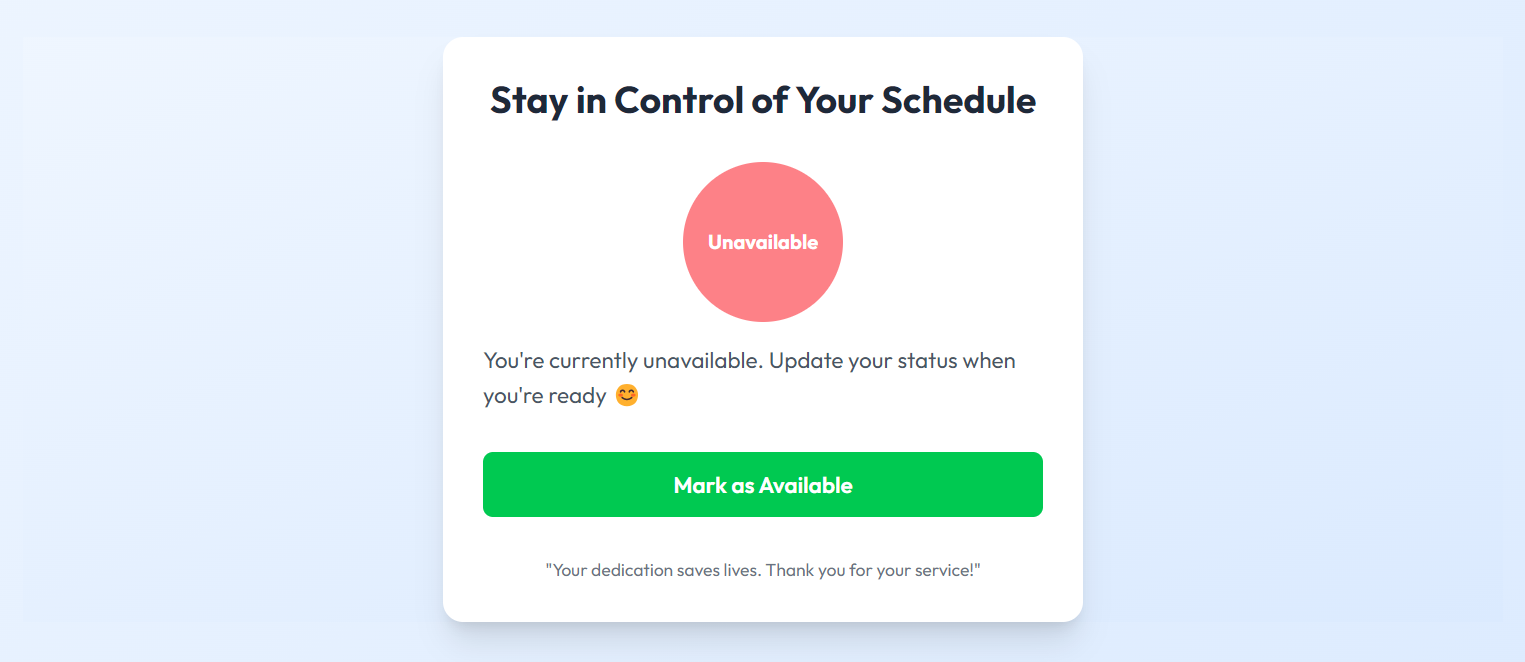
1. View Upcoming appointments:



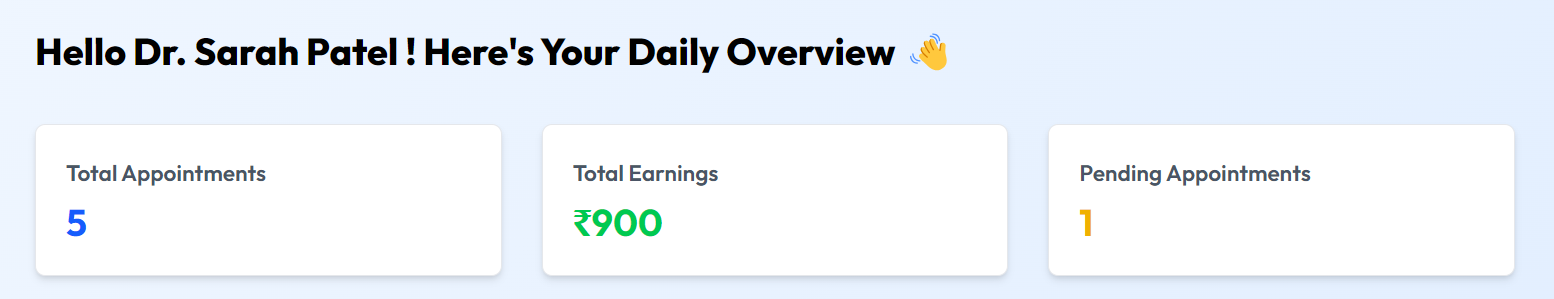
1. Click on an appointment to view detailed patient information:
2. Fill in suggestions & prescribe medication:

