# Report

**Project Title:**

## Hospital Management System

## Healspire Medical Center

**Group: AH**

**Module Leader: Prof. Chaminda Wijesinghe**

| Student_ID | Name |
|---|---|
| M.B.M Afrij | 34175 |
| R.F MOHAMED FAWAS | 34250 |
| J. MUHAMMED AAFIR | 32238 |
| AA ANOAF | 34185 |
| A.M.Azad | 32771 |
| B.M.T.U.D WIJIEMANNA | 34195 |
| P.ABISHANA | 34342 |
| A H M SATHEEK | 32270 |
| M.I.M Ifas | 32800 |
| M.Z Ahmed | 32884 |

# Healspire Medical Center

## Project Report: Hospital Management System

### Introduction

The modern healthcare landscape demands efficient, secure, and scalable solutions to manage hospital operations. The Hospital Management System (HMS) developed in this project addresses critical challenges faced by small to mid-sized hospitals by leveraging core Java EE technologies (Java Servlets, JSP, and JDBC). This web-based system replaces outdated paper-based workflows and fragmented software, offering a centralized platform to streamline patient care, administrative tasks, and financial operations.

## Problem Statement

Traditional hospital management methods often rely on manual record-keeping, isolated software tools, or obsolete systems. These practices lead to inefficiencies such as:

**1. Delays and Errors:** Manual data entry increases the risk of human errors, misplaced records, and delayed access to patient information.

**2. Poor Coordination:** Disconnected systems for appointments, billing, and medical histories hinder real-time communication between staff and doctors.

**3. Security Risks:** Physical records or outdated digital systems lack robust security measures, compromising patient confidentiality.

**4. Operational Costs:** Time-consuming administrative tasks divert resources from patient care, reducing overall service quality.

To overcome these challenges, the HMS project aims to digitize and unify hospital workflows, ensuring accuracy, accessibility, and cost-effectiveness.

## System Overview

The HMS is designed as a modular, role-based web application with the following features:

**1.Centralized Patient Management:**

- Stores patient demographics, medical histories, and treatment records in a secure database.
- Enables instant retrieval of patient data via search filters, reducing time spent on manual record searches.

**2.Appointment Scheduling:**

- Allows admin to book, veiw,cancel appointments based on doctor availability.

- Sends automated reminders to patients to minimize no-shows.
- Patient book appionment, pay payment, downlord invoice

**3.Billing and Insurance Integration:**

- Generates itemized bills for consultations, procedures, and medications.
- Supports insurance claim processing by linking patient policies to transactions.

**4.Medical History Tracking:**

- Maintains a chronological log of diagnoses, prescriptions, and lab reports for informed decision-making.

**Technology Stack**

The system adheres to Java EE standards for scalability and portability:

- **Java Servlets:** Handle HTTP requests, manage business logic (e.g., appointment booking, billing), and interact with the database.

- **JavaServer Pages (JSP):** Render dynamic user interfaces for admin dashboards, patient forms, and reports.

- **JDBC:** Connects to a MySQL database for persistent storage of patient records, appointments, and transactions.

- **Apache Tomcat:** Serves as the web server to deploy the application.

The use of open-source tools (NetBeans, MySQL, Tomcat) ensures low implementation costs, aligning with the budget constraints of smaller healthcare facilities.

## 2. Requirements

2.1 Target Users

This hospital management system is designed specifically for three fundamental classes of users: patients, doctors, and administrative staff.

Patients are the primary end-users of the system, and they utilize the platform to register, safely log in, view doctor availability, schedule appointments, pay for consultations via the internet, and view or download their test reports and bills. The interface is made intuitive, mobile-friendly, and fast to facilitate eradication of customary paper-based appointment and reporting delays.

The platform is utilized by doctors to manage their day-to-day schedules, review assigned appointment lists, and access patient test reports. This gives them better time management and access to necessary information at the point of care. Doctors also have secure access and real-time updates on schedules and reports.

Administrators (admins) act as the system coordinators. They deal with user account creation, management of accounts (doctors, nurses, and patients), uploading patient test reports, appointment slot management, user log management, and overall data consistency and workflow. They also upload test results and handle invoice generation for patient consultation.

### 2.2 Core Features & Justification

Authentication System: Role-based secure login was employed to differentiate access control for admins, doctors, and patients. This prevents unauthorized access and grants data confidentiality throughout the system.

Appointment Management: Booking, viewing, and cancellation of appointments is a critical feature as timely access to medical consultancy is important in healthcare operations. Patients and admins can book; doctors can view and manage them.

Test Report Handling: Admins publish patient test reports (images or files), and these are made available to doctors and patients. This allows for faster diagnostics and transparency.

Invoice Generation: Patients can download and see consultation invoices as PDF documents. This allows for better personal financial tracking and record-keeping.

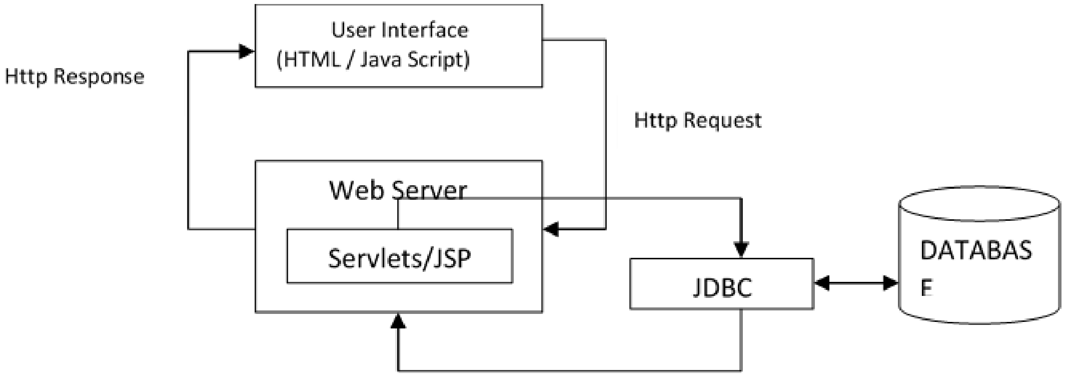
Schedule Management: Doctors and admins can manage availability, minimizing booking clashes and optimizing consultation time.

User Administration: Admins can add, modify, and remove users (patients, doctors, and nurses) that update the user directory.

## 3. Overview of the Design

**Architecture:**

- The project uses **MVC (Model-View-Controller)** architecture
- Java Servlets and JSP are employed for views and backend logic, with **JDBC** handling database connections.



**Key Components:**

- **Servlets:** Controllers that accept HTTP requests and handle communication between view and model.
- **JSPs:** Are dynamic views employed to render data.
- **DAOs:** Independent database logic for modularity and reusability.
- **Models:** Plain Java objects to model things like Patient, Doctor, Appointment, etc.

**Database Schema:**

- Tables are users, doctors, patient, appointments, schedules, test_reports, and invoices.

**Component Interaction:**

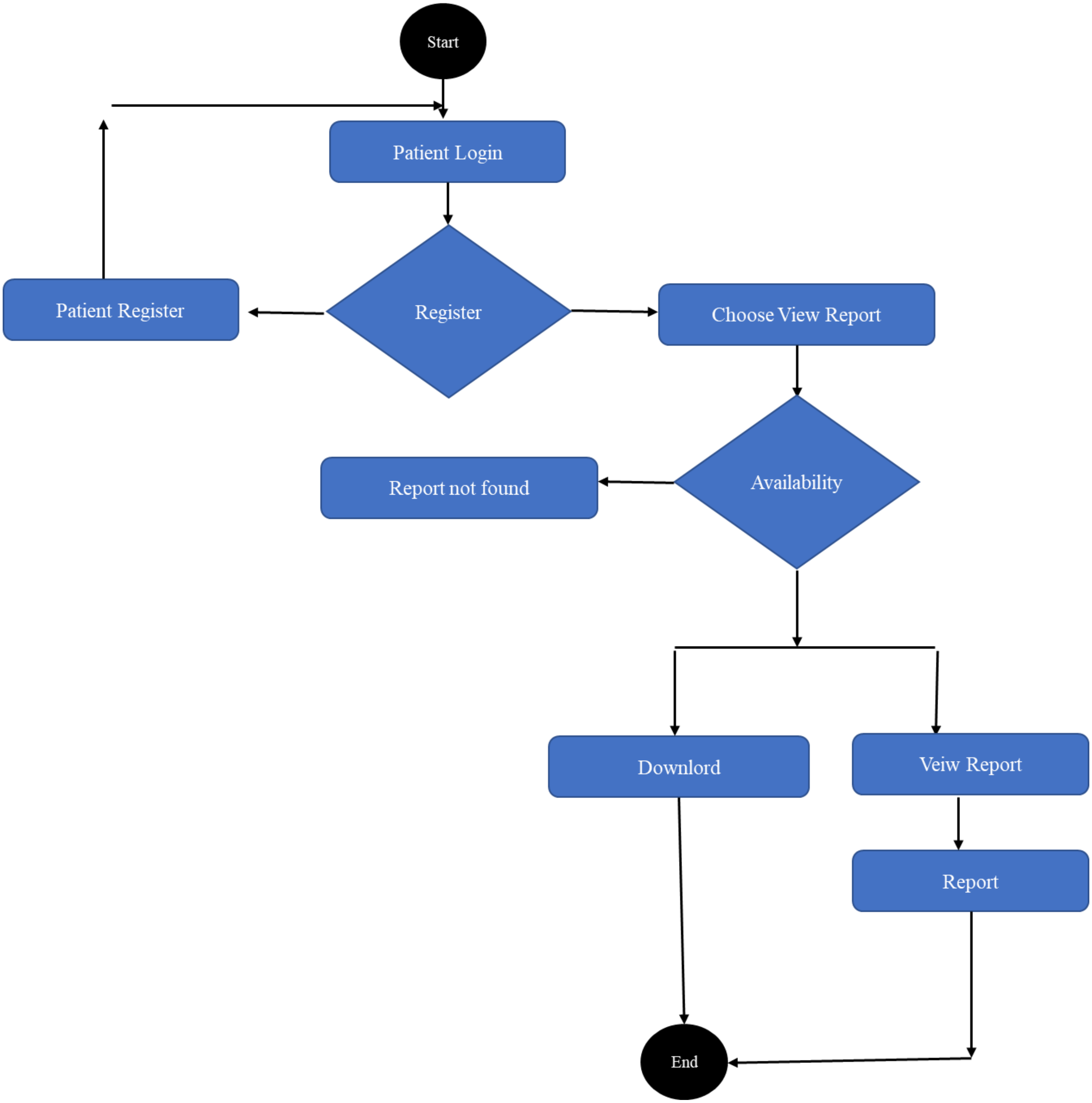
- User submits request from JSP → Servlet processes the request → DAO queries DB → Response is returned back to JSP.

