****

**Name:**

**Muhammad Ahsan Ali**

**Muhammad Jawad**

**Roll No:**

**233569**

**233557**

**Section &Program:**

**BS CS F-23(A)**

**Subject:**

**Mobile Computing Project**

**Course Instructor:**

**Prof Rashaf Jamil Sb**

**Semester:**

**4th**

**Dr KOTTNIS HOSPITAL**

**Appointment Booking App – Full Documentation**

* This Flutter-based appointment booking application is designed to streamline the interaction between users and doctors through a structured system of login, appointment scheduling, profile management, and real-time status tracking of appointments. The app provides an intuitive user interface that guides the user from authentication to booking and managing appointments, ensuring both patients and doctors have seamless access to vital scheduling tools.
* The application begins with a **Login Page**, which is the gateway to all user functionalities. Here, users are required to enter their credentials, such as an email or username and a secure password. The login form includes validation checks to ensure inputs meet the required format. This authentication layer guarantees that only registered users gain access to the core functionalities of the application.
* Upon successful login, the user is directed to the **Home Screen**, which serves as the central navigation hub of the app. From this screen, users can access their profile, settings, appointment scheduling options, and view existing appointments. The layout is designed for ease of use, ensuring that users can quickly find and utilize the functions they need without unnecessary complexity.
* The **User Profile Screen** provides access to the user’s personal information such as name, email address, and other optional details. This screen not only allows users to view their data but also offers an editing option, enabling users to update their profile information as needed. One of the key features accessible through this section is the ability to change the account password.
* Through the **Change Password** section within the profile, users can update their credentials securely. They must enter their current password along with a new password that meets strength requirements. This feature ensures ongoing security of user accounts, and password validation is handled both on the frontend and backend for optimal protection.
* The **Settings Page** includes a set of customizable preferences for the user. Although the document provides limited visual detail on this page, typically, this would contain options like notification settings, theme customization (light or dark mode), language preferences, and possibly account-related options like deactivation or logout.
* Central to the app’s purpose is the **User Appointment Screen**, where users can view all their scheduled appointments. This screen lists the status of each appointment, showing whether it is still pending, confirmed by the doctor, or has been cancelled. It provides a real-time view into the appointment lifecycle and keeps the user informed about their upcoming interactions.
* To schedule a new appointment, users navigate to the **Book Appointment Screen**. Here, they can choose from a list of available doctors, select a preferred date and time, and submit a booking request. This request is then sent to the selected doctor for confirmation. The user-friendly interface includes dropdowns, date pickers, and validation to make the booking process efficient and error-free.
* Once an appointment is booked, users are redirected to a screen that shows the **status of the booking request**. At this stage, the appointment is marked as “pending,” awaiting action from the doctor. This ensures that the user has clear visibility over their request and can expect an update soon.
* On the **Doctor Appointment Screen**, doctors are able to view all pending appointment requests sent by users. The screen provides appointment details, including user information and requested times. Doctors are given the option to either confirm or cancel these appointments. This screen acts as a decision center for doctors to manage their availability and commitments.
* After a doctor has responded to a request, the **User Appointment Screen** is updated to reflect the doctor's decision. If the appointment is confirmed, it is marked as scheduled, giving the user assurance of their booking. If the appointment is cancelled, the user is notified, possibly along with a reason for the cancellation. This real-time feedback loop ensures transparency and improves user trust in the platform.
* In conclusion, this application provides a robust platform for managing appointments between users and doctors. It combines secure authentication, profile and password management, real-time appointment tracking, and responsive interfaces for both users and doctors. The application can be further enhanced by integrating features like push notifications, in-app messaging, or calendar synchronization to improve communication and usability.

## 🧩 **Problem Statement**

In many healthcare environments, especially in developing regions, managing doctor appointments is still handled manually or with inefficient systems. This often leads to:

* Long waiting times
* Missed or forgotten appointments
* No real-time updates for users
* Poor communication between doctors and patients

There is a need for a **digital solution** that enables **secure, fast, and efficient appointment scheduling**, real-time status tracking, and simplified doctor-patient interaction.

**Problem:** Lack of a user-friendly, real-time, and automated system to manage doctor appointments efficiently for both patients and healthcare providers.

## 🎯 **Objectives of the Application**

The primary goals of this Flutter-based appointment booking system are:

1. **User Authentication**
   * Secure login for users and doctors with input validation.
2. **Profile Management**
   * Allow users to view and update personal details.
3. **Password Management**
   * Provide functionality for users to change passwords securely.
4. **Appointment Booking**
   * Enable patients to book appointments with available doctors based on time slots.
5. **Real-Time Appointment Status**
   * Let users track their appointment status (pending, confirmed, cancelled).
6. **Doctor Dashboard**
   * Doctors can view appointment requests and respond (confirm or cancel).
7. **User Experience**
   * Offer a clean, intuitive interface with dropdowns, date pickers, and dynamic updates.
8. **Scalability** (Future Enhancements)
   * Incorporate push notifications, in-app chat, and calendar sync.

## 🔍 **Background Study – Similar Applications**

Here are some popular appointment booking applications and a brief comparison:

### 🟢 **Oladoc:**

* Allows booking doctor appointments, ordering medicine, and consulting online.
* Strengths: Multi-feature integration, doctor ratings, verified listings.
* Limitation: Slightly complex for users just needing appointment booking.

### 🟢 **Zocdoc:**

* US-based platform for scheduling doctor visits.
* Strengths: Insurance compatibility check, real-time availability.
* Limitation: Region-locked and lacks offline support in many countries.

### 🟢 **Marham.App(Pakistan)**

* Primarily a health & fitness tracker but includes doctor consultation.
* Strength: Integration with fitness and diet data.
* Limitation: Not primarily focused on appointments.

### 🔵 **Comparison with Proposed App**

| **Feature** | **Oladoc** | **Zocdoc** | **Proposed App** |
| --- | --- | --- | --- |
| * User-friendly UI | * ✅ | * ✅ | * ✅ |
| * Real-time Status Tracking | * ✅ | * ✅ | * ✅ |
| * Simple Doctor Dashboard | * ❌ | * ❌ | * ✅ |
| * Lightweight Flutter App | * ❌ | * ❌ | * ✅ |
| * Offline/Low-data Friendly | * ❌ | * ❌ | * ✅ (planned) |
| * Chat / Push Notifications | * ✅ | * ✅ | * ❌ (planned) |

# 📱 DR KOTTNIS HOSPITAL

# Appointment App

## 📋 Overview

This Flutter-based application allows users to manage appointments with doctors. It includes a login system, user profile management, booking capabilities, and real-time doctor confirmation or cancellation of appointments.

## 🔐 1. Login Page

* **Functionality**: Secure user login.
* **Fields**: Email/username and password.
* **Validation**: Ensures proper input formatting and authenticates against backend/user database.

## 🏠 2. Home Screen

* **Functionality**: Acts as the main dashboard after login.
* **Features**:
  + Navigation to profile, settings, appointment booking.
  + Displays general app functionality or user-specific information.

## 👤 3. User Profile

* **Functionality**: Displays user information (e.g., name, email, etc.).
* **Actions**:
  + Edit personal details.
  + Access password change.

## 🔐 4. Change Password

* **Location**: Accessed through User Profile.
* **Functionality**:
  + Allows the user to change their current password.
  + Includes input validation (e.g., old password match, new password strength).

