

UNIVERSITY OF RWANDA

COLLEGE OF SCIENCE AND TECHNOLOGY

SCHOOL OF ICT

**COMPUTER AND SOFTWARE ENGINEERING DEPARTMENT [NYARUGENGE
CAMPUS/GAKO]**

MODULE CODE/TITLE: INTERACTIVE WEB DEVELOPMENT

AccessHealth: Digital Healthcare Management Platform

1. 224008709
2. 224006132
3. 224004620

Year of study: 2

Date: 14/12/2025

Table of Contents

- **Chapter 1. Introduction**
 - 1.1. Historical background of the case study
 - 1.2. Problem statement
 - 1.3. Proposed Solution
 - 1.4. Purpose
 - 1.5. Objectives
- **Chapter 2. System Analysis and Design**
 - 2.1. System analysis
 - 2.1.1. Functional requirements
 - 2.1.2. Users of the project
 - 2.1.3. Partners
 - 2.2. System design
 - 2.2.1. UI design
 - 2.2.2. Database design
- **Chapter 3. Implementation**
 - 3.1. Introduction
 - 3.2. Tools and Technology
 - 3.3. Screenshots
- **Chapter 4. Conclusion and Recommendation**
 - 4.1. State of implementation
 - 4.2. Recommendations
- **Appendix**

Chapter 1. Introduction

1.1. Historical Background of the Case Study

Healthcare service delivery has traditionally relied on face-to-face consultations, paper-based patient records, and fragmented communication between patients, healthcare providers, and health institutions. In many developing countries, including Rwanda, access to timely and specialized medical care has often been limited by geographical distance, shortage of healthcare professionals, long waiting times, and inefficient referral systems.

Patients commonly struggle to identify appropriate doctors, schedule appointments, and receive follow-up care, especially in rural and underserved areas. While health systems have begun adopting digital solutions such as electronic medical records and national health databases to improve efficiency, many of these systems operate in isolation, limiting their effectiveness in providing integrated, patient-centered care.

AccessHealth is a web-based healthcare management platform designed to connect patients, doctors, and healthcare institutions through a unified digital solution. The platform aims to support hospitals, clinics, and independent healthcare providers by offering doctor discovery services, appointment scheduling, wearable device integration, and secure patient record management.

1.2. Problem Statement

Access to quality healthcare services remains a major challenge, particularly in developing countries where healthcare resources are limited and unevenly distributed. Many patients experience difficulties in finding appropriate medical specialists, booking appointments, and receiving timely medical advice.

Long waiting times, lack of early symptom assessment, and poor coordination between healthcare providers often lead to delayed diagnosis and treatment. Additionally, patient health information is frequently stored in fragmented systems or paper-based records, making it difficult for doctors to access complete medical histories. The absence of real-time health monitoring further limits the ability of healthcare providers to detect critical conditions early.

1.3. Proposed Solution

AccessHealth is designed to address the identified challenges through a comprehensive digital healthcare platform. The solution includes:

- Patient-Centric Services: Allowing patients to easily discover qualified doctors and book appointments.
- Integrated Health Records: Providing secure electronic patient records accessible to authorized professionals.
- Real-Time Health Monitoring: Integrating with wearable health devices for tracking vital signs like heart rate and blood pressure.

- Notification System: Delivering real-time notifications for appointments, health alerts, and updates.

1.4. Purpose

The main purpose of this project is to design and develop an intelligent digital healthcare platform that improves access to medical services, supports early diagnosis, and enhances patient care through technology. The system serves as a decision-support and healthcare coordination tool to connect patients, doctors, and health institutions.

1.5. Objectives

The specific objectives of the AccessHealth system are:

- To enable patients to easily find and book appointments with doctors based on location, specialty, availability, cost, and insurance coverage.
- To implement an AI-powered triage system that assesses patient symptoms and provides recommendations.
- To integrate wearable health devices for real-time monitoring of patient vital signs.
- To maintain accurate and secure electronic patient medical records accessible to authorized healthcare professionals.
- To facilitate secure data sharing with national health systems and provide multilingual interfaces for users.

Chapter 2. System Analysis and Design

2.1. System Analysis

The system analysis phase focuses on identifying the specific functional requirements necessary to address the problems outlined above. This includes the authentication flows, appointment management, and data tracking capabilities required by the different user roles.

2.1.1. Functional Requirements

2.1.1.1. Login Functionality

The system provides secure login functionality for both doctors and patients. The login process includes multi-step verification: verifying if the email exists, authenticating the password, and checking the specific profile attribute (e.g., patient_profile or doctor_profile) to enforce role-based access.

2.1.1.2. User Registration Functionality

The system supports distinct registration workflows for doctors and patients.

- Doctor Registration: Collects professional details such as specialization, license number, and hospital affiliation.
- Patient Registration: Collects personal health information, contact details, and medical history. Both processes include form validation for email uniqueness and password strength.

2.1.1.3. Logout Functionality

Secure logout functionality is provided to terminate user sessions. This ensures the security of sensitive medical data, particularly on shared devices (implied by Django authentication standards).

2.1.1.4. View, Update Functionality

- Dashboards: Doctors can view their upcoming appointments, recent patients, and notifications on a personalized dashboard. Patients can view their health metrics, appointment history, and active devices.
- Profile Management: Users can update their personal information and manage password changes through the account management features.

