

Functional Design Document for Medicare Hub

Medicare Hub

Project Description:

Medicare Hub is a comprehensive healthcare management system that facilitates seamless interaction between patients and healthcare providers. It incorporates role-based access control with three distinct roles: Admin, Doctor, and User. The system is designed to enhance efficiency in managing appointments, user profiles, and healthcare information.

Prepared by: Dhruv Bansal

Table of Contents

1. Project Overview

- Project Name
- Project Description

2. Roles and Functionalities

- 2.1. Admin Role
- 2.2. Doctor Role
- 2.3. User Role

3. User Interface Design

- 3.1. Admin Dashboard Interface
- 3.2. Doctor Dashboard Interface
- 3.3. User Dashboard Interface

4. System Architecture

- 4.1. Technology Stack
- 4.2. Data Flow

5. Functional Requirements

- 5.1. Role Management
- 5.2. Appointment Handling
- 5.3. Profile Management
- 5.4. Department Management

6. Non-Functional Requirements

- 6.1. Security
- 6.2. Scalability
- 6.3. Performance
- 6.4. Usability

7. Conclusion

1. Project Overview

Project Name: Medicare Hub

Project Description:

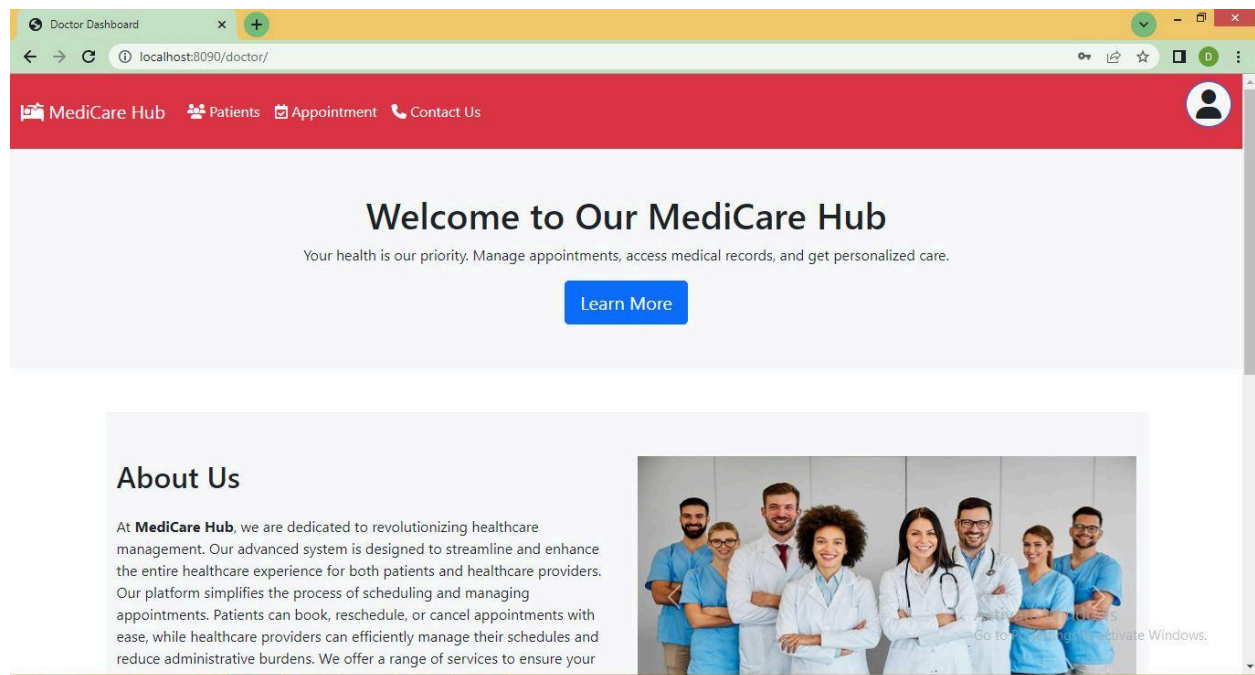
Medicare Hub is a healthcare management system with role-based access control. It includes three main roles: Admin, Doctor, and User. Each role has distinct functionalities that allow users to manage appointments, view lists (patients, doctors, departments), and maintain profiles. The system is designed to streamline the interaction between patients and healthcare providers.

2. Roles and Functionalities

2.1. Admin Role

Dashboard:

The Admin Dashboard is the central control panel where the admin can access various functionalities.



- Manage User Roles:
 - Assign roles to users (Admin, Doctor, User).
 - Update or change user roles as required.

| ID | Full Name | Email | Role |
|----|---------------|---------------------------|---|
| 13 | dhruv bansall | dhruvbansal0505@gmail.com | User <input type="button" value="Update Role"/> |
| 14 | Admin | admin1@gmail.com | Admin <input type="button" value="Update Role"/> |
| 15 | Doctor | doctor1@gmail.com | Doctor <input type="button" value="Update Role"/> |
| 16 | abc | abc1@gmail.com | User <input type="button" value="Update Role"/> |
| 18 | rg | rg@gmail.com | User <input type="button" value="Update Role"/> |
| 19 | doctor2 | doctor2@gmail.com | Doctor <input type="button" value="Update Role"/> |
| 20 | Raj Sharma | raj_doctor@gmail.com | Doctor <input type="button" value="Update Role"/> |
| 21 | Anshul Jain | anshul_doctor@gmail.com | Doctor <input type="button" value="Update Role"/> |
| 22 | Nakul Chalana | nakul_doctor@gmail.com | Doctor <input type="button" value="Update Role"/> |
| 23 | Anurag sharma | anurag_doctor@gmail.com | Doctor <input type="button" value="Update Role"/> |

Patient List:

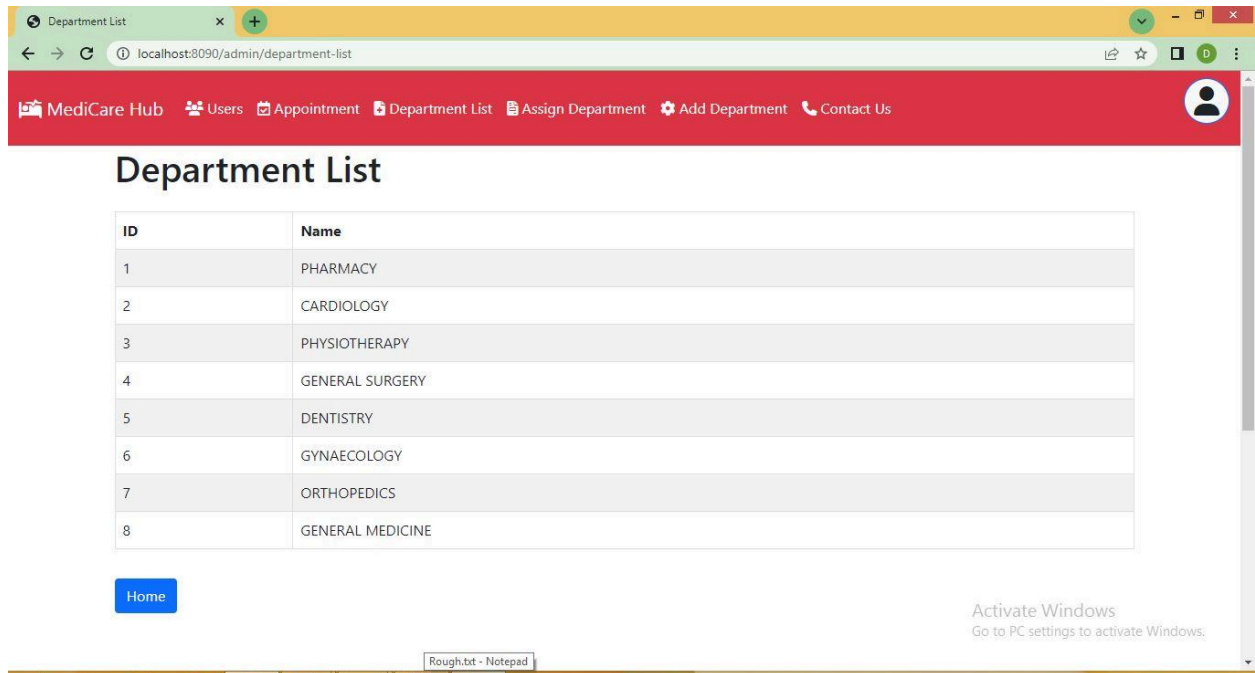
- View a list of all patients registered in the system.
- Access patient details (e.g., name, contact info, appointment history).

Doctor List:

- View a list of all doctors registered in the system.
- Access doctor details (e.g., name, specialization, department).

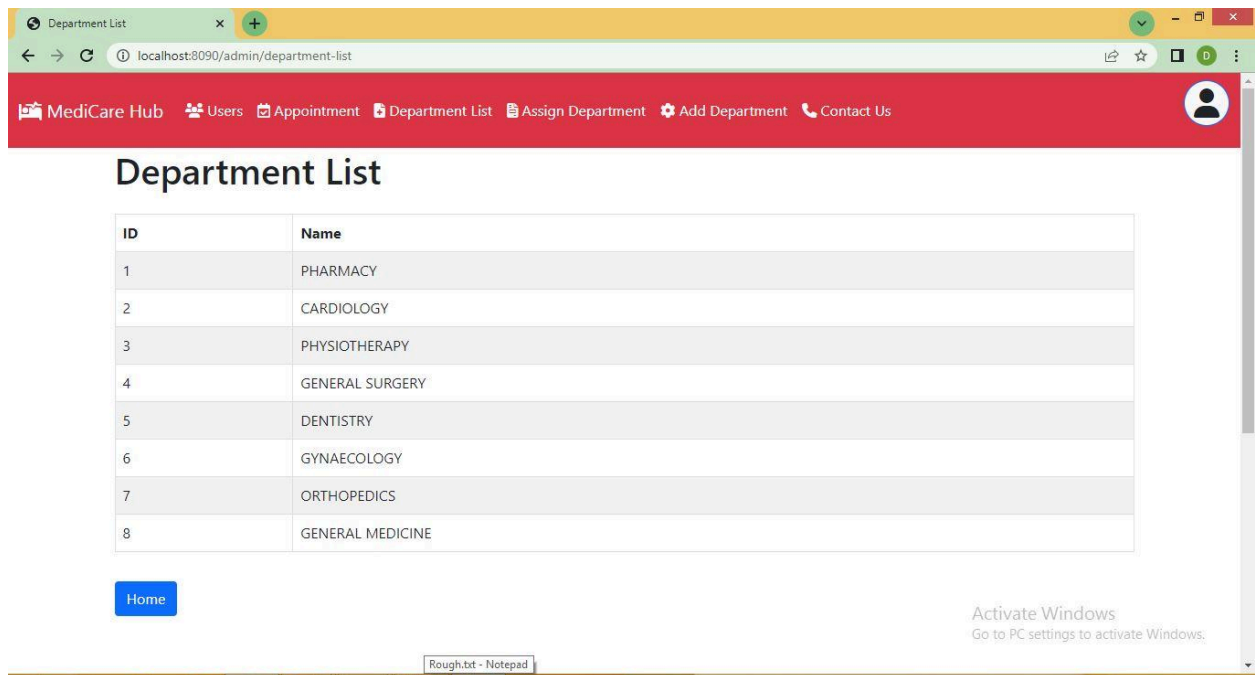
Appointments List:

- View all appointments that have been approved by doctors.



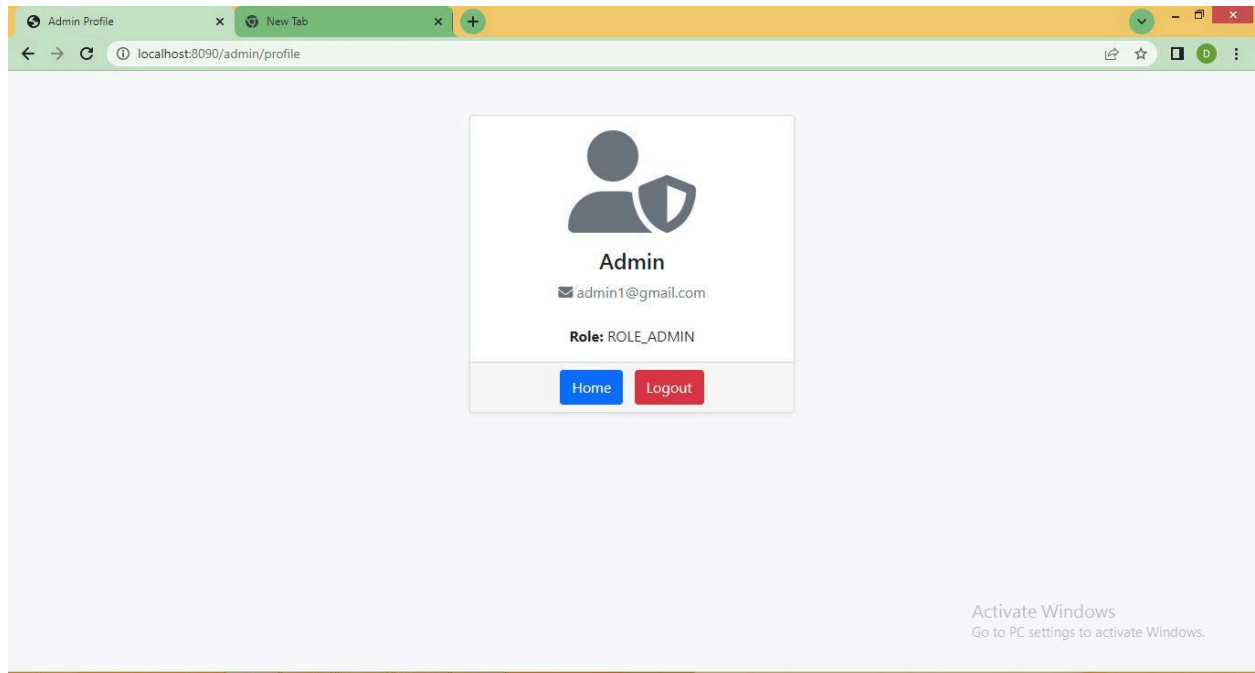
Department Management:

- View the list of departments within the hospital.
- Add new departments to the system.
- Assign departments to specific doctors.



Profile Management:

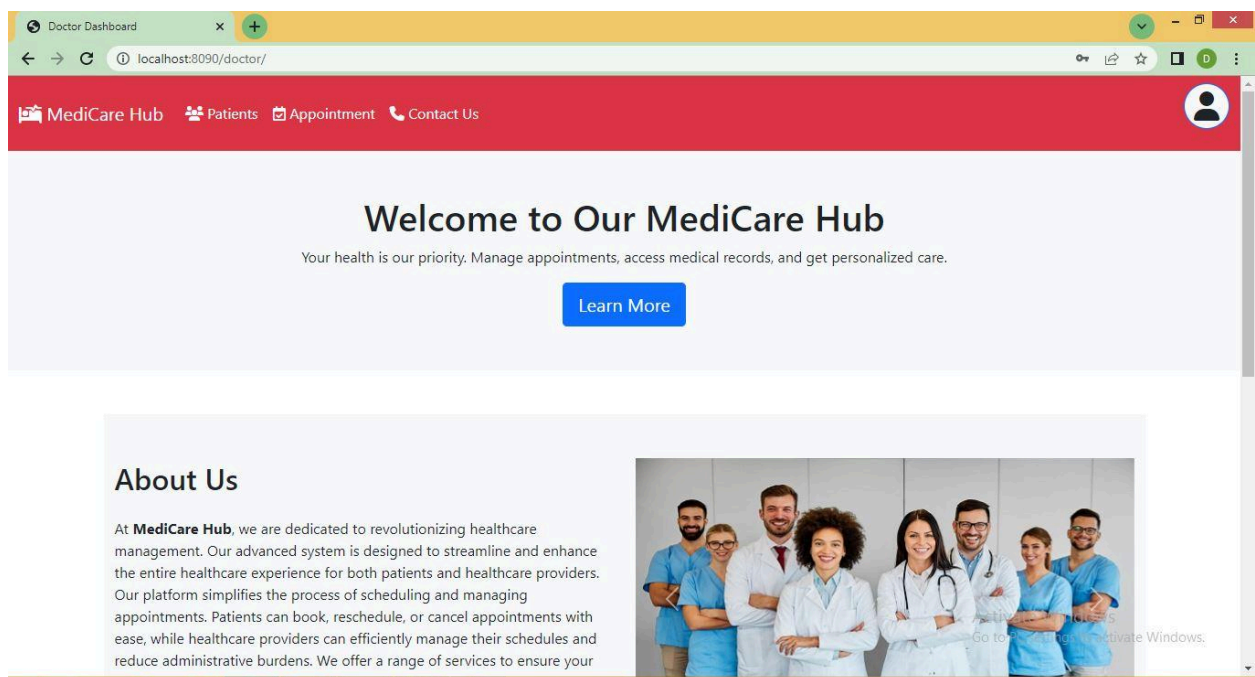
- View the admin profile.



2.2. Doctor Role

Dashboard:

- The Doctor Dashboard provides an overview of the doctor's appointments and patient list.



Manage Appointments:

- View appointment requests from users.
- Approve or reject appointment requests.

The screenshot shows a web browser window with the URL `localhost:8090/appointments/doctor-list`. The page has a red header with navigation links: "MediCare Hub", "Patients", "Appointment", and "Contact Us". A user profile icon is in the top right. The main content area is titled "Doctor Appointment List" and contains a table with the following data:

| User | Appointment Date | Status | Update Status |
|---------------|------------------|----------|---|
| Keshav Porwal | 2024-08-27T16:06 | APPROVED | Approved <input type="button" value="Update Status"/> |
| Keshav Porwal | 2024-08-27T16:06 | PENDING | Pending <input type="button" value="Update Status"/> |

Below the table is a blue "Home" button. At the bottom of the page is a "Contact Us" section with a message: "For any inquiries or support, feel free to reach out to us. We are here to assist you with your needs and provide the necessary information to ensure a smooth experience with our services." It includes an "Address" field with "123 MediCare St, Health City, HC 12345" and "Name" and "Email address" input fields. An "Activate Windows" watermark is visible in the bottom right corner.

Patient List:

- View a list of patients assigned to the doctor.
- Access patient details.

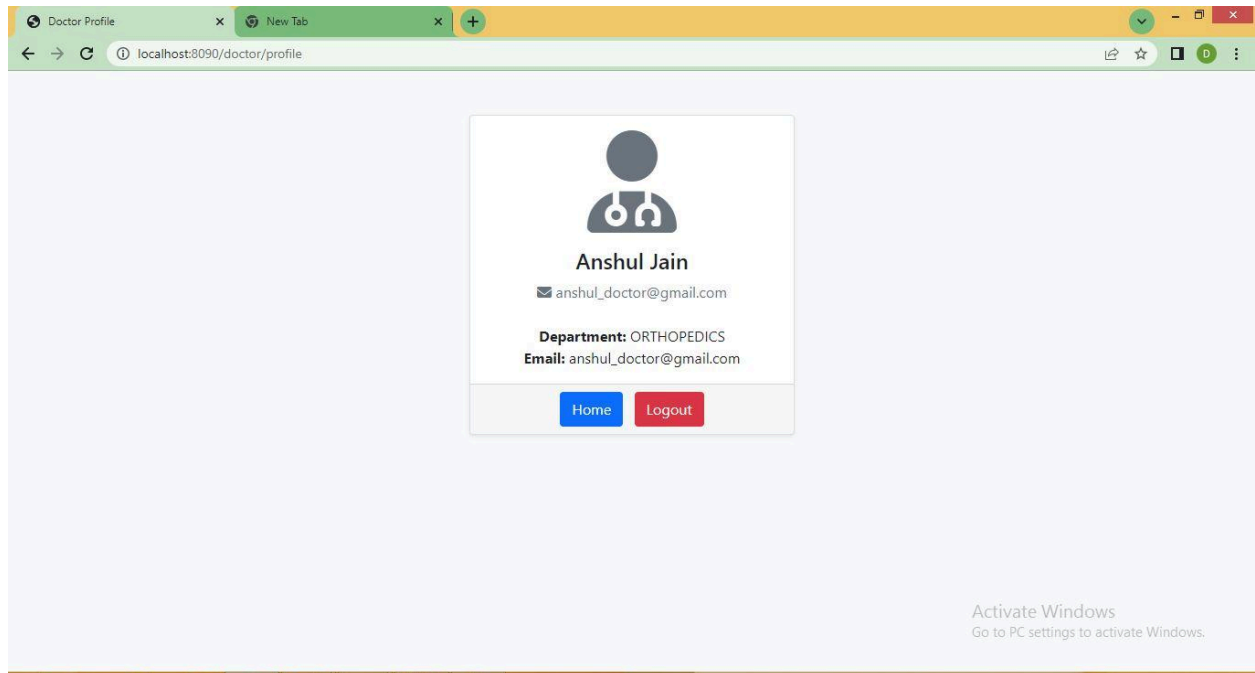
The screenshot shows a web browser window with the URL `localhost:8090/doctor/users`. The page has a red header with navigation links: "MediCare Hub", "Patients", "Appointment", and "Contact Us". A user profile icon is in the top right. The main content area is titled "List of Users" and contains a table with the following data:

| ID | Name | Email |
|----|---------------|---------------------------|
| 13 | dhruv bansall | dhruvbansal0505@gmail.com |
| 16 | abc | abc1@gmail.com |
| 18 | rg | rg@gmail.com |
| 25 | Mayank Pant | mayank@gmail.com |
| 26 | Vansh Kabra | kabra@gmail.com |
| 27 | Parth Dadich | parth@gmail.com |
| 28 | Rohan Jha | rohan@gmail.com |
| 29 | Keshav Porwal | porwal@gmail.com |
| 30 | Harsh Agarwal | harsh02@gmail.com |
| 31 | Harsh Agarwal | harsh02@gmail.com |

Below the table is a blue "Home" button. At the bottom of the page is an "Activate Windows" watermark.

Profile Management:

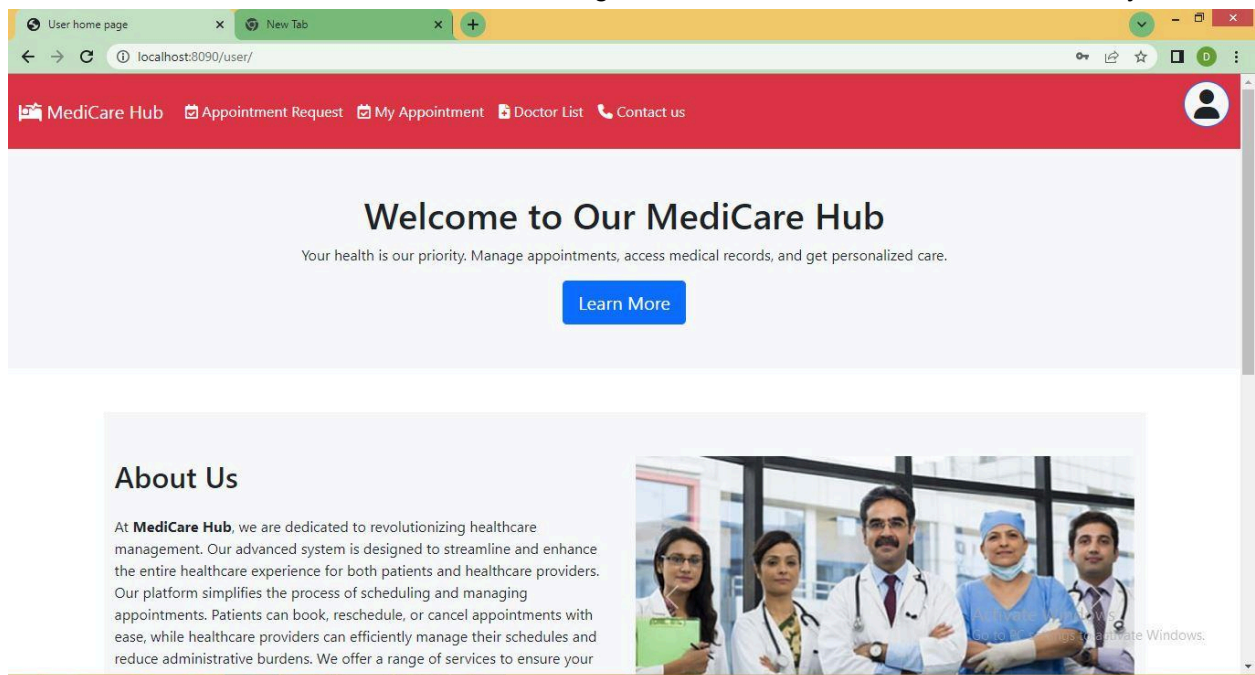
- View and update the doctor profile (e.g., name, contact info, specialization, password).



2.3. User Role

Dashboard:

- The User Dashboard allows users to manage their interactions with the healthcare system.



Request Appointment:

- Submit a request for an appointment with a doctor.
- Select the department and doctor for the appointment.

The screenshot shows a web browser window with the URL `localhost:8090/appointments/request-form`. The page has a red header with navigation links: [MediCare Hub](#), [Appointment Request](#), [My Appointment](#), [Doctor List](#), and [Contact us](#). A user profile icon is in the top right. The main content area features a white box titled "Appointment Request" with the following fields: "Department:" with a dropdown menu showing "PHARMACY", "Doctor:" with a dropdown menu showing "Select a doctor", and "Appointment Date and Time:" with a date and time picker showing "dd-mm-yyyy --:--". A red "Request Appointment" button is at the bottom of the form. Below the form is a blue "Home" button. In the bottom right corner, there is a "Activate Windows" watermark with the text "Go to PC settings to activate Windows."

Appointment Status:

- View the status of submitted appointments (e.g., pending, approved, rejected).

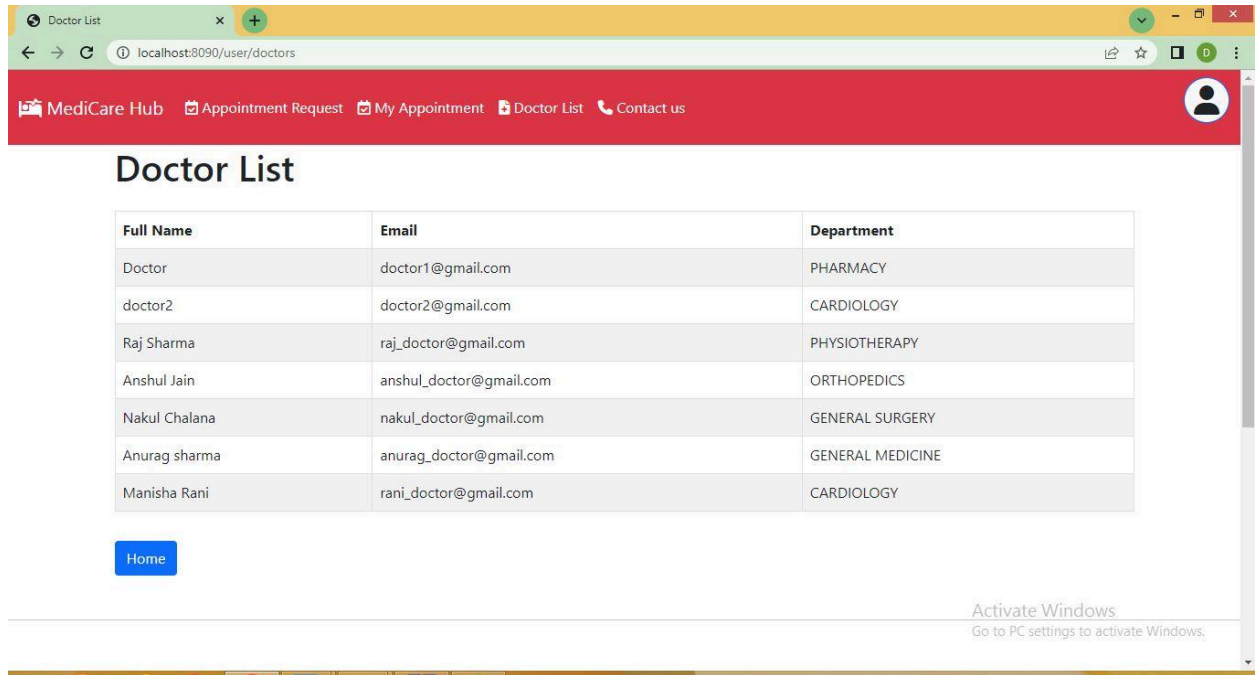
The screenshot shows a web browser window with the URL `localhost:8090/appointments/user-list`. The page has a red header with navigation links: [MediCare Hub](#), [Appointment Request](#), [My Appointment](#), [Doctor List](#), and [Contact us](#). A user profile icon is in the top right. The main content area features a white box titled "My Appointments" containing a table with the following data:

| ID | Doctor | Department | Date AND Time | Status |
|----|-------------|-------------|------------------|----------|
| 11 | Anshul Jain | ORTHOPEDICS | 2024-08-27T16:06 | APPROVED |
| 12 | Anshul Jain | ORTHOPEDICS | 2024-08-27T16:06 | PENDING |

Below the table is a blue "Home" button. At the bottom of the page, there is a "Contact Us" section with a form for "Name" and "Email address". The "Name" field contains "Your Name" and the "Email address" field contains "Your Email". In the bottom right corner, there is a "Activate Windows" watermark with the text "Go to PC settings to activate Windows."

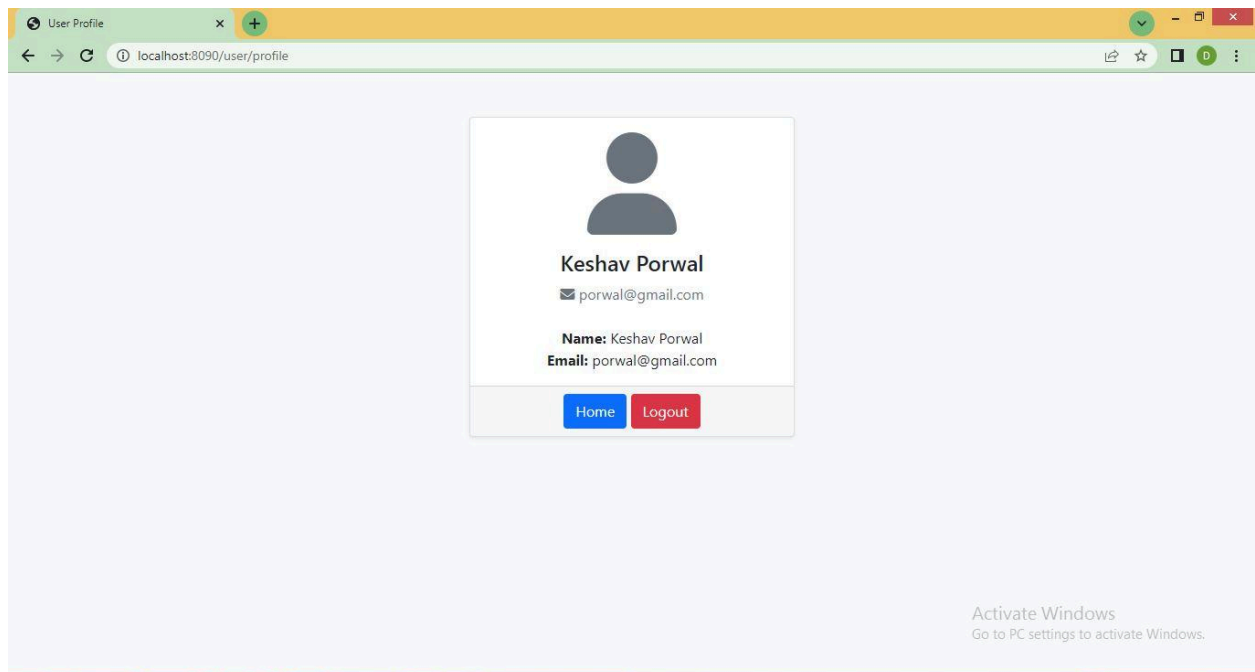
Doctor List:

- View a list of doctors in the system.
- Filter doctors by department or specialization.



Profile Management:

- View and update the user profile (e.g., name, contact info, password).



3. User Interface Design

3.1. Admin Dashboard Interface

- **Navigation Menu:** Access to manage users, patients, doctors, departments, appointments, and profile.
- **Main Panel:** Displays overview and statistics related to users, appointments, and departments.

3.2. Doctor Dashboard Interface

- **Navigation Menu:** Access to manage appointments, view patient list, and profile.
- **Main Panel:** Displays upcoming appointments and patient overview.

3.3. User Dashboard Interface

- **Navigation Menu:** Access to request appointments, view appointment status, view doctor list, and profile.
- **Main Panel:** Displays appointment history and available doctors.

4. System Architecture

4.1. Technology Stack

- **Frontend:** HTML, CSS, JavaScript
- **Backend:** Java, Spring Boot
- **Database:** MySQL/SQL Database
- **Security:** Spring Security for role-based authentication and authorization

4.2. Data Flow

- **User Authentication:**

Users log in through a common login interface. Based on their credentials, they are redirected to the appropriate dashboard (Admin, Doctor, User).

- **Role-Based Access:**

Each role has specific endpoints and functionalities that are secured using Spring Security.

- **Database Management:**

Data related to users, appointments, doctors, and departments are stored in a relational database. CRUD operations are managed through the backend services.

5. Functional Requirements

5.1. Role Management

- Admin can assign and modify user roles.

5.2. Appointment Handling

- Users can request appointments.
- Doctors can approve/reject appointments.
- Admin can view all approved appointments.

5.3. Profile Management

- Each role can view and update their profile details.

5.4. Department Management

- Admin can manage departments and assign them to doctors.

6. Non-Functional Requirements

6.1. Security

- Role-based access control ensures that users can only access the functionalities specific to their role.
- Passwords and sensitive data are encrypted in the database.

6.2. Scalability

- The system is designed to handle a growing number of users, doctors, and appointments without performance degradation.

6.3. Performance

- Efficient data handling ensures quick response times for dashboard interactions.

6.4. Usability

- The user interface is intuitive, making it easy for all users to navigate and perform their respective tasks.

7. Conclusion

This document outlines the functional design of the Medicare Hub project, detailing the roles, features, and system architecture. The system is designed to provide a seamless experience for users, doctors, and admins, ensuring efficient management of healthcare services.