## ⚙️ 5. Settings

* **Functionality**: Customize user preferences.
* **Potential Features**:
  + Notification toggles.
  + Theme selection.
  + Account or privacy settings.

## 📆 6. User Appointment Screen

* **Functionality**: Shows all user-booked appointments.
* **States**:
  + Pending (awaiting doctor response).
  + Confirmed or cancelled by the doctor.

## 🩺 7. Book Appointment Screen

* **Functionality**:
  + User can select a doctor, date, and time.
  + Submits request to the backend for doctor confirmation.
* **Features**:
  + Dropdown or search for doctors.
  + Calendar/time picker.

## ✅ 8. Post-Booking Screen (User View)

* **Functionality**: Shows current status of the appointment.
* **Status Display**: Appointment is submitted and pending doctor action.

## 👨‍⚕️ 9. Doctor Appointment Screen

* **Functionality**: Doctor views pending appointments.
* **Actions**:
  + Confirm or cancel appointments.
  + View user info and details.

## ☑️ 10. After Doctor Response (User View)

* **Functionality**: Updated view showing doctor’s response:
  + Confirmed: Displays scheduled appointment.
  + Cancelled: Notifies user of cancellation and reason (if applicable).

## 🛠️ Technologies Used

* **Flutter**: For frontend UI.
* **Dart**: Programming language.
* **Firebase / Custom Backend (assumed)**: Authentication and database services.
* **State Management**: Likely Provider / Riverpod / GetX (please specify if used).

## 📈 Future Enhancements (Optional)

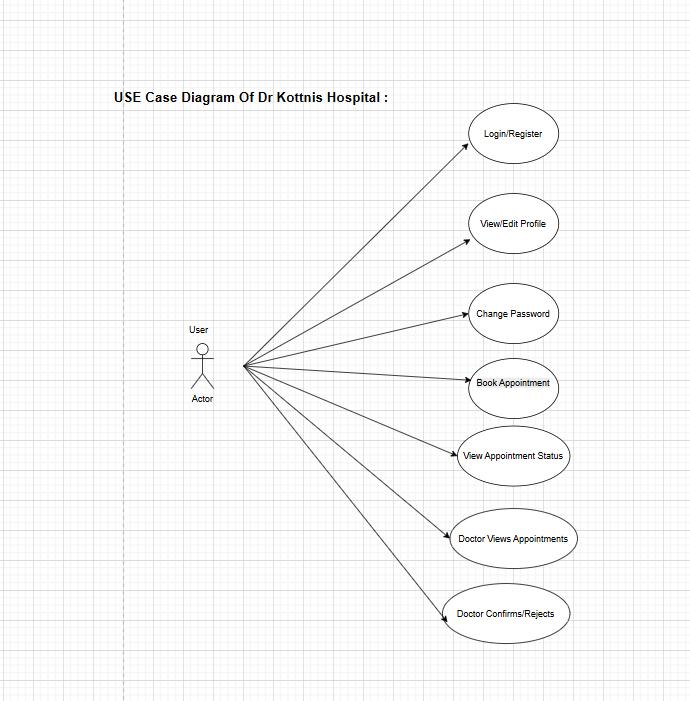
* Chat between user and doctor.
* Push notifications for updates.
* In-app calendar integration.
* Doctor availability scheduling.

**Use Case Diagram Of Dr Kottnis Hospital:**

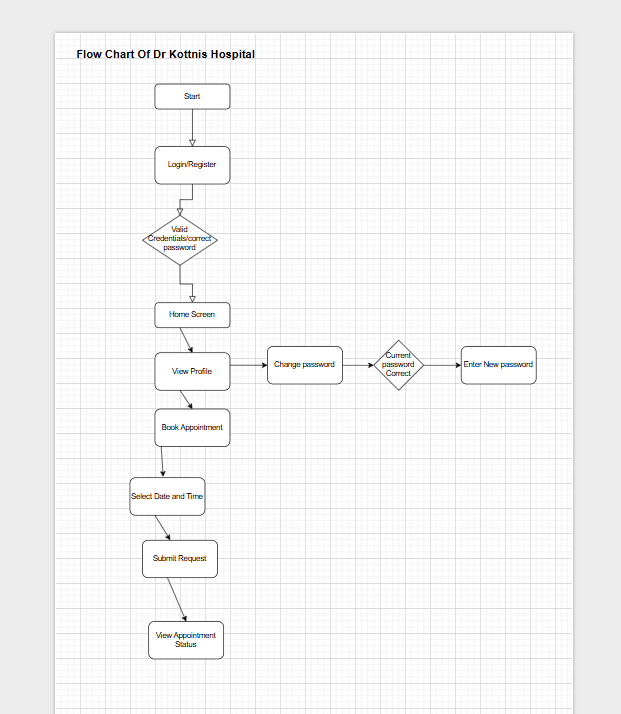
**Actors**: User, Doctor, System

**Use Cases**:

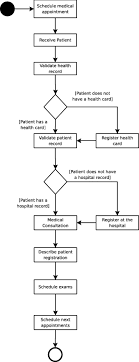
* Login/Register
* View/Edit Profile
* Change Password
* Book Appointment
* View Appointment Status
* Doctor Views Appointments
* Doctor Confirms/Rejects



