VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590 014

A Mini - Project Report

On

"DOCTOR APPOINTMENT SYSTEM"

Submitted in partial fulfillment of the requirements for the **MINI PROJECT (BCD586)** course of the 5th semester

Bachelor of Engineering

In

Computer Science & Engineering (DATA SCIENCE)

Submitted by

Mr. Chinmay C S (4AI22CD013)

Mr. K Shivadarshan (4AI22CD028)

Mr. Sunil kumar A B (4AI22CD056)

Mr. Yashas Xavier (4AI22CD062)

Under the guidance of

Mrs. SHALINI I S B.E., M.Tech.

Assistant Professor

Department of CS&E (DATA SCIENCE)

Adichunchanagiri Institute of Technology

CHIKKAMAGALURU - 577102

2024-25

ABSTRACT

The Doctor Appointment System is an online platform designed to streamline the process of scheduling and managing doctor appointments. It provides a convenient and efficient solution for patients to book appointments with their preferred healthcare providers at their convenience. The system allows users to search for doctors based on their specialization, availability. Patients can view detailed profiles of doctors, including their qualifications, experience, and specialization. Upon selecting a doctor, patients can choose a suitable date and time for their appointment, which will be confirmed through the system. Doctors can manage their schedules, track appointments, and view patient details. Additionally, the system includes features for appointment cancellations and rescheduling, ensuring flexibility for both patients and doctors. The platform may also provide reminders via email or SMS to ensure that appointments are not missed. This system reduces the time spent on manual scheduling, minimizes the risk of errors, and enhances the overall experience for both patients and healthcare providers. It aims to improve healthcare accessibility and efficiency by digitizing and automating appointment booking processes.

ACKNOWLEDGEMENTS

We express our humble pranamas to his holiness **Divine Soul Parama Poojya Jagadguru Padmabushana Sri Sri Sri Dr.Balagangadharanatha Maha Swamiji** and **Parama Poojya Jagadguru Sri Sri Sri Dr. Nirmalanandanatha Maha Swamiji Pontiff, Sri Adichunchanagiri Maha Samsthana Matt and Sri Sri Gunanatha Swamiji,** Chikkamagaluru branch, Sringeri who have showered their blessings on us.

The completion of any project involves the efforts of many people. We have been lucky enough to have received a lot of help and support from all quarters during the making of this project, so with gratitude, we take this opportunity to acknowledge all those whose guidance and encouragement helped us emerge successful.

We express our gratitude to **Dr. C K Subbaraya**, Director, Adichunchanagiri Institute of Technology.

We express our sincere thanks to our beloved principal**, Dr. C T Jayadeva** for having supported us in our academic endeavors.

We are also indebted to **Dr. Adarsh M J,** HOD of CS&E (DATA SCIENCE) Department, for the facilities and support extended towards us.

We thank our project coordinator **Mrs. Shilpa K V** Asst. Professor, Department of CS&E (DATA SCIENCE), for her lively correspondence and assistance in carrying on with this project.

We are thankful to the resourceful guidance, timely assistance and graceful gesture of our guide **Mrs. Shalini I S,** Asst. Professor, Department of CS&E (DATA SCIENCE), who has helped us in every aspect of our project work.

We would be very pleased to express our heart full thanks to all the teaching and non-teaching staff of CS&E (DATA SCIENCE) Department and our friends who have rendered their help, motivation and support.

Chinmay C S

K Shivadarshan

Sunil Kumar A B

Yashas Xavier

CONTENTS

CONTENTS

List of Tables

List of Snapshots

Chapter 1

Introduction

1.1 Background

- **Context**: A doctor appointment system is a software application or platform designed to facilitate and streamline the process of scheduling, managing, and tracking medical appointments between patients and healthcare providers.
- **Problem**: To design and develop an Online Doctor Appointment System that allows patients to book, reschedule, and cancel appointments with doctors online, reducing no-show rates, improving patient experience, and enhancing healthcare efficiency.
- **Opportunity**: The background opportunity for a Doctor Appointment System is based on the convergence of several key factors: growing demand for healthcare services, patient-centric expectations, regulatory and security requirements, and technological advancements.

1.2 Problem Statement

- **Overview of the Problem**: The current methods of appointment systems in many, clinics and hospitals often rely on manual entry, which increases the chances of wait time and consumes valuable more time. Doctors and patients alike struggle with the process of consulting patients.
- **Specific Issues**:
    - Overbooking and Double Bookings
    - Time-consuming manual processes.
    - No-Shows and Late Cancellations.
    - Inefficient Use of Resources (Doctors, Nurses, Rooms).
    - Poor User Experience (UX) for Patients.

1.3 Objective of the System

- The primary objective of the **Doctor Appointment System** enable patients to easily book, reschedule, or cancel appointments online ,provides seamless experience, minimize the waiting time of patients, reduce no-show rates by reminding patients of their appointments.
- **Key Goals**:
- Develop a user-friendly interface for appointment booking.
- Implement a flexible calendar system for doctors and patients.
- Allow patients to view available time slots in real-time.
- Implement an efficient scheduling algorithm to optimize appointment slots.
- Allow patients to check in online and update their status in real-time.
- Allow patients to confirm, cancel, or reschedule their appointments through reminders.

1.4 Significance of the System

- **Efficiency**: Reduces the time staff spend on phone calls and manual booking and allows patients to quickly find and book available time slots, reducing wait times.
- **Accuracy**: Timely access to appointments and efficient management of chronic conditions through regular follow-ups improve health outcomes. This is supported by numerous studies in healthcare management.
- **Real-Time Monitoring**: Patients receive real-time updates about appointment availability, wait times, and any changes to their schedules.
- **Data Analytics**: Analyze the times and days when appointments are most frequently booked to optimize availability.
- **Cost-Effective**: Decreased number of administrative staff needed for appointment management, reduced time spent on phone calls and paperwork.

1.5 Scope of the Project

- **In Scope**:
  - Development of a web-based application.
  - Role-based access (Admin, Doctor, Patient).
  - Display real-time availability of doctors and appointment slots.
  - Enable patients to reschedule or cancel appointments online.
  - Maintain detailed patient profiles with medical history, contact information, and appointment history.
  - Allow new patients to register and create profiles.

1.6 Methodology

- **Approach**: The system will be developed using a **web-based platform**, leveraging modern technologies such as **HTML, CSS,** and a backend language like **PHP**. The system will interact with a **relational database** (like **MySQL** or **PostgreSQL**) to store and retrieve attendance data.
- **Agile Development**: The system will follow an **Agile development** methodology, involving iterative design and feedback cycles to ensure that the system meets the needs of users at each stage of development.
- Testing: The system will be tested through a combination of unit testing, integration testing, and user acceptance testing to ensure functionality and user satisfaction.