2.1.1.5. Delete Functionality (Cancellation)

The system allows for the cancellation ("deletion") of scheduled appointments. Doctors or patients can cancel pending appointments, which updates the status in the database and triggers notifications to the relevant parties. Admins also retain the ability to delete any record (User, Appointment, or Log) via the admin interface.

2.1.1.6. Consultation Start Functionality ("Check-in")

Note: Adapted from "Check-in". Doctors can initiate consultations for patients who have "checked in" for their appointment. This involves accessing the Consultation model to record start times, chief complaints, and examination notes.

2.1.1.7. Consultation Completion Functionality ("Check-out")

Note: Adapted from "Check-out". Upon finishing a visit, the doctor completes the consultation record. This process includes saving the diagnosis, treatment plan, medications, and follow-up instructions, effectively "checking out" the patient and updating their medical history.

2.1.1.8. Check Doctor Availability Functionality

Note: Adapted from "Check room availability". Patients can check the availability of doctors before booking. The system checks the DoctorHospital availability days and ensures no conflicting appointments exist for the selected time slot before confirming a booking.

2.1.1.9. Booking Functionality

Patients can search for doctors by specialty or location and book appointments. The booking workflow validates that the requested date is in the future and that the doctor is available. Once validated, an appointment record is created with a "pending" status.

2.1.2. Users of the Project

- Admins: Have full control over all models, including hospitals, doctors, and system configurations.
- Doctors: Can manage their schedules, view patient records, conduct consultations, and access wearable device data for their patients.
- Patients: Can search for doctors, book appointments, view their own medical records, and monitor their health stats via wearable integrations.

2.1.3. Partners

The system is designed to be used by various partners in the healthcare ecosystem:

- Hospitals and Clinics: Partner facilities where doctors practice and where patients are referred.
- National Health Systems: Integration partners such as e-Ubzima and NHIC for insurance and data reporting.
- Business Partners: Potential adopters who may use the system as a foundation for further development or internal hospital management.

2.2. System Design

2.2.1. UI Design

The user interface is designed to be responsive and user-friendly, utilizing HTML5, CSS3, and Bootstrap 5.

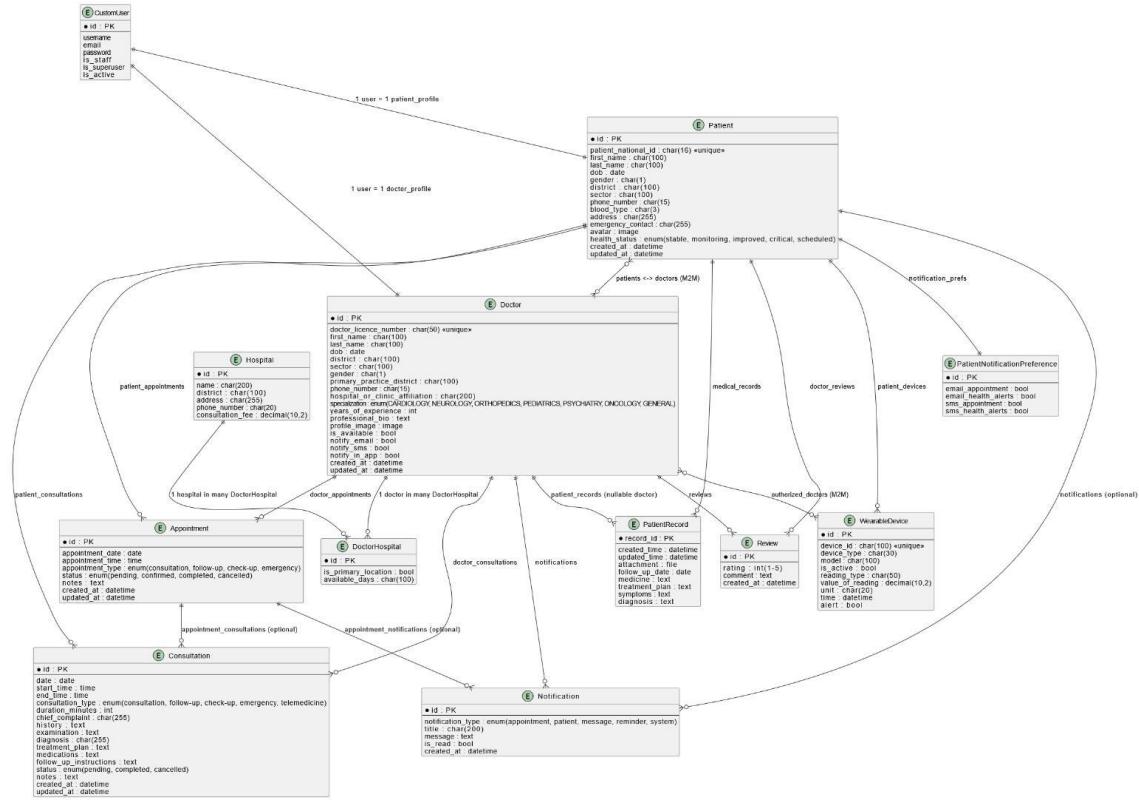
- Patient Dashboard: Focuses on clear visualizations of health metrics (Heart Rate, BP) and easy access to "Find Doctors".
- Doctor Dashboard: Focuses on practice management, displaying lists of appointments and recent patients for quick access.
- Landing Page: A public-facing page displaying key features and call-to-action buttons for registration.