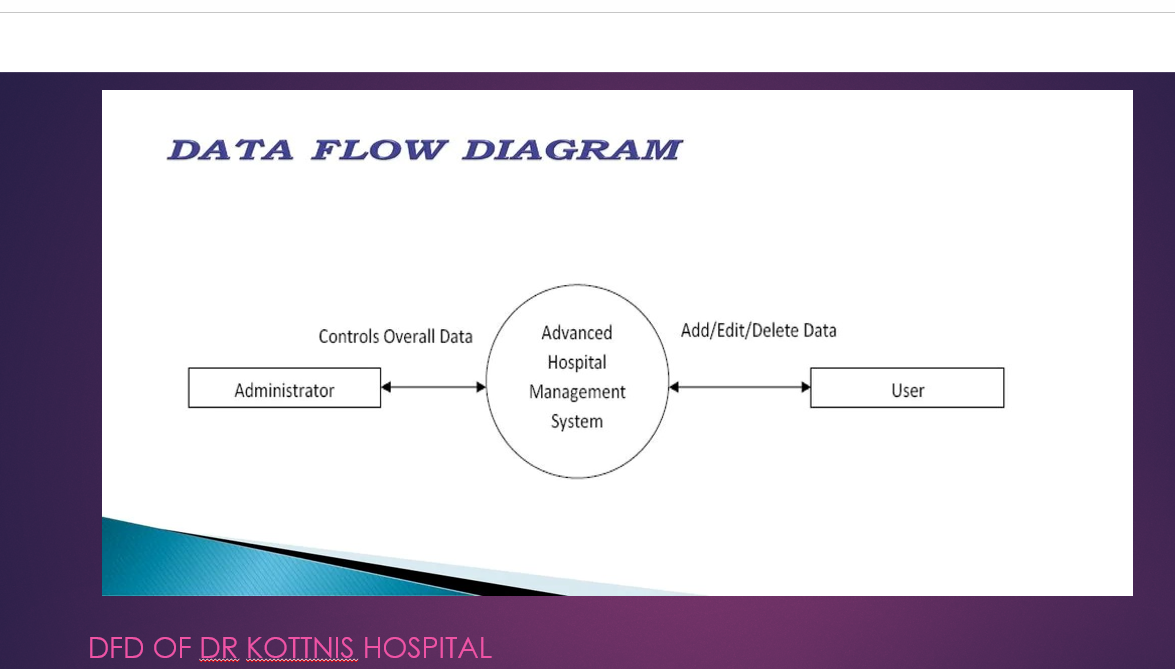
**FlowChart of Dr Kottnis Hospital:**



**Activity Diagram OF Dr Kottnis Hospital:**

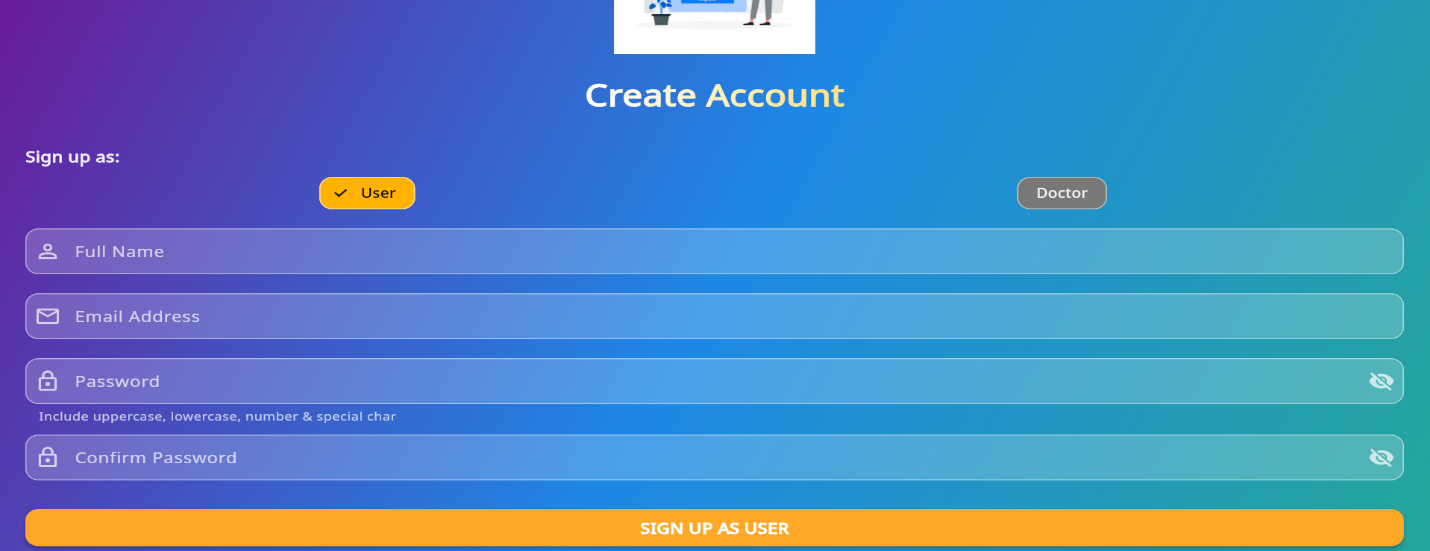
****

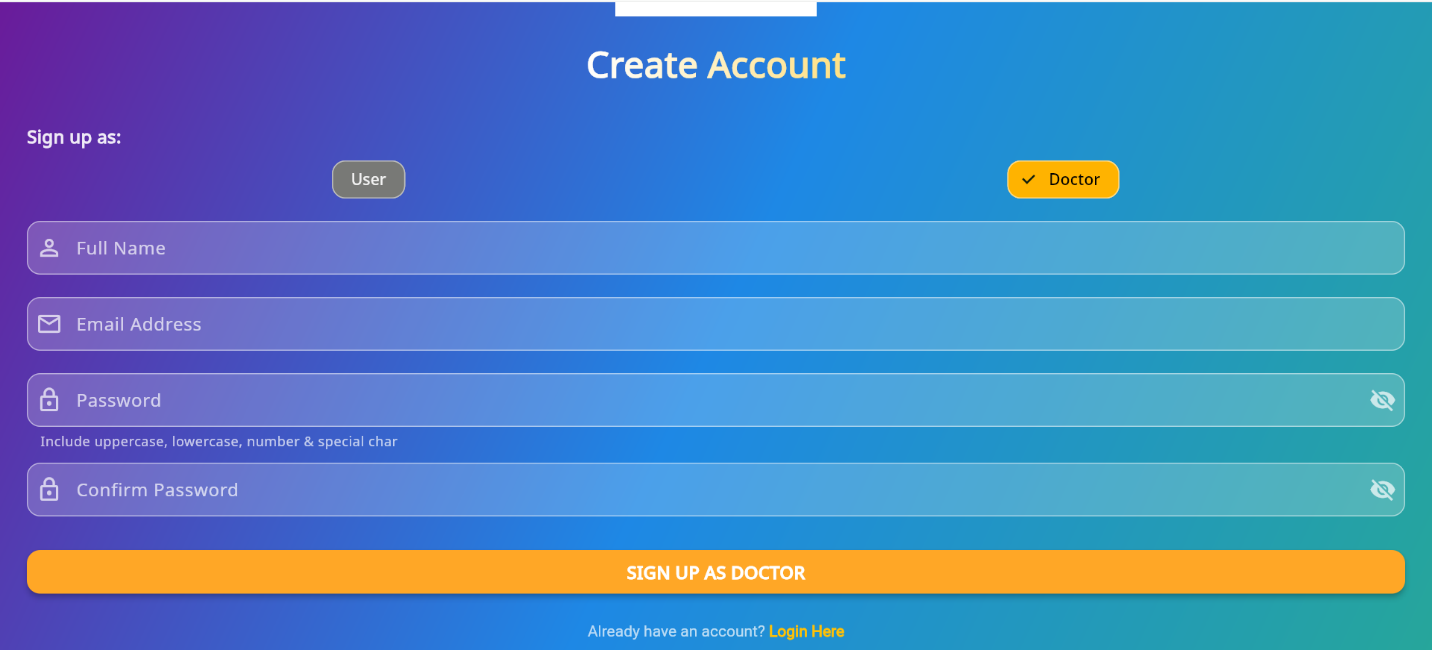