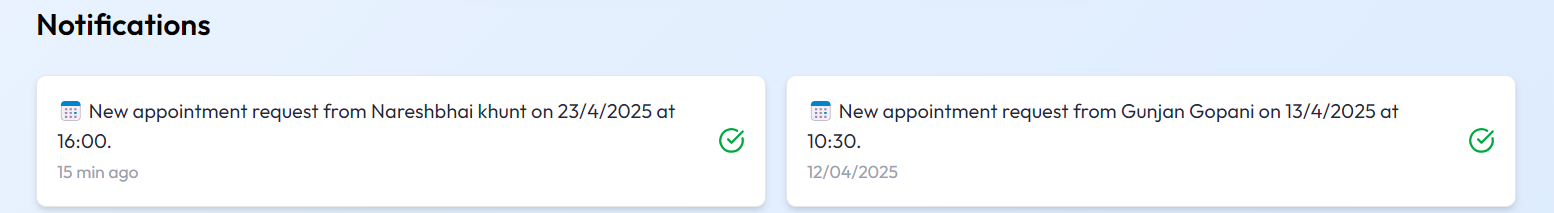


5. Update Availability status By Doctor:

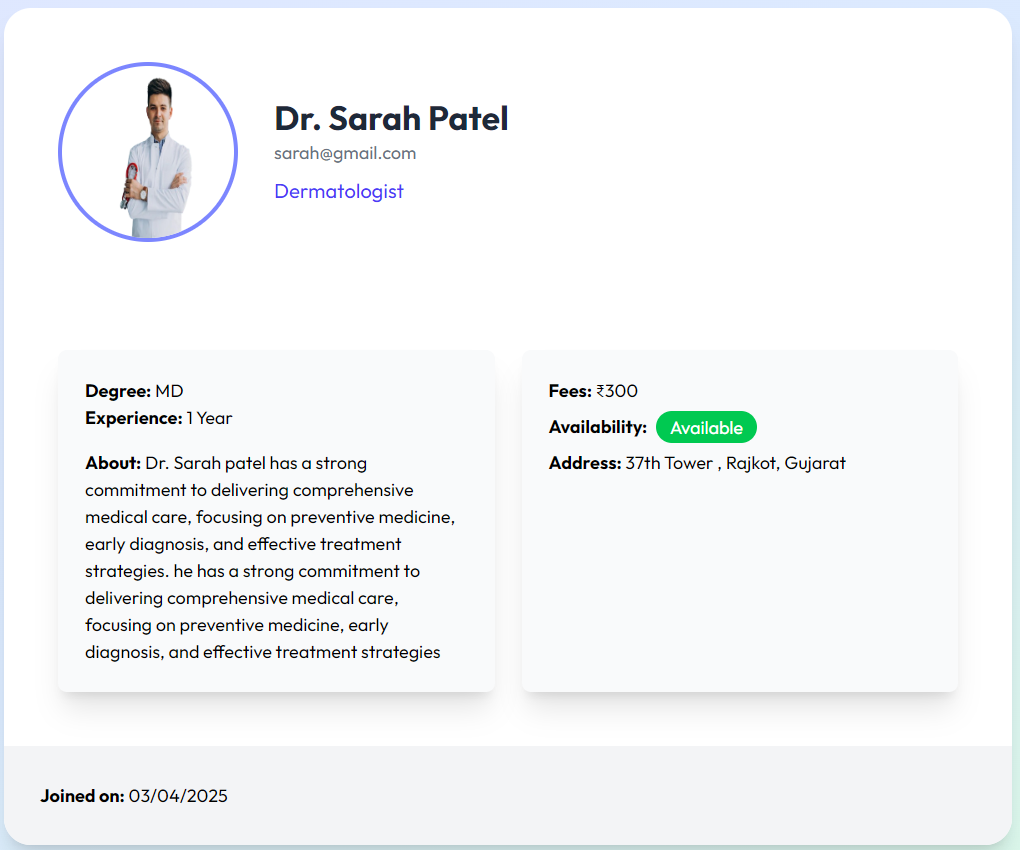


6.View earnings & Total and Pending Appointments and Notifications:

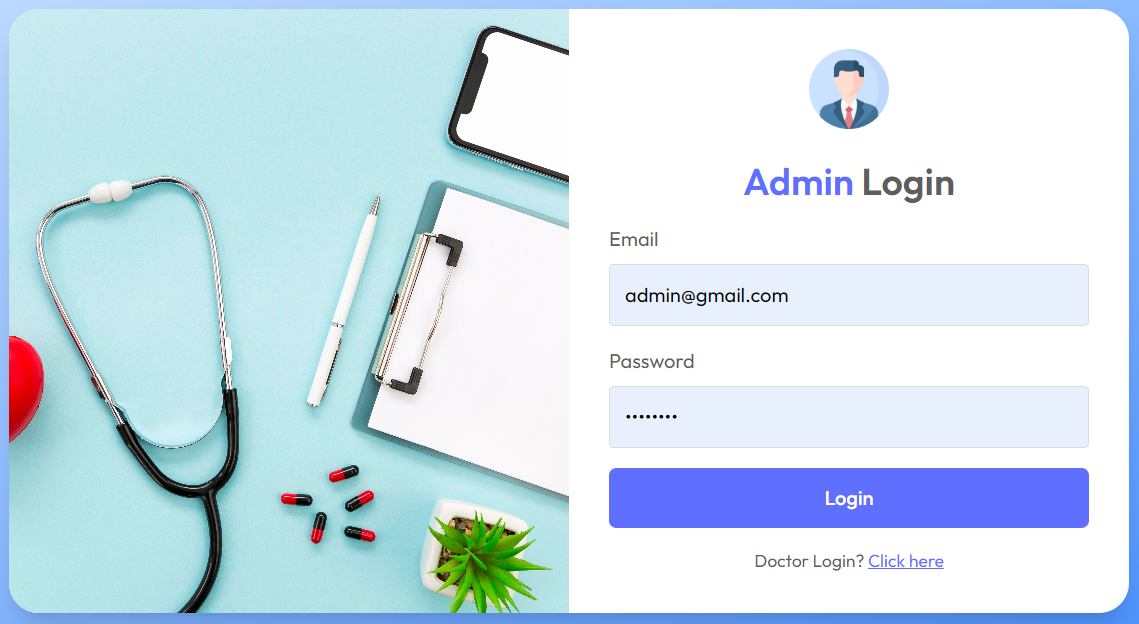




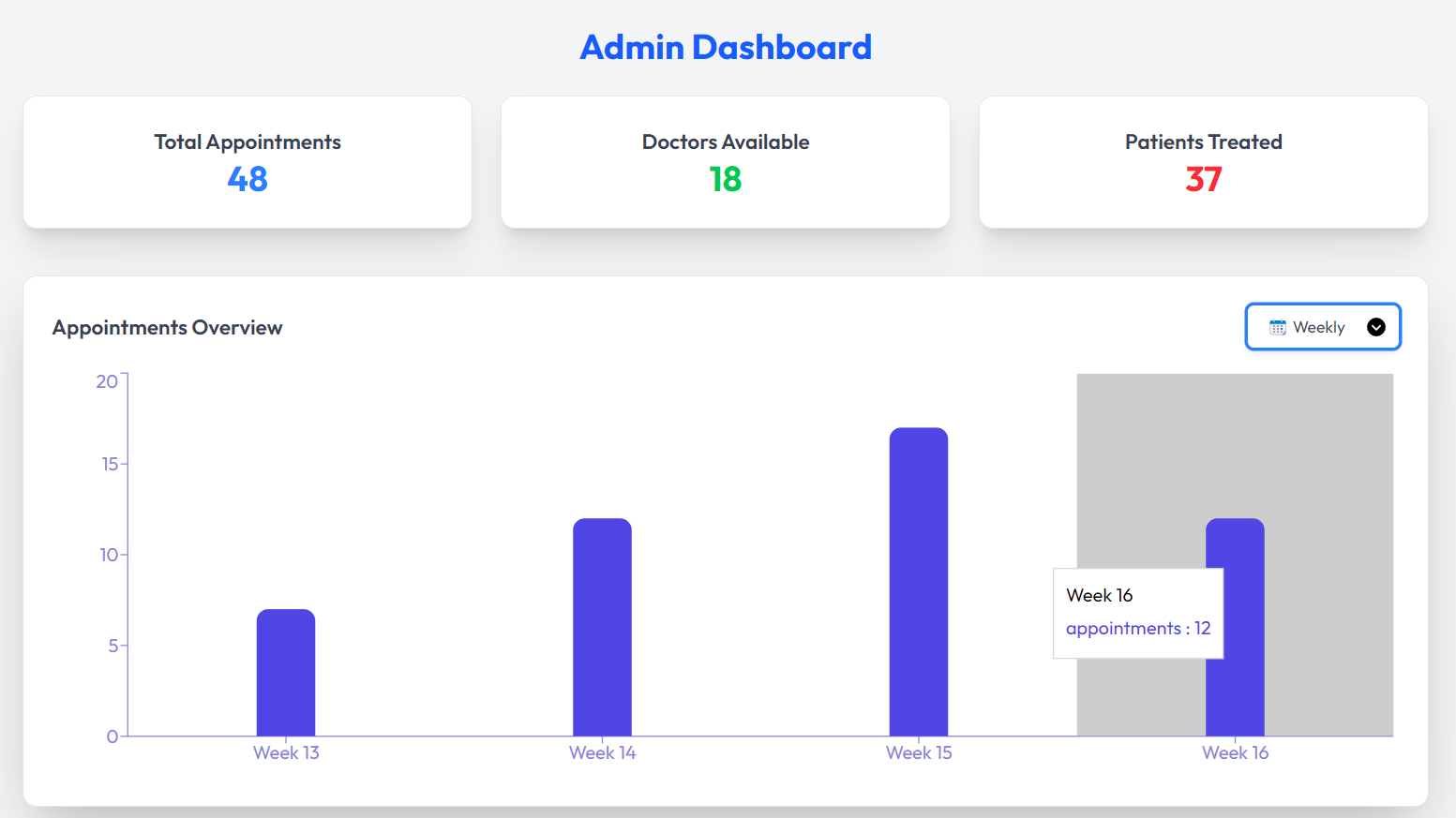
7.Doctor Profile:



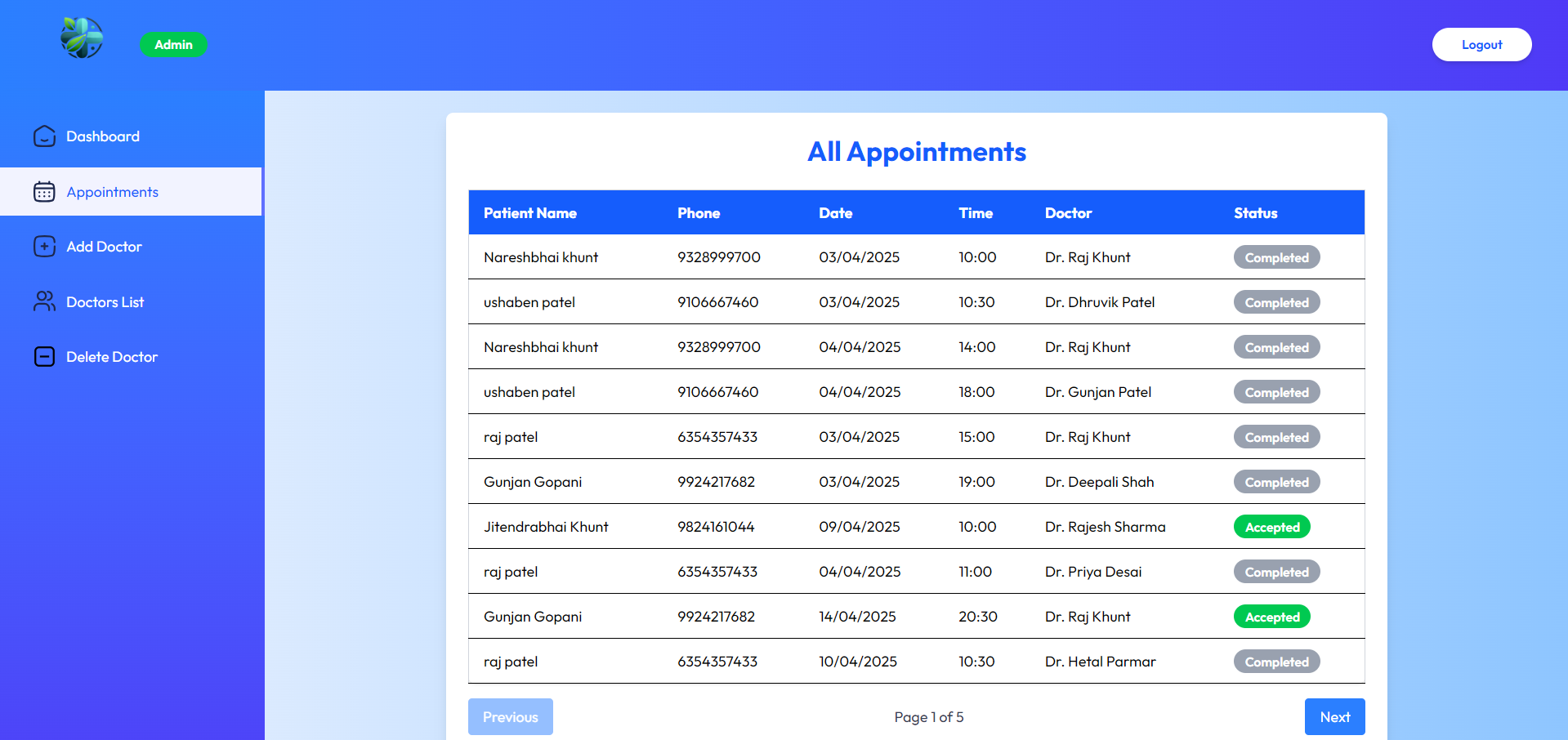
* **Admin:**

1. Login to Admin Panel:

2. Dashboard:

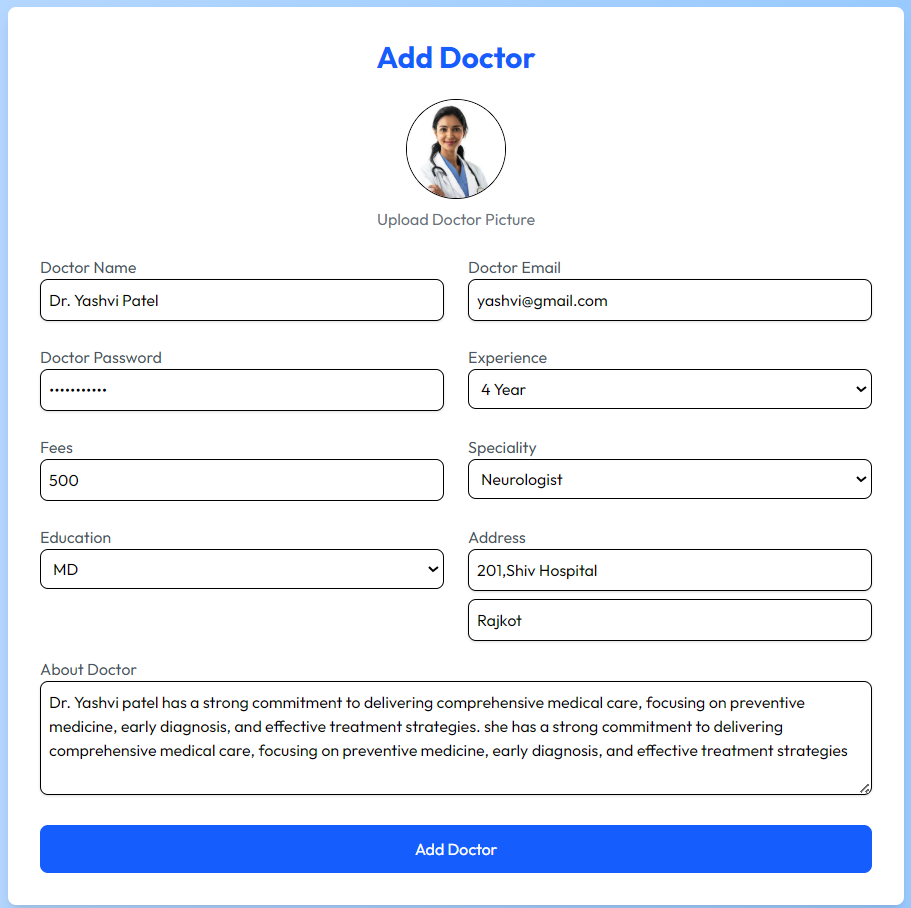