2.2.2. Database Design

The system uses a relational database (SQLite for development, PostgreSQL for production) with the following core entities:

- CustomUser: Extends the base Django user model.
- Doctor & Patient: Profile models linked one-to-one with CustomUser.
- Appointment: Links Doctors and Patients with date, time, and status.
- WearableDevice: Stores health readings linked to a patient.
- Notification: Manages system alerts for users.
- Hospital: Stores facility details and links to doctors via a Many-to-Many relationship.

Figure 1: Database Entity Relationship Diagram - Shows the relationships between Users, Doctors, Patients, and Appointments.



Chapter 3. Implementation

3.1. Introduction

The implementation phase involved developing the core web system, including authentication, dashboards, appointment management, and medical record handling. Approximately 95% of the planned core features have been fully implemented and tested.

3.2. Tools and Technology

The following technology stack was used to develop AccessHealth:

- Backend Framework: Django 5.2.8.
- Database: SQLite (Development) / PostgreSQL (Production).
- Frontend: HTML5, CSS3, Bootstrap 5, JavaScript ES6+.
- Image Processing: Pillow 12.0.0.
- Development Tools: VS Code.

3.3. Screenshots and Visual Documentation

3.3.1. Landing Page

The landing page serves as the entry point to AccessHealth, displaying the system overview, key features, and call-to-action buttons for new users.

Figure 2: AccessHealth Landing Page - "Healthcare at Your Fingertips"

 AccessHealth

EN Log In Sign Up

View in Preview

Healthcare at Your Fingertips

Find affordable doctors, get AI-powered health guidance, and monitor your health with connected devices across Rwanda.

[Get Started](#) [Learn More](#)



 24/7 Support
Always here for you.

5,000+ Doctors Across Rwanda	50+ Hospitals Partner Facilities	All 30 Districts Coverage
--	--	-------------------------------------

Core Features

Comprehensive healthcare solutions designed specifically for Rwanda's healthcare system



Find Doctors

Search for doctors by location, specialty, cost, and availability. Our AI recommends the best match for your needs.



AI Health Assistant

Get instant health guidance through our intelligent chatbot that helps determine your care needs.



Wearable Integration

Connect your health devices to share real-time data with your healthcare providers.

3.3.2. Patient Authentication

The patient login interface provides secure email and password authentication with comprehensive validation and error handling.

Figure 3: Patient Login Page

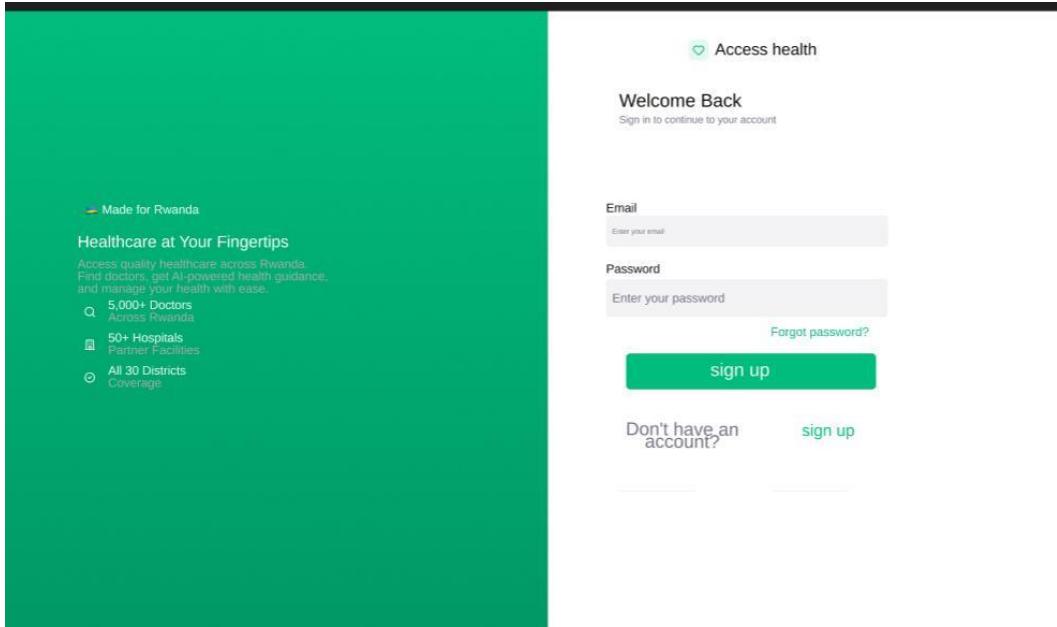
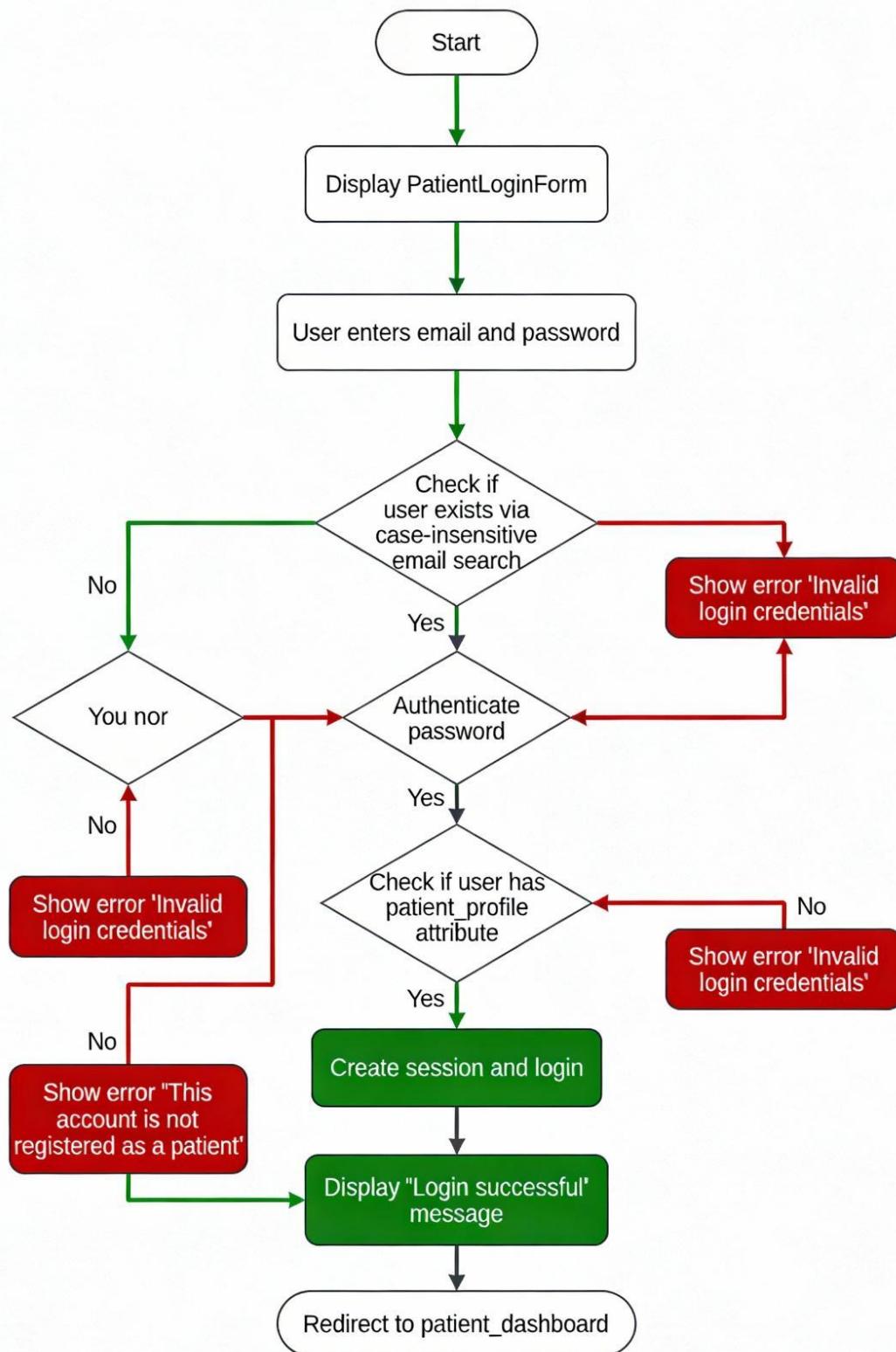


Figure 4: Patient Login Authentication Flowchart



3.3.3. Doctor Authentication

Similar to patient login, the doctor login page includes personalized welcome messages and role-based access control.

Figure 5: Doctor Login Page

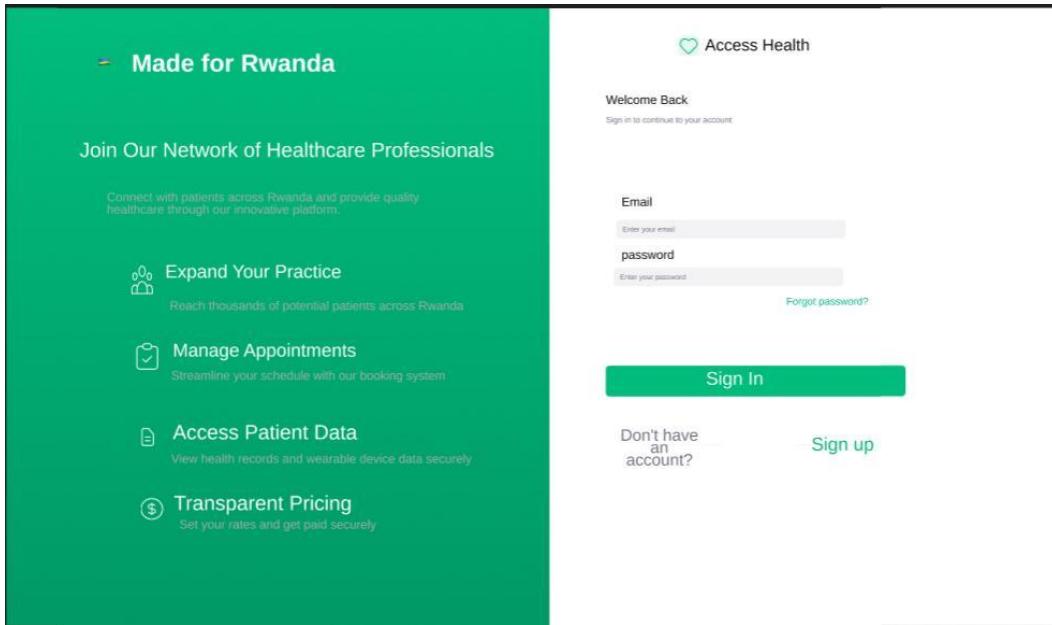
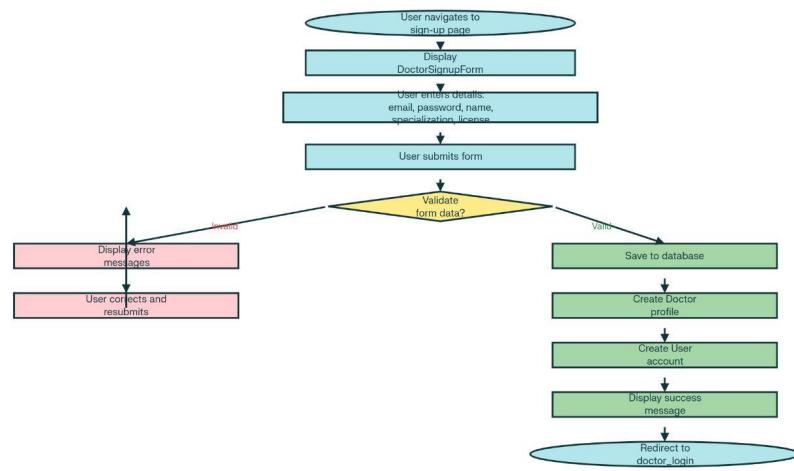


Figure 6: Doctor Registration Process Flowchart

Doctor Registration Process Flow

Complete workflow with validation and error handling paths



3.3.4. Patient Dashboard

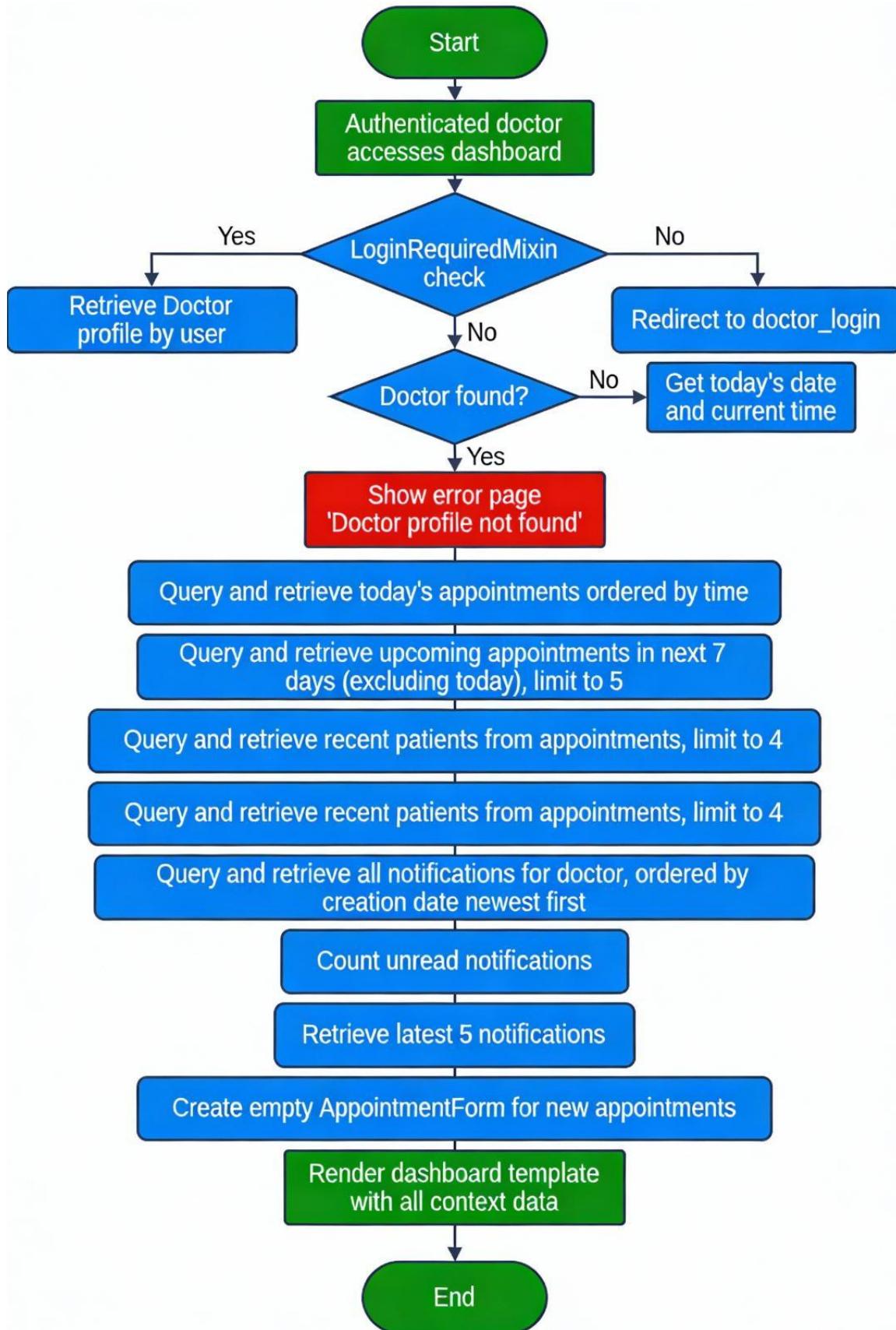
The patient dashboard provides a comprehensive health overview displaying vital signs from wearable devices, appointments, and system notifications.

Figure 7: Patient Dashboard View

The screenshot shows the AccessHealth Patient Dashboard. At the top, there's a navigation bar with 'AccessHealth' logo, 'Dashboard' (highlighted in green), 'Find Doctors', 'AI Triage', and language/username settings ('EN', 'Hello, test1'). Below the navigation is a header 'Dashboard' with the sub-instruction 'Monitor your health data and upcoming appointments'. The main content area is divided into several sections:

- Health Overview:** Contains four cards: 'Current Heart Rate' (No data), 'Blood Pressure' (No data), 'Connected Devices' (0/0, No active devices), and 'Next Appointment' (Dec. 24, 2025 5:30 p.m., Dr. test2 test2).
- Recent Vital Signs:** A table showing three rows: Heart Rate (No data), Blood Pressure (No data), and Temperature (No data). Each row has a 'More' button on the right.
- Health Alerts:** A section with a green header bar stating 'No health alerts at the moment.' followed by a small circular icon and the text 'No health alerts at the moment.'

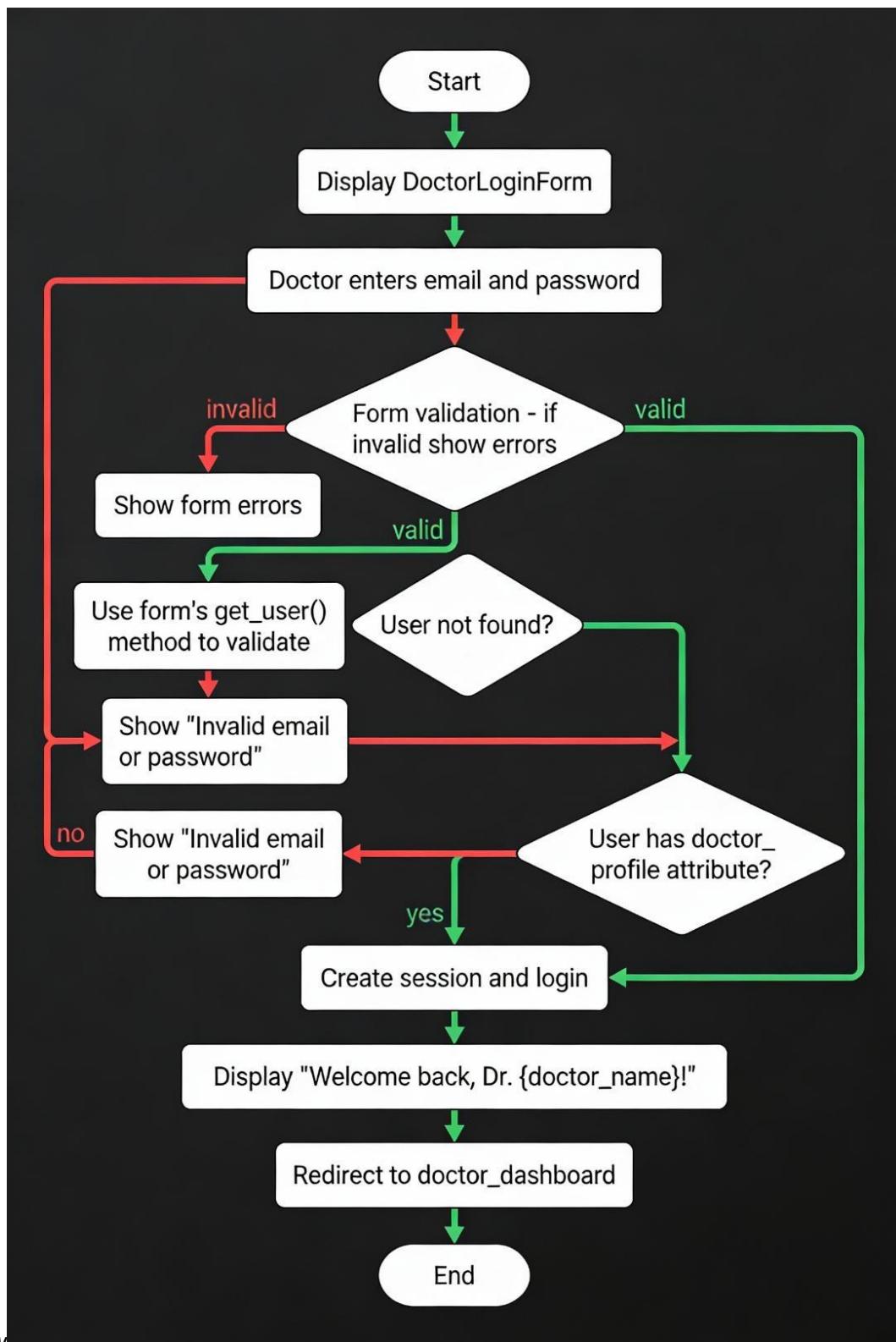
Figure 8: Patient Dashboard Data Flow Diagram



3.3.5. Doctor Dashboard

The doctor dashboard offers personalized practice management with appointments, recent patient records, and notification management.

Figure 9: Doctor Dashboard

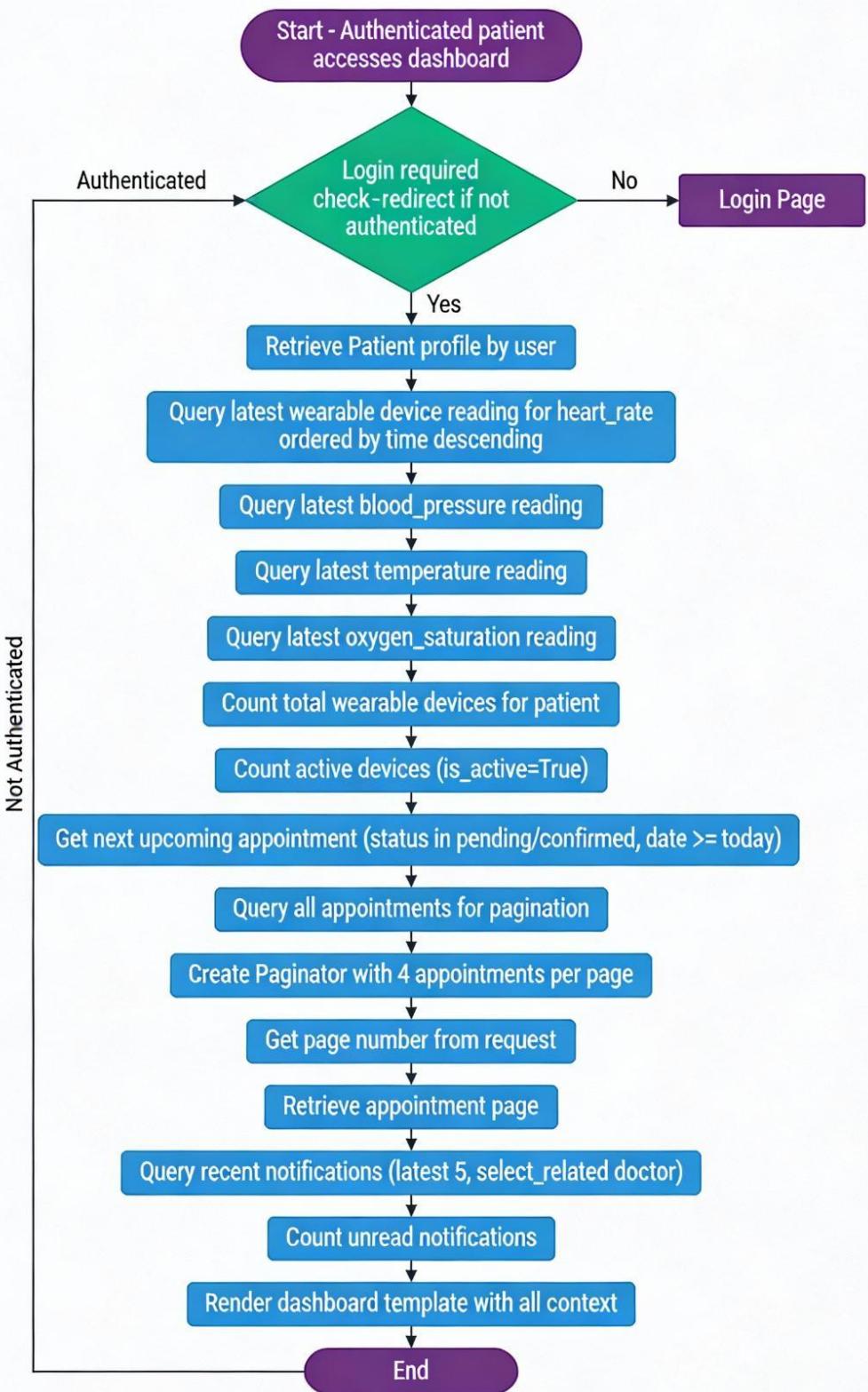


View

3.3.6. Doctor Discovery and Search

Comprehensive search and filtering interface allowing patients to find doctors by specialty, location, consultation fees, and ratings.