1.7 Target Audience

- **Doctors**: Doctors will use the system to view their appointments according to their availability in their session.
- **Patients**: Patients will be able to view the availability of doctors with their specialization and session and can book their appointments.
- **Administrators**: Administrators will manage the system, handle user accounts, and oversee the entire bookings, availability of doctors, patients registered across the platform.

1.8 Overview of the Report

This report is structured into several chapters that detail the development and design of the **Student Attendance System**. The following chapters include:

- **Chapter 2: System Design** – Describes the architecture and design of the system.
- **Chapter 3: Implementation** – Discusses the system's development and the technologies used.
- **Chapter 4: Testing and Validation** – Details the testing process and results.
- **Chapter 5: Results and Discussions** – Presents and results obtained and discusses the limitations
- **Chapter 6: Conclusion and Future enhancement -** Summarizes the project and suggests future improvements.

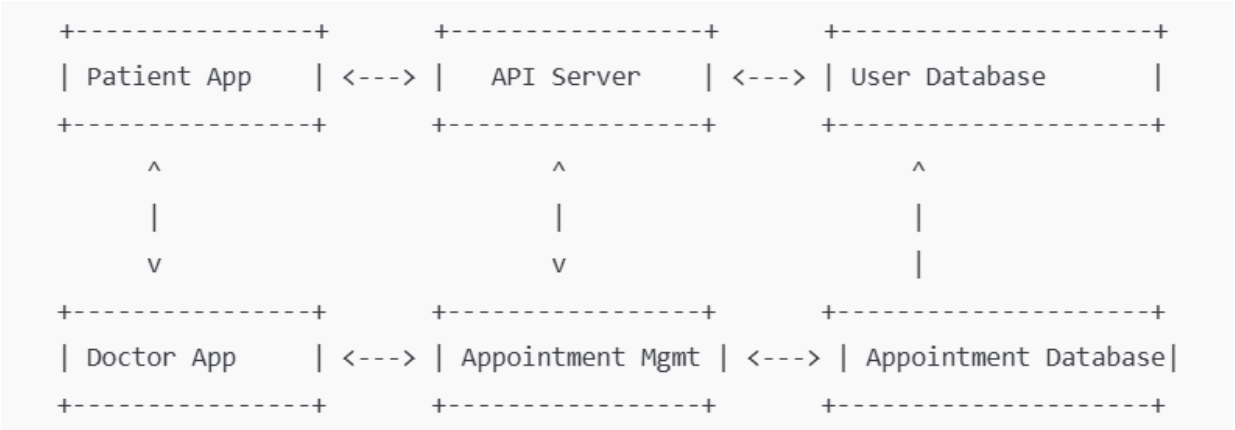
Chapter 2

System Design

This chapter describes the technical design of the Doctor Appointment System, explaining its architecture, components, and how they work together to track and manage all the data in the admin dashboard. The design approach aims to make appointments accurate, efficient, and user-friendly.

2.1 System Architecture

- **High-Level Overview**: The system follows a client-server model where users access the system through a web interface. The backend processes requests, manages business logic, and interacts with a database to store and retrieve data.
- **Architecture Diagram**:



➢ **Frontend (UI):**

Patients:

- Web portal for patients to book, cancel, or reschedule appointments, view doctors' schedules.
- Integration with calendar appointment reminders.

Admin/Support Staff:

- Admin portal for managing appointments, doctor availability, patient data, and generating reports.

Healthcare Providers:

- Web portal for doctors to manage their schedules, view patient records, and conduct virtual consultations.

➢ **Backend server:**

- **Appointment Scheduling**

Service:

- Manages the logic for appointment booking, cancellations, rescheduling, and real-time slot availability.

Doctor Availability Service:

- Manages doctor availability (working hours, holidays, and specific time slots) and integrates with the appointment scheduling system**.**

Patient Management Service:

- Manages patient profiles, medical history, and personal details.

Doctor Management Service:

- Manages doctor profiles, specialties, certifications, and availability.

Database:

Relational Database (e.g., MySQL, PostgreSQL):
Stores structured data such as patient profiles, doctor schedules, appointment history.

Components:

- **Frontend**: A web interface where doctors, patients, and admins interact with the system.
- **Backend Server**: It is responsible for handling the core business logic, processing data, ensuring communication between different components, and interacting with the database and external services
- **Database**: Stores user data, appointment records.

2.2 Module Design

- The system is divided into functional modules, each handling a specific task.

2.2.1 User Authentication Module

1. User Roles & Permissions:

The authentication system must distinguish between different user roles and grant appropriate permissions. Common roles include:

- **Patient**: Can book, cancel, and view appointments, as well as access their medical history.
- **Doctor**: Can view their appointments and access patient records for consultation.
- **Admin**: Can manage doctors, patients, appointments, and oversee system operations

(e.g., adding/removing doctors, verifying patient identity, etc.).

2. **Registration (Sign-Up):**

- **Patient Registration**:
    - Collects basic details like name, email, phone number, password, and medical details (if needed).
    - Optionally, users can sign up using **social logins** (Google, etc.) to simplify the process.

2.2.2 Patient Management Module

1. Patient Registration & Profile Management

New patients can register on the platform by providing their basic details such as name, contact information (phone, email), gender, date of birth, and medical history.

2. Appointment Booking:
- Patients can book appointments with doctors based on available time slots.
- They can select the **doctor's specialty**, view **doctor profiles**, and check **availability** before booking an appointment.

3. Appointment History:
- Patients can view past and upcoming appointments with doctors, including appointment date, time, status (confirmed, cancelled, completed), and consultation details.

4. Appointment Modifications and Cancellations:
- Patients can **modify** or **cancel** appointments according to the system's policies (e.g., canceling within a specific time frame).

2.2.3 Appointment Tracking Module:

1. Appointment Booking and Confirmation
- Appointment Scheduling:
- Patients can choose available time slots for a doctor based on the doctor's calendar and availability.
- The system may allow the patient to book for online consultations or in-person visits, depending on the doctor's availability and patient preferences.

