MediGuideBD

A Web-Based Healthcare Guidance Platform for Bangladesh

by

Ayan Bilas Roy
ID: CSE2302029036

MD Macksudur Hasan
ID: CSE2302029061

Fardous Hossain
ID: CSE2302029099

Apurbo Krishna Roy
ID: CSE2302029060

Shraboni Bairagi
ID: CSE2302029101

Supervised by
Professor Bulbul Ahamed

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)

September-2026

MediGuideBD

A Web-Based Healthcare Guidance Platform for Bangladesh

by

Ayan Bilas Roy
ID: CSE2302029036

MD Macksudur Hasan
ID: CSE2302029061

Fardous Hossain
ID: CSE2302029099

Apurbo Krishna Roy
ID: CSE2302029060

Shraboni Bairagi
ID: CSE2302029101

Supervised by
Professor Bulbul Ahamed

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)

September-2026

APPROVAL

The project titled **"MediGuideBD A Web-Based Healthcare Guidance Platform for Bangladesh"** submitted by **Ayan Bilas Roy** (CSE2302029036)**, MD Macksudur Hasan** (CSE2302029061), **Fardous Hossain** (CSE2302029099), **Apurbo Krishna Roy** (CSE2302029060), **Shraboni Bairagi** (CSE2302029101) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Professor Bulbul Ahamed
Head of the Department
Department of Computer Science and Engineering
Sonargaon University (SU)

Supervisor

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 1

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 2

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 3

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Professor Bulbul Ahamed,** Head**,** Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this thesis has been or is being submitted elsewhere for the award of any degree.

Countersigned

(Professor Bulbul Ahamed)
Supervisor

Signature

Ayan Bilas Roy
ID: CSE2302029036

MD Macksudur Hasan
ID: CSE2302029061

Fardous Hossain
ID: CSE2302029099

Apurbo Krishna Roy
ID: CSE2302029060

Shraboni Bairagi
ID: CSE2302029101

ABSTRACT

MediGuideBD is a comprehensive web-based healthcare guidance platform developed to address the critical challenge of specialist identification in Bangladesh's healthcare system, where patients frequently struggle to determine the right medical expert for their symptoms, leading to wasted time, unnecessary costs, and delayed care. This three-tier client-server web application intelligently bridges that gap through three core services: a symptom-based specialist search using a weighted mapping algorithm to match symptoms with appropriate specialties and display available doctors with real-time availability; a disease-based hospital search that ranks facilities by specialization level (Best, Good, or Standard); and emergency services that locate ambulances, blood banks, pharmacies, and emergency centers, complemented by first-aid guides. Built with HTML5, CSS3, and Vanilla JavaScript for a responsive, bilingual (English and Bengali) interface, the platform leverages Node.js and Express.js for backend logic and routing, with a MySQL database storing all healthcare data, including doctors, hospitals, diseases, symptoms, specialties, time slots, and mappings. Additional features include a doctor dashboard for profile and schedule management and an administrative panel for system oversight. Importantly, MediGuideBD does not diagnose or treat conditions; it solely guides users to the appropriate place and professional, eliminating middlemen and reducing patient confusion. Developed as a BSc final year project at Sonargaon University, it demonstrates how technology can create meaningful, real-world impact in Bangladesh's healthcare sector by simplifying access to specialized care and emergency support.

KEYWORDS: Healthcare Guidance, Specialist Identification, Symptom-Based Search, Weighted Mapping Algorithm, Hospital Ranking, Emergency Services, Bilingual Interface

ACKNOWLEDGEMENT

At the very beginning, we would like to express my deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully within the schedule time.

We are auspicious that we had the kind association as well as supervision of **Professor Bulbul Ahamed**, Head, Department of Computer Science & Engineering, Sonargaon University whose hearted and valuable support with best concern and direction acted as necessary recourse to carry out our project.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to my parents for their support, encouragement, and endless love.

LIST OF ABBREVIATIONS

API	Application Programming Interface
BMDC	Bangladesh Medical and Dental Council
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
GPS	Global Positioning System
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
ID	Identifier
JSON	JavaScript Object Notation
MySQL	My Structured Query Language
REST	Representational State Transfer
SQL	Structured Query Language
SSL	Secure Sockets Layer
UAT	User Acceptance Testing
UI	User Interface
URL	Uniform Resource Locator

TABLE OF CONTENTS

Title	Page No.
DECLARATION ……...……	iii
ABSTRACT ……...……	iv
ACKNOWLEDGEMENT ……...……	v
LIST OF ABBREVIATIONS ……...……	vi
CHAPTER 1	
INTRODUCTION	**1-4**
1.1 Background……	1
1.2 Problem Statement…….	1
1.3 Objectives……..	2
1.3.1 Core Guidance Features……...	2
1.4 Project Scope…….	3
1.4.1 In Scope (System Features & Capabilities) …….	3
1.4.2 Out of Scope (System Exclusions) ……..	3
1.5 Expected Outcome……..……	4
CHAPTER 2	
LITERATURE REVIEW	**5-6**
2.1 Introduction……..……..	5
2.2 Existing Healthcare Systems……..……	5
2.2.1 Comparative Analysis……..…….	5
2.2.2 Research Gap……..……...	6
2.3 Overview of the Healthcare Landscape in Bangladesh……..……	6
CHAPTER 3	
FEASIBILITY ANALYSIS AND REQUIREMENT GATHERING	**7-12**
3.1 Introduction……..……..	7
3.2 Purpose of Feasibility Analysis……..……	7
3.3 Technical Feasibility……..……	8
3.4 Operational Feasibility……..…….	8
3.4.1 Operational Workflow……..…….	9
3.5 Economic Feasibility……..……	9
3.6 Feasibility Summary……..…….	10

3.7 Requirement Gathering……. 10
3.8 Stakeholder Identification……. 10
3.9 Functional Requirements……..…….. 11
3.10 Non-Functional Requirements…….…… 12

CHAPTER 4
SYSTEM ANALYSIS & DESIGN **13-26**
4.1 Introduction…….…….. 13
4.2 System Architecture…….……. 13
4.3 System Components…….……. 14
4.4 System Modules…….……... 14
4.4.1 Home Module…….…… 14
4.4.2 Disease-Based Search Module……. 15
4.4.3 Symptom-Based Search Module…….. 15
4.4.4 Emergency Search Module…….. 16
4.4.5 Admin Module……. 16
4.5 Use Case Diagram……. 17
4.6 Data Flow Diagram (DFD) ……... 17
4.6.1 Level 0 DFD……. 18
4.6.2 Level 1 DFD……. 18
4.7 Database Design…… 19
4.8 Entity Relationship Diagram (ERD) …… 19
4.9 Database Table Design……. 20
4.10 Search Mechanism and System Logic…… 25
4.10.1 Disease Search Logic……. 25
4.10.2 Symptom Search Logic…….. 25
4.10.3 Emergency Search Logic……... 26

CHAPTER 5
IMPLEMENTATION **27-40**
5.1 Frontend Implementation…….. 27
5.1.1 Key Page Implementations……... 27
5.1.2 Responsive Design Strategy……. 34
5.2 Backend Implementation……... 34
5.2.1 Server Architecture…….. 34
5.2.2 RESTful API Design…… 35
5.2.3 Database Connection…… 37
5.3 Database Implementation…….. 38
5.3.1 Database Schema Design……. 38

5.3.2 Table Creation SQL……. 38
5.3.3 Initial Data Population……. 39
5.4 User Module Implementation…… 39
5.4.1 General User Module……... 39
5.4.2 Doctor Module……. 40
5.4.3 Admin Module……. 40

CHAPTER 6
TESTING, RESULTS, AND DEMONSTRATION **41-46**
6.1 Testing Methodology…… 41
6.1.1 Test Environment……. 42
6.2 Test Cases and Results…….. 42
6.2.1 Functional Test Result Discussion……... 43
6.3 Demonstration……... 43
6.3.1 System Walkthrough…… 43
6.3.2 Key Demonstration: From Symptom to Doctor……... 44
6.3.3 Doctor Availability and Appointment Contact…… 45
6.3.4 Disease-Based Hospital Search…… 45
6.3.5 Emergency Services Demonstration…… 46
6.4 Results Summary……... 46
6.5 Testing Considerations and Limitations…… 46

CHAPTER 7
CONCLUSION & FUTURE SCOPE **47**
7.1 Conclusion……. 47
7.2 Future Scope…….. 47

REFERENCES **48-49**

LIST OF TABLES

Table No.	Title	Page
Table 2.1	Comparison of Existing Healthcare Systems with MediGuideBD	6
Table 3.1	Feasibility Areas and Main Concern	8
Table 3.2	Technology Stack and Purpose	8
Table 3.3	Economic Feasibility	9
Table 3.4	Feasibility Summary	10
Table 3.5	Stakeholder and their role in system with core interest	11
Table 3.6	Functional Requirements	11
Table 3.7	Non-Functional Requirements	12
Table 4.1	Users Table	20
Table 4.2	Doctors Table	21
Table 4.3	Specialties Table	21
Table 4.4	Divisions Table	21
Table 4.5	symptom_specialty_map Table	21
Table 4.6	doctor_time_slots Table	22
Table 4.7	diseases Table	22
Table 4.8	hospitals Table	22
Table 4.9	hospital_specialties Table	22
Table 4.10	ambulance_services Table	23
Table 4.11	blood_banks Table	23
Table 4.12	pharmacies Table	24
Table 4.13	emergency_centers Table	24
Table 5.1	Authentication APIs	35
Table 5.2	Doctor Search APIs	35
Table 5.3	Doctor Profile APIs	36
Table 5.4	Hospital and Disease APIs	36
Table 5.5	Emergency Services APIs	36
Table 6.1	Test Environment	42
Table 6.2	Scenario and Test Result	42-43

LIST OF FIGURES

Figure No.	Title	Page
Figure 3.1	Feasibility Analysis of MediGuideBD	7
Figure 3.2	MediGuideBD Operational Workflow	9
Figure 3.3	Requirement Gathering Process	10
Figure 3.4	Stakeholder Interaction with MediGuideBD	10
Figure 4.1	Overall System Architecture	14
Figure 4.2	Use Case Diagram	17
Figure 4.3	Level 0 DFD Diagram	18
Figure 4.4	Level 1 DFD Diagram	18
Figure 4.5	MediguideBD Database Entity Relationship Diagram	20
Figure 4.6	Disease Search Diagram	25
Figure 4.7	Symptom Search Diagram	25
Figure 4.8	Emergency Search Diagram	26
Figure 5.1	Homepage	27
Figure 5.2	Symptom-Based Search Page	28
Figure 5.3	Dynamic Dropdown	28
Figure 5.4	Disease-Based Search Page	29
Figure 5.5	Disease-Based Search Dropdown	30
Figure 5.6	Doctor Dashboard	30
Figure 5.7	Doctor's Schedule Management Page	31
Figure 5.8	Doctor's Time Slot Management Page	31
Figure 5.9	Admin Dashboard	32