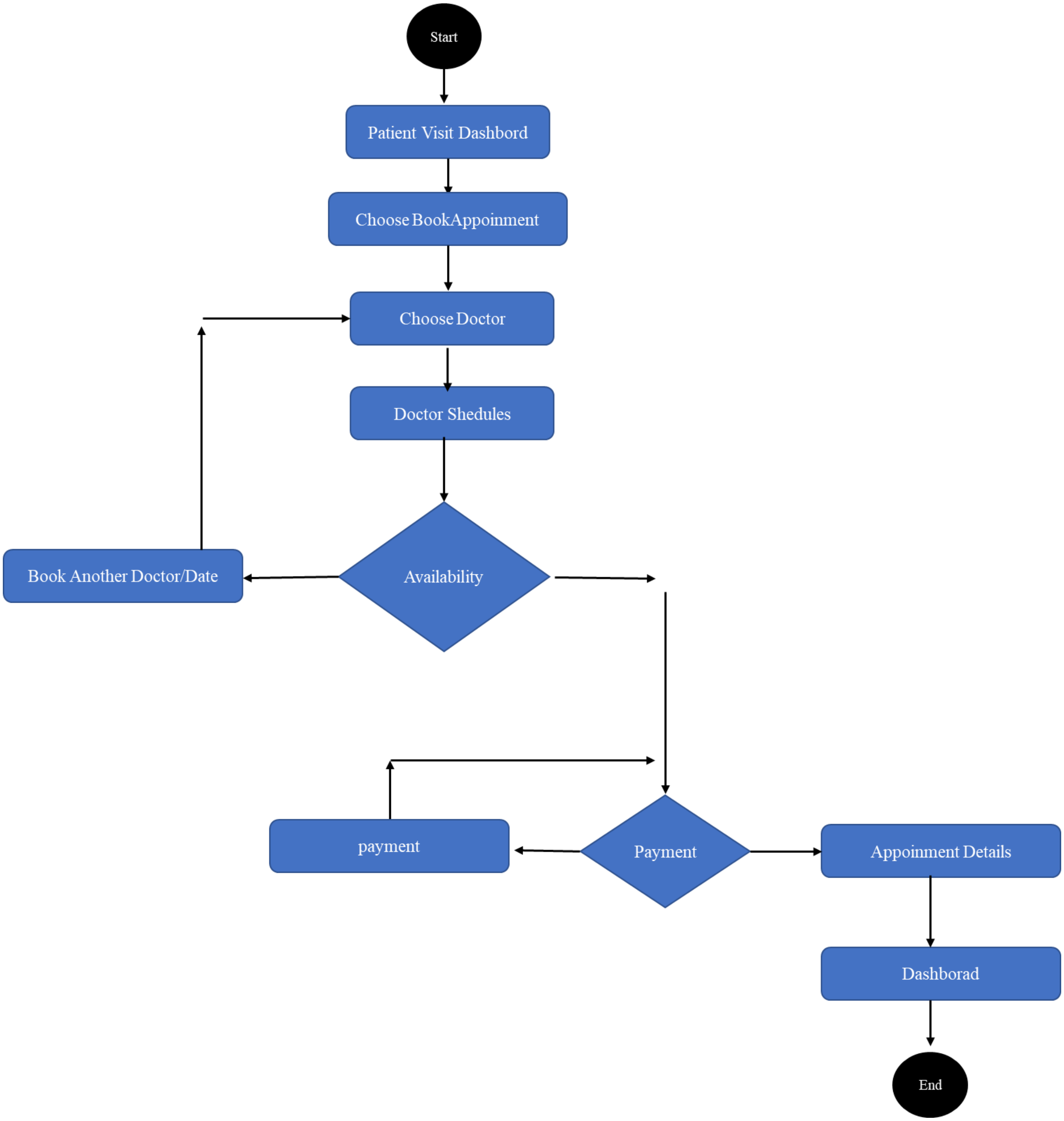
**Design Justification:**

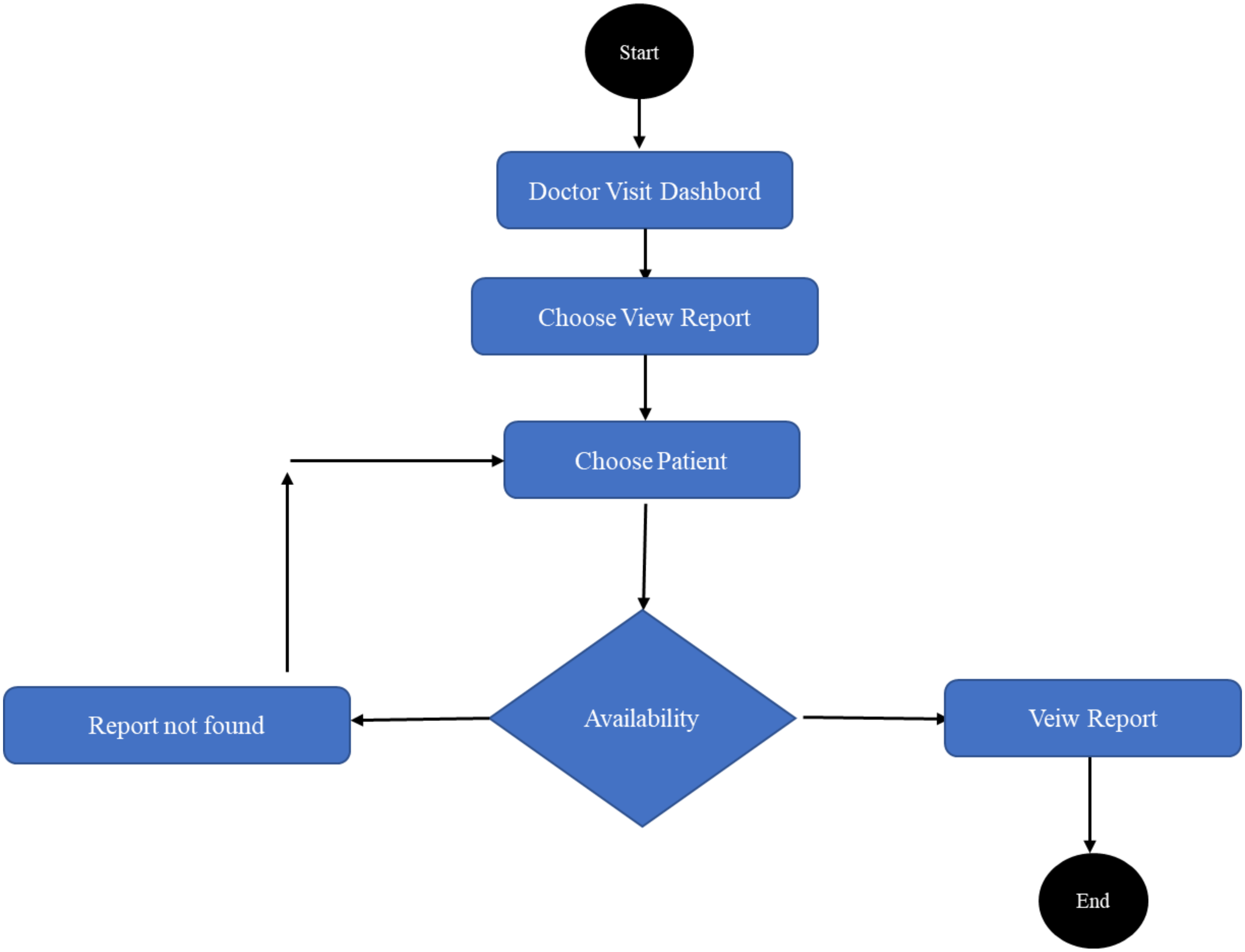
- MVC structure enables maintainability and separation of concerns.
- It is simple to migrate to ORM or NoSQL using DAOs.
- Standard relational schema provides scalability and compatibility.

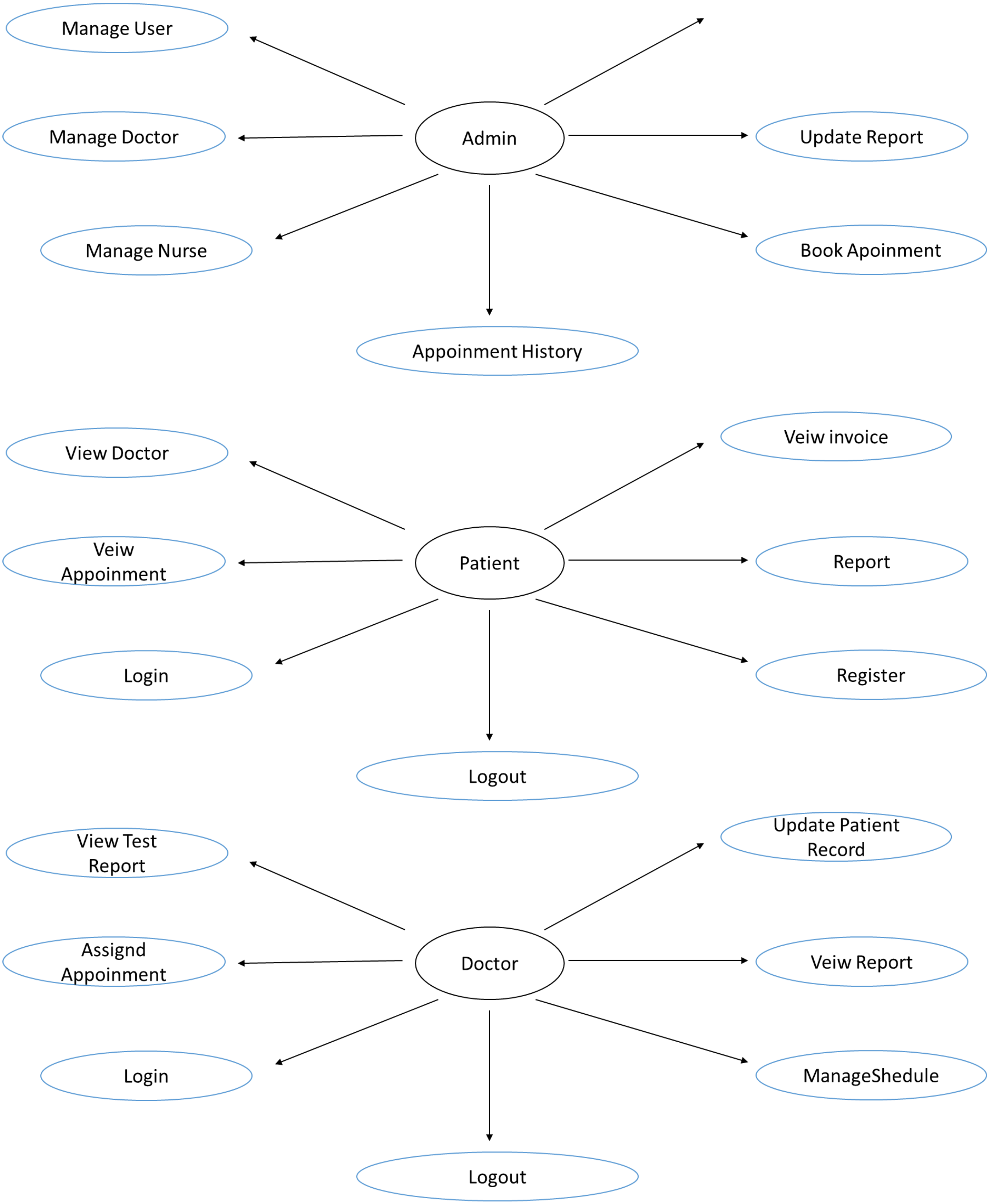
**UML Diagrams (refer Appendices):**

- Activity Diagram: Appointment Booking & Report Viewing.


Start
Patient Login
Patient Register
Register
Choose View Report
Report not found
Availability
Downlord
Veiw Report
Report
End


Start
Patient Visit Dashbord
Choose BookAppoinment
Choose Doctor
Doctor Shedules
Book Another Doctor/Date
Availability
payment
Payment
Appoinment Details
Dashborad
End


Start
Doctor Visit Dashbord
Choose View Report
Choose Patient
Availability
Report not found
Veiw Report
End


Manage User
Manage Doctor
Manage Nurse
Admin
Update Report
Book Apoinment
Appoinment History
View Doctor
Veiw invoice
Veiw
Appoinment
Patient
Report
Login
Register
Logout
View Test
Report
Update Patient
Record
Assignd
Appoinment
Doctor
Veiw Report
Login
ManageShedule
Logout

## Structure of Automated Tool for Medicare

- **CRISP-DM Framework**: A six-stage, cross-industry process for data mining, beginning with Business Understanding (defining high-level goals) and ending with Deployment.
- **Healthcare Adaptation**: Hospital-level objectives—like optimizing resource use, preventing over-occupancy, and improving operating-theater scheduling—are translated into data-mining goals (e.g. predicting bed occupancy, estimating patient stay length, forecasting surgery delays).
- **Generic Objectives & Data**: Because most hospitals collect the same core data (admission/discharge dates, reasons for admission, assigned services, etc.), a tool can ship with a library of pre-defined DM objectives and data requirements.
- **Semi-Automation Across Sites**: The main variation is each hospital's database schema. Once you map its HIS to a standard warehouse schema (Stage 2), the downstream steps (data preparation, modeling, evaluation, deployment) can largely be automated and reused from one hospital to the next.

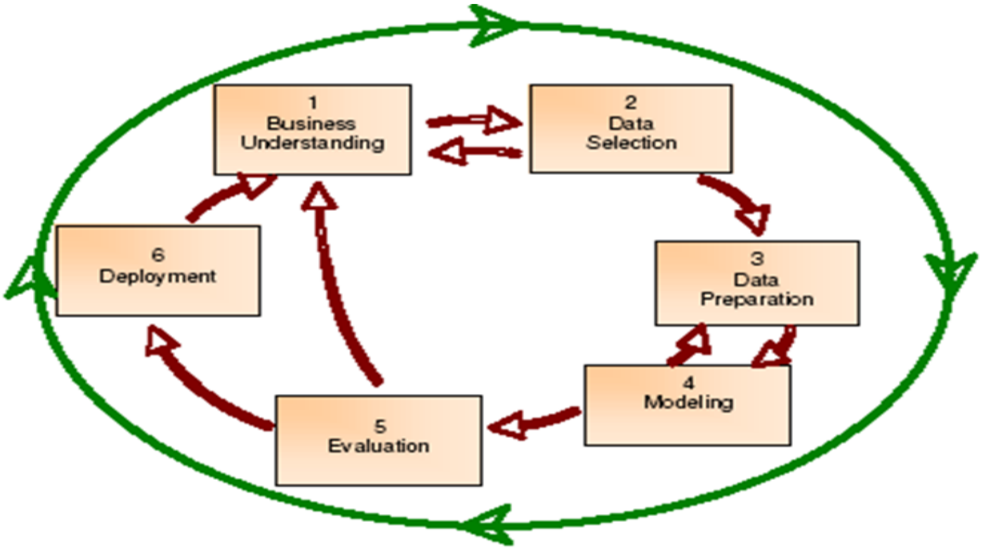


**Figure 1:** Life cycle of a data mining project.

## 4. Personal Reflection

What Worked Well

Modular MVC Structure

- Employing a solid Model–View–Controller architecture allowed us to neatly compartmentalize issues. Our servlets did request routing and business, DAOs encapsulated database interaction, and JSPs/CSS handled presentation. This modularity allowed us to easily isolate and debug, unit test our data-access layer, and swap out discrete pieces without blanket refactoring.

Consistent JDBC Usage

- Leaning on plain JDBC for database operations gave us explicit control over SQL queries, transactions, and connection pooling. While less concise than higher-level ORMs, JDBC was solid: our SELECT/INSERT/UPDATE routines were predictable, and we never received any "magic" behavior or lazy-loading surprises.

Strong PDF Generation using PDFBox

- Invoicing and receipt generation on the fly with Apache PDFBox gave us a final quality artifact with our hospital logo, formatted tables, and automatic page breaks. Installing our InvoicePDFGenerator utility, further fields (e.g., itemized details, patient details) could be included through minor template changes, saving time on follow-up enhancements.

Responsive Frontend using JSP + CSS

- Although we employed server-side JSPs rather than single-page app frameworks, careful CSS design (media queries, adaptive layouts, animations) yielded a modern and branded-looking interface. We could change colors, typography, and backgrounds globally quickly, so login form, dashboards, and tables were uniformly styled.

Challenges Faced

- File Paths in Production
- Hard-coding image file (logo, background) and uploaded document absolute paths resulted in redundant deployment issues. We eliminated this by relocating file-storage configuration to a central properties file and using ServletContext.getRealPath() to create portable, context-relative URLs.

Session and Authentication Complexity

- Having three discrete roles (Admin, Doctor, Patient) required a robust session design with role checking in every servlet or JSP. Initially, there were unintended permission leaks that allowed users to access unauthorized pages. There was comprehensive testing and a centralized "auth filter" later enforced consistent access control.

Branching and Collaboration with Git

- Our project utilized feature branches and pull requests, but merging was difficult when long-lived branches branched far away from main. We minimized conflicts by encouraging shorter-lived branches, employing descriptive commit messages, and performing regular "rebase" updates to keep linear histories in branches.

Lessons Learned

- Design Before Code: Designing UML diagrams (use-cases, class, sequence) and wireframes prior to writing a single line of Java or SQL avoided numerous cycles of speculative programming.

- Database Schema First: Carefully thought-out relational design with properly defined primary/foreign keys and indexing provisions upfront prevented later performance problems—most notably in appointment-lookup queries.
- Consistent Versioning: Atomic, frequent commits with useful messages allowed us to safely roll back breaking changes without having to cut back on healthy features.
- User Experience Matters Early: Spending time on responsive CSS and form validation up front paid off in user experience and reduced post-launch adjustments.

## Future Work

Several significant upgrades are suggested as part of the Hospital Management System's ongoing development to improve user experience, performance, and scalability. With these improvements, the system will be ready for practical use in extensive healthcare settings.

### 1. Switching to RESTful Services and Spring Boot

Although the current system, which was constructed with Servlets and JSP, is appropriate for educational settings, it lacks the resilience required for scalable implementation. Making the switch to the Spring Boot framework will facilitate greater security, better dependency management, and quicker development cycles. More interoperability and flexibility will be promoted by integrating RESTful APIs, which will enable the application to communicate with external systems including insurance services, pharmacy systems, and mobile apps.

### 2. Putting Role-Based Dashboards and Notifications into Practice.

Personalised dashboards for various user roles, including administrators, physicians, nurses, and patients, should be implemented in order to increase user engagement. These dashboards would show pertinent data and role-specific actions. Communication between the hospital and its stakeholders would also be greatly improved by incorporating a notification system that sends out emails or SMS alerts for impending appointments, lab results, or system upgrades. Administrators could also benefit from the addition of analytical tools to track patient trends, physician productivity, and hospital performance.

### 3. Adopt a Microservices Framework

As the application expands, the current monolithic structure may become a constraint. Patient data, billing, and appointment scheduling are just a few of the modules that can function independently by restructuring the system using a microservices architecture. Better fault tolerance, simpler updates, and better resource management are made possible by this separation. Additionally, it allows each service to be scaled according to demand, which is perfect for busy healthcare settings.

### 4. Leveraging Advanced Technology

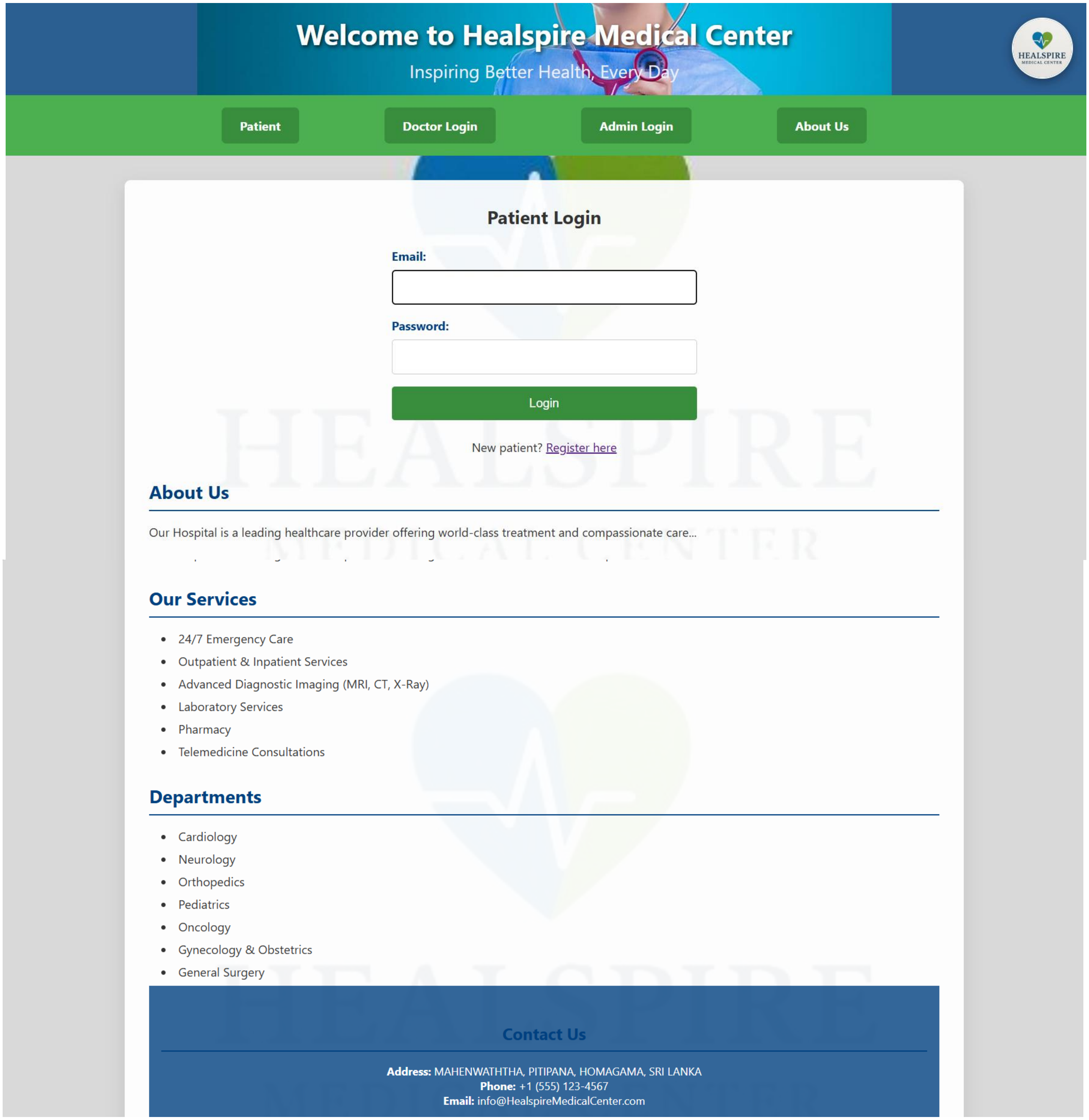
To improve the platform’s functionality, patient records and medical documents can be safely stored on cloud-based services such as Amazon S3 or Google Cloud Storage. Implementing biometric authentication methods, like fingerprint or facial recognition, can enhance security while providing faster access for healthcare providers and patients. Moreover, integrating an SMS gateway can enable real-time alerts and reminders, thereby enhancing the patient experience and minimizing missed appointments.

**conclusion**

The scalability, security, and user experience of the HMS will all be much improved by these suggested changes, establishing it as a solution that is ready for the future and able to satisfy the needs of contemporary healthcare facilities.

Git Hub Link :- https://github.com/Fawas-Mohamed/Healspire-Medical-Center.git

# HOME PAGE



Welcome to Healspire Medical Center

Inspiring Better Health, Every Day

Patient Login

Email:

Password:

New patient? Register here

About Us

Our Hospital is a leading healthcare provider offering world-class treatment and compassionate care...

Our Services

- 24/7 Emergency Care
- Outpatient & Inpatient Services
- Advanced Diagnostic Imaging (MRI, CT, X-Ray)
- Laboratory Services
- Pharmacy
- Telemedicine Consultations

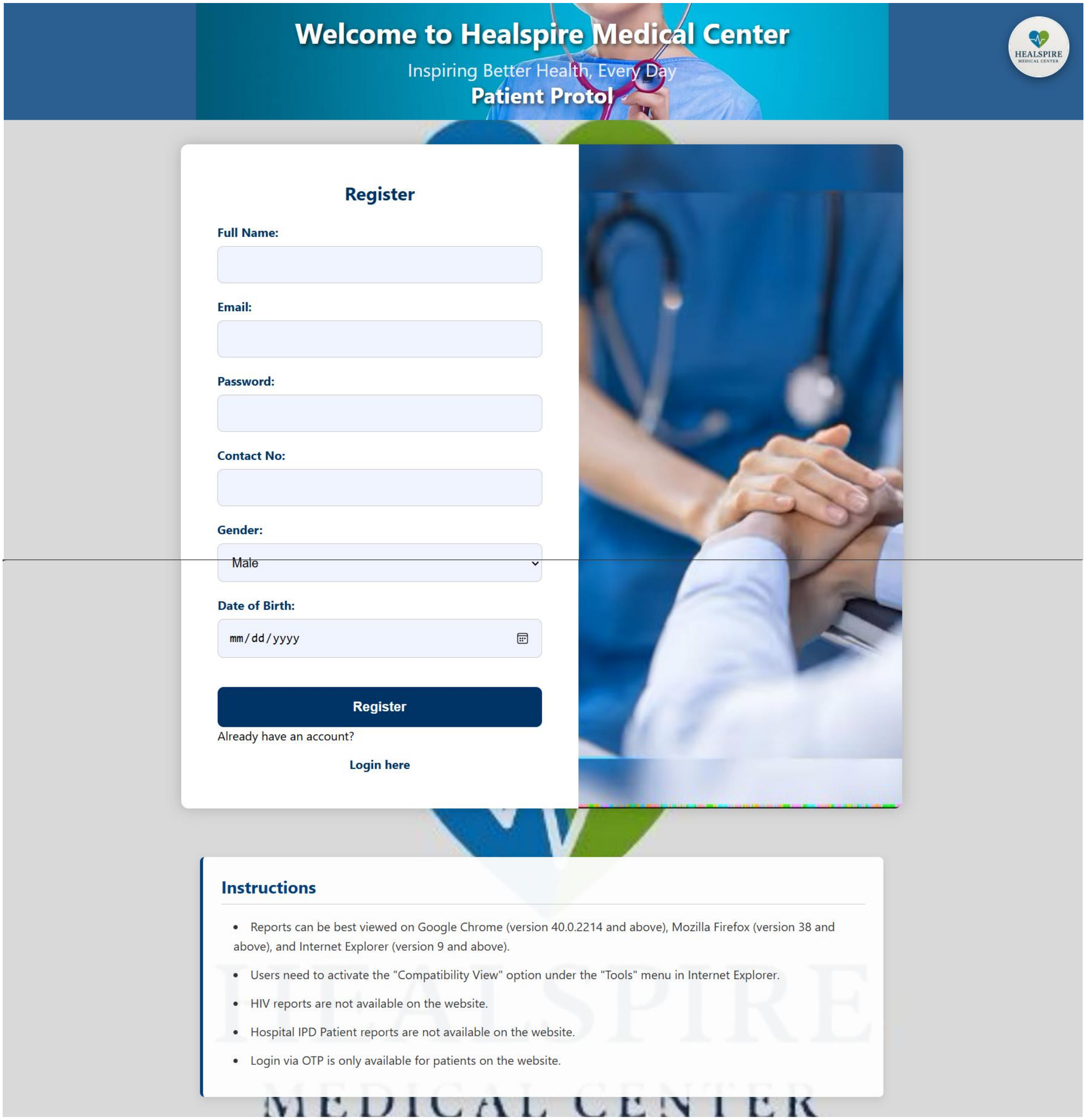
Departments

- Cardiology
- Neurology
- Orthopedics
- Pediatrics
- Oncology
- Gynecology & Obstetrics
- General Surgery

Contact Us

**Address:** MAHENWATHTHA, PITIPANA, HOMAGAMA, SRI LANKA
**Phone:** +1 (555) 123-4567
**Email:** info@HealspireMedicalCenter.com

# Patient Registration



Welcome to Healspire Medical Center
Inspiring Better Health, Every Day
Patient Protol
HEALSPIRE
MEDICAL CENTER

## Register

Full Name:

Email:

Password:

Contact No:

Gender: Male

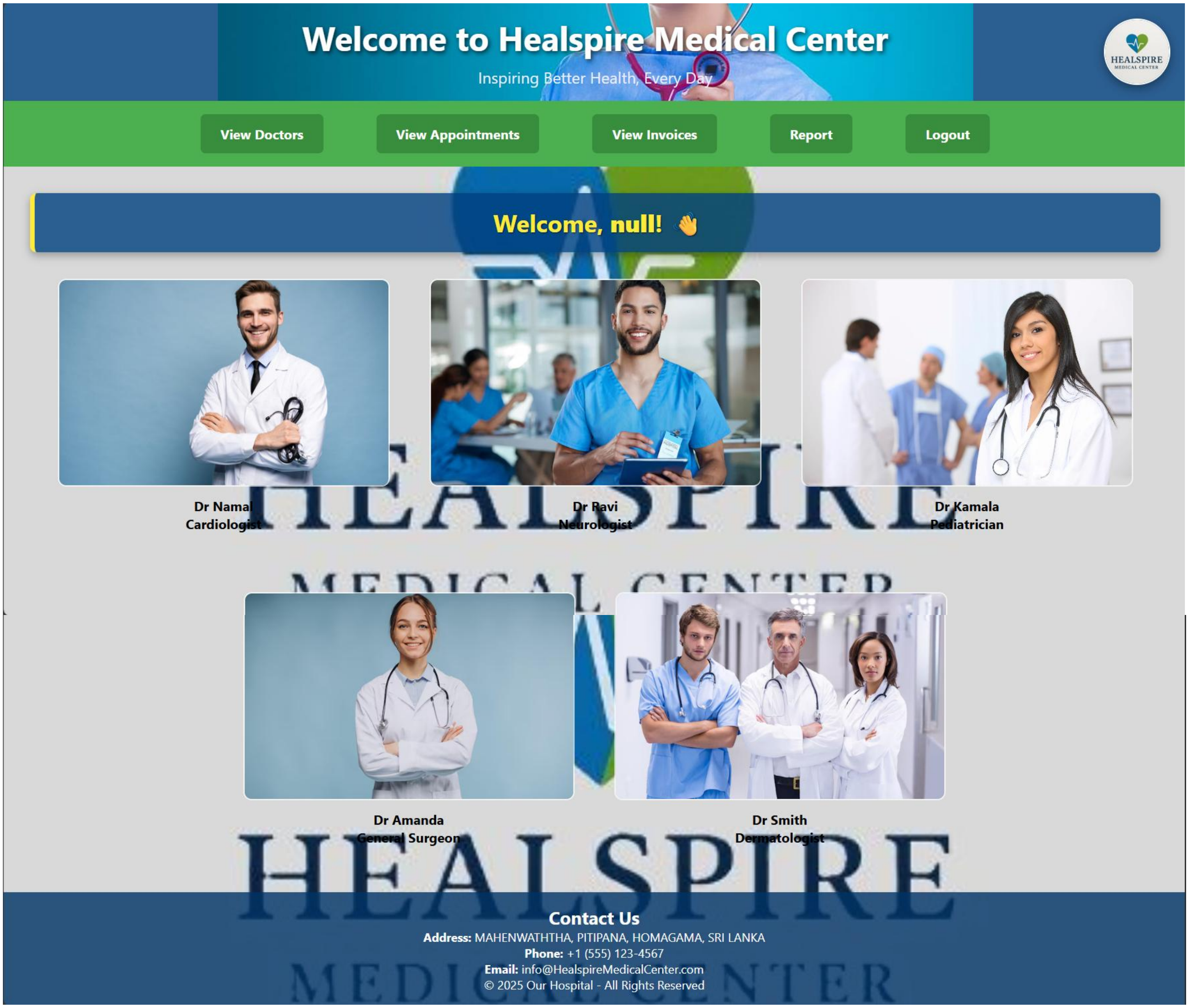
Date of Birth: mm/dd/yyyy

Already have an account?

## Instructions

- Reports can be best viewed on Google Chrome (version 40.0.2214 and above), Mozilla Firefox (version 38 and above), and Internet Explorer (version 9 and above).
- Users need to activate the "Compatibility View" option under the "Tools" menu in Internet Explorer.
- HIV reports are not available on the website.
- Hospital IPD Patient reports are not available on the website.
- Login via OTP is only available for patients on the website.

# Patient Dashborad



Welcome to Healspire Medical Center

Inspiring Better Health, Every Day

Welcome, null! 👋

Dr Namal
Cardiologist

Dr Ravi
Neurologist

Dr Kamala
Pediatrician

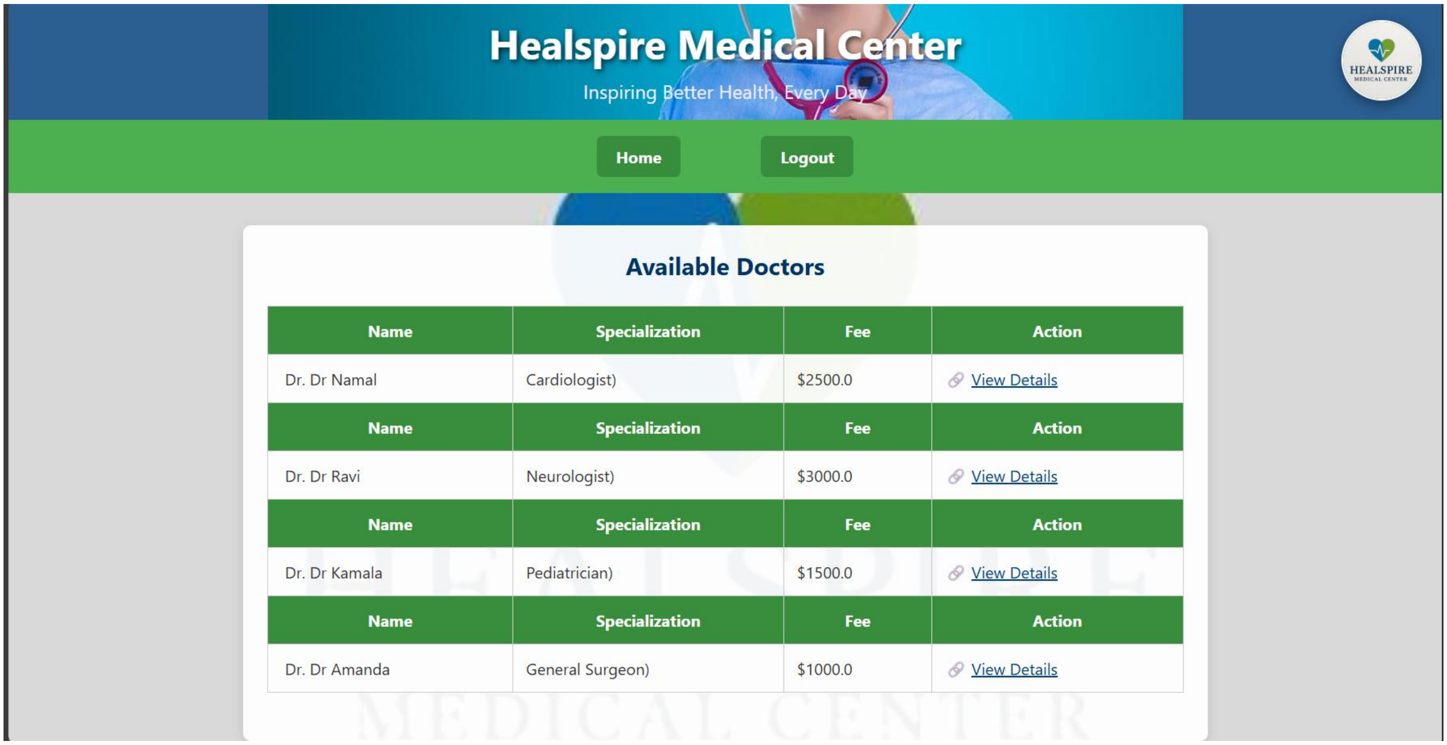
Dr Amanda
General Surgeon

Dr Smith
Dermatologist

**Contact Us**

**Address:** MAHENWATHTHA, PITIPANA, HOMAGAMA, SRI LANKA
**Phone:** +1 (555) 123-4567
**Email:** info@HealspireMedicalCenter.com
© 2025 Our Hospital - All Rights Reserved

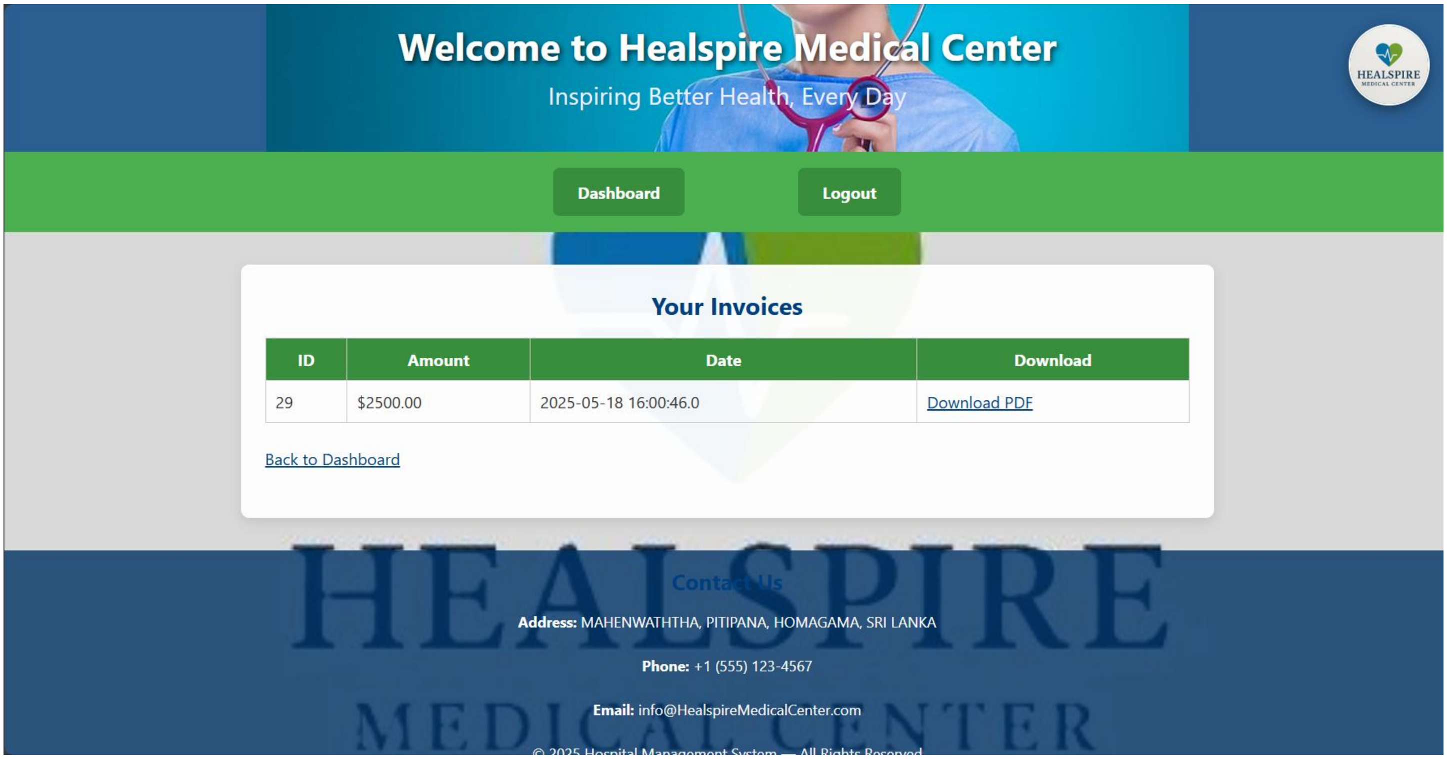
# ViewDoctors



Available Doctors

| Name | Specialization | Fee | Action |
|---|---|---|---|
| Dr. Dr Namal | Cardiologist) | $2500.0 | View Details |
| Name | Specialization | Fee | Action |
| Dr. Dr Ravi | Neurologist) | $3000.0 | View Details |
| Name | Specialization | Fee | Action |
| Dr. Dr Kamala | Pediatrician) | $1500.0 | View Details |
| Name | Specialization | Fee | Action |
| Dr. Dr Amanda | General Surgeon) | $1000.0 | View Details |

# Veiw / Downlord Invoice


Welcome to Healspire Medical Center
Inspiring Better Health, Every Day
HEALSPIRE
MEDICAL CENTER
Dashboard
Logout

Your Invoices

| ID | Amount | Date | Download |
|---|---|---|---|
| 29 | $2500.00 | 2025-05-18 16:00:46.0 | Download PDF |

Back to Dashboard

Contact Us

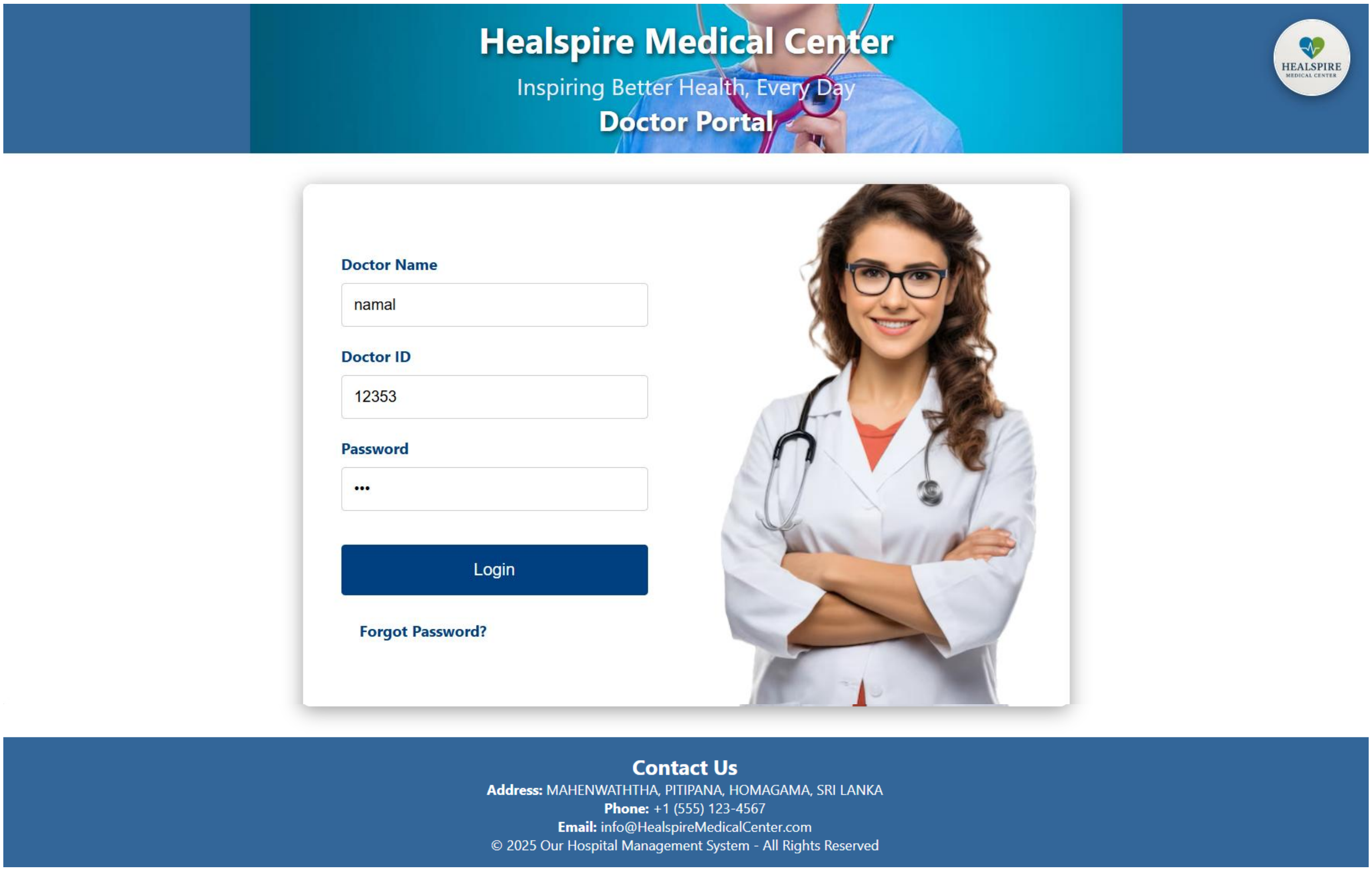
**Address:** MAHENWATHTHA, PITIPANA, HOMAGAMA, SRI LANKA

**Phone:** +1 (555) 123-4567

**Email:** info@HealspireMedicalCenter.com

© 2025 Hospital Management System — All Rights Reserved

## Doctor Login Page



Healspire Medical Center

Inspiring Better Health, Every Day

Doctor Portal

Doctor Name: namal

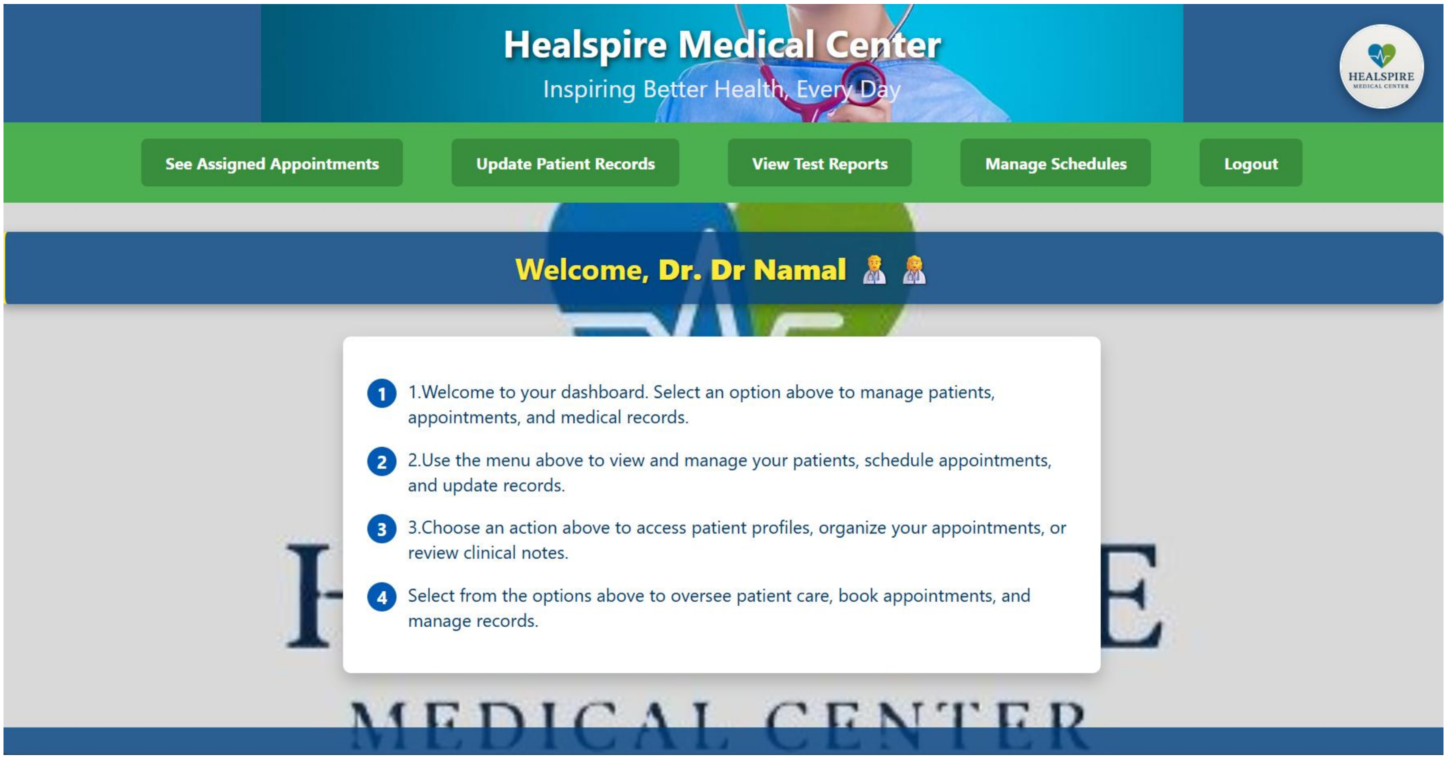
Doctor ID: 12353

Password: •••

Contact Us

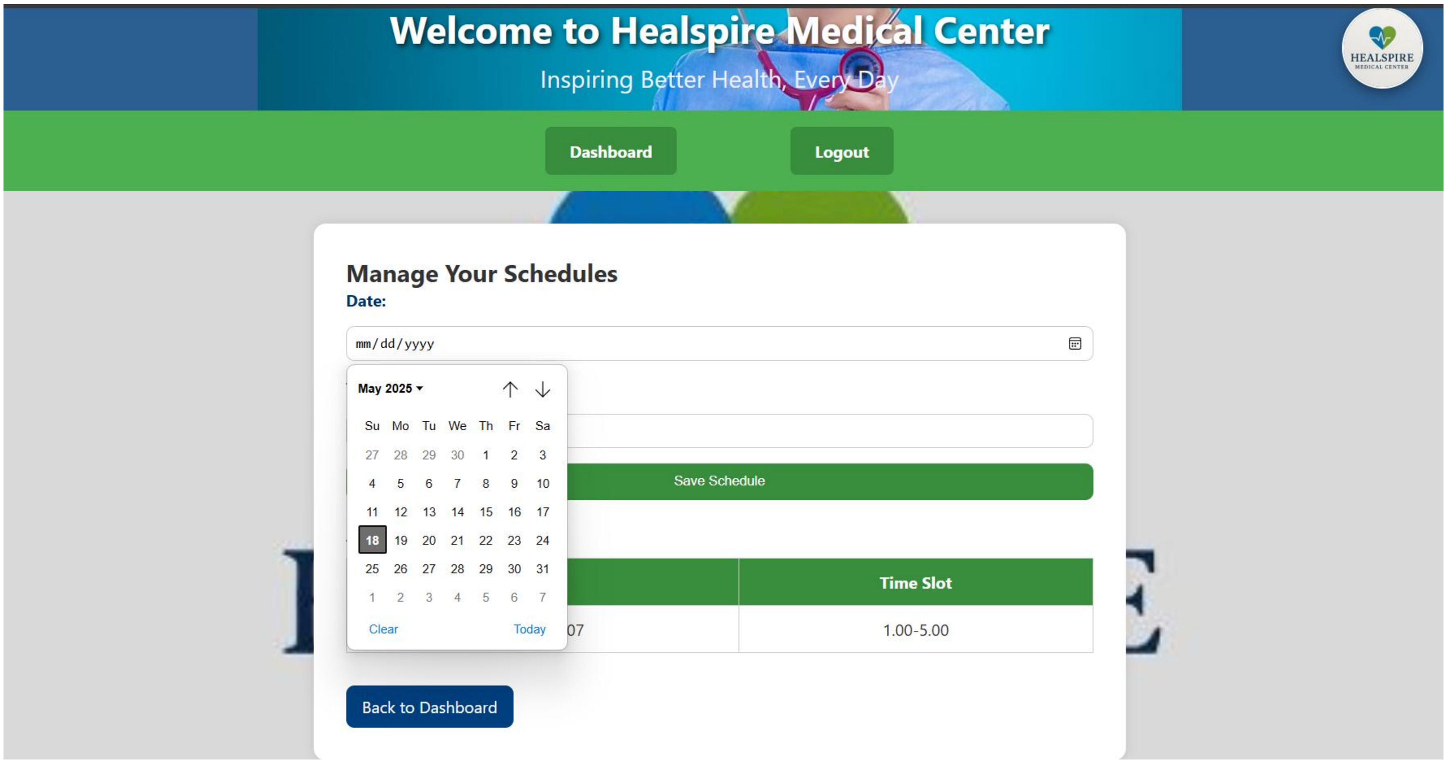
**Address:** MAHENWATHTHA, PITIPANA, HOMAGAMA, SRI LANKA
**Phone:** +1 (555) 123-4567
**Email:** info@HealspireMedicalCenter.com
© 2025 Our Hospital Management System - All Rights Reserved

# Doctor Dashborad Page



Welcome, Dr. Dr Namal 🧑‍⚕️ 👩‍⚕️

1. 1.Welcome to your dashboard. Select an option above to manage patients, appointments, and medical records.
2. 2.Use the menu above to view and manage your patients, schedule appointments, and update records.
3. 3.Choose an action above to access patient profiles, organize your appointments, or review clinical notes.
4. Select from the options above to oversee patient care, book appointments, and manage records.



Welcome to Healspire Medical Center
Inspiring Better Health, Every Day
HEALSPIRE MEDICAL CENTER
Dashboard
Logout

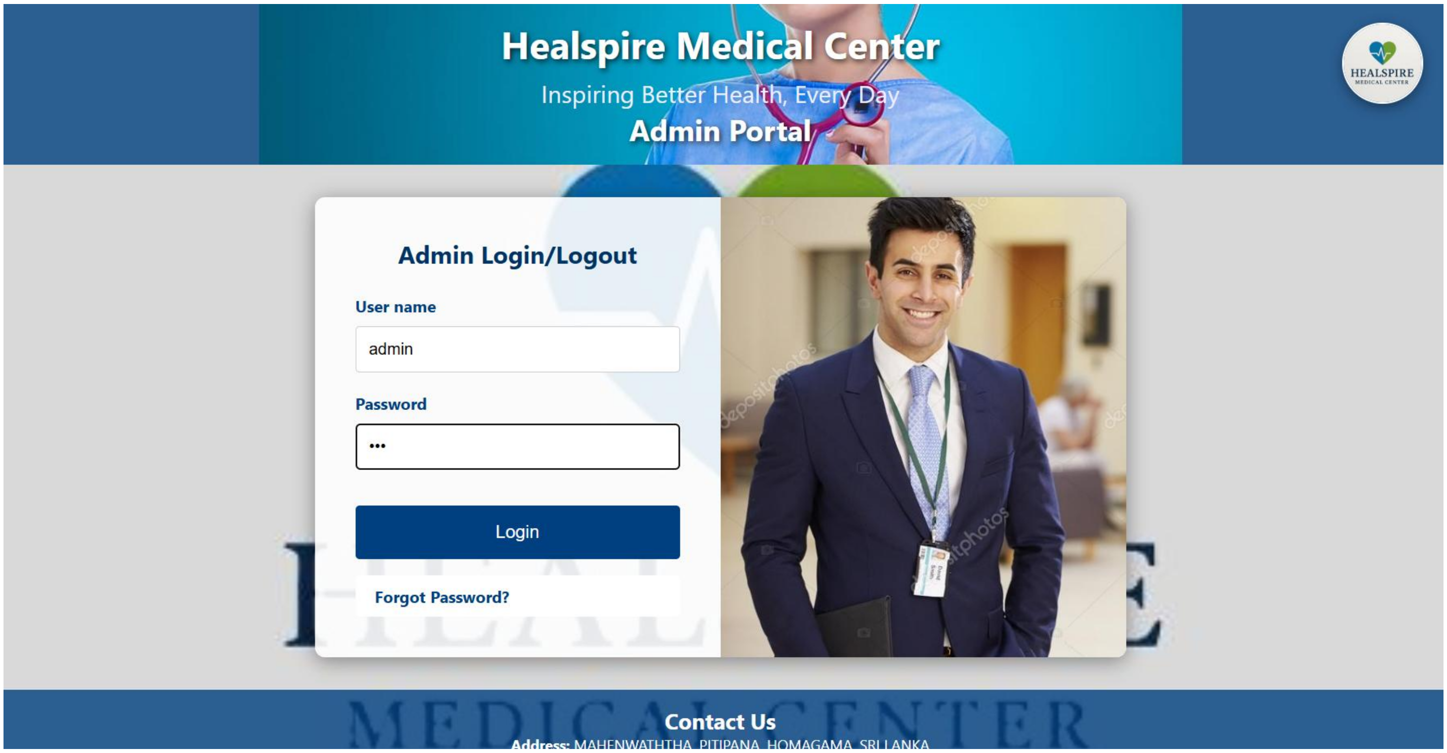
Manage Your Schedules

Date:

mm/dd/yyyy

| | Time Slot |
|---|---|
| 07 | 1.00-5.00 |

# Admin login page



Healspire Medical Center

Inspiring Better Health, Every Day

Admin Portal

Admin Login/Logout

User name

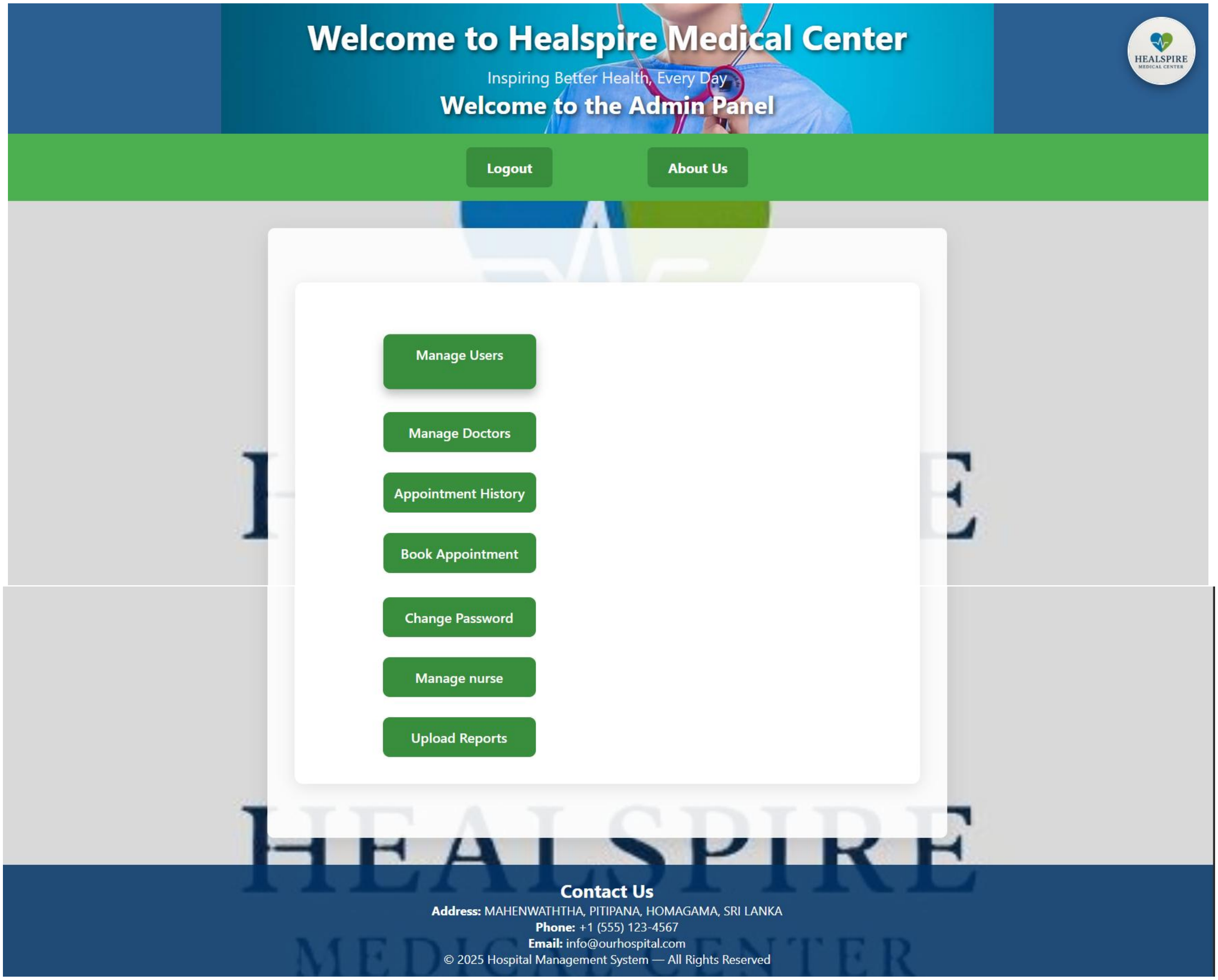
admin

Password

•••

Contact Us

**Address:** MAHENWATHTHA PITIPANA HOMAGAMA SRI LANKA



# Welcome to Healspire Medical Center

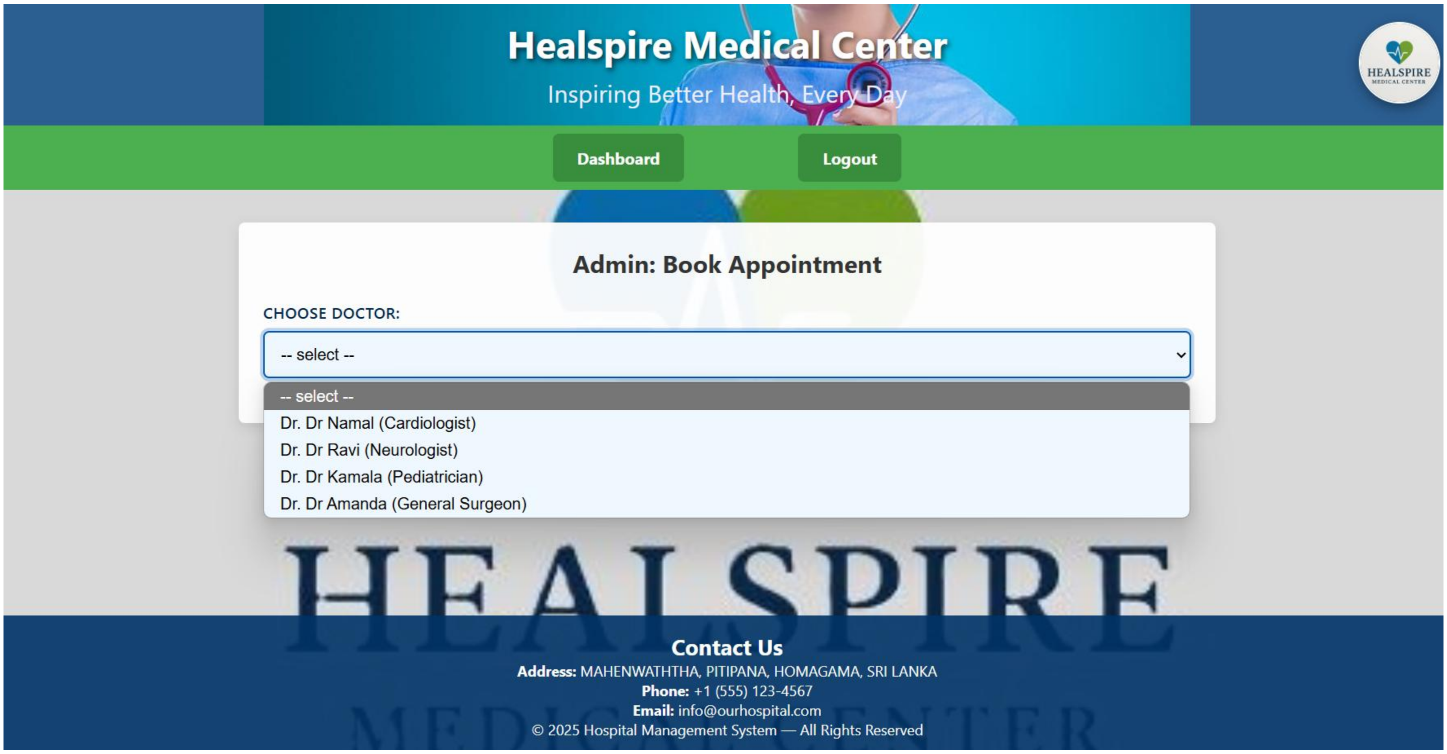
Inspiring Better Health, Every Day

## Welcome to the Admin Panel

### Contact Us

**Address:** MAHENWATHTHA, PITIPANA, HOMAGAMA, SRI LANKA
**Phone:** +1 (555) 123-4567
**Email:** info@ourhospital.com
© 2025 Hospital Management System — All Rights Reserved

# Book appoinment(admin)



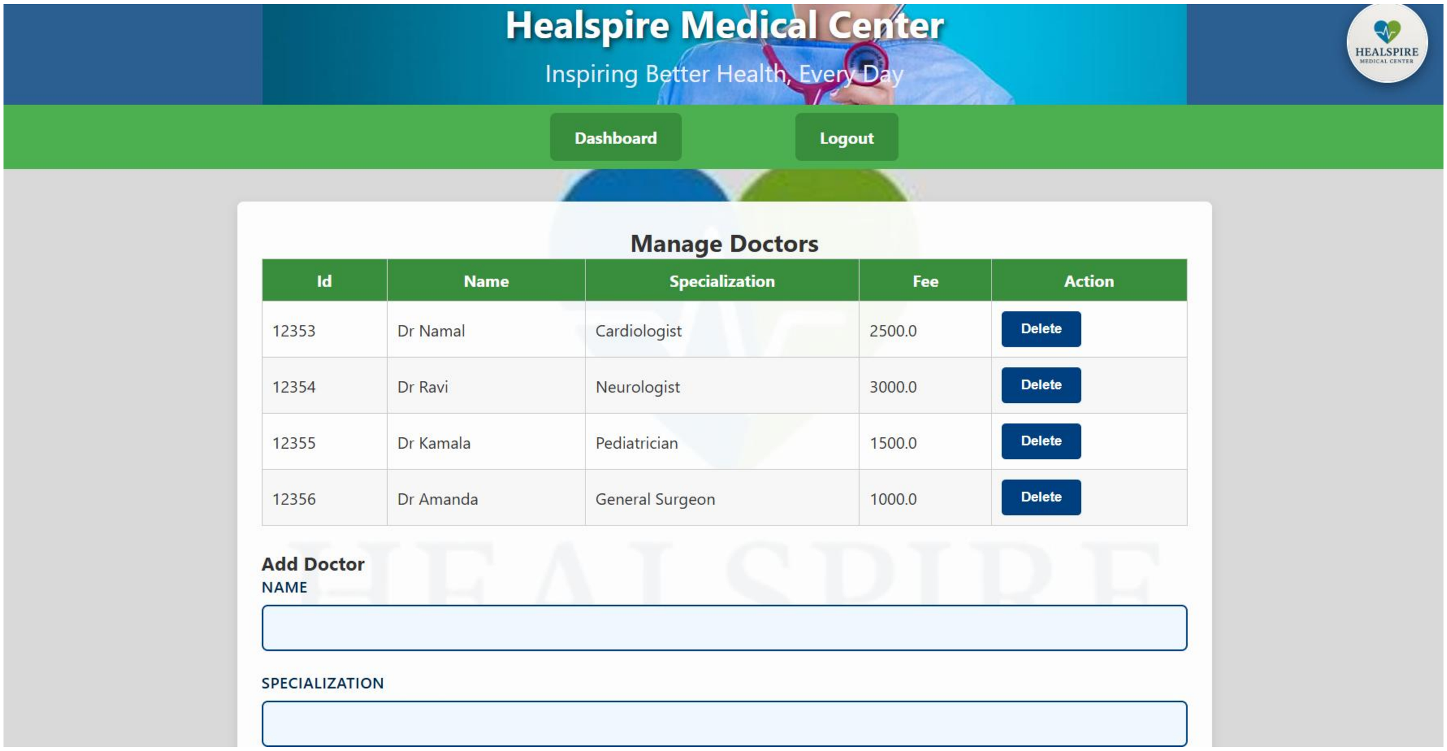
## Admin: Book Appointment

CHOOSE DOCTOR:

-- select --

- -- select --
- Dr. Dr Namal (Cardiologist)
- Dr. Dr Ravi (Neurologist)
- Dr. Dr Kamala (Pediatrician)
- Dr. Dr Amanda (General Surgeon)

**Contact Us**
**Address:** MAHENWATHTHA, PITIPANA, HOMAGAMA, SRI LANKA
**Phone:** +1 (555) 123-4567
**Email:** info@ourhospital.com
© 2025 Hospital Management System — All Rights Reserved



## Manage Doctors

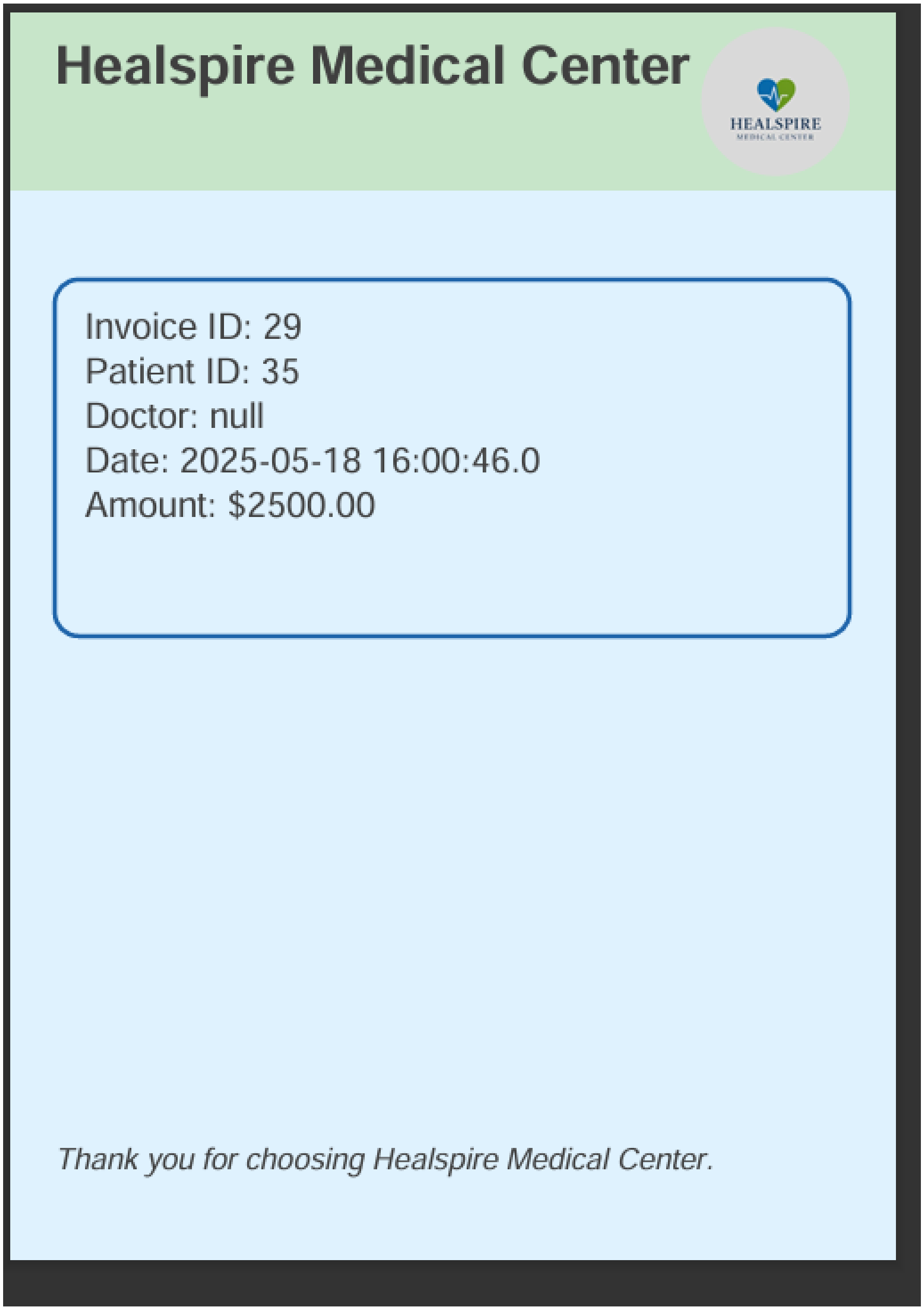
| Id | Name | Specialization | Fee | Action |
|---|---|---|---|---|
| 12353 | Dr Namal | Cardiologist | 2500.0 | Delete |
| 12354 | Dr Ravi | Neurologist | 3000.0 | Delete |
| 12355 | Dr Kamala | Pediatrician | 1500.0 | Delete |
| 12356 | Dr Amanda | General Surgeon | 1000.0 | Delete |

**Add Doctor**

NAME

SPECIALIZATION

### Hospital Invoice

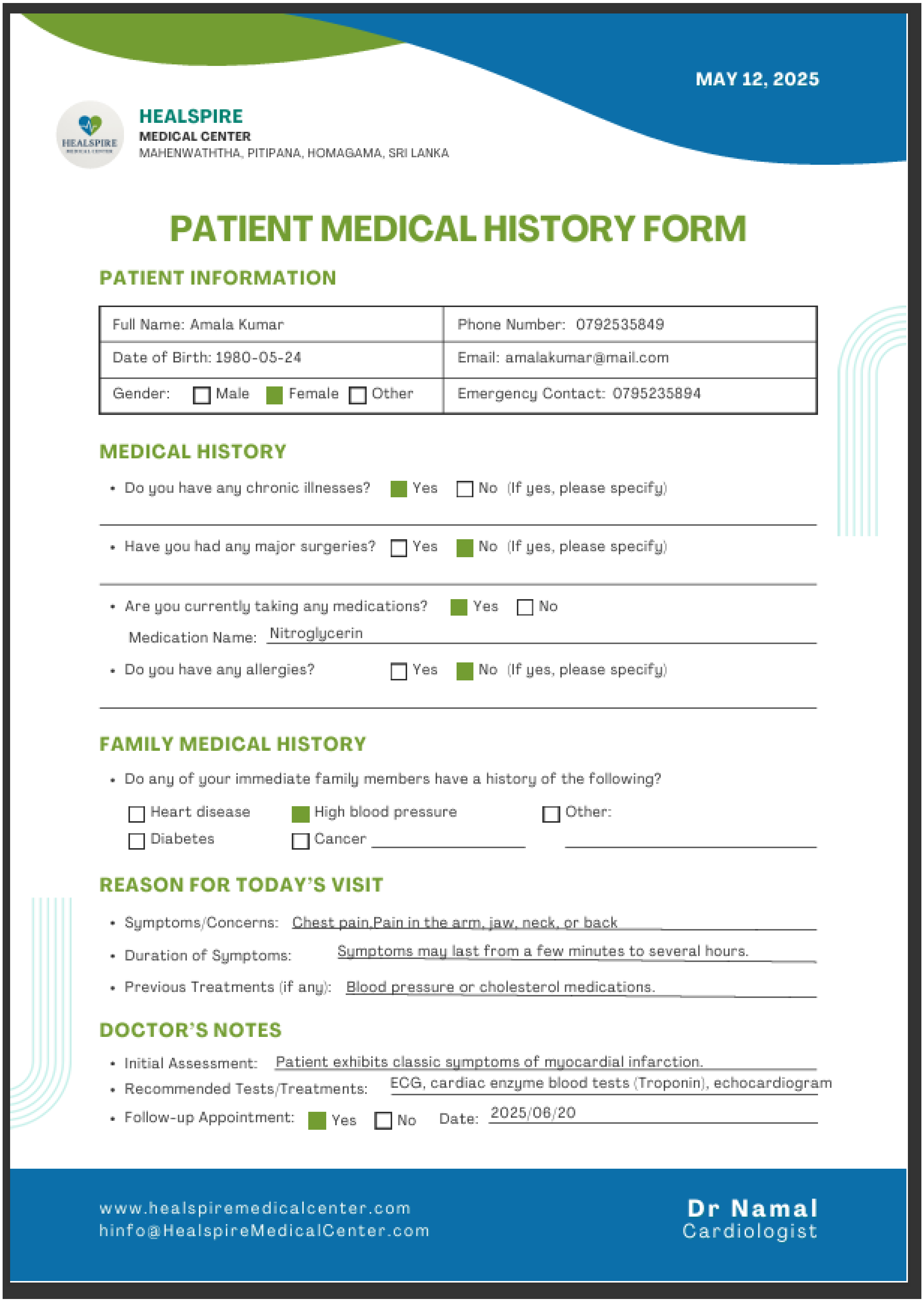


## Healspire Medical Center

Invoice ID: 29
Patient ID: 35
Doctor: null
Date: 2025-05-18 16:00:46.0
Amount: $2500.00

*Thank you for choosing Healspire Medical Center.*

# Example Ptient Report



MAY 12, 2025

HEALSPIRE
MEDICAL CENTER
MAHENWATHTHA, PITIPANA, HOMAGAMA, SRI LANKA

## PATIENT MEDICAL HISTORY FORM

### PATIENT INFORMATION

| | |
|---|---|
| Full Name: Amala Kumar | Phone Number: 0792535849 |
| Date of Birth: 1980-05-24 | Email: amalakumar@mail.com |
| Gender: □ Male ☑ Female □ Other | Emergency Contact: 0795235894 |

### MEDICAL HISTORY

- Do you have any chronic illnesses? ☑ Yes □ No (If yes, please specify)
- Have you had any major surgeries? □ Yes ☑ No (If yes, please specify)
- Are you currently taking any medications? ☑ Yes □ No

  Medication Name: Nitroglycerin
- Do you have any allergies? □ Yes ☑ No (If yes, please specify)

### FAMILY MEDICAL HISTORY

- Do any of your immediate family members have a history of the following?

  □ Heart disease ☑ High blood pressure □ Other:
  □ Diabetes □ Cancer

### REASON FOR TODAY'S VISIT

- Symptoms/Concerns: Chest pain,Pain in the arm, jaw, neck, or back
- Duration of Symptoms: Symptoms may last from a few minutes to several hours.
- Previous Treatments (if any): Blood pressure or cholesterol medications.

### DOCTOR'S NOTES

- Initial Assessment: Patient exhibits classic symptoms of myocardial infarction.
- Recommended Tests/Treatments: ECG, cardiac enzyme blood tests (Troponin), echocardiogram
- Follow-up Appointment: ☑ Yes □ No Date: 2025/06/20

www.healspiremedicalcenter.com
hinfo@HealspireMedicalCenter.com

**Dr Namal**
Cardiologist