**DFD Diagram of Dr Kottnis Hospital:**

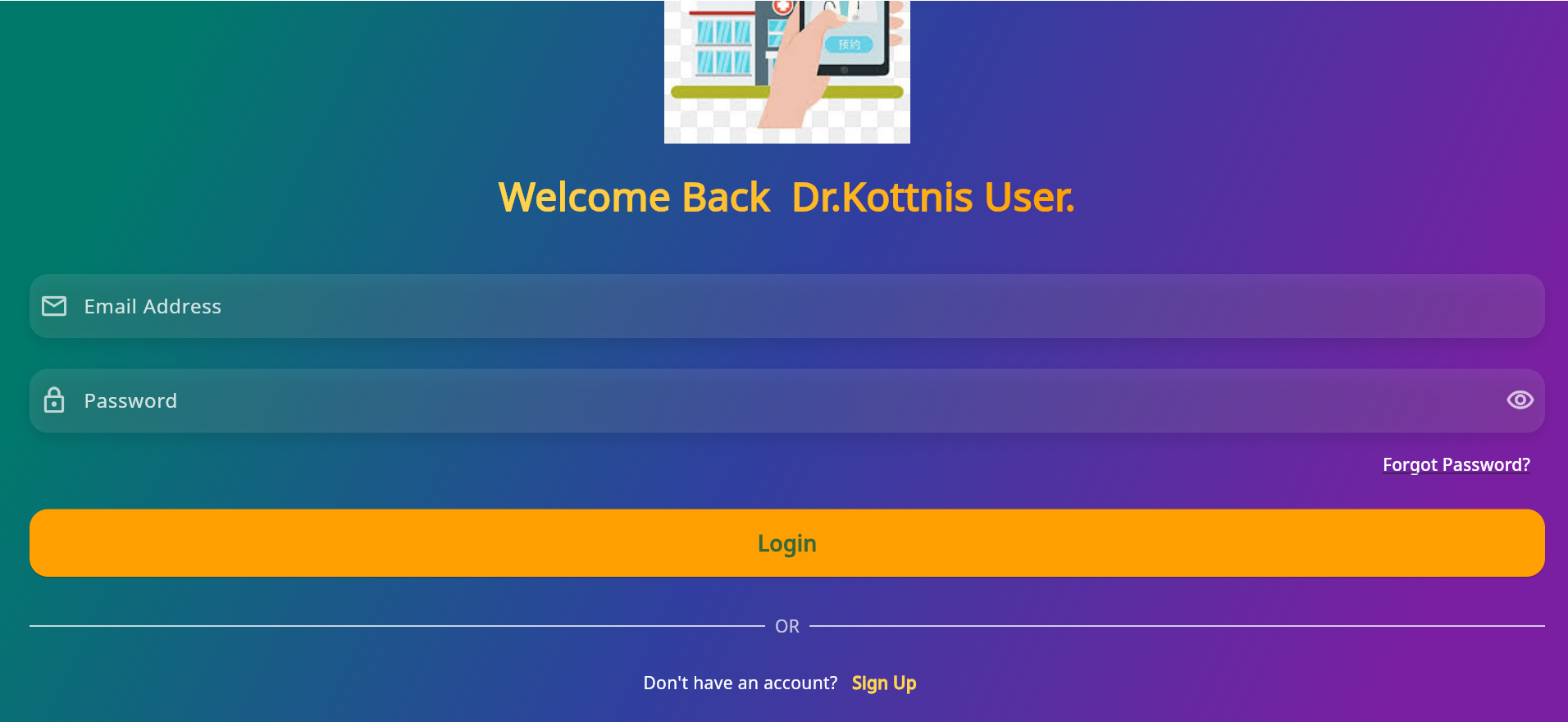


**ScreenShots of App:**

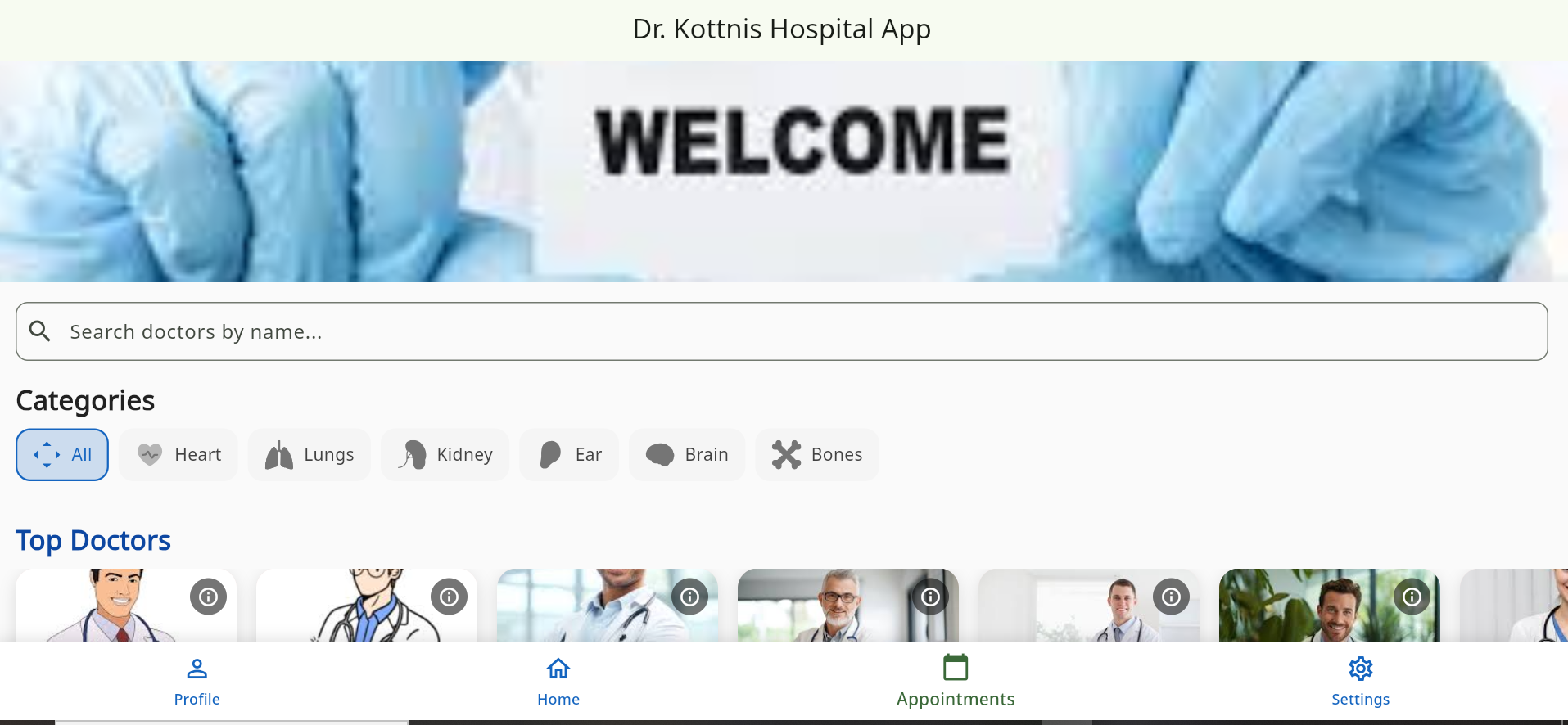
* **Register As a User**



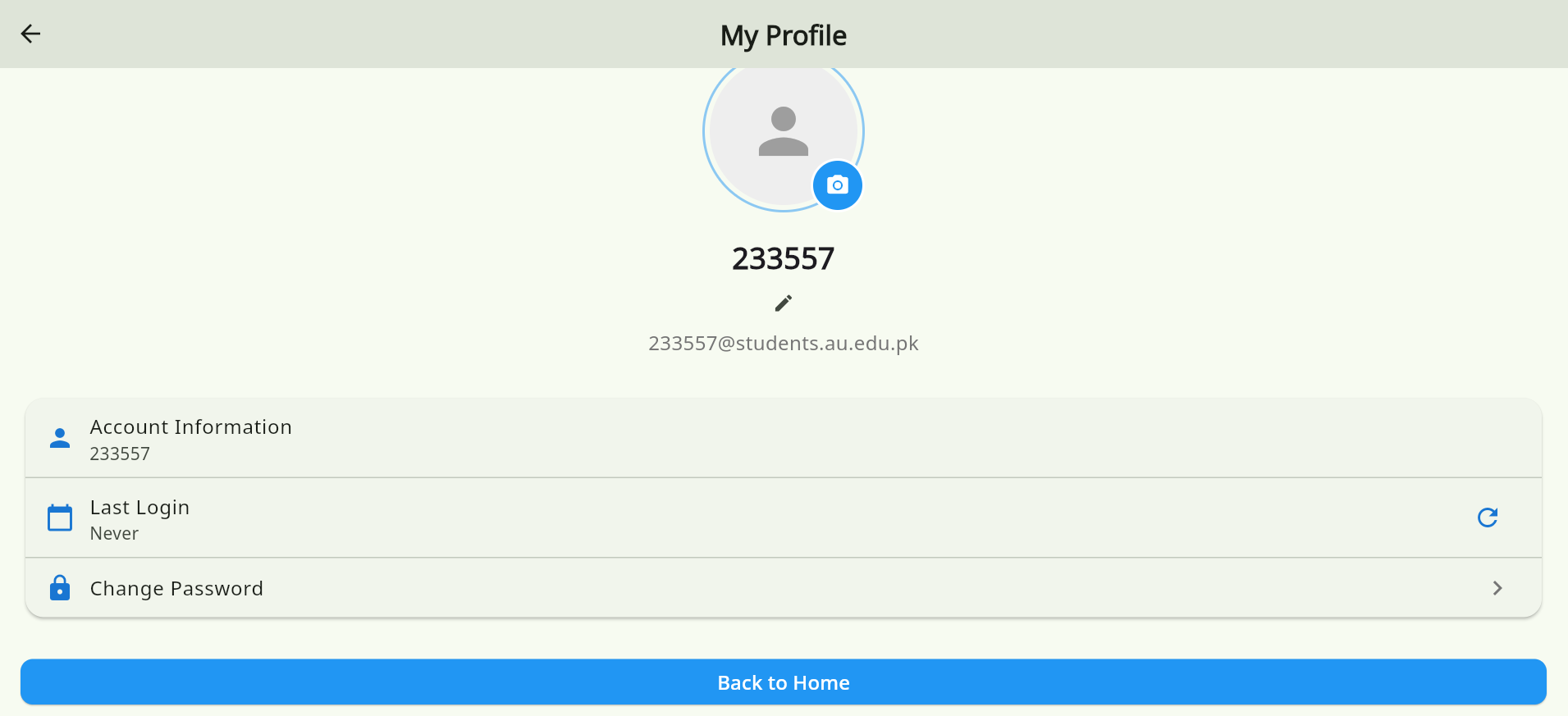
* **Register As a Doctor:**
* **Login page**



