CHAPTER 1
COMPANY PROFILE

1.1 Overview of the Organization

DLithe is an EdTech company serving IT Companies and Academic Institutions, since the year 2018. With experiences drawn from corporate time, the foundation of DLithe is built to innovate products that transform the upcoming generation. Our expertise in Embedded Systems, Robotics, Internet of Things, Cyber Security, and Artificial Intelligence is helping academics institutions to align with industry needs. Since inception, we have established 8 development centers enabling student community to work on research and development. Our services to IT companies have reduced the hiring cycle time and led to cost effective measures to source the best talent from on and off campus. We have transformed many lives by imparting 360 degree learning – Domain, Process & Technology, keeping focus on Customer Experience and Operational Excellence objectives. We are proud to say, DLithe is a bootstrap company with strong foundation, experience, trust and commitment to build an agile workforce towards industry need.

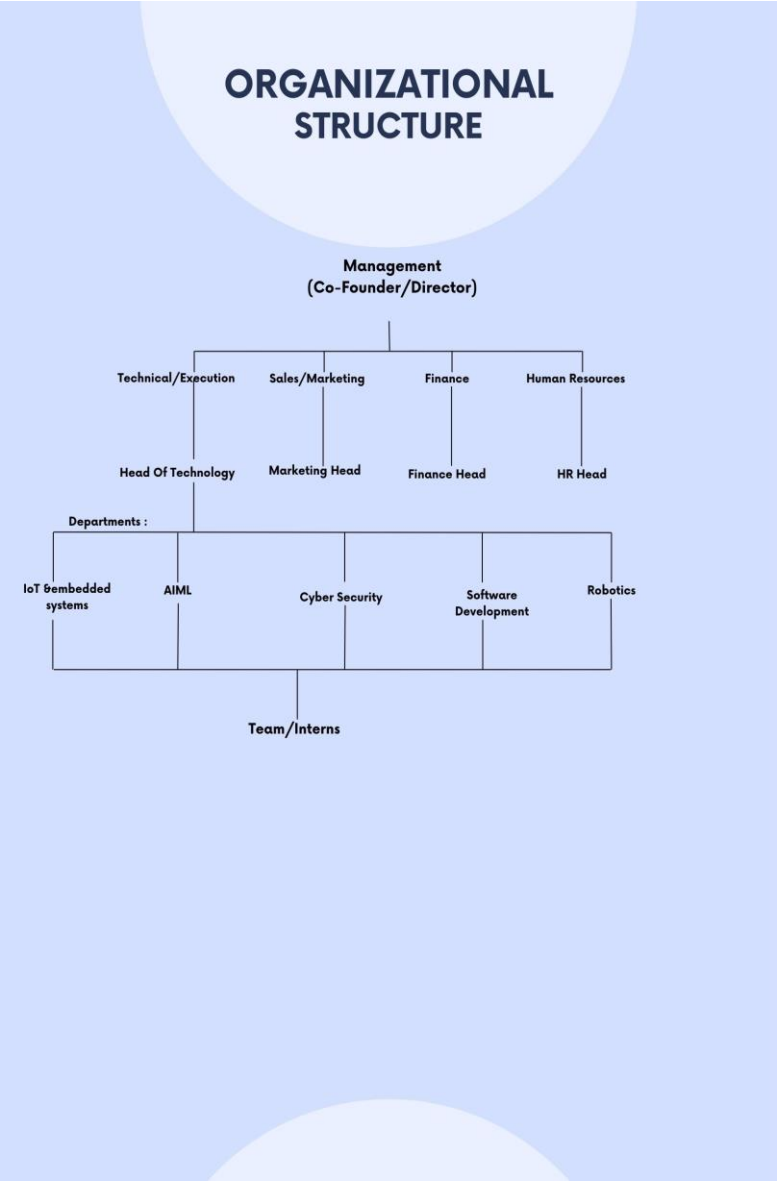
Reasons why DLithe is Best:

- Cost Effective
- Innovations
- We keep our promises
- Corporate Culture
- Proof of concepts
- Expert Base

1.2 Vision and Mission of the organization

- To support business functions to achieve Customer Experience and Operational Excellence.
- To build an agile workforce that is competent in Domain, Technology & Personality, Catering towards global industry needs.
- We are committed to providing cost-effective quality services to corporates and academics to achieve their goals.

1.3 Organization Structure



1.4 Roles and Responsibilities of the Personnel in the Organization

- Fulfilling tasks assigned by a supervisor.
- Performing clerical duties.
- Event handling or planning.
- Job shadowing.
- Learning technical skills related to the industry.
- Assist and Contribute to the team
- Managing Social Media and Emails
- Picking up hard skills
- Brushing up on our soft skills
- Build the network
- Make a career call

1.5 Product and market performance

1.5.1 Products

- **Microcontroller Trainer kits (ARM Cortex M3) DLM-1.0:**

Microcontroller Trainer kit provides a hardware platform for developing embedded systems using SAM913X8E ARM Cortex M3 Series MCU. These features make DLM-1.0 board ideal for understanding Architecture of ARM Microprocessors, GPIO, Peripherals, Sensor Interfacing, Communication protocols and other demanding application areas where ease of access and technology support with in-circuit hardware upgradeability is of very important.

- **Iocube-IoT dev kit:**

Rapidly and easily get started with learning IoT using the iocube Wi-Fi board, which have a great appeal to STEM learners, Makers, Engineers and schools alike. No prior experience is required, as the iocube dev board introduces both coding and electronics through innovative, engaging, and hands-on projects. You

can use this board to teach students about fundamentals of programming, Interfacing sensors, Cloud connectivity, Sensor node, Edge IoT, Automation and Robotics. This dev board enables any beginner to learn and understand about sensors and actuators and how to understand both digital and analog signals.

1.5.2 Services

- **To corporates:**

We provide services such as recruitment support, bootcamp, and technological improvements to IT companies.

Bootcamps:

Corporate often spends a significant amount of time training resources using internal resources. However, full-time availability and attention to detail in identifying issues with resources or identifying scope for improvements become the major challenges here. Bootcamps enable corporates to engage DLithe full-time in resource competence up skilling. The ultimate goal is to meet the needs of the customer, and the time and effort invested are critical.

Leadership Transformation training program:

Leadership transformation training program is a critical investment for corporate organizations that want to develop their workforce and optimize their business results. This program is designed to equip leaders with the necessary skills and knowledge to lead effectively in a rapidly changing and competitive business environment. DLithe, with its unique methodology of execution, helps participants to identify their strengths and weaknesses, and develop the necessary skills to lead and manage their teams effectively, also highlighting different aspects of leadership such as strategic thinking, communication, collaboration, innovation, and change management.

Train and Deploy:

50-60% of resources on campus fail to find work for a variety of reasons. It doesn't mean they're not capable. They are lacking in preparation and direction. Many resources from on and off campus have been placed in various companies thanks to our Train & Deploy programme.

- **To Academic institutions(Competency development):**

We have worked with, and are trusted by major academic institutions, allowing us to provide services such as Faculty development programs, internships, and placement training with the goal of up skilling students to meet the demands of the corporate world. Our R&D/Incubation centres are housed in the majority of these institutions.

CAMPS@CAMPUS:

Enabling the student community to bridge the gap between academic and industry. Creating an agile workforce, placement training and mentoring hackathons, and internship platforms are making resources employable.

- **DLithe's major Services to Academics:**

- Technical training - form 3rd Semester onwards
- Microcontroller Lab set up
- Robotics lab setup
- Pre placement training
- Webinars/Seminars
- Internships
- Faculty Development Program
- Cyber Forensic Lab setup
- Hackathon
- Workshops

- Certification Programs
- IOT Lab setup

- **DLithe's trusted Academic Partners:**

- NMAMIT Nitte
- Manipal Academy of Higher Education
- IBMR Business School
- SDMIT Ujire
- SJBIT Bangalore
- BMSIT Bangalore
- Dhanalakshmi Srinivasan University
- SMVCE Pondicherry
- Alvas Education Foundations
- MITE
- KL University
- Mangalore University
- BITM

- **DLithe's Trusted Corporate Partners:**

- Renewin
- Softvent Technologies
- Tietoevry
- Mobiez
- Senseops
- International Datalyzer
- Meal Meister
- C-Square
- Pacewisdom

CHAPTER-2

ON JOB TRAINING –1

2.1 Introduction

A Patient Management System (PMS) is a comprehensive software solution designed to streamline and enhance the efficiency of healthcare facilities by digitizing patient data, scheduling appointments, managing medical records, and facilitating communication between healthcare providers.

This system serves as a centralized platform for storing, retrieving, and updating patient information securely, ensuring accuracy and confidentiality. By integrating various functionalities such as billing, prescription management, and reporting, it empowers healthcare professionals to deliver personalized care, optimize resource utilization, and improve patient outcomes. With its user-friendly interface and customizable features, a Patient Management System revolutionizes the healthcare delivery process, promoting seamless coordination and collaboration across all levels of care.

Moreover, a Patient Management System offers advanced analytics capabilities that enable healthcare administrators to gain valuable insights into patient demographics, treatment trends, and operational performance. By harnessing data-driven decision-making, healthcare providers can identify areas for improvement, implement targeted interventions, and ultimately enhance the quality of care delivered. Additionally, these systems often integrate with electronic health records (EHRs) and other healthcare IT solutions, fostering interoperability and ensuring continuity of care across different healthcare settings. With the increasing complexity of healthcare delivery and the growing emphasis on patient-centric care, a robust Patient Management System has become indispensable for modern healthcare organizations striving to deliver efficient, high-quality services while maximizing patient satisfaction and safety.

2.2 Analysis:

2.2.1 System Analysis:

The patient management system (PMS) functions as a centralized hub, offering a comprehensive solution for managing all aspects of patient care within a healthcare organization. Its user-friendly interface streamlines critical tasks like appointment scheduling, patient record management, and secure communication. Healthcare providers can leverage robust reporting features to gain valuable insights into patient demographics, treatment trends, and overall practice performance. By centralizing patient data and automating administrative processes, the PMS enhances efficiency, improves patient care coordination, and empowers healthcare professionals to deliver exceptional service.

2.2.1.1 Advantages:

- **Improved Efficiency**: By digitizing patient records, scheduling appointments, and automating administrative tasks, a Patient Management System significantly reduces the time and effort required to manage patient information.
- **Enhanced Patient Care**: With centralized access to comprehensive patient histories, treatment plans, and medication records, healthcare providers can deliver more personalized and coordinated care.
- **Better Decision Making**: Patient Management Systems offer robust analytics and reporting capabilities, allowing healthcare administrators to analyze trends, monitor key performance indicators, and make data-driven decisions.
- **Increased Patient Engagement**: Many Patient Management Systems include patient portals that empower individuals to access their medical records, schedule appointments, and communicate securely with healthcare providers.

2.2.1.2 Disadvantages:

- **Initial Implementation Costs**: Implementing a Patient Management System can involve significant upfront costs, including software licensing fees, hardware infrastructure upgrades, and staff training expenses.
- **Learning Curve**: Transitioning to a new Patient Management System requires healthcare staff to learn how to use the software effectively. This learning curve can lead to temporary decreases in productivity and potential errors during the adaptation period, especially for employees who are not tech-savvy.
- **System Downtime**: Like any computerized system, Patient Management Systems are susceptible to technical issues and downtime, which can disrupt normal operations and impact patient care.
- **Data Security Risks**: Despite employing security measures, Patient Management Systems are still vulnerable to data breaches, hacking attempts, and unauthorized access.

2.2.2 System Specification:

The system requirements for a Patient Management System can vary depending on the specific software solution chosen and the needs of the healthcare facility. However, here are some general requirements typically necessary to ensure smooth operation:

2.2.2.1 Hardware Requirements:

- **Processor (CPU):**
    - **Minimum**: An Intel Core i3 or AMD Ryzen 3 processor (current or previous generation) will suffice.
    - **Recommended:** For smoother performance, consider an Intel Core i5 or AMD Ryzen 5 processor.
- **Memory (RAM):**
    - **Minimum:** 8GB of RAM is the baseline for running the patient management system efficiently.
    - **Recommended:** 16GB of RAM is ideal for handling multiple concurrent tasks and large datasets.

- **Storage**: A solid-state drive (SSD) with at least 256GB of storage is highly recommended for faster data access and system responsiveness.