2. **Appointment Status Tracking**
- **Tracking Status**:
    - Each appointment has a **status** that tracks its current state, such as:
        - **Booked**: The appointment has been scheduled.
        - **Confirmed**: The appointment is confirmed by both the patient and the doctor.
        - **In Progress**: The doctor is actively consulting the patient.
        - **Completed**: The consultation has ended.
        - **Cancelled**: The appointment was cancelled by the patient or doctor

3. **Doctor's Calendar**:
- The system integrates with the doctor's personal or practice calendar to avoid double-booking and ensure that available slots are reflected in real-time.

Key Entities & Their Attributes:

1. Patient

Stores information about the patients who book appointments.

Field Name	Data Type	Description
PatientID	INT	Primary Key, Unique Patient ID
FirstName	VARCHAR(50)	Patient's first name
LastName	VARCHAR(50)	Patient's last name
Gender	VARCHAR(10)	Gender (Male/Female/Other)
DateOfBirth	DATE	Patient's Date of Birth
Email	VARCHAR(100)	Email Address
PhoneNumber	VARCHAR(20)	Contact Number
Address	TEXT	Patient's Address

2. Doctor

Stores information about the doctors available for appointments.

Field Name	Data Type	Description
DoctorID	INT	Primary Key, Unique Doctor ID
FirstName	VARCHAR(50)	Doctor's first name
LastName	VARCHAR(50)	Doctor's last name
Gender	VARCHAR(10)	Gender (Male/Female/Other)
SpecializationID	INT	Foreign Key, references Specialization
DepartmentID	INT	Foreign Key, references Department
PhoneNumber	VARCHAR(20)	Contact Number
Email	VARCHAR(100)	Doctor's email
Availability	TEXT	Doctor's availability schedule

3. Specialization

Stores the specializations that doctors can have (e.g., Cardiologist, Neurologist).

Field Name	Data Type	Description
SpecializationID	INT	Primary Key, Unique Specialization ID
Name	VARCHAR(100)	Name of the Specialization (e.g., Cardiologist)

4. Appointment Status

Stores the status of appointments (e.g., Scheduled, Completed, Canceled).

Field Name	Data Type	Description
StatusID	INT	Primary Key, Unique Status ID
StatusName	VARCHAR(50)	Status name (Scheduled, Completed, Canceled, etc.)
Description	TEXT	Detailed description of the status

2.3 User Interface (UI) Design

- **Main Screens**:
  - **Login Screen**: Users enter credentials to access the system.
  - **Dashboard**: A central hub for accessing system features, adjusted based on user roles.

Common UI Elements:

- Navigation Bar: For easy access to different sections like appointments, profile, doctors, and settings.
- Buttons: Clear action buttons for booking appointments, confirming, canceling, etc.
- Forms: Simple forms for entering patient information, appointment details, and other inputs.
- Calendar/Date Picker: For scheduling appointments, viewing availability, and managing schedules.
- Search Bar: Allow searching for doctors, specializations, or appointment histories.

Patient Interface Design

Upon login, patients are greeted with a dashboard that includes:

- Upcoming Appointments: A list of upcoming appointments with time, doctor details, and status.
- Quick Book Appointment: Button to quickly book a new appointment.
- Notifications: Recent updates or reminders about appointments.
- Search Doctor: Search functionality for finding doctors by specialization or department.
- Profile Overview: A quick view of the patient's profile details (e.g., name, contact, insurance).

Screens:

- Login/Registration Screen
    - Username, password fields, and login button.
    - Registration option for new users (name, email, password, phone).
- Search & Book Appointment
    - Search Bar: To search for doctors by name, specialization, or department.
    - Filters: Specialization, doctor availability, and department filters.
    - Doctor Profile: Displays doctor's details (name, specialization, availability).
    - Calendar: Select a date and time for the appointment.
    - Appointment Confirmation: Show confirmation details with time, doctor, and reason for visit.
- Appointment History
    - Upcoming Appointments: List upcoming appointments with doctor name, date, and status.

Admin Interface Design

Admin users should have access to more detailed management tools, such as:

- Manage Doctors: Add, remove, or modify doctor details (specializations, availability).
- Manage Appointments: View and manage all appointments, including rescheduling and cancellation.
- Patient Management: View and edit patient information.
- Reports: View system reports (e.g., appointment statistics, doctor performance).

Screens

- Login Screen
    - Admin username and password fields with login button.
- Doctor Management
    - Add/Edit Doctor: Forms to add or update doctor details, including specialty, department, availability.
    - Doctor List: A list of all doctors with the ability to search, edit, or delete doctors.
- Appointment Management
    - Appointment List: View and manage all appointments.
    - Appointment Status: Change status of any appointment (e.g., reschedule, cancel).
    - Search Appointments: Find specific appointments by patient, date, or doctor.

Patient Management

- **Patient List**: View, search, and edit patient records.
- **Patient Details**: Access detailed patient information, including appointment history.

2.4 Technology Stack

- **Frontend**: HTML, CSS for an interactive user interface.
- **Backend**: **PHP** (Hypertext Preprocessor) is a popular server-side scripting language that is widely used for backend development, particularly for web applications.
- **Database**: MySQL or PostgreSQL for reliable data storage and retrieval.

Chapter 3

Implementation

This chapter outlines the steps taken to implement the Doctor Appointment System, covering the backend, frontend, database, and integration processes. It describes the technologies used, the structure of the codebase, and any special development techniques.

3.1 Backend Implementation

The Online Doctor Appointment System is a web application that facilitates patients in scheduling appointments with doctors. The backend of this system is implemented in PHP, with MySQL as the database.

System Architecture

1. **Users**: Includes patients, doctors, and administrators.
2. **Functionalities**:
    - Admin, Patient, Registration and Login
    - Doctor Profile Management
    - Appointment Scheduling
    - Appointment Management
3. **Tech Stack**:
    - **Language**: PHP
    - **Database**: MySQL
    - **Server**: Apache
    - **Communication**: RESTful APIs

3.2 Frontend Implementation

The frontend provides the user interface for teachers, students, and administrators to interact with the system.

User Interface (UI) Components

- **Logo**: Represents the system's branding.
- **Navigation Menu**: Links to Make Appointment, Login/Sign-up, etc.
- **Search Bar**: For searching doctors by specialty, name.

Registration Form

- **Components**:
- **Fields**: Full Name, Email, Phone Number, Password, Confirm Password, Date of Birth.
- **Buttons**: "Register", "Reset".

- **Validation**: Ensures that the entered data is accurate and meets the required criteria.
- **Purpose**: Allows new users (patients) to create an account on the system.