3. View all appointments:

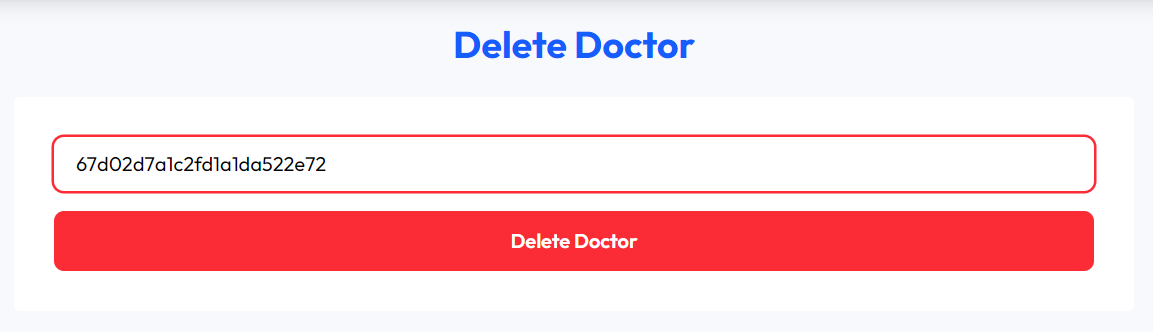


4. Add/Delete doctors:

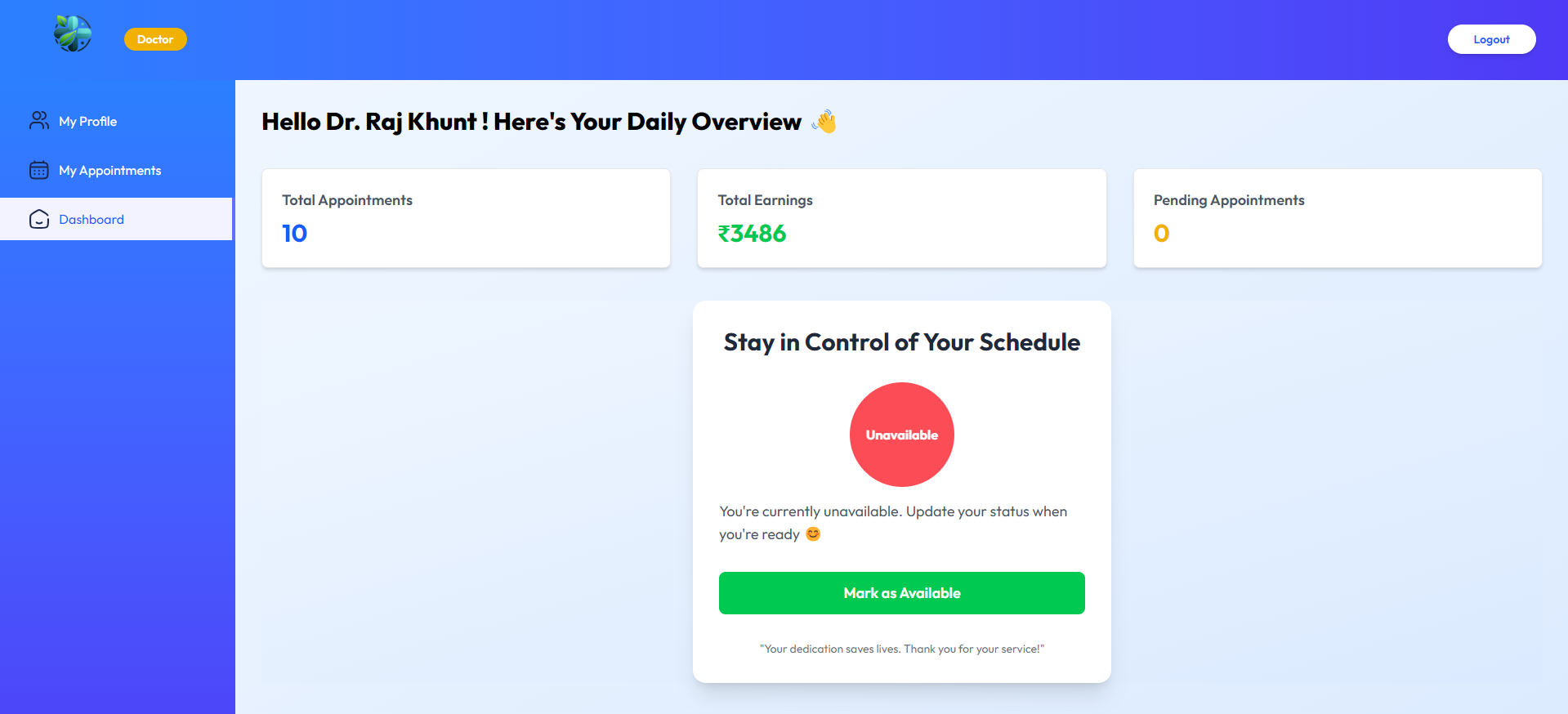
* Enter Doctor Details:



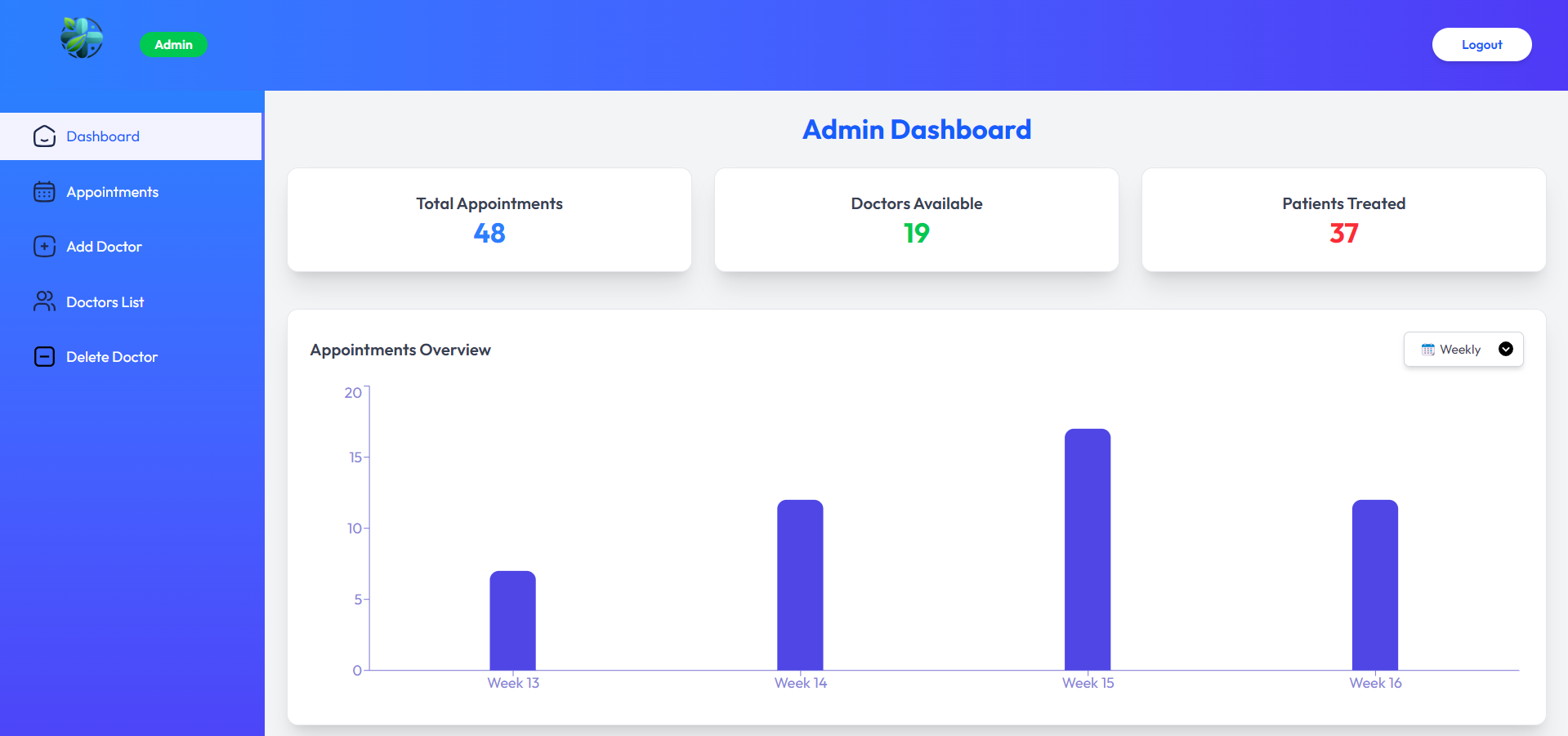
* Enter Doctor ID for Delete:



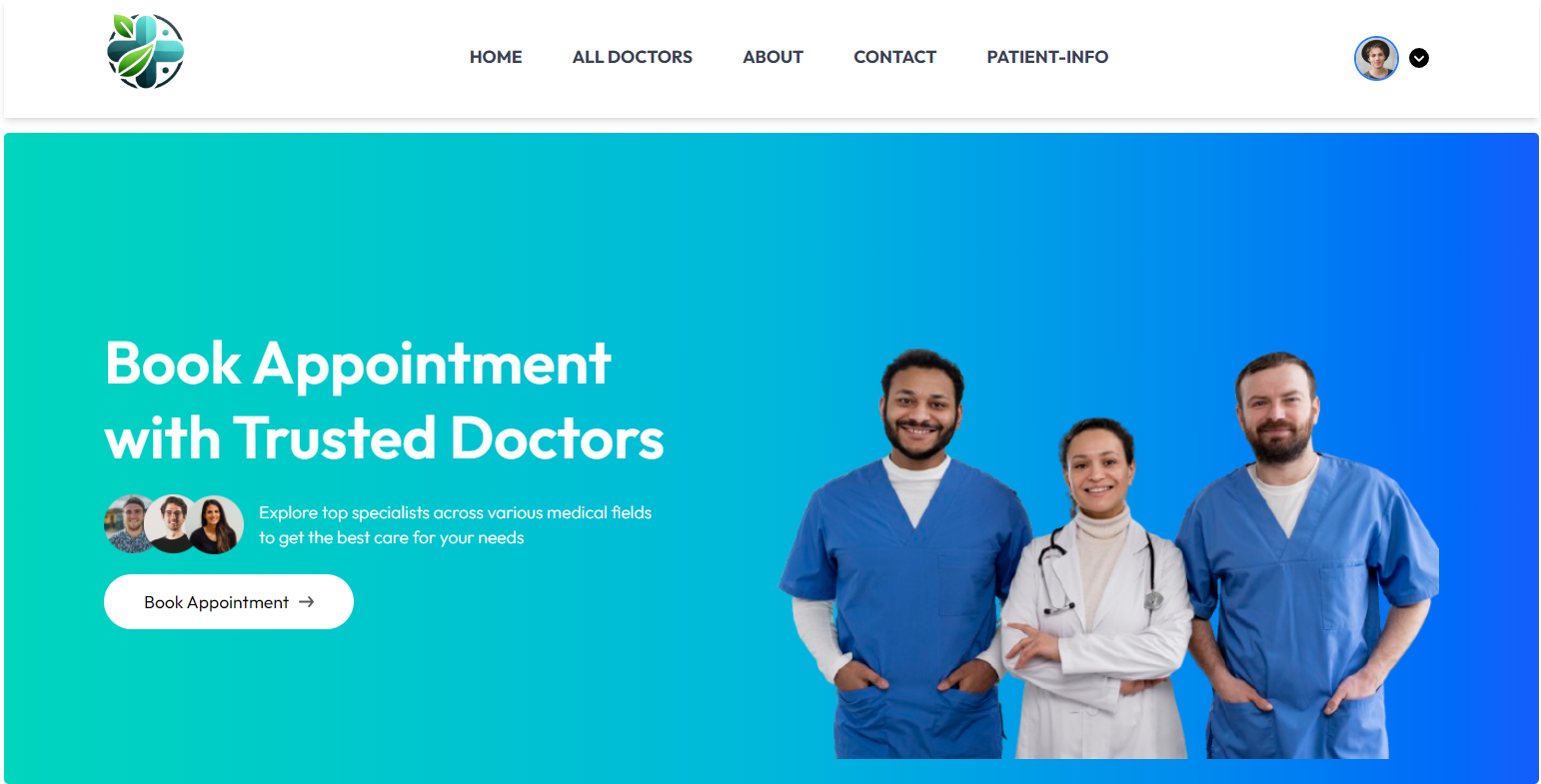
* Doctor Dashboard:



* Admin Management Page:



* Home Page:



* **Data Dictionary**

1. **Users Collection:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Description** |
| Userid | varchar(20) | Primary Key | Unique identifier for each user |
| Name | varchar(20) | Unique, Not Null | Full name of the user |
| Email | varchar(20) | Unique, Not Null | Email of the user (unique) |
| Password | varchar(20) | Not Null | Hashed password for authentication |
| Phone | int(10) | Not Null | User's phone number |
| Gender | varchar(6) | Not Null | Gender of the user (Male, Female, Other) |

1. **Doctors Collection:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Description** |
| doctorid | varchar(24) | Primary Key | |  | | --- | |  |   Unique identifier for each doctor |
| Name | varchar(20) | Not Null | Full name of the doctor |
| Email | varchar(20) | Not Null | Doctor's email (unique) |
| password | varchar(20) | Not Null | Hashed password for authentication |
| Specialization | varchar(20) | Not Null | Doctor's specialization (e.g., Cardiologist) |
| Phone | int(10) | Not Null | Doctor's phone number |
| Experience | varchar(20) | Not Null | Years of experience |
| Availability | Boolean | Not Null | Availability status (true/false) |
| Earnings | int(10) | Not Null | Total earnings from appointments |
| createdAt | Datetime | Default | Timestamp when the doctor was added |
| Userid | Varchar(20) | Foreign Key | Reference to the user |

**3.Appointments Collection:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Description** |
| \_id | varchar(24) | Primary Key | Unique identifier for each appointment |
| patientName | varchar(15) | Not Null | Name of the patient |
| patientPhone | Int(10) | Not Null | Phone number of the patient |
| patient Email | varchar(15) | Not Null | Email of the patient |
| disease Type | varchar(20) | Not Null | Type of disease mentioned by the patient |
| Description | varchar(200) | Not Null | Detailed description of the patient’s condition (200 words max) |
| doctorId | varchar(24) | Foreign Key | Reference to the assigned doctor |
| doctor Name | varchar(20) | Not Null | Name of the assigned doctor |
| appointment Date | Date | Not Null | Scheduled date of the appointment |
| Status | varchar(20) | Not Null | Appointment status (Pending, Completed, Cancelled) |
| Suggestions | varchar(20) | Not Null | Doctor’s suggestions (entered post-consultation) |
| medicines | varchar(20) | Not Null | Medicine prescription given by the doctor |

**4.Admin Collection:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Description** |
| Admin Email | varchar(20) | Unique, Not Null | Email of the admin(unique) |
| password | varchar(20) | Not Null | Password for Authentication |