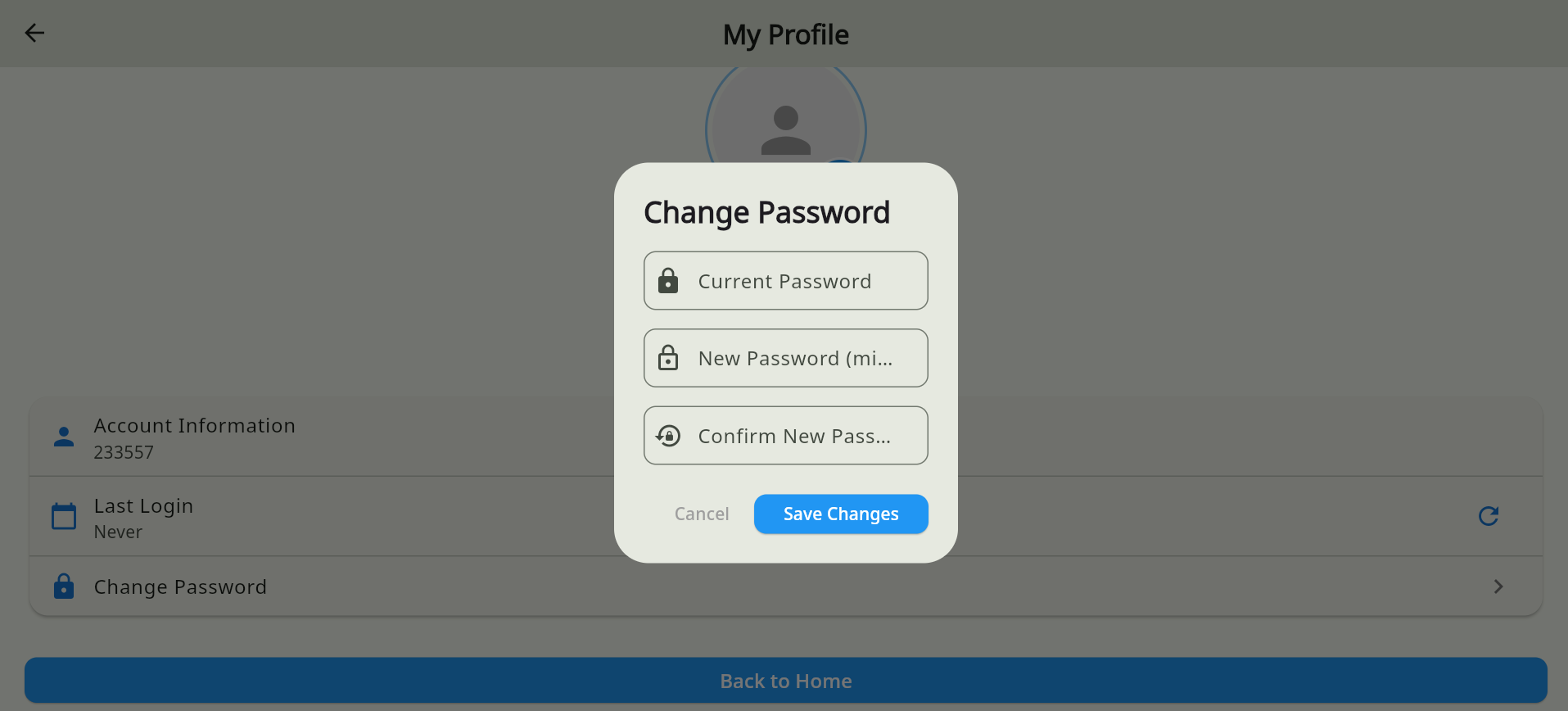
* **Homescreen**



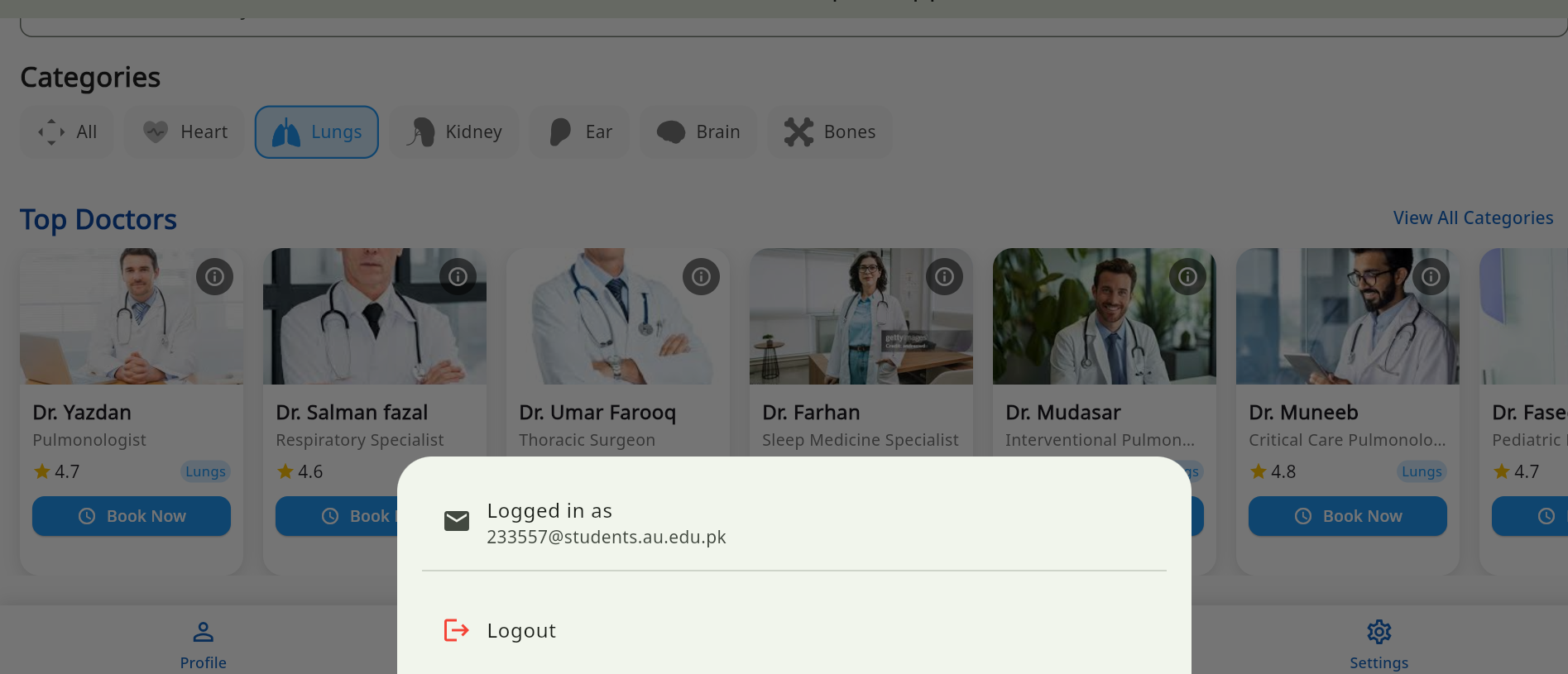
* **Userprofile:**



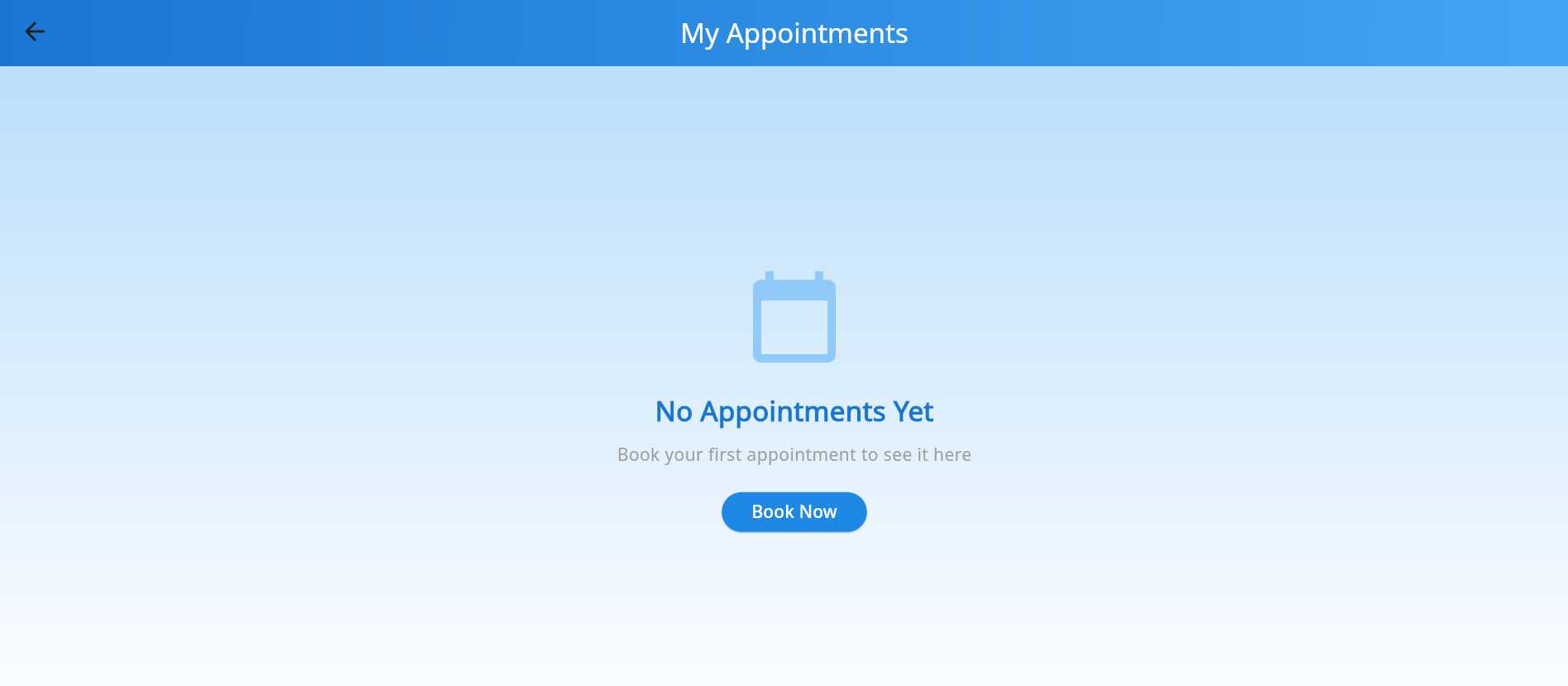
* **Change password in user profile:**



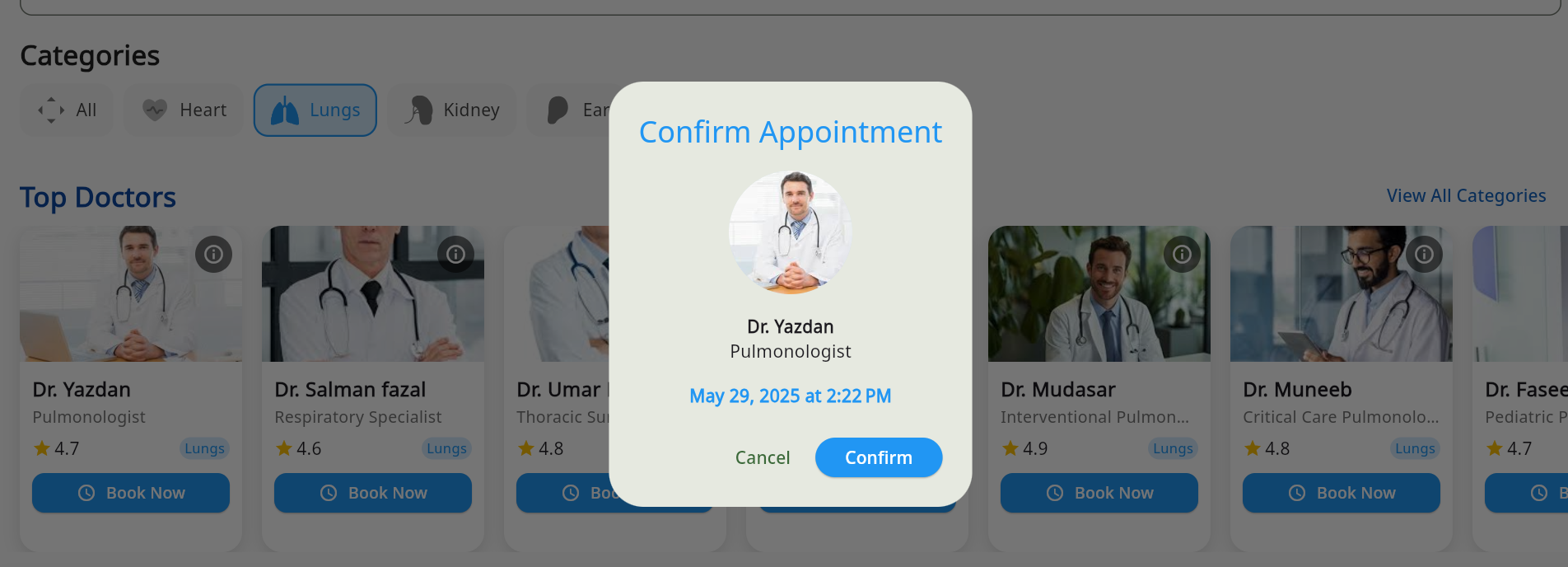
* **Setting:**



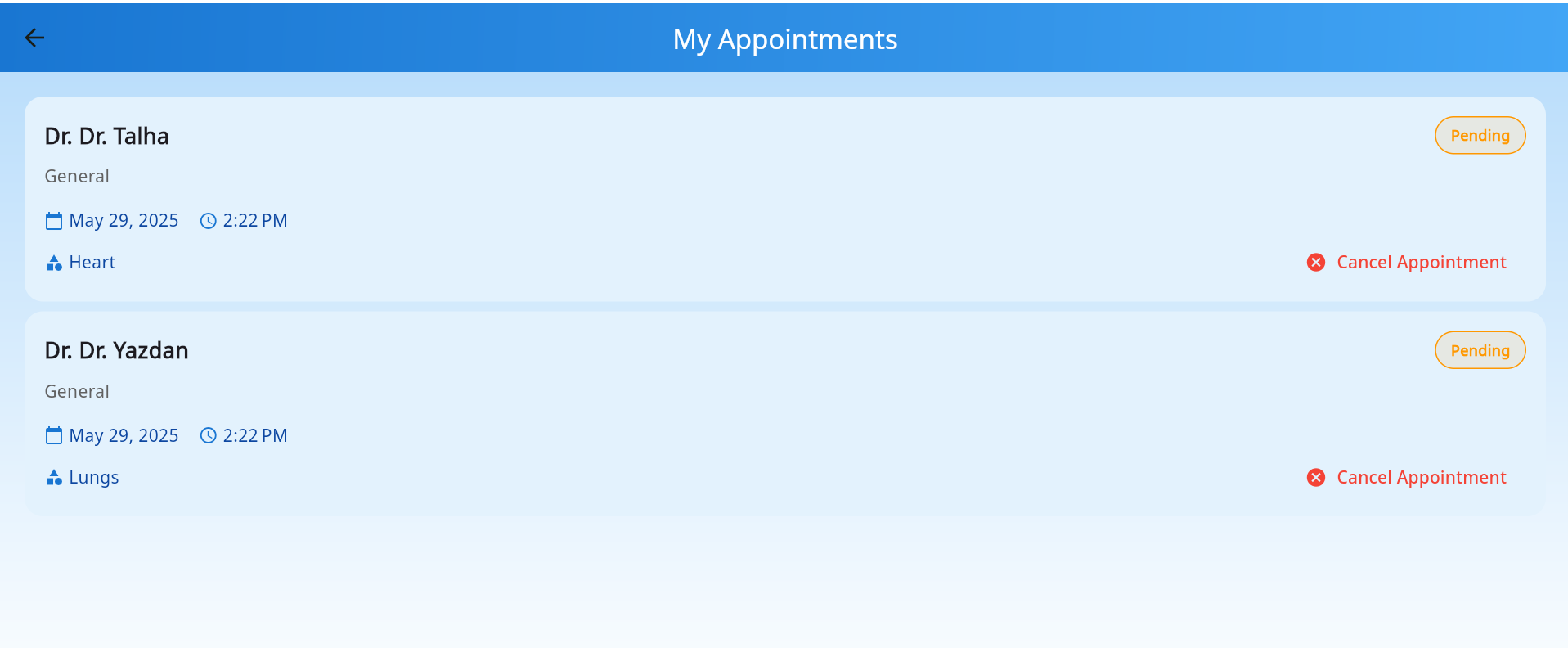
* **User appointment screen :**



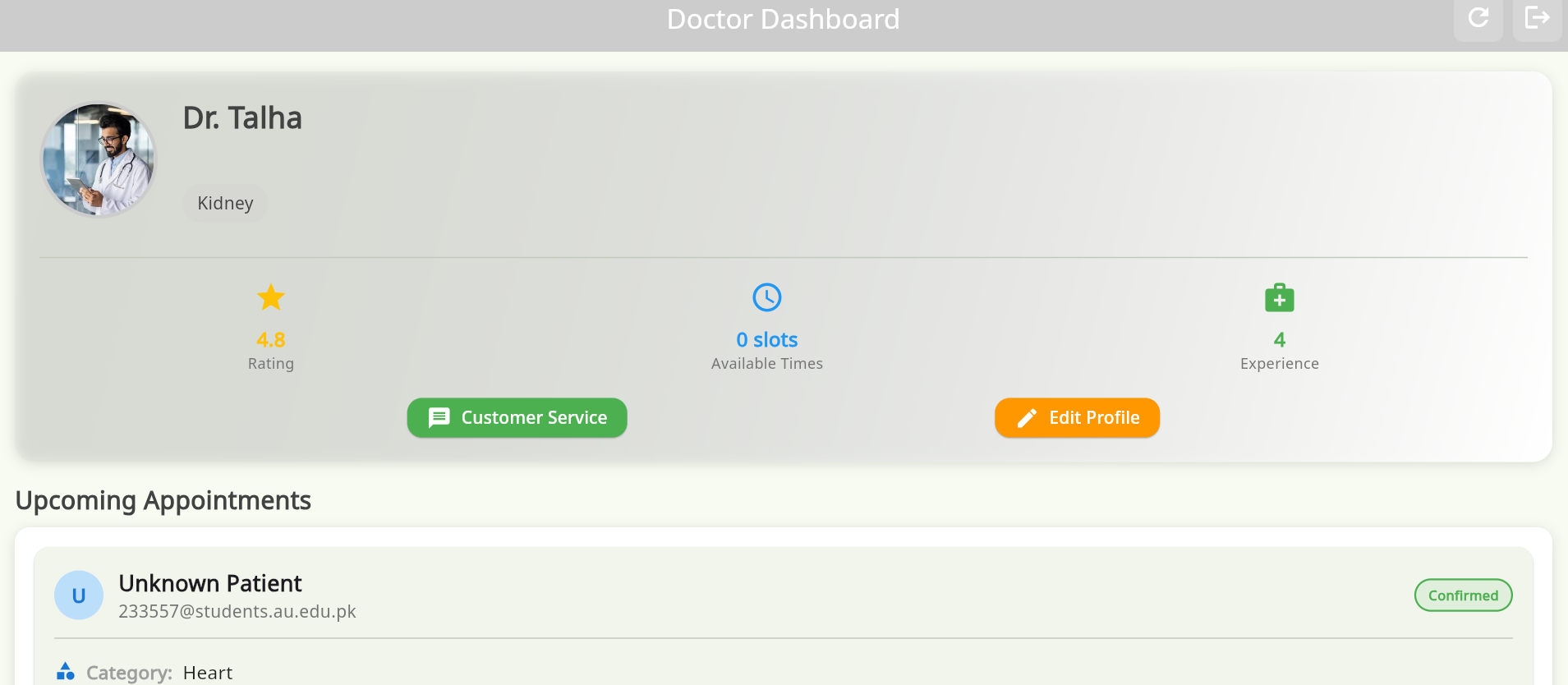
* **Book appoimnet user:**



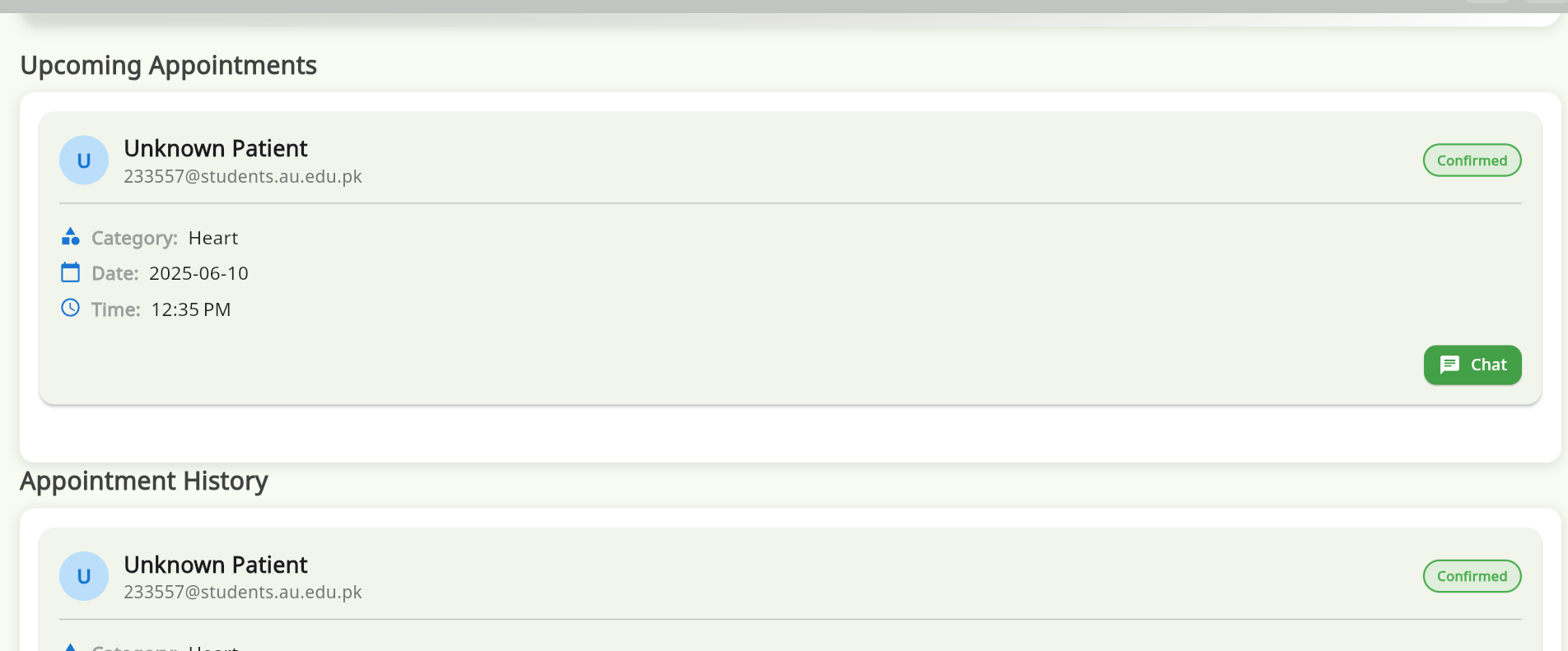