3.3 Database Implementation

- **Database Setup**: Used MySQL as the database system to store admin and doctor, patient data in a relational structure.
- **Database Schema**:
  - **User Table**: Stores information about all users (patients, doctors, and administrators)
  - **Doctor Table**: Stores additional information specific to doctors.
  - **Appointment Table**: Stores details of appointments between patients and doctors.
  - **Patient Table**: Stores additional information specific to doctors.

Chapter 4

Testing

This chapter covers the testing processes and methodologies applied to the Doctor Appointment System. Testing is essential to identify and correct any issues, validate that the system meets functional and non-functional requirements, and ensure that it performs reliably under various conditions.

4.1 Testing Objectives

- Verify that the system functions as intended by executing a series of test cases.
- Ensure that all user roles (Admin, Doctor, Patient) can access the intended features without any errors.
- Test the accuracy of appointment records and data retrieval.
- Confirm that the system is secure and handles invalid inputs or unauthorized access appropriately.
- Evaluate the system's performance and reliability under load.

4.2 Testing Environment

- **Hardware**: Laptop/PC with minimum 8GB RAM and multi-core processor.
- **Software**:
  - Backend and frontend hosted on local servers (PHP for backend, CSS and HTML for frontend).
  - Database: MySQL
  - Testing Tools: phpMyAdmin for Database management and query testing.
- **Operating System**: Windows 10/macOS/Linux.
- **Browser**: Google Chrome, Mozilla Firefox, and Microsoft Edge for cross-browser testing.

4.3 Types of Testing

4.3.1 Unit Testing

- **Objective**: To test individual components or functions in isolation to verify their correctness.
- **Example Test Cases**:
  - **User Authentication**: Verifies that the login function correctly authenticates users based on their credentials.
  - **Report Generation**: Confirms that the report function generates data correctly for given filters.

4.3.2 Integration Testing

- **Objective**: To test the interaction between different modules of the system (e.g., frontend and backend, backend and database).

4.3.3 Functional Testing

- **Objective**: To test the system against functional requirements to ensure it meets specified user needs.
- **Test Scenarios**:
  - **Login and Registration**: Tests login and registration processes for all user roles and checks that users are directed to the appropriate dashboard.
  - **Appointment Management**: Verifies that doctors can view appointment for each patient and that the system accurately reflects the booking status.

4.4 Test Cases

Table 4.1: Test Cases

Test Case ID	Test Case Description	Test Steps	Expected Result	Status
TC-001	Verify user login with valid credentials	1. Open the login page. 2. Enter valid email and password. 3. Click login.	User is redirected to their respective dashboard (e.g., patient, doctor, admin).	Pass/Fail
TC-002	Verify login with invalid credentials	1. Open the login page. 2. Enter invalid email/password. 3. Click login.	System displays "Invalid credentials" error message.	Pass/Fail
TC-003	Verify registration with missing fields	1. Open the registration page. 2. Leave required.	System highlights missing fields and displays appropriate error messages.	Pass/Fail

Table 4.2: Appointment Booking

Test Case ID	Test Case Description	Test Steps	Expected Result	Status
TC-101	Book an appointment with an available doctor	1. Login as a patient. 2. Search for a doctor. 3. Select an available	Appointment is successfully created and visible in both patient and doctor dashboards.	Pass/Fail

		time slot and confirm.		
TC-102	Attempt booking on an unavailable slot	1. Login as a patient. 2. Select a time slot already booked by another patient.	System displays "Slot unavailable" error message.	Pass/Fail
TC-103	Cancel a booked appointment	1. Login as a patient. 2. View appointments. 3. Select an appointment and cancel.	Appointment status is updated to "Canceled," and slot becomes available for others.	Pass/Fail

Table 4.3: Doctor Availability

Test Case ID	Test Case Description	Test Steps	Expected Result	Status
TC-201	Add availability slots as a doctor	1. Login as a doctor. 2. Navigate to availability settings. 3. Add slots.	Slots are saved and visible to patients when searching for doctors.	Pass/Fail
TC-202	Update existing availability	1. Login as a doctor. 2. Edit an existing slot.	Updated slot information is saved and displayed correctly.	Pass/Fail
TC-203	Delete an availability slot	1. Login as a doctor.	Slot is removed and no longer visible to patients.	Pass/Fail

Chapter 5

Results and Discussion

This chapter summarizes the results of the Doctor Appointment System project, discussing its effectiveness, reliability, and alignment with the intended objectives. The chapter also covers any challenges encountered, key insights, and recommendations for future improvements.