**SYSTEM TESTING**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test case ID** | **Scenario** | **Purpose** | **Steps to Perform** | **Expected Result** | **Actual Result** | **Status** |
| **TC-01** | Admin login | Verify admin login with correct credentials | 1. Enter valid email & password  2. Click "Login" | Admin should be redirected to the dashboard | Works as expected | PASS |
| **TC-02** | Admin login | Verify admin login with incorrect credentials | 1. Enter invalid email/password  2. Click "Login" | Error message should appear | Works as expected | PASS |
| **TC-03** | Doctor login | Check doctor login functionality | 1. Enter correct doctor email & password  2. Click "Login" | Doctor should be redirected to their dashboard | Works as expected | PASS |
| **TC-04** | Patient Information | Ensure patient can book an appointment | 1. Fill patient details  2. Select doctor & date  3. Click "Book Appointment" | Appointment should be saved in database | Works as expected | PASS |
| **TC-05** | Appointment View | Verify doctor can view their appointments | 1. Login as doctor  2. Navigate to "Appointments" | List of assigned appointments should be displayed | Works as expected | PASS |
| **TC-06** | Prescription & Suggestions | Doctor should add prescriptions & suggestions | 1. Login as doctor  2. Select an appointment  3. Enter suggestions & medicines  4. Click "Save" | Prescription should be stored in the database | Works as expected | PASS |
| **TC-07** | Patient Profile | Verify patient details are stored properly | 1. Patient fills details  2. Click "Save" | Data should be stored in MongoDB | Works as expected | PASS |
| **TC-08** | Auto Status Update | Verify that past appointments change to "Completed" automatically | 1. Book an appointment for a past date  2. Check status the next day | Status should be updated to "Completed" automatically | Works as expected | PASS |
| **TC-09** | Appointment Update | Doctor should update appointment status to "Completed" | 1. Login as doctor  2. Select appointment  3. Click "Mark as Completed" | Status should update in the database | Works as expected | PASS |
| **TC-10** | Patient Profile | Verify patient can view their profile | 1. Login as patient  2. Navigate to "My Profile" | Patient's stored details should be displayed | Works as expected | PASS |
| **TC-11** | Patient Appointments | Verify patient can view their booked appointments | 1. Login as patient  2. Go to “My Appointments” | List of booked appointments should be displayed | Works as expected | PASS |
| **TC-12** | Cancel Appointment | Verify patient can cancel an appointment | 1. Login as patient  2. Go to "My Appointments"  3. Click “Cancel” | Appointment should be removed from active list | Works as expected | PASS |
| **TC-13** | Admin - Add Doctor | Verify admin can add a new doctor | 1. Login as admin  2. Go to "Add Doctor" page  3. Enter doctor details  4. Click “ADD” | New doctor should be added to the database | Works as expected | PASS |
| **TC-14** | Admin - Delete Doctor | Verify admin can delete a doctor | 1. Login as admin  2. Go to “Delete Doctor” page  3. Write a doctorID 4. Click "Delete" | Doctor should be removed from the database | Works as expected | PASS |
| **TC-15** | Admin - Change Doctor Availability | Verify admin can update a doctor's availability | 1. Login as admin  2. Go to “doctor’s list” page 3. Change availability status | Availability should update in the database | Works as expected | PASS |
| **TC-16** | Admin - View All Appointments | Verify that admin can see a list of all booked appointments | 1. Login as Admin  2. Navigate to "All Appointments" page | A list of all patient appointments should be displayed | Works as expected | PASS |
| **TC-17** | Doctor - Change Availability | Verify that a doctor can update their availability status | 1. Login as Doctor  2. Navigate to "Dashboard"  3. Change availability status as “Mark as Available or Not” | Availability should be updated in the database | Works as expected | PASS |
| **TC-18** | Doctor - View Total Appointments | Verify that the doctor can see the total number of appointments on the dashboard | 1. Login as Doctor  2. Navigate to "Doctor Dashboard" | The total count of booked appointments should be displayed | Works as expected | PASS |
| **TC-19** | Doctor - View Pending Appointments | Verify that the doctor can see the number of pending appointments | 1. Login as Doctor  2. Navigate to "Doctor Dashboard" | The number of pending appointments should be displayed | Works as expected | PASS |
| **TC-20** | Doctor - View Total Earnings and Notifications | Verify that the doctor can see their total earnings and notifications on the dashboard | 1. Login as Doctor  2. Navigate to "Doctor Dashboard" | The total earnings should be displayed based on completed appointments | Works as expected | PASS |

**FUTURE ENHANCEMENT**

**1.Token-based Authentication**

Implement secure login for Admin, Doctor, and User using JWT (JSON Web Token) to ensure session safety and prevent unauthorized access.

**2.Mobile App Integration**

Develop a mobile application (using React Native or Flutter) to allow users to book and manage appointments conveniently.

**3.Real-Time Notifications**

Use web sockets to push real-time notifications when appointment status is updated, or reminders need to be sent.

**4.Chat or Messaging System**

Enable patients to chat with doctors before or after the appointment for quick consultation or follow-up.

**References**

* **References**
* **MongoDB** – <https://www.mongodb.com/docs/>
* **Express.js** – <https://expressjs.com/>
* **Node.js** – <https://nodejs.org/en/docs>
* **React.js** – <https://react.dev/>
* **Tailwind CSS** – <https://tailwindcss.com/docs>
* Additional Tools & Software:
* **GitHub Repository -** <https://github.com/Raj-310504/hospital-appointment-system>
* **Postman (For API Testing)**

**THANK YOU**