* **After book appoiment screen now:**



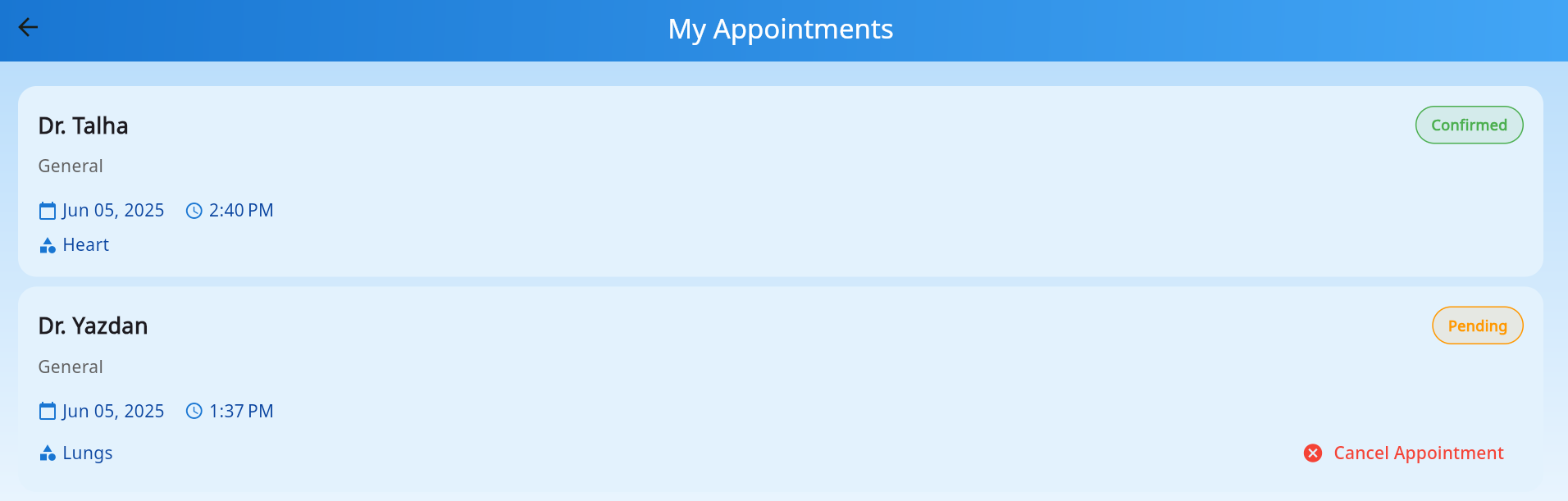
DOCTOR DASHBOARD:



* **Doctor appoimnet screen after user book showing in doctor appoimnet screen:**



* **After confirm the user appointment by doctor now user appoimner screen is this:**



* **Github Repository Link:** [https://github.com/muhammadjawad557/Mobile-Computing-App-File](%20https:/github.com/muhammadjawad557/Mobile-Computing-App-File)

**THE END!**