5.1 Results

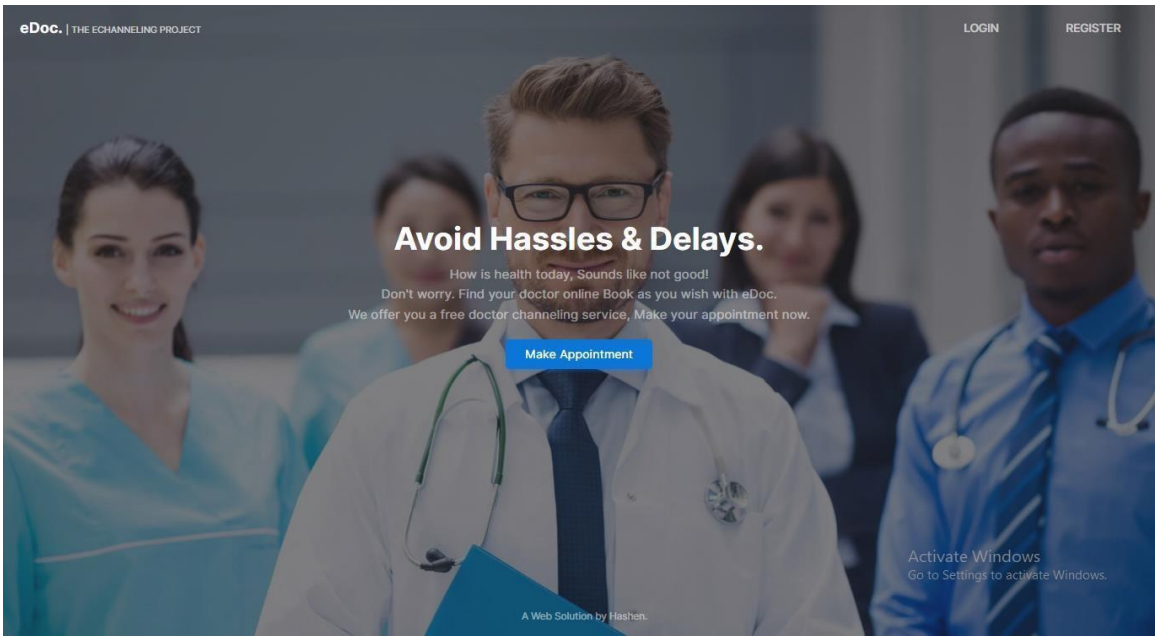


Fig 5.1: Front page

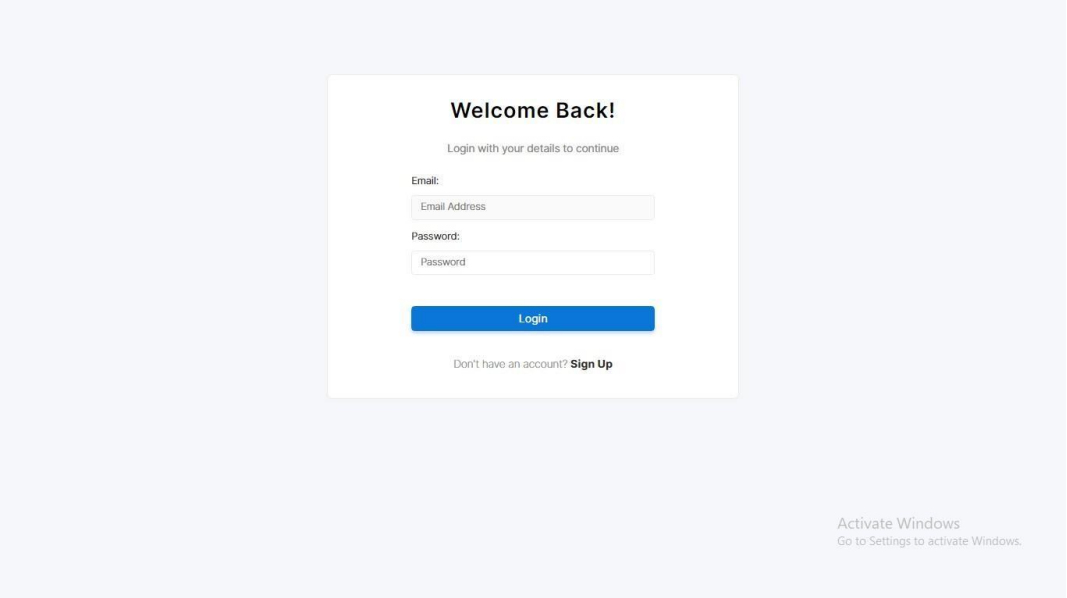


Fig 5.2: Login page

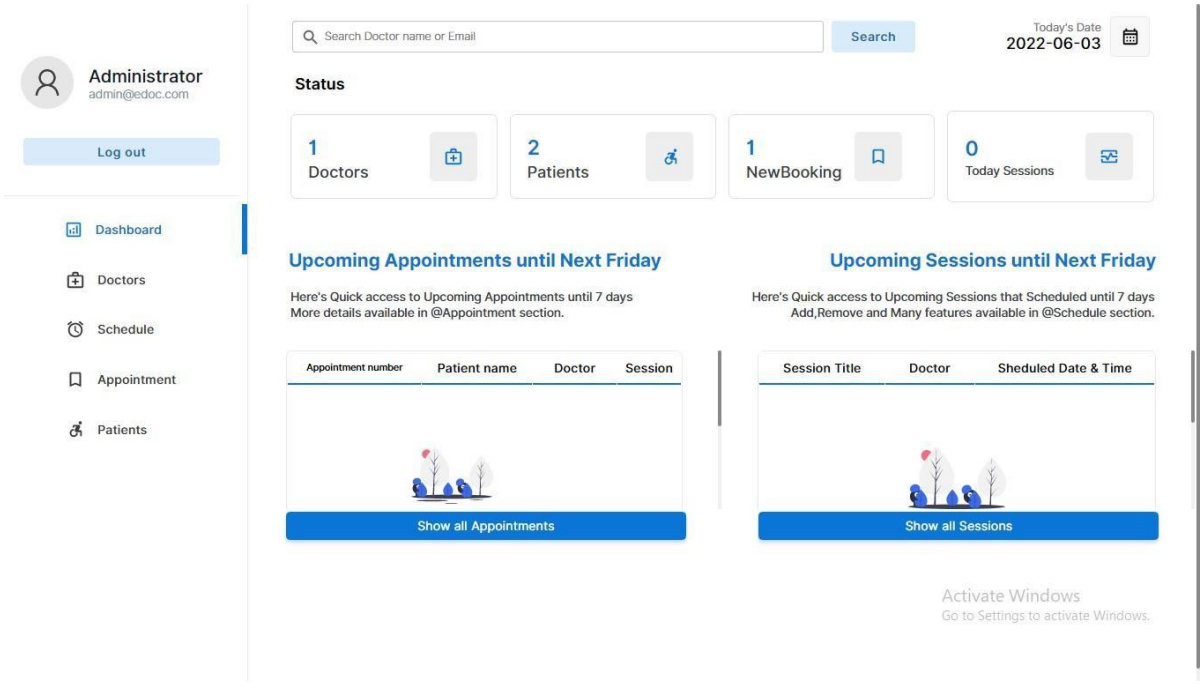


Fig 5.3 : Admin dashboard

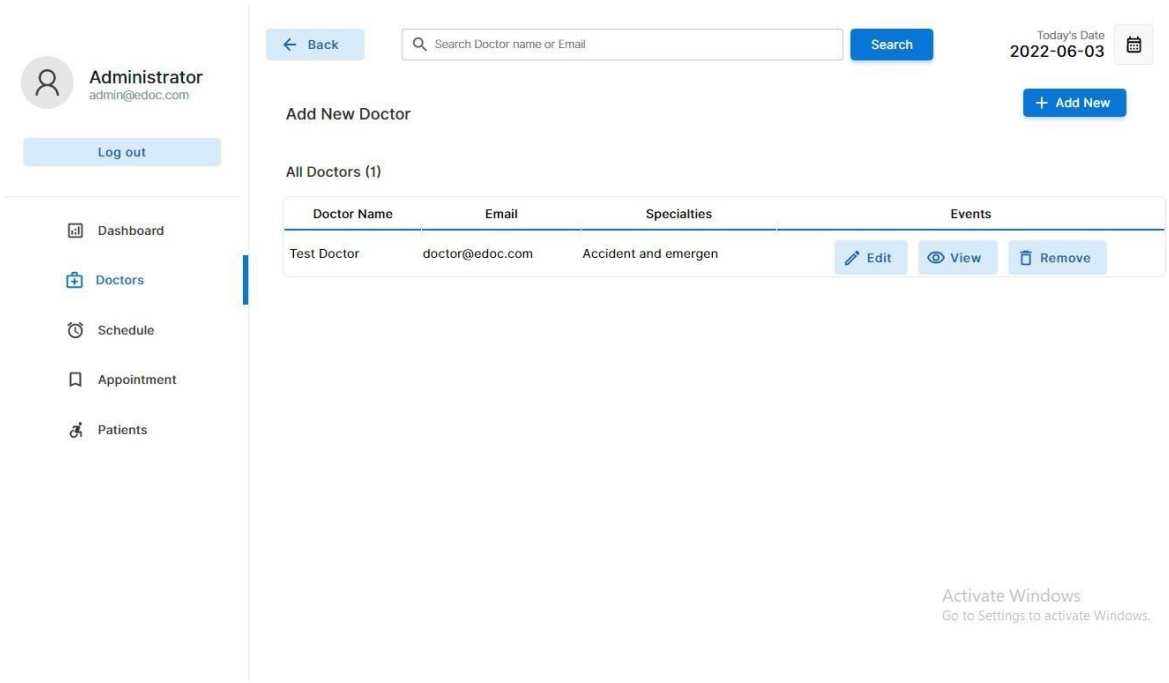


Fig 5.4: View doctors

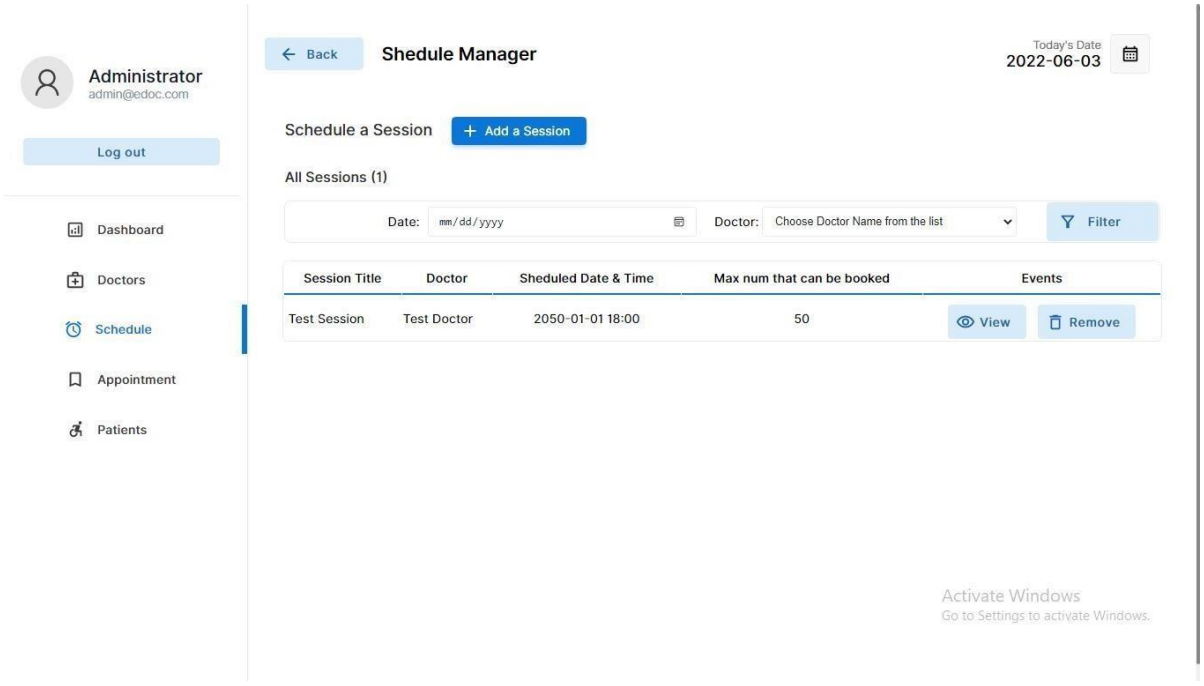


Fig 5.5: View schedules

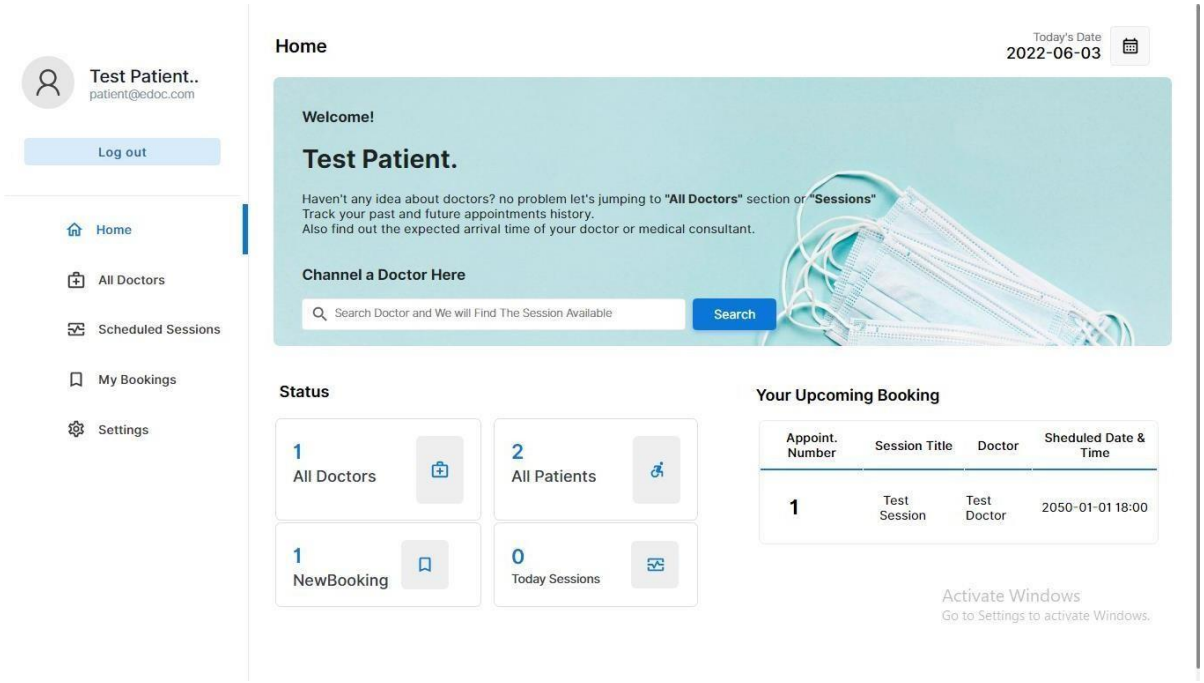


Fig 5.6:View patient

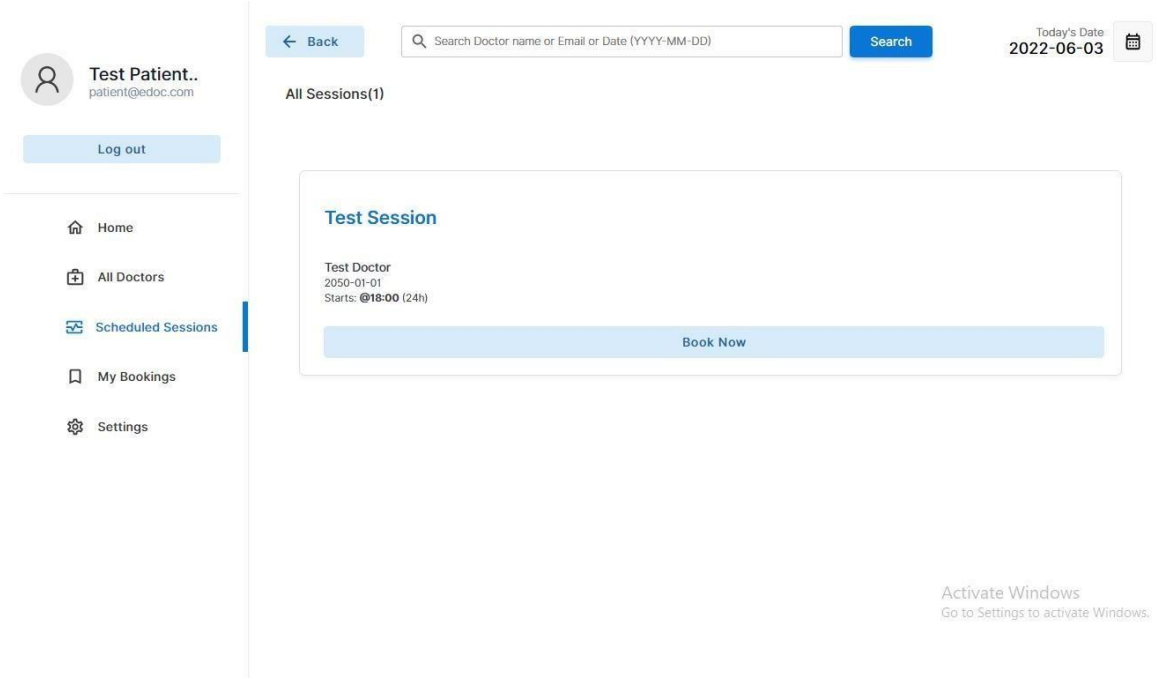


Fig 5.7: Schedule session of patient

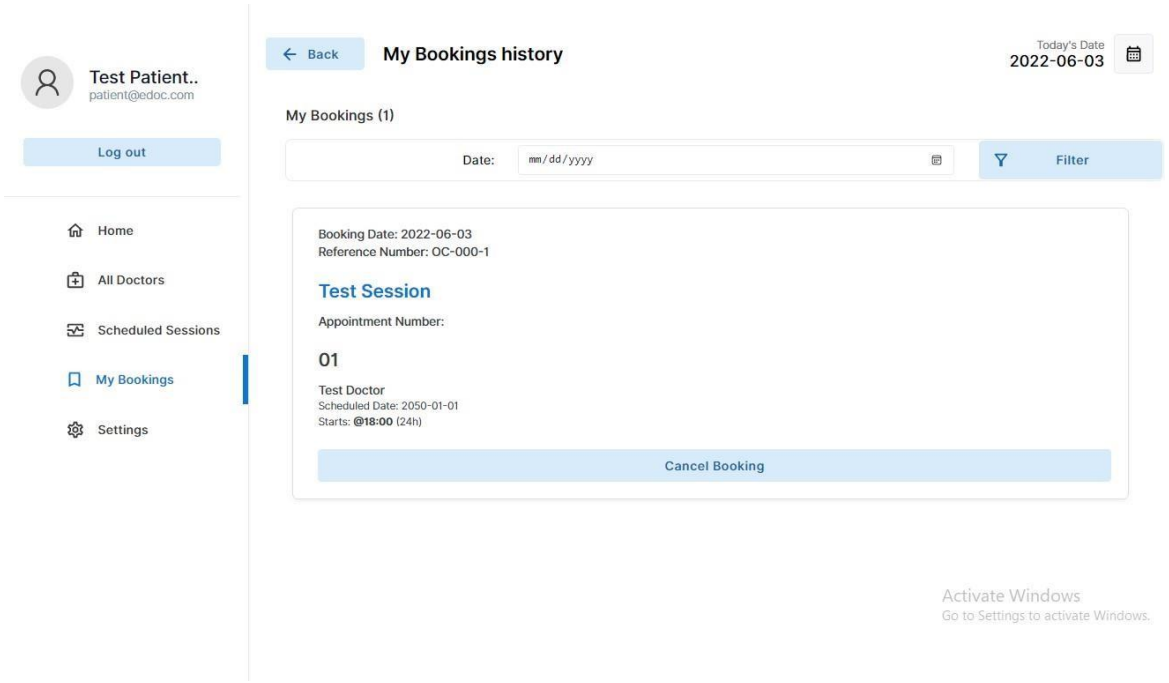


Fig 5. 8: Booking of patient

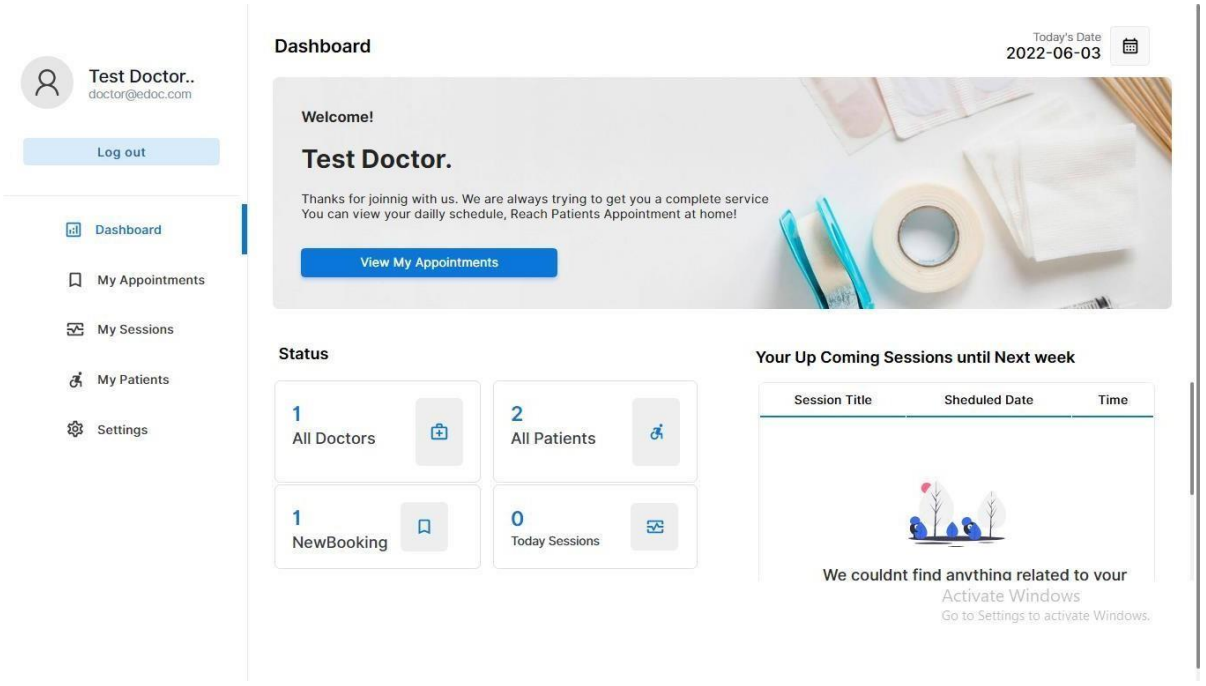


Fig 5.9: Doctors dashboard

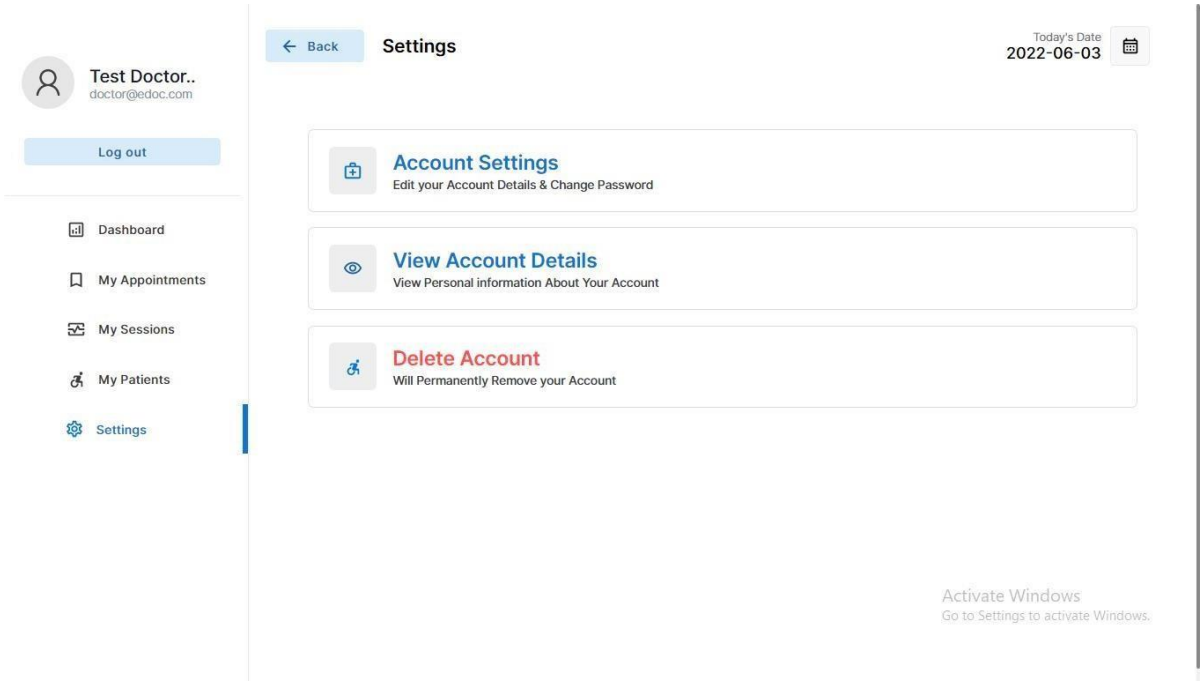


Fig 5.10: Settings