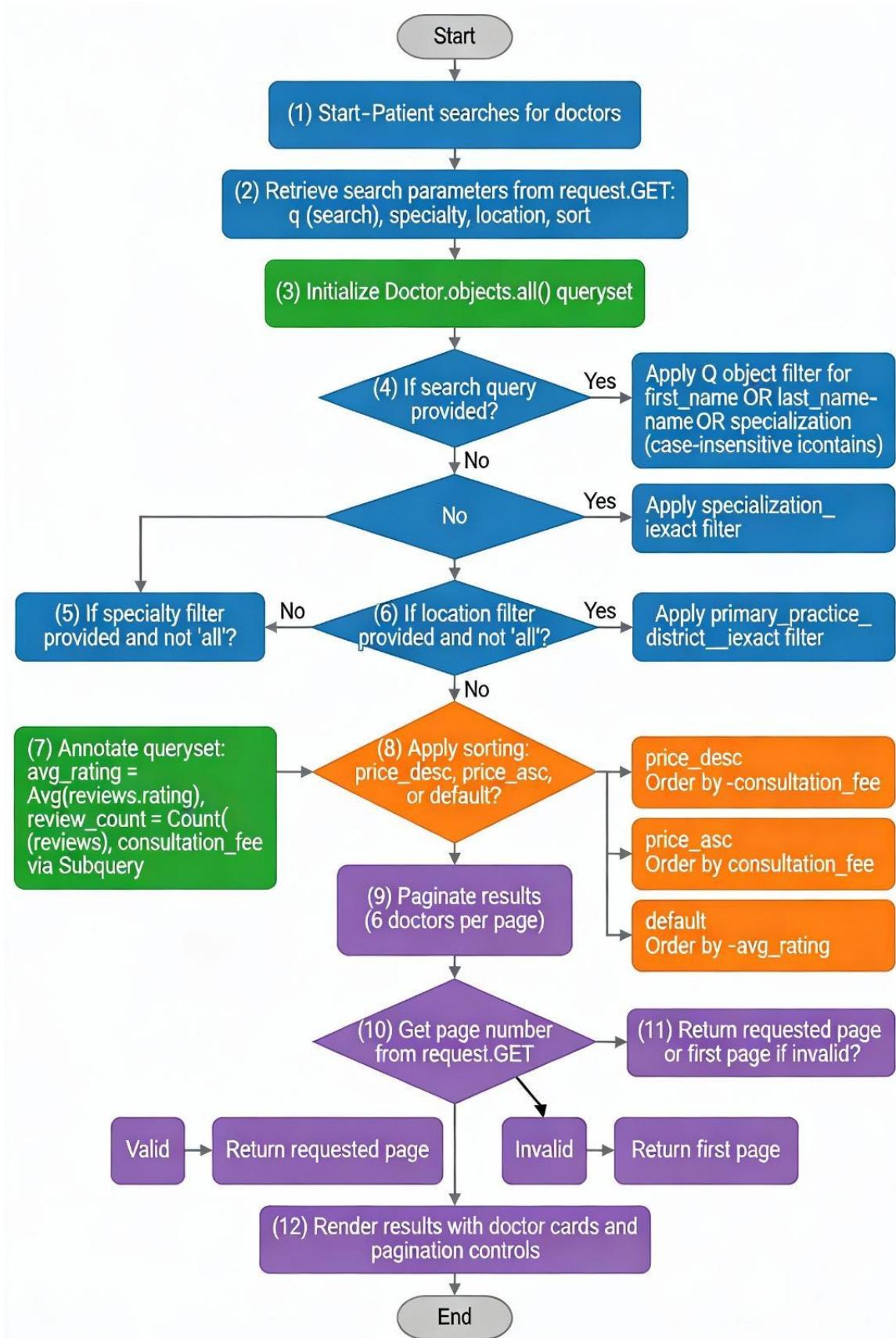
Figure 10: Doctor Discovery and Search Interface



3.3.7. Appointment Booking

Complete workflow showing appointment creation, validation, availability checking, and confirmation process.

Figure 11: Patient Appointment Booking Flow



Chapter 4. Conclusion and Recommendation

4.1. State of Implementation

AccessHealth has been successfully developed as a functional healthcare management platform. The completed features include user authentication, role-based dashboards, doctor discovery with advanced filtering, appointment booking, and a wearable device integration framework.

However, some advanced features like real-time IoT device streaming and the fully trained AI triage model are currently partially implemented prototypes due to time and resource constraints.

4.2. Recommendations

For Future Development:

- AI-Powered Features: Fully implement symptom analysis and triage recommendations to assist doctors.
- Telemedicine: Add video consultation capabilities to support remote patients.
- Payment Integration: Integrate secure payment processing for consultation fees.
- Mobile Applications: Develop native iOS and Android apps to improve accessibility for patients in rural areas.

For Implementation: It is recommended to start with a phased rollout, beginning with internal use for appointment management (Phase 1) before enabling AI features and wider public deployment (Phase 2 & 3).

Appendix

A.1 Test Credentials

User Type	Credentials
Patient	Email: tes3@gmail.com Password: 12345678
Doctor	Email: test2@gmail.com Password: 12345678
Admin	Email: admin Password: admin

A.2 Technology Stack

Component	Technology
Backend Framework	Django 5.2.8
Database	SQLite (Dev) / PostgreSQL (Prod)
Frontend	HTML5, CSS3, Bootstrap 5, JavaScript ES6+
Image Processing	Pillow 12.0.0

A.3 Repository and Documentation

GitHub Repository: <https://github.com/sfsf02/AccessHealth>