Other considerations

- **Operating System:** The patient management system should be compatible with popular operating systems such as Windows, macOS, or Linux, based on the preferences and requirements of the users.
- **Network Connectivity:** Stable internet connectivity is essential for accessing online resources, updating software, and facilitating communication with external systems such as laboratories or insurance providers.
- **Peripheral Devices:** Consideration should be given to peripherals such as printers, scanners, or barcode readers, depending on the specific needs of the healthcare facility.
- **Security Features:** The hardware should support security measures such as encryption, biometric authentication, and regular system updates to ensure patient data confidentiality and compliance with healthcare regulations.

2.2.2.2 Software requirement:

Front-end:

- **HTML**: It is used to structure a web page and its content.
- **CSS**: It is used to style and layout web pages.
- **JavaScript**: It is used to make web pages interactive.

Additional Tools:

- **Version Control System (VCS):** Like Git, to track code changes and collaborate effectively.
- **Web Server:** Software that allows your web application to run on the internet. Common options include Apache and Nginx.

Deployment Platform:

- **Web Hosting Platform:** A service to host our application on the internet. Popular options include AWS, Google Cloud Platform (GCP), and Azure.

2.3 Project Modules:

The Patient Management System web application consists of three modules. They are:

- Admin Module
- Doctor Module
- Patient Module

2.3.1 Admin Module

- **Login:** Admin can securely access their accounts by providing their credentials, typically a combination of username/email and password. This login module ensures that only authorized individuals can access the admin dashboard.
- **Doctor List:** Administrators can access a comprehensive list of all registered doctors within the system. This functionality enables administrators to review and manage doctor profiles efficiently.
- **Add Doctor:** This feature allows administrators to seamlessly add new doctors to the system. Administrators can input essential details such as doctor's name, specialty, contact information, and other relevant information.
- **Patient List:** Administrators can view a detailed list of all registered patients, facilitating efficient management of patient data. This functionality provides administrators with access to patient profiles and medical records as necessary.
- **Add Patient:** This functionality empowers administrators to include new patients into the system seamlessly. It's like creating a new file for a patient in the digital cabinet. Admins input essential details such as the patient's

name, date of birth, contact information, medical history, and any other patient information.

- **Appointments:** Administrators can monitor and manage all appointments within the system, including both visited and cancelled appointments. This functionality enables administrators to track patient appointments and ensure smooth scheduling processes.
- **Admin List:** This feature provides administrators with a comprehensive overview of all registered admin users. Administrators can review and manage admin profiles, including permissions and access levels.
- **Add Admin:** Administrators can add new admin users to the system, facilitating the expansion of administrative roles and responsibilities. This functionality includes setting up login credentials and defining permissions for new admin users.
- **Logout:** This module enables administrators to securely log out of the admin dashboard, terminating their current session and preventing unauthorized access to administrative functionalities.

2.3.2 Doctor Module

- **Login:** Doctor can securely access their accounts by providing their credentials, typically a combination of username/email and password. This login module ensures that only authorized individuals can access the doctor dashboard.
- **View Appointment:** Doctors or staff members can access a comprehensive list of all appointments scheduled by patients. This functionality allows them to review appointment details and take actions such as accepting or declining appointments directly from the dashboard.
- **Patients List:** Doctors or staff members can view a detailed list of all registered patients within the system. This feature provides access to patient profiles, medical histories, and other relevant information necessary for providing healthcare services.

- **Prescription:** This feature enables doctors or staff members to generate prescriptions for patients based on their appointments and medical needs. It allows them to document prescribed medications, dosages, and other treatment instructions.
- **Logout:** This module enables doctors or staff members to securely log out of the dashboard, terminating their current session and ensuring data security.

2.3.3 Patient Module

- **Registration:** Patients can register for an account by providing necessary details such as username, email, password, and other relevant information. This registration module enables new users to create accounts and access the patient dashboard.
- **Login:** Patients can securely access their accounts by providing their credentials, typically a combination of username/email and password. This login module ensures that only authorized individuals can access the patient dashboard.
- **Book Appointment:** Patients can access a feature to schedule appointments with doctors. This functionality allows them to select preferred dates and times for appointments, providing convenience in booking healthcare services.
- **Appointment History:** Patients can view a comprehensive history of all past appointments, including both cancelled and deleted appointments. This feature enables patients to track their appointment records and histories efficiently.
- **My Appointments:** Patients can access a dedicated section to view all upcoming appointments, including those that are approved and awaiting confirmation. This functionality provides patients with visibility and control over their scheduled appointments.

- **Prescriptions:** Patients can access a repository of all prescriptions prescribed by specific doctors. This feature enables patients to review and access their prescribed medications and treatment instructions conveniently.
- **Logout:** This module enables patients to securely log out of the dashboard, terminating their current session and ensuring data privacy.

2.4 Implementation

Filename: admin_interface.html

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Admin Interface</title>

<link rel="stylesheet" href="admin_interface.css">

<link rel="icon" href="../images/favicon.png" type="image/x-icon">

<link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600
,600i,700,700i|Raleway:300,300i,400,400i,500,500i,600,600i,700,700i|Poppins:3
00,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.1/css/all.min.css" integrity="sha512-
DTOQO9RWCH3ppGqcWaEA1BIZOC6xxalwEsw9c2QQeAIftl+Vegovlnee1c9
QX4TctnWMn13TZye+giMm8e2LwA==" crossorigin="anonymous"
referrerpolicy="no-referrer" />

</head>

<body onload="loadDashboard()">

<header id="header">
```

```
<a href="../index.jsp" id="logo" target="_blank"> Aarogya Healthcare </a>
<nav>
<div class="search-filter">
<textarea id="search-filter" placeholder="Search Email Address..."></textarea>
</div>
<div class="search-user-filter">
<textarea id="search-user-filter"
placeholder="Search Patient Email..."></textarea>
</div>
<div class="search-doctor-filter">
<textarea id="search-doctor-filter"
placeholder="Search Doctor Email..."></textarea>
</div>
<div class="logout-container"
onclick="window.location.replace('../logins/admin_login.html')">
<i class="fa-solid fa-right-from-bracket"></i>
<button>Logout</button>
</div>
</nav>
</header>
<main id="main">
<div id="sidebar">
<div class="sidebar-options" onclick="loadDashboard()">
<i class="fa-solid fa-house-chimney"></i>
```

```
<p>Dashboard</p>
</div>
<div class="sidebar-options" onclick="loadDoctorList()">
<i class="fa-solid fa-user-doctor"></i>
<p>Doctor List</p>
</div>
<div class="sidebar-options" onclick="loadAddDoctor()">
<i class="fa-solid fa-hand-holding-medical"></i>
<p>Add Doctor</p>
</div>
<div class="sidebar-options" onclick="loadPatientList()">
<i class="fa-solid fa-user"></i>
<p>Patient List</p>
</div>
<div class="sidebar-options" onclick="loadAddPatient()">
<i class="fa-solid fa-user-plus"></i>
<p>Add Patient</p>
</div>
<div class="sidebar-options" onclick="loadAppointments()">
<i class="fa-solid fa-calendar-check"></i>
<p>Appointments</p>
</div>
<div class="sidebar-options" onclick="loadAdminList()">
<i class="fa-solid fa-user-tie"></i>
```

```
<p>Admin List</p>
</div>
<div class="sidebar-options" onclick="loadAddAdmin()">
<i class="fa-solid fa-user-plus"></i>
<p>Add Admin</p>
</div>
</div>
<div id="right-body"></div>
</main>
<script src="admin_interface.js" defer></script>
</body>
</html>
```

2.5 Conclusion

In conclusion, our patient management system provides a robust and intuitive platform to enhance healthcare delivery and patient care. With features like appointment scheduling, prescription management, and comprehensive patient records, our system streamlines administrative tasks and facilitates efficient patient-doctor interactions.

Imagine spending less time on manual paperwork and more on providing personalized care to your patients. Picture a seamless experience for both healthcare providers and patients, with easy access to medical histories, appointments, and prescriptions.

Embrace the future of healthcare management with our patient management system. Improve patient outcomes, increase operational efficiency, and elevate the quality

of care provided. With our system, you can focus on what truly matters – the health and well-being of your patients.

Don't just manage patients, elevate healthcare delivery. Embrace our patient management system and experience the difference.

2.6 Future Enhancement

- **AI-Powered Appointment Scheduling:** Implement an AI-powered appointment scheduling system that analyses patient preferences, doctor availability, and clinic capacity to optimize appointment scheduling
- **Virtual Health Assistant:** Integrate a virtual health assistant powered by natural language processing (NLP) and machine learning algorithms. This assistant can provide patients with personalized health advice, medication reminders, and guidance on managing chronic conditions, enhancing patient engagement and self-care.
- **Predictive Analytics for Patient Outcomes:** Utilize predictive analytics to forecast patient outcomes based on historical data, treatment plans, and demographic factors. Healthcare providers can use this information to proactively identify high-risk patients, tailor treatment strategies, and improve clinical decision-making.
- **Block chain-based Health Records Management:** Implement blockchain technology to securely manage and share patient health records across healthcare providers. Blockchain offers tamper-proof record-keeping, data integrity, and interoperability, ensuring patient data privacy and facilitating seamless information exchange between healthcare organizations.
- **Remote Monitoring and Telehealth Integration:** Expand telehealth capabilities by integrating remote monitoring devices and telehealth platforms into the patient management system. Patients can transmit vital signs, monitor chronic conditions, and consult with healthcare providers remotely, enabling proactive care management and reducing hospital readmissions.

CHAPTER-3

ON JOB TRAINING –2

3.1 Introduction

A Patient Management System (PMS) is a comprehensive software solution designed to streamline and optimize the processes involved in managing patient information within healthcare facilities. These systems play a crucial role in modern healthcare settings, providing healthcare professionals with efficient tools to record, organize, and access patient data securely. From appointment scheduling and medical records management to billing and insurance processing, a PMS serves as the backbone of administrative and clinical operations in hospitals, clinics, and other healthcare institutions.

One of the primary benefits of a PMS is its ability to centralize patient information, eliminating the need for paper-based records and disparate electronic systems. By consolidating data into a unified platform, healthcare providers can access patient records quickly and accurately, facilitating better decision-making and enhancing patient care outcomes. Additionally, PMS software often includes features such as electronic health records (EHR), which allow clinicians to capture, store, and share patient data in real-time, promoting collaboration and continuity of care across different healthcare settings.

Furthermore, Patient Management Systems play a pivotal role in improving operational efficiency and reducing administrative burdens within healthcare organizations. These systems automate routine tasks such as appointment scheduling, prescription management, and billing, freeing up staff time to focus on delivering quality patient care. Moreover, PMS software often integrates with other healthcare technologies, such as telemedicine platforms and diagnostic tools, enabling seamless data exchange and enhancing the overall patient experience.

3.2 Analysis:

3.2.1 System Analysis:

The Patient Management System (PMS) functions as a pivotal tool in modern healthcare organizations, offering a centralized solution for managing various aspects of patient care. Its intuitive interface facilitates crucial tasks such as appointment scheduling, patient record maintenance, and secure communication among healthcare professionals. Robust reporting features empower healthcare providers to extract valuable insights into patient demographics, treatment patterns, and overall practice performance. Through the consolidation of patient data and automation of administrative workflows, the PMS enhances operational efficiency, fosters better patient care coordination, and enables healthcare professionals to deliver superior service.

3.2.1.1 Advantages:

- **Streamlined Workflow:** The digitization of patient records and automation of administrative tasks significantly reduce the time and effort required for managing patient information
- **Personalized Care Delivery:** With access to comprehensive patient histories, treatment plans, and medication records, healthcare providers can tailor treatments to individual needs.
- **Informed Decision-Making:** Patient Management Systems offer robust analytics and reporting capabilities, enabling healthcare administrators to analyze trends, monitor key performance indicators, and make informed decisions.
- **Enhanced Patient Engagement:** Many Patient Management Systems include patient portals, empowering individuals to access their medical records, schedule appointments, and communicate securely with healthcare providers.

3.2.1.2 Disadvantages:

- **Initial Investment:** The implementation of a Patient Management System entails significant upfront costs, including software licensing fees, hardware upgrades, and staff training expenses.
- **Learning Curve:** Transitioning to a new Patient Management System requires healthcare staff to adapt to the software's functionalities, which can lead to temporary decreases in productivity and potential errors.
- **System Downtime:** Patient Management Systems are susceptible to technical issues and downtime, which can disrupt normal operations and affect patient care.
- **Data Security Risks:** Despite stringent security measures, Patient Management Systems remain vulnerable to data breaches, hacking attempts, and unauthorized access.

3.2.2 System Specification:

The system requirements for a Patient Management System can vary depending on the specific software solution chosen and the needs of the healthcare facility. However, here are some general requirements typically necessary to ensure smooth operation:

3.2.2.1 Hardware Requirements:

- **Processor (CPU):**
    - **Minimum**: An Intel Core i3 or AMD Ryzen 3 processor (current or previous generation) will suffice.
    - **Recommended:** For smoother performance, consider an Intel Core i5 or AMD Ryzen 5 processor.
- **Memory (RAM):**
    - **Minimum:** 8GB of RAM is the baseline for running the patient management system efficiently.
    - **Recommended:** 16GB of RAM is ideal for handling multiple concurrent tasks and large datasets.

- **Storage**: A solid-state drive (SSD) with at least 256GB of storage is highly recommended for faster data access and system responsiveness.

Other considerations

- **Operating System:** The patient management system should be compatible with popular operating systems such as Windows, macOS, or Linux, based on the preferences and requirements of the users.
- **Network Connectivity:** Stable internet connectivity is essential for accessing online resources, updating software, and facilitating communication with external systems such as laboratories or insurance providers.
- **Peripheral Devices:** Consideration should be given to peripherals such as printers, scanners, or barcode readers, depending on the specific needs of the healthcare facility.
- **Security Features:** The hardware should support security measures such as encryption, biometric authentication, and regular system updates to ensure patient data confidentiality and compliance with healthcare regulations.

3.2.2.2 Software requirement

Back-end

- **Java**: It is used to create backend web application using servlets or JSP.
- **Apache HTTP Server**: It is used to accept HTTP requests from visitors and send them back the requested information.
- **Apache Tomcat Server**: It is used as a servlet container in the webserver, and is used to serve dynamic webpages.
- **MySQL**: It is used to store data in tables that map to objects.

Additional Tools

- **Version Control System (VCS):** Like Git, to track code changes and collaborate effectively.
- **Web Server:** Software that allows your web application to run on the internet. Common options include Apache and Nginx.

Deployment Platform

- **Web Hosting Platform:** A service to host your application on the internet. Popular options include AWS, Google Cloud Platform (GCP), and Azure.

3.3 Project Modules

The Patient Management System web application consists of three modules. They are:

- Admin Module
- Doctor Module
- Patient Module

3.3.1 Admin Module

- **Login:** Admin can securely access their accounts by providing their credentials, typically a combination of username/email and password. This login module ensures that only authorized individuals can access the admin dashboard.
- **Doctor List:** Administrators can access a comprehensive list of all registered doctors within the system. This functionality enables administrators to review and manage doctor profiles efficiently.
- **Add Doctor:** This feature allows administrators to seamlessly add new doctors to the system. Administrators can input essential details such as doctor's name, specialty, contact information, and other relevant information.
- **Patient List:** Administrators can view a detailed list of all registered patients, facilitating efficient management of patient data. This functionality provides administrators with access to patient profiles and medical records as necessary.
- **Add Patient:** This functionality empowers administrators to include new patients into the system seamlessly. It's like creating a new file for a patient in the digital cabinet. Admins input essential details such as the patient's

name, date of birth, contact information, medical history, and any other patient information.

- **Appointments:** Administrators can monitor and manage all appointments within the system, including both visited and cancelled appointments. This functionality enables administrators to track patient appointments and ensure smooth scheduling processes.
- **Admin List:** This feature provides administrators with a comprehensive overview of all registered admin users. Administrators can review and manage admin profiles, including permissions and access levels.
- **Add Admin:** Administrators can add new admin users to the system, facilitating the expansion of administrative roles and responsibilities. This functionality includes setting up login credentials and defining permissions for new admin users.
- **Logout:** This module enables administrators to securely log out of the admin dashboard, terminating their current session and preventing unauthorized access to administrative functionalities.

3.3.2 Doctor Module

- **Login:** Doctor can securely access their accounts by providing their credentials, typically a combination of username/email and password. This login module ensures that only authorized individuals can access the doctor dashboard.
- **View Appointment:** Doctors or staff members can access a comprehensive list of all appointments scheduled by patients. This functionality allows them to review appointment details and take actions such as accepting or declining appointments directly from the dashboard.
- **Patients List:** Doctors or staff members can view a detailed list of all registered patients within the system. This feature provides access to patient profiles, medical histories, and other relevant information necessary for providing healthcare services.

- **Prescription:** This feature enables doctors or staff members to generate prescriptions for patients based on their appointments and medical needs. It allows them to document prescribed medications, dosages, and other treatment instructions.
- **Logout:** This module enables doctors or staff members to securely log out of the dashboard, terminating their current session and ensuring data security.

3.3.3 Patient Module

- **Registration:** Patients can register for an account by providing necessary details such as username, email, password, and other relevant information. This registration module enables new users to create accounts and access the patient dashboard.
- **Login:** Patients can securely access their accounts by providing their credentials, typically a combination of username/email and password. This login module ensures that only authorized individuals can access the patient dashboard.
- **Book Appointment:** Patients can access a feature to schedule appointments with doctors. This functionality allows them to select preferred dates and times for appointments, providing convenience in booking healthcare services.
- **Appointment History:** Patients can view a comprehensive history of all past appointments, including both cancelled and deleted appointments. This feature enables patients to track their appointment records and histories efficiently.
- **My Appointments:** Patients can access a dedicated section to view all upcoming appointments, including those that are approved and awaiting confirmation. This functionality provides patients with visibility and control over their scheduled appointments.

- **Prescriptions:** Patients can access a repository of all prescriptions prescribed by specific doctors. This feature enables patients to review and access their prescribed medications and treatment instructions conveniently.
- **Logout:** This module enables patients to securely log out of the dashboard, terminating their current session and ensuring data privacy.

3.4 Implementation

Filename: adminDashboard.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ page import="java.sql.*"%>
<%
Connection con = null;
Statement st = null;
ResultSet rs = null;

int totalPatients = 0;
int totalDoctors = 0;
int totalAppointments = 0;
int totalUnreadQueries = 0;
int totalReadQueries = 0;

String opname = request.getParameter("opname");
String queryId = request.getParameter("qid");

try {
Class.forName("com.mysql.cj.jdbc.Driver");
con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/aarogya_healthcare_
db","root","trickortreat");
```

```
st = con.createStatement();

String getTotalPatients = "SELECT COUNT(*) FROM registered_users";
String getTotalDoctors = "SELECT COUNT(*) FROM registered_doctors";
String getTotalAppointments = "SELECT COUNT(*) FROM
appointment_records";
String getTotalUnreadQueries = "SELECT COUNT(*) FROM query_records
WHERE message_status = \"unread\"";
String getTotalReadQueries = "SELECT COUNT(*) FROM query_records
WHERE message_status = \"read\"";

rs = st.executeQuery(getTotalPatients);
rs.next();
totalPatients = rs.getInt(1);

rs = st.executeQuery(getTotalDoctors);
rs.next();
totalDoctors = rs.getInt(1);

rs = st.executeQuery(getTotalAppointments);
rs.next();
totalAppointments = rs.getInt(1);

rs = st.executeQuery(getTotalUnreadQueries);
rs.next();
totalUnreadQueries = rs.getInt(1);

rs = st.executeQuery(getTotalReadQueries);
rs.next();
totalReadQueries = rs.getInt(1);
```

```
String unreadQueryRecordsSQL = "SELECT * FROM query_records WHERE
message_status = \"unread\" ORDER BY query_id DESC";
String readQueryRecordsSQL = "SELECT * FROM query_records WHERE
message_status = \"read\" ORDER BY query_id DESC";

if (queryId != null) {
String updateQueryStatus = "UPDATE query_records SET message_status =
\"read\" WHERE query_id = " + queryId;
st.executeUpdate(updateQueryStatus);
}

if (opname.equals("unread")) {
rs = st.executeQuery(unreadQueryRecordsSQL);
} else if (opname.equals("read")) {
rs = st.executeQuery(readQueryRecordsSQL);
}
%>

<div class="dashboard-top-container">
<div class="view-details" style="border-left: 10px solid orange;">
<p id="appointments">Total Appointments: <%= totalAppointments %></p>
</div>

<div class="view-details"
style="border-left: 10px solid rgb(1, 234, 255);">
<p id="doctors">Total Doctors: <%= totalDoctors %></p>
</div>

<div class="view-details" style="border-left: 10px solid greenyellow;">
<p id="patients">Total Patients: <%= totalPatients %></p>
</div>
</div>
```

```
<div class="dashboard-bottom-container">
<h3>Query List</h3>

<div class="container-tabs">
<p class="container-tabs-options" onclick="loadUnreadQueries()">Unread
Queries</p>
<p class="container-tabs-options" onclick="loadReadQueries()">Read
Queries</p>

<div id="query-count-container">
<h3 id="query-count">
Number of queries:
<%
int queriesTotal = 0;
if (opname.equals("unread")) {
queriesTotal = totalUnreadQueries;
%>
<%= totalUnreadQueries %>
<%
} else if (opname.equals("read")) {
queriesTotal = totalReadQueries;
%>
<%= totalReadQueries %>
<%
}
%>
</h3>
</div>
</div>
<div class="query-records-container">
<table>
```

```
<tr>
<th>Date</th>
<th>Name</th>
<th>Subject</th>
<th>View Message</th>
<th>Email Address</th>
</tr>
<%
for (int i = 1; i <= queriesTotal; i++) {
rs.next();
%>
<tr>
<td style="text-align: center"><%= rs.getString(7) %></td>
<td><%= rs.getString(3) %></td>
<td><%= rs.getString(4) %></td>
<td style="text-align: center">
<%
if (rs.getString(6).equals("unread")) {
%>
<i class="fa-solid fa-envelope" onclick="showMessage(this)"></i>
<%
} else if (rs.getString(6).equals("read")) {
%>
<i class="fa-solid fa-envelope-open" onclick="showMessage(this)"></i>
<%
}
%>
<div class="message" id="<%= rs.getString(1) %>">
<i class="fa-solid fa-square-xmark" id="<%= rs.getString(6) %>"
onclick="closeMessage(this)"></i>
<p><%= rs.getString(5) %></p>
</div>
```

```
</td>
<td><%= rs.getString(2) %></td>
</tr>
<%
}
%>
</table>
</div>
</div>
<%
con.close();
} catch(Exception e) {
e.printStackTrace();
}
%>
```

3.5 Conclusion

In conclusion, our patient management system epitomizes the pinnacle of modern healthcare administration, offering a comprehensive and intuitive platform designed to enhance healthcare delivery and elevate the standard of patient care. By seamlessly integrating features such as appointment scheduling, prescription management, and detailed patient records, our system serves as a catalyst for streamlining administrative processes and fostering efficient communication between healthcare providers and patients.

Imagine a healthcare landscape where the tedious burden of manual paperwork is replaced by a dynamic and responsive system, allowing healthcare professionals to devote more time and energy to delivering personalized care to their patients. Envision a future where the patient experience is characterized by accessibility, transparency, and empowerment, with easy access to medical histories, appointment scheduling, and prescription management at their fingertips.

Embracing our patient management system signifies a commitment to embracing the forefront of healthcare innovation. It represents an opportunity to catalyse tangible improvements in patient outcomes, operational efficiency, and the overall quality of care provided. With our system as a cornerstone of healthcare administration, healthcare providers can reallocate resources towards what truly matters – nurturing the health and well-being of their patients.

In essence, our patient management system transcends mere software; it embodies a transformative shift in healthcare delivery. By leveraging cutting-edge technology and intuitive design, our system empowers healthcare providers to deliver exceptional care experiences that prioritize patient-centricity, efficiency, and effectiveness. Embrace our patient management system, and embark on a journey towards redefining the future of healthcare administration and patient care.

3.6 Future Enhancement

- **Smart Prescription Management:** Introduce a smart prescription management feature that utilizes machine learning algorithms to analyse patient health records, medication histories, and drug interactions. This system can provide personalized medication recommendations, dosage adjustments, and alerts for potential adverse drug reactions, promoting safer medication practices and optimizing treatment outcomes.
- **Remote Patient Education Modules:** Develop interactive patient education modules within the system to deliver tailored health information, lifestyle recommendations, and treatment protocols remotely. These modules can cover various topics such as chronic disease management, preventive care measures, and post-treatment recovery plans, empowering patients to take an active role in their healthcare journey and improve health literacy.
- **Wearable Device Integration:** Integrate wearable health monitoring devices, such as smart watches and fitness trackers, with the patient management system to enable real-time tracking of patient health metrics,

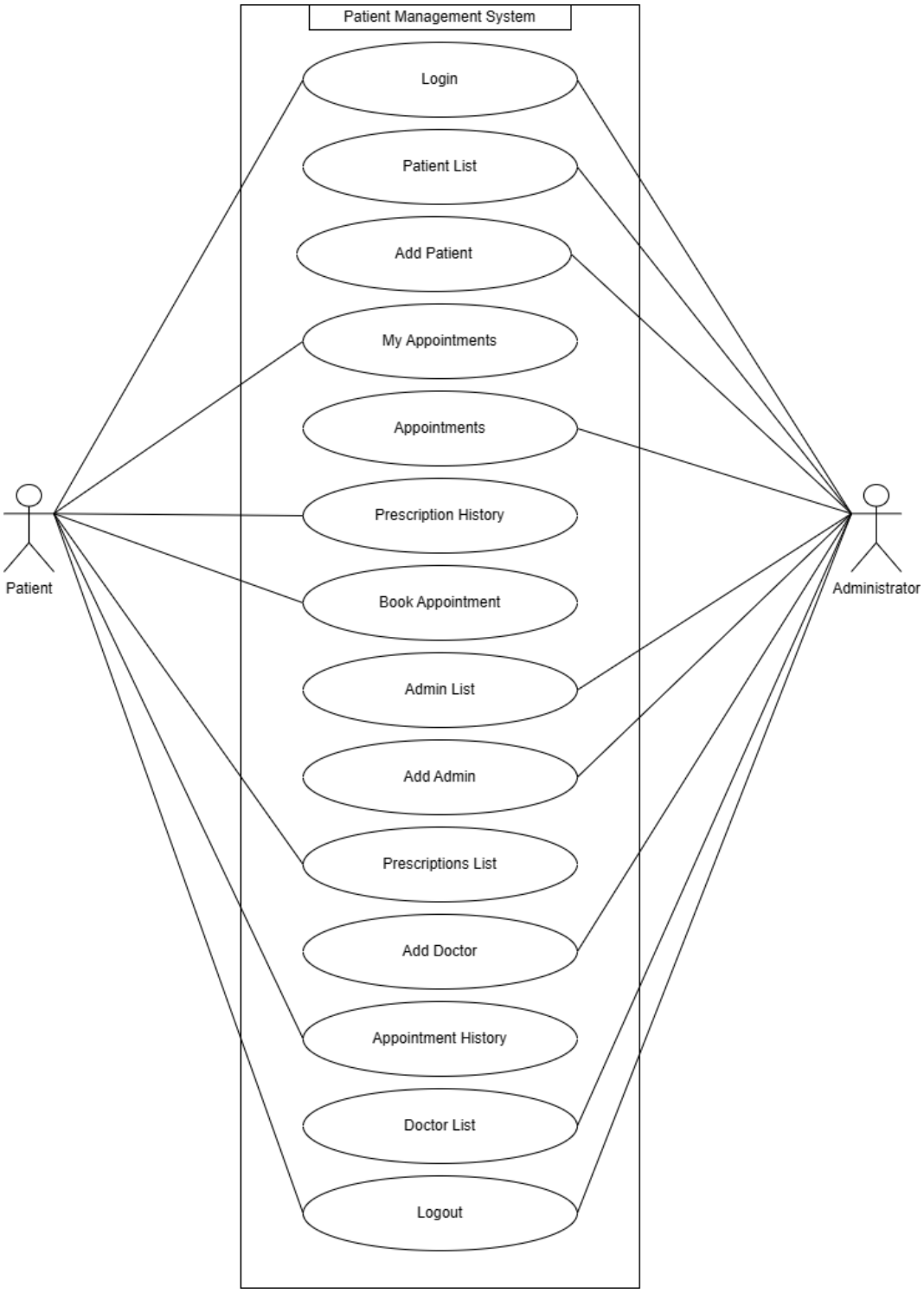
activity levels, and adherence to treatment plans. Healthcare providers can remotely monitor patient progress, identify trends, and intervene proactively in case of anomalies, fostering continuous care management and early intervention strategies.

- **Personalized Health Risk Assessment:** Develop a personalized health risk assessment tool that evaluates individual health risks based on genetic predispositions, lifestyle factors, and environmental influences. By analysing comprehensive patient data, including family history, biomarkers, and behavioural patterns, this tool can generate personalized risk profiles, preventive recommendations, and tailored wellness plans to empower patients to make informed health decisions and mitigate future health risks.

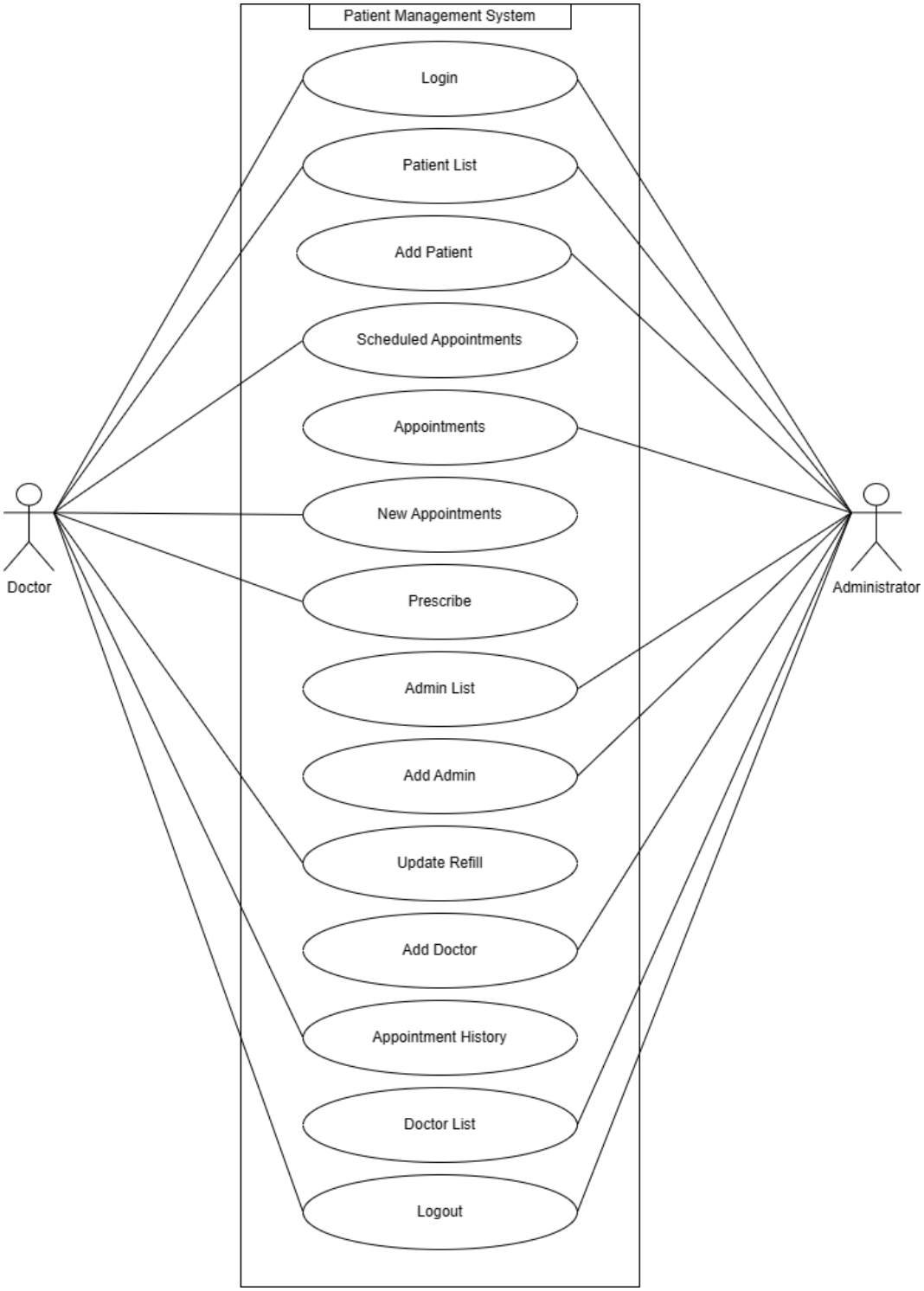
CHAPTER-4

USE CASE DIAGRAM

4.1 USE CASE-1



4.2 USE CASE-2


Patient Management System
Login
Patient List
Add Patient
Scheduled Appointments
Appointments
New Appointments
Prescribe
Admin List
Add Admin
Update Refill
Add Doctor
Appointment History
Doctor List
Logout
Doctor
Administrator

STUDENT RESUME

DISHANTH SHETTY

Student

Reg. No: 309CS21012
Computer Science
Diploma in Computer Science and Engineering
NRAM Polytechnic Nitte, Karkala.

Basral House, Ranganpalke Post,
Karkala Taluk, Udupi, Karnataka
+91 8296369142
dishanthshetty120@gmail.com

Skills

- Database Management System
- Proficiency in Programming Languages
- Basic knowledge in Programming Languages like JAVA etc.
- Basics of DSP (Data structures with Python)

Personal Projects

Patient Management System (PMS) Using Java Full Stack (Both Frontend and Backend Development).

VS Code, MySQL, JSP, Servlets, JDBC, GitHub.

Developed a web application that streamlines the patient management system process for Administrator, Doctors and Patients modules (Contributed to development of an efficient output patient management web application.)

Experience

Internship / DLithe Consultancy Services Pvt. Ltd., Bengaluru
From 10 Jan 2024 To 15 April 2024 – (Java Full Stack)

Education

APRIL, 2019
Govt. Pre University College, Bailur
SSLC – NCERT – Karnataka Secondary School Examination Board, Karnataka - 484/625 –77.44%

JULY, 2021
PUC – HEBA – Department Of Pre-University Education, Karnataka - 456/600 – 76.00%

N.R.A.M POLYTECHNIC NITTE, Karkala.
Diploma 3rd year in Computer Science and Engineering – 2024, 384/400 – 96.00%

Certificates

- Learning Full Stack Development, Infosys Springboard 2023
- Learning HTML, CSS and JavaScript for beginners, Infosys Springboard 2023
- Learning Dependency Injection, Infosys Springboard 2023

Languages

English - *Professional Proficiency*
Kannada - *Full Professional Proficiency*
Tulu - *Full Professional Proficiency*

PHOTO GALLERY

Home Landing Page

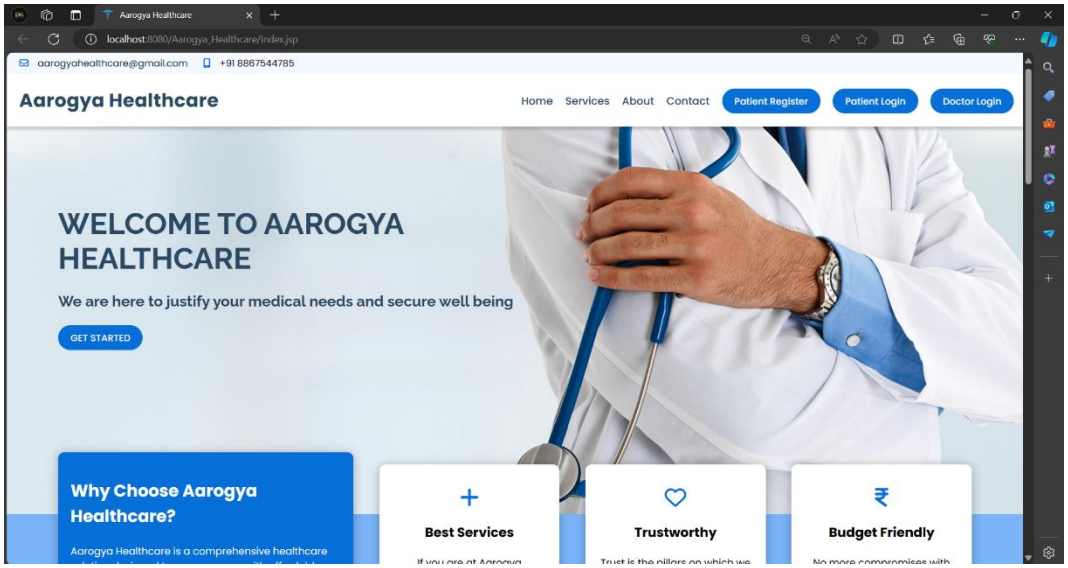


Figure 1 Home Landing Page

1. Administrator

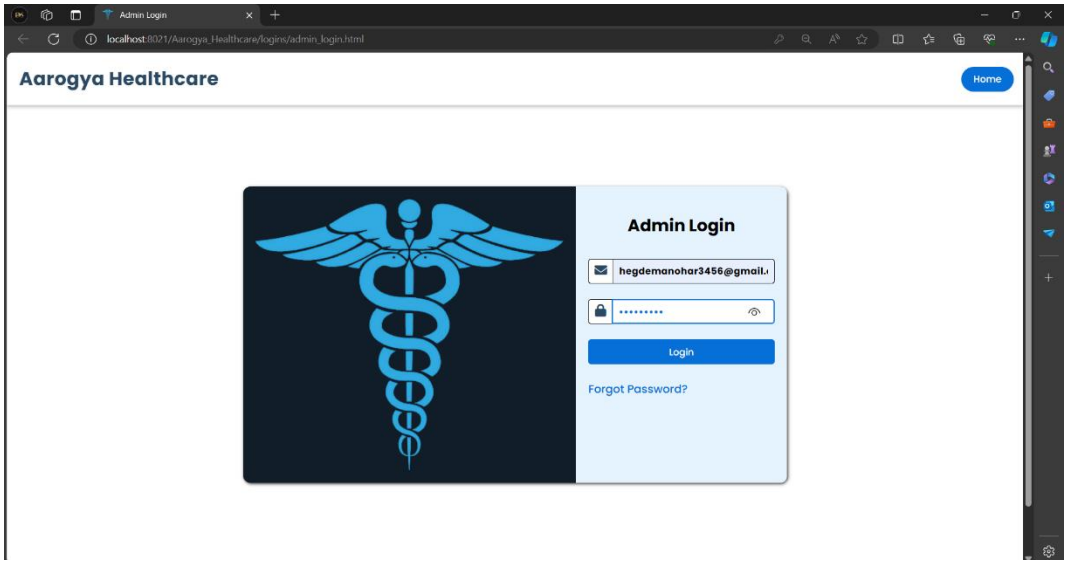


Figure 1.1 Admin Login

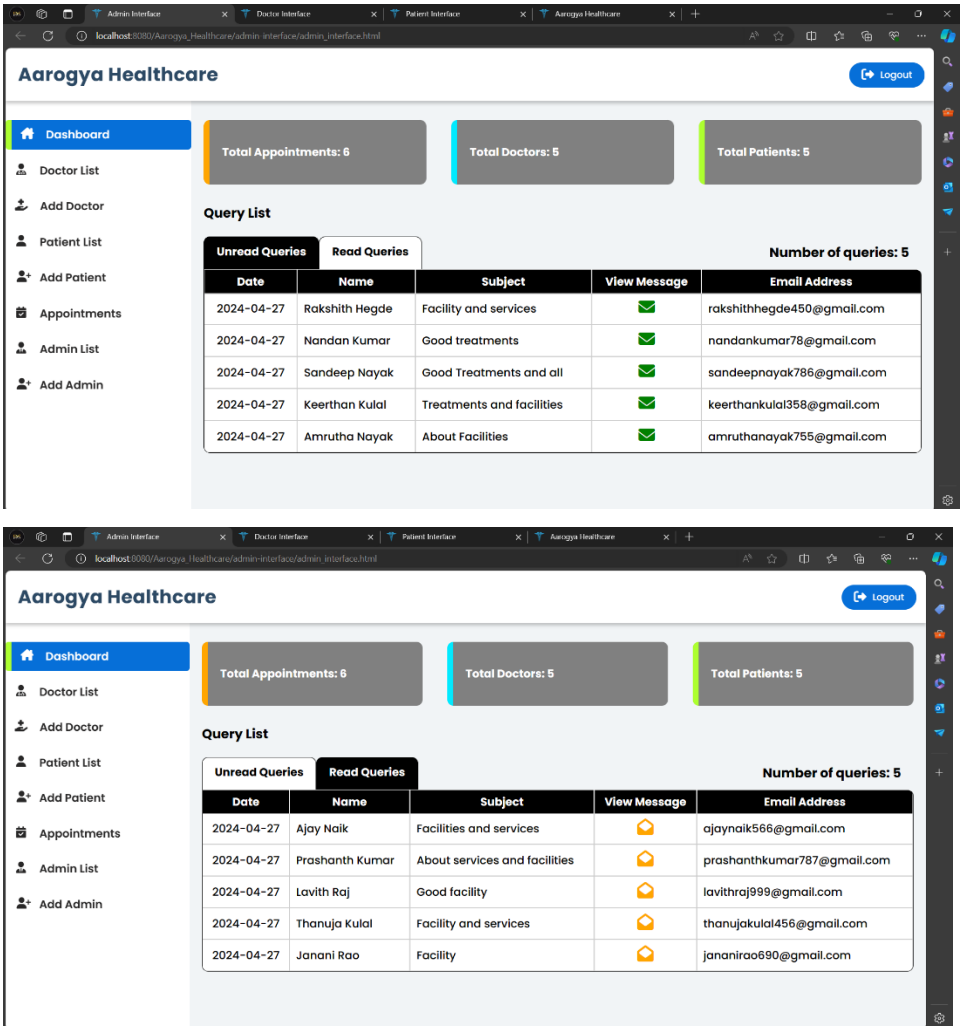


Figure 1.2 Admin Dashboard

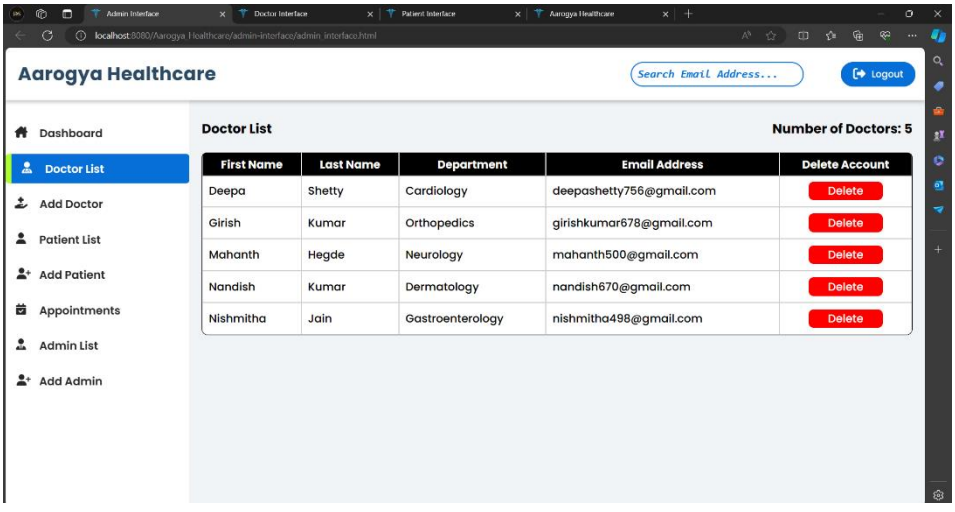


Figure 1.3 Doctor List

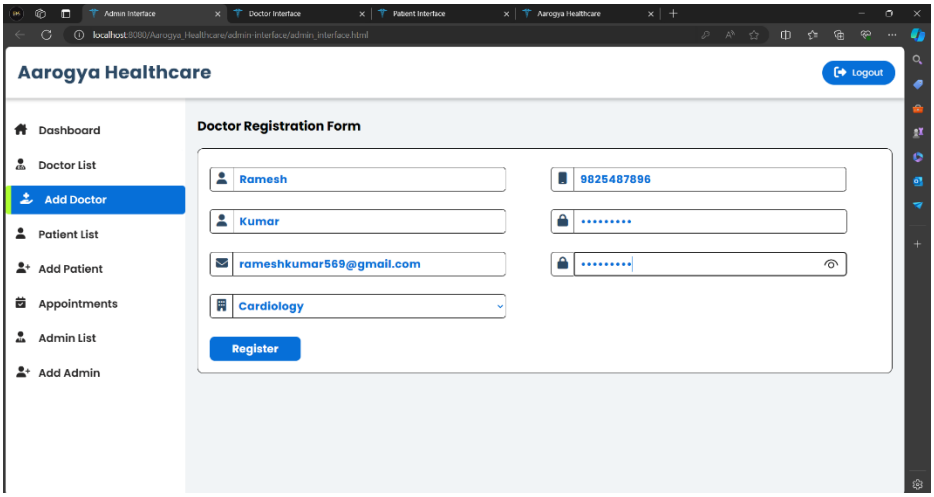


Figure 1.4 Add Doctor

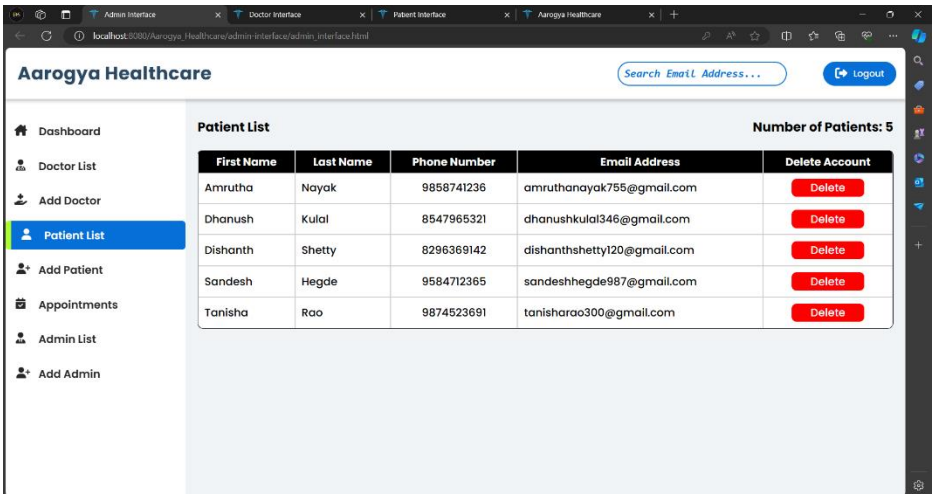


Figure 1.5 Patient List

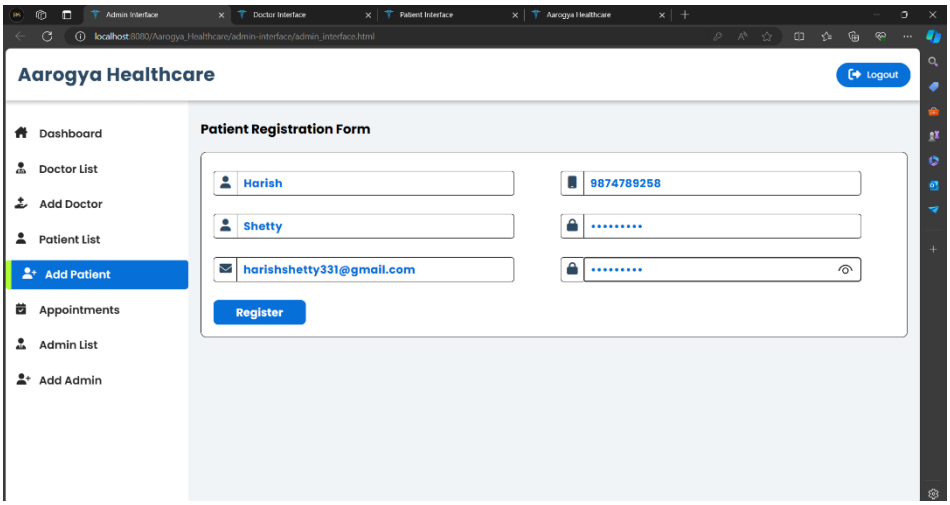


Figure 1.6 Add Patient

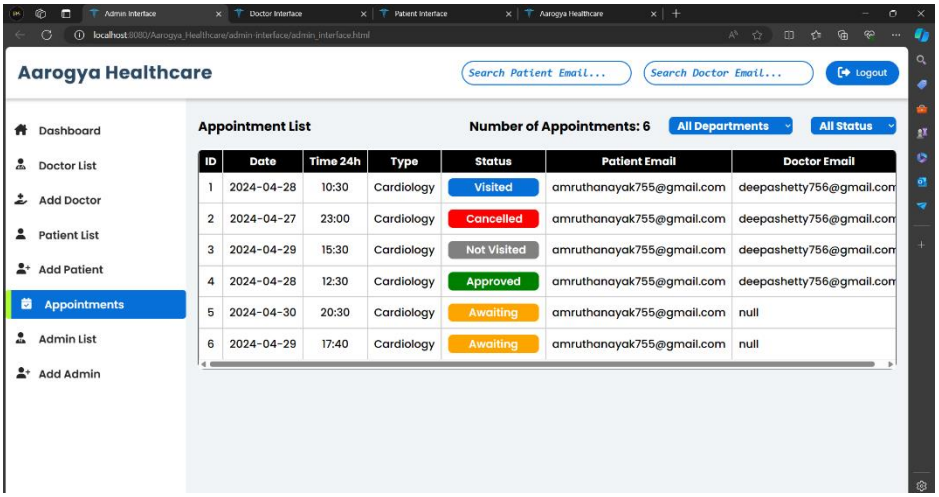


Figure 1.7 Appointments

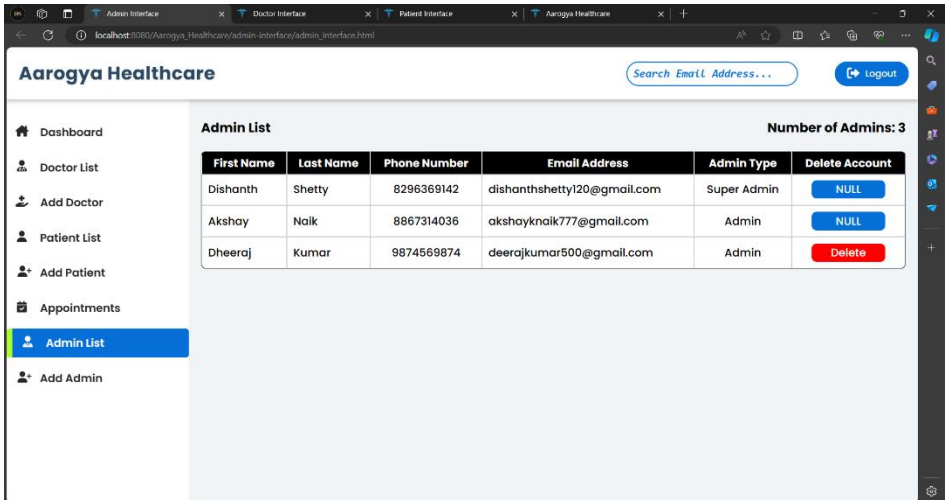


Figure 1.8 Admin List

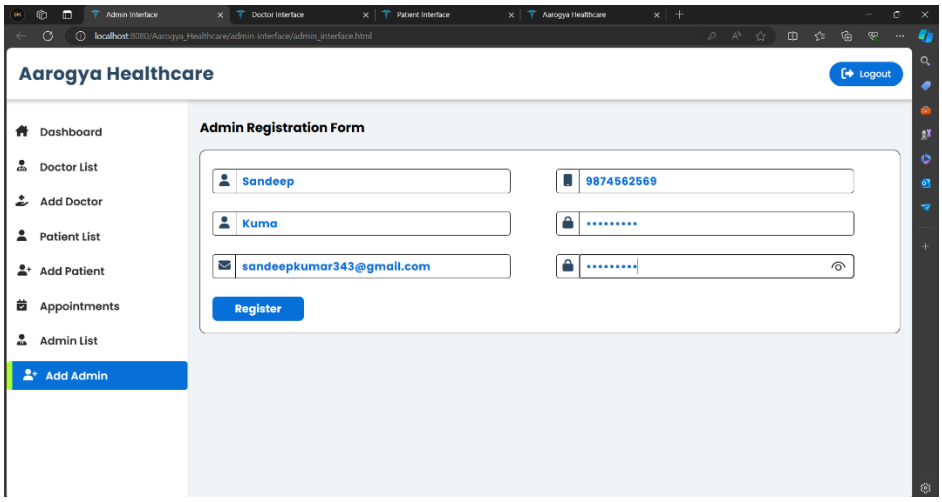


Figure 1.9 Add Admin

2. Doctor

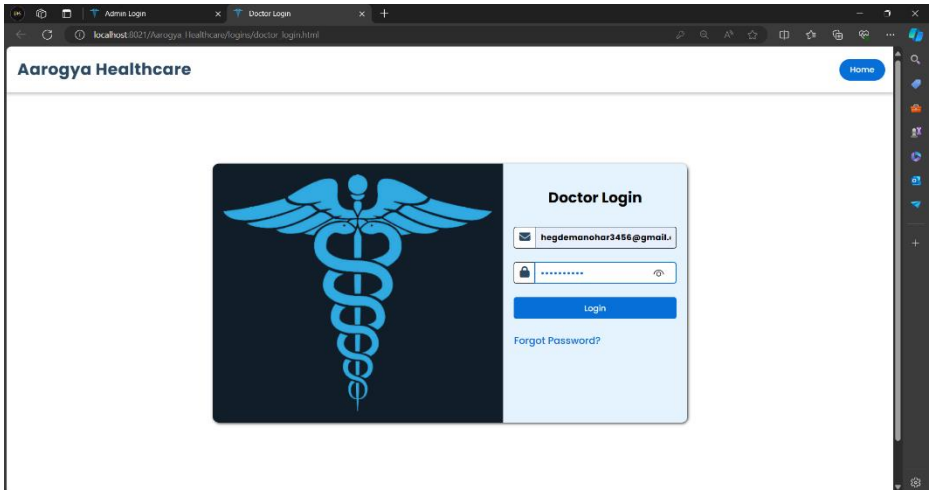


Figure 2.1 Doctor Login

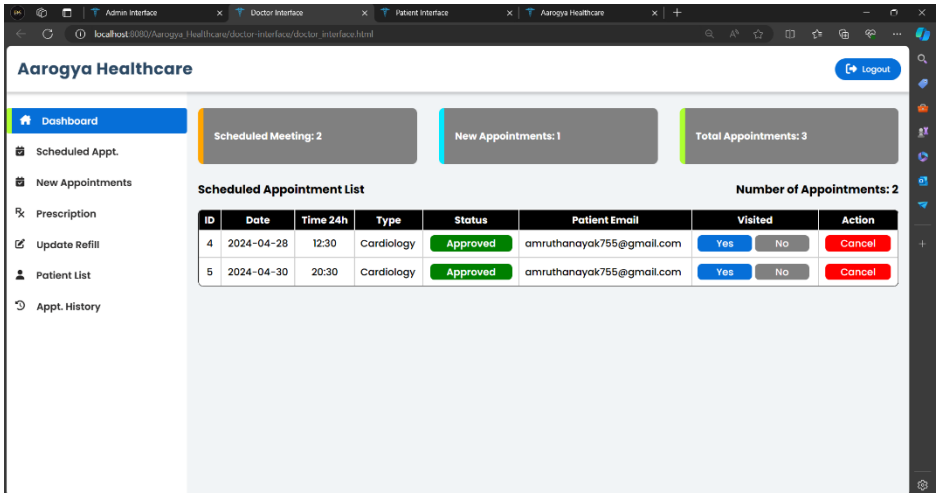


Figure 2.2 Doctor Dashboard

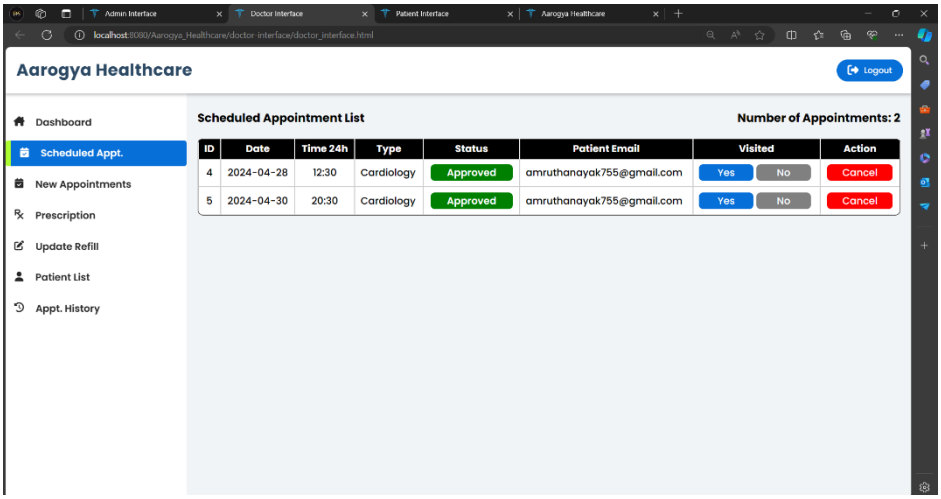


Figure 2.3 Scheduled Appointments

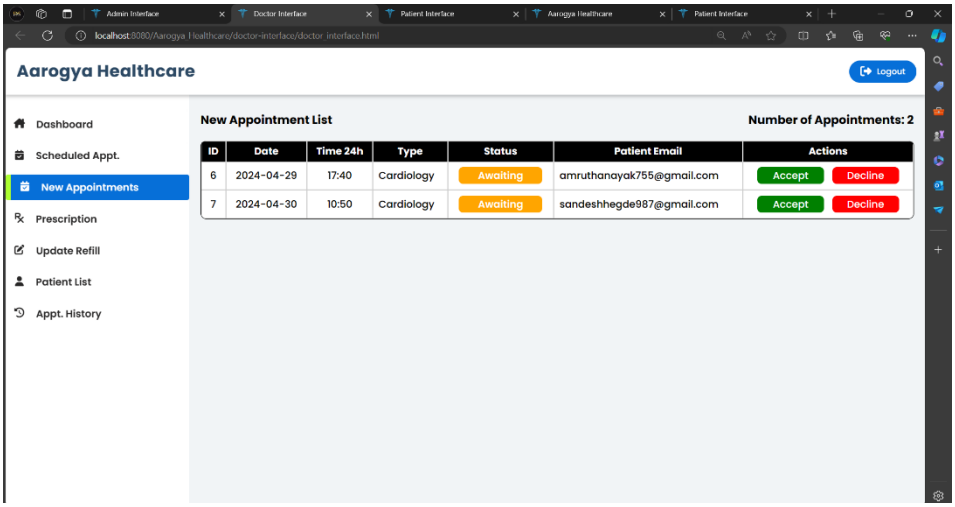


Figure 2.4 New Appointments

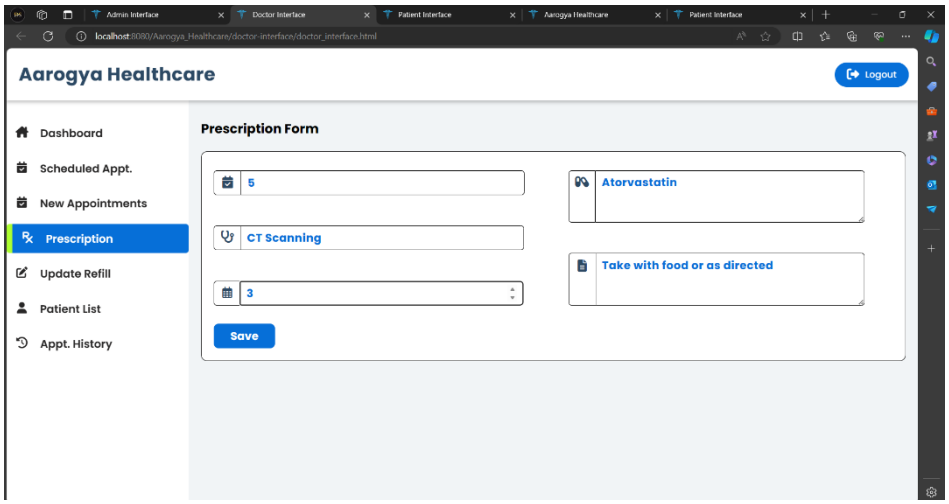


Figure 2.5 Prescription

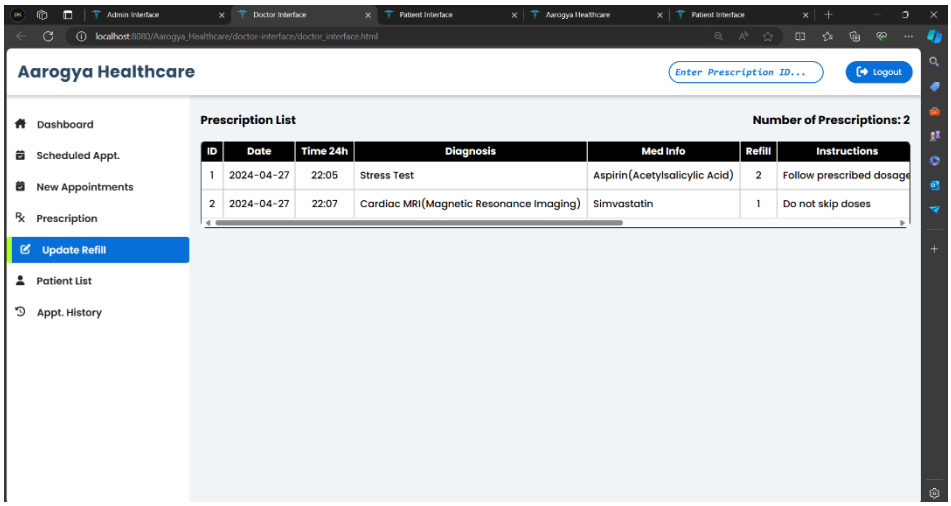


Figure 2.6 Update Refill

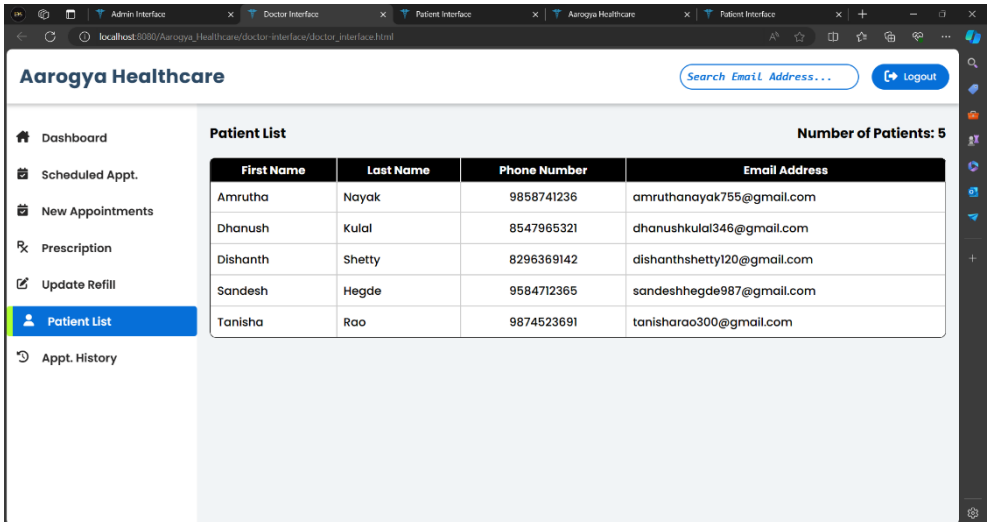


Figure 2.7 Patient List

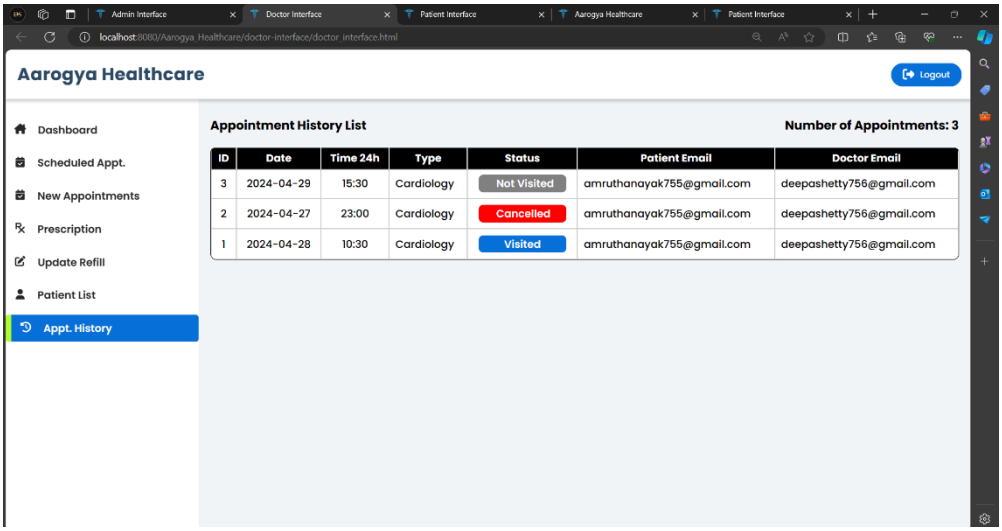


Figure 2.8 Appointment History

3. Patient

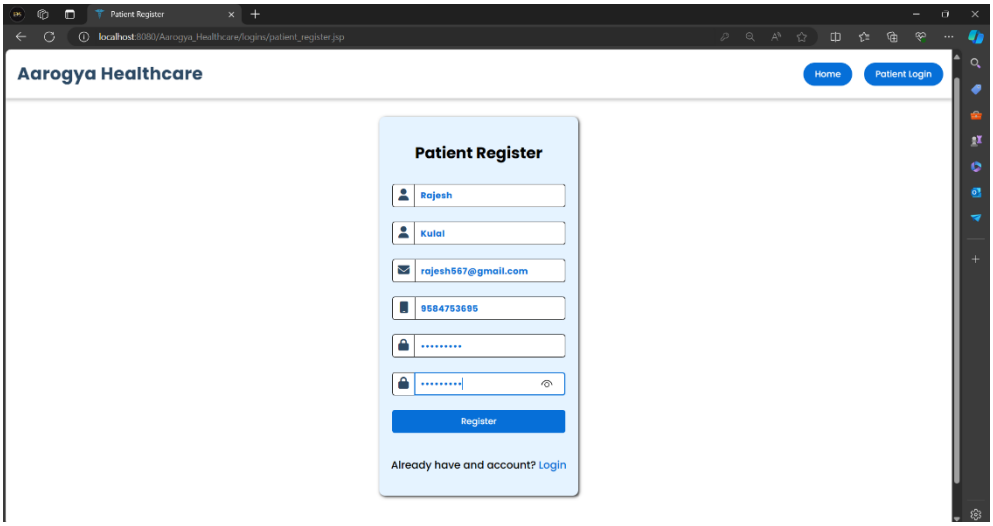


Figure 3.1 Patient Register

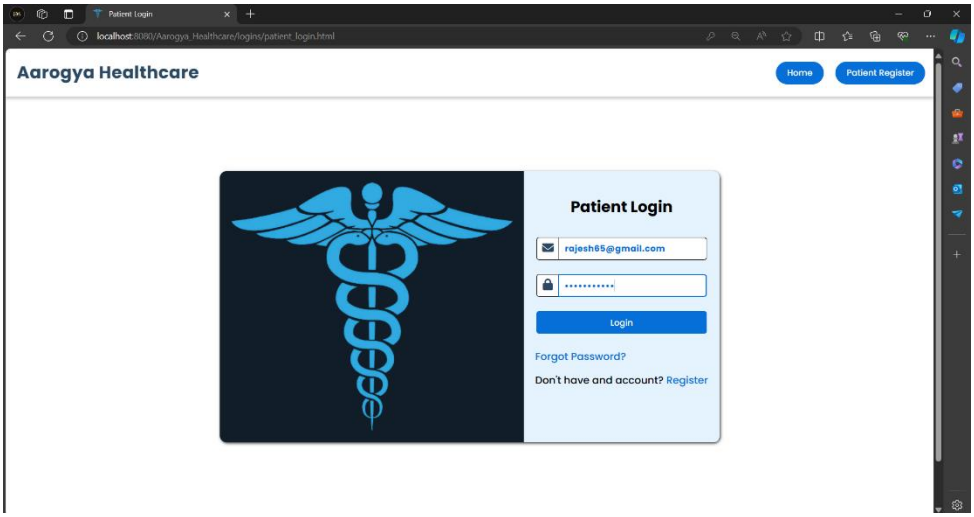


Figure 3.2 Patient Login

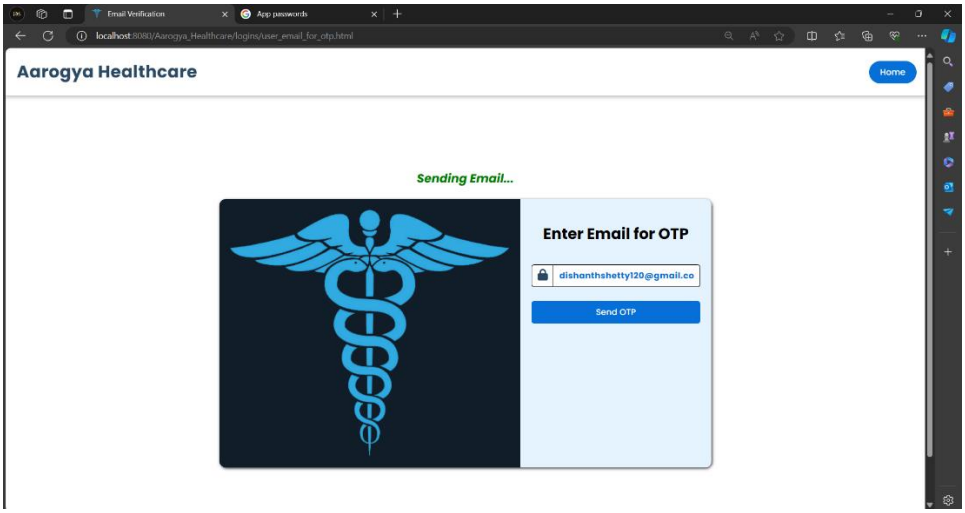


Figure 3.3 Forgot Password

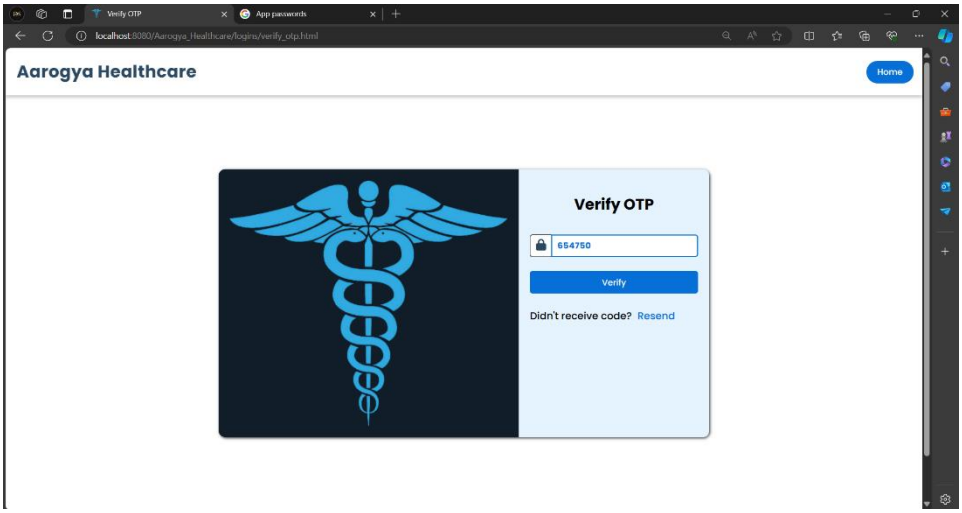


Figure 3.4 Verify OTP

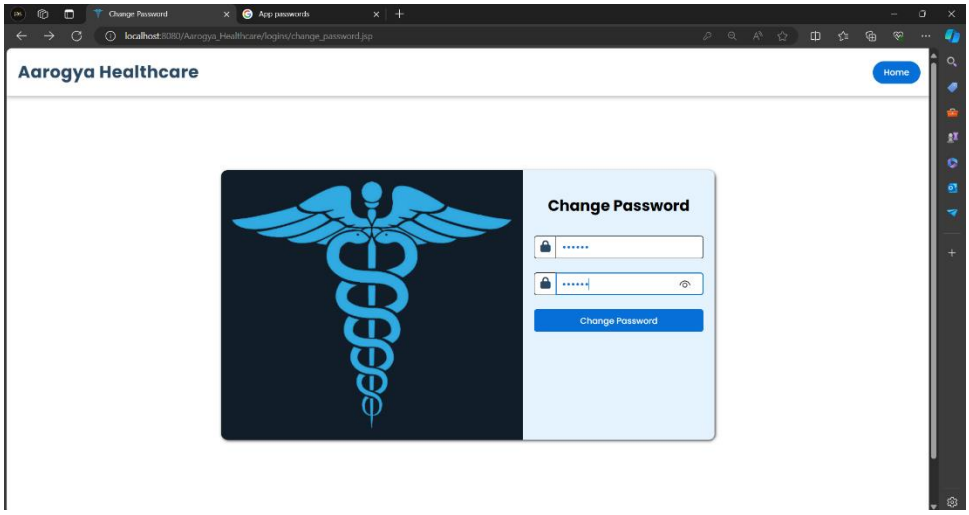


Figure 3.5 Change Password

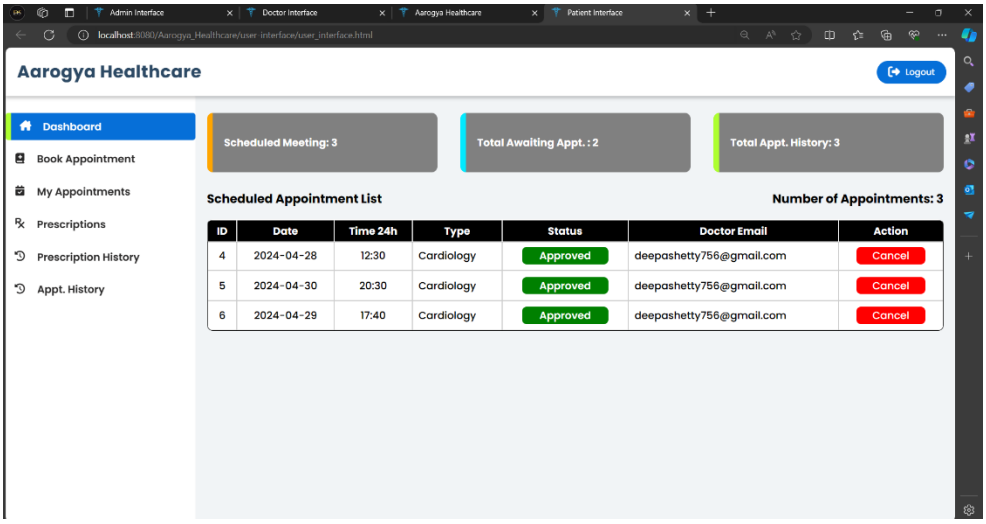


Figure 3.6 Patient Dashboard

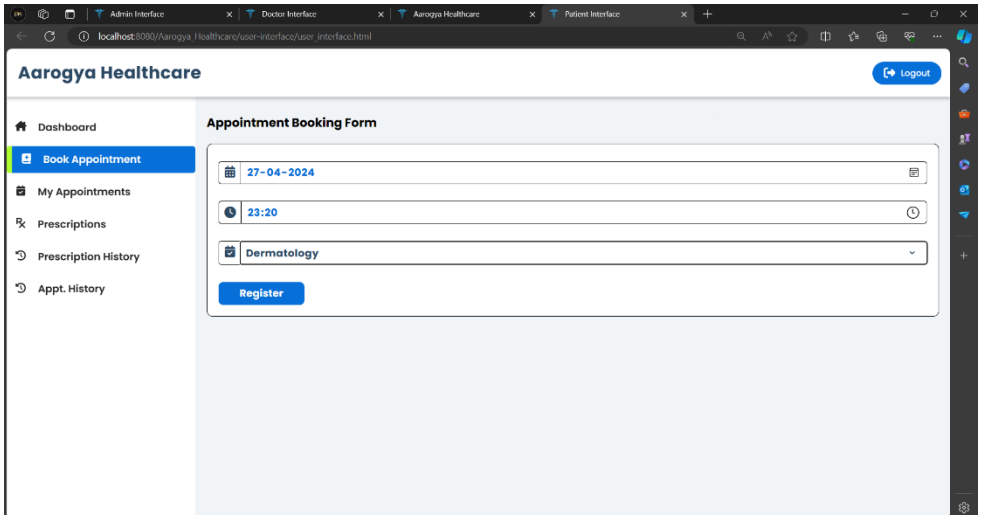


Figure 3.7 Book Appointment

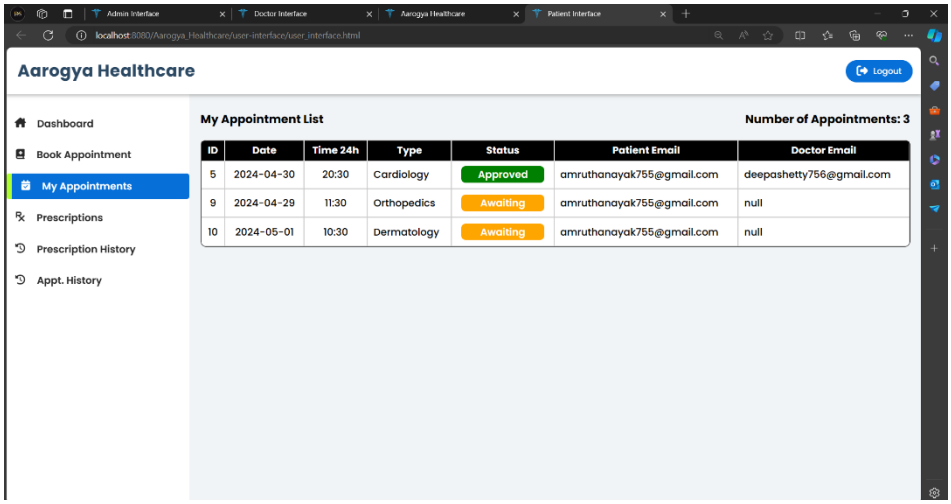


Figure 3.8 My Appointments

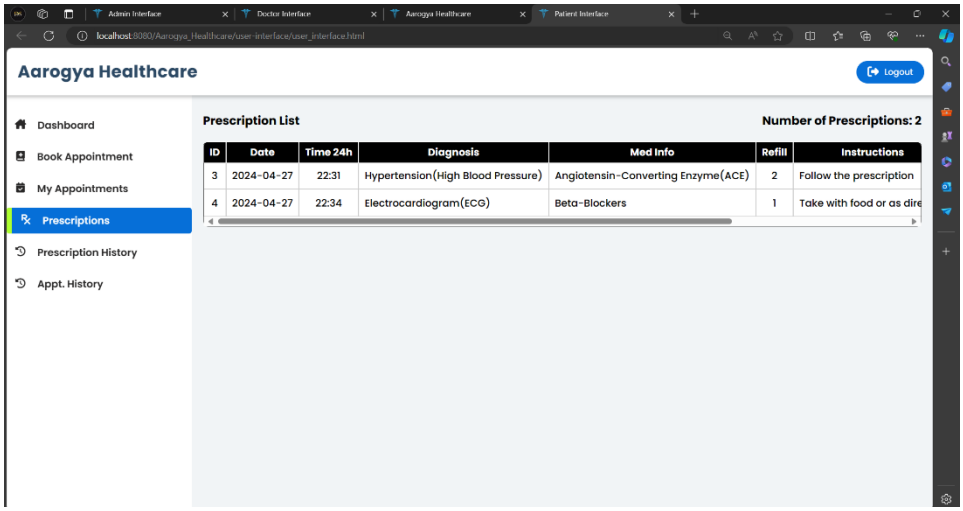


Figure 3.9 Prescriptions

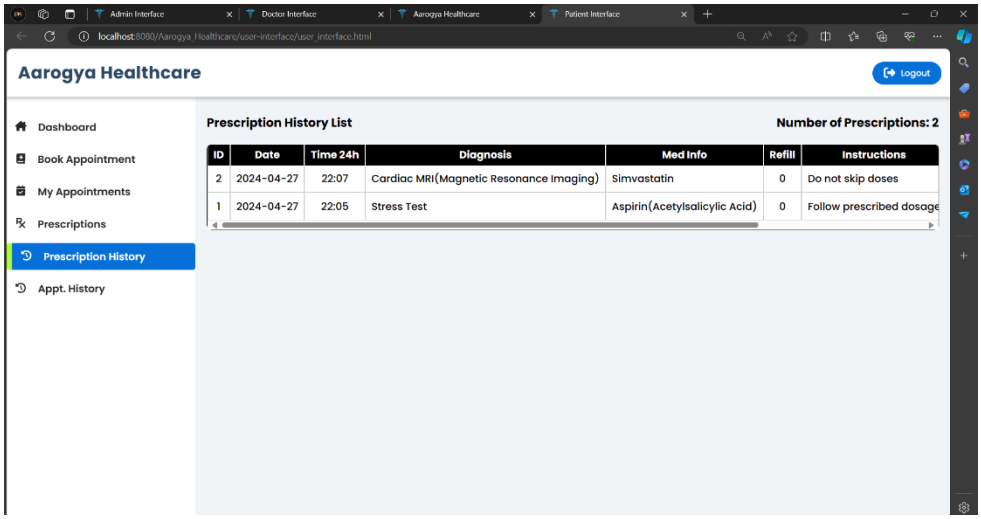


Figure 4.0 Prescription History

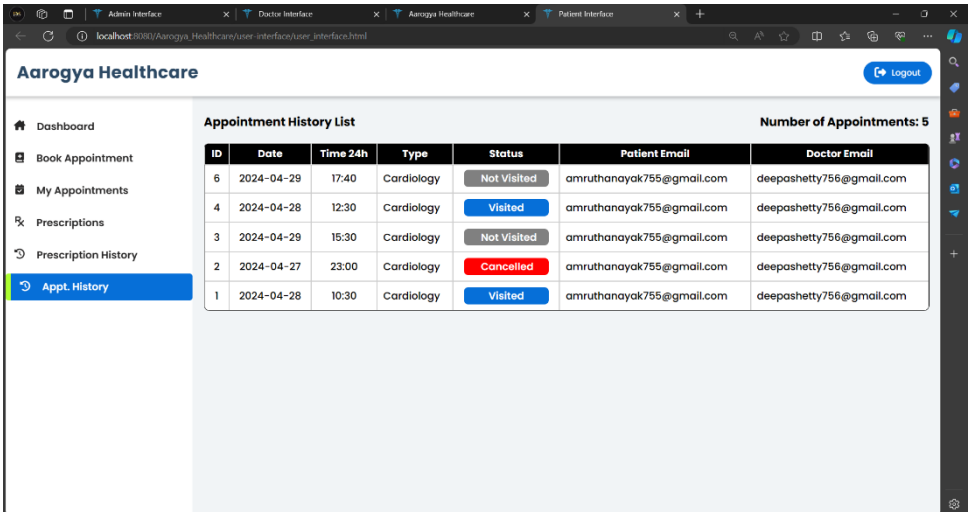


Figure 4.1 Appointment History

APPENDICES

- **https://graffersid.com/case-studies/hospital-management-system-dashboard/**
- **Patient Management System Website Templates | ThemeForest**
- **https://www.hhs.gov/hipaa/index.html**
- **https://www.ama-assn.org/practice-management/medicare medicaid/meaningful-use-electronic-health-record-ehr-incentive**
- **https://www.cdc.gov/nchs/icd/icd10cm_browsertool.htm**
- **https://www.ama-assn.org/practice-management/medicare-medicaid/meaningful-use-electronic-health-record-ehr-incentive**
- **https://www.cdc.gov/nchs/icd/icd10cm_browsertool.htm**
- **https://www.ama-assn.org/practice-management/cpt**
- **Medical Record Management System In Java Using JSP And Servlet With Source Code – Codebun**
- **patient-management-system · GitHub Topics · GitHub**