5.2 Discussion

Effectiveness of the System

- The Doctor Appointment System effectively achieved its goal of streamlining attendance management, enhancing accuracy, and improving access to booking.
- The successful implementation of user roles allowed each type of user to access only the necessary features, improving both security and user experience.
- Overall, the system demonstrated reliable performance, accurate data management, and ease of use, making it a valuable tool for hospitals.

Chapter 6

Conclusion and Future Enhancements

6.1 Conclusion

- The **Online Doctor Appointment System** is a robust, user-friendly, and efficient platform designed to streamline the process of scheduling medical consultations. It bridges the gap between patients and healthcare providers by offering a seamless and accessible interface for appointment booking, management, and communication.
- The system incorporates key functionalities such as user authentication, appointment scheduling, doctor availability management, notifications, and feedback mechanisms. These features ensure that users—patients, doctors, and administrators—can interact with the system effectively while maintaining data integrity, security, and privacy.

6.2 Future Enhancements

- As technology evolves and the healthcare landscape changes, there are numerous opportunities to improve the **Online Doctor Appointment System**.
- Future enhancements in the **Online Doctor Appointment System** will focus on improving user convenience, accessibility, and system intelligence.
- These enhancements will not only streamline appointment management but also bring a more comprehensive and supportive healthcare environment.

REFERENCES

Citation Format:

- **www.healthcaresolutions.com.**
- **https://www.scribd.com**
- **https://www.studocu.com**
- **HTML, CSS, PHP <<https://www.w3schools.com/>>**
- **Neinstein, L. S. 1982. Lowering broken appointment rates at a teenage health center. J.**

Web Development Technologies

- PHP Documentation: https://www.php.net/docs.php
- MySQL Documentation: https://dev.mysql.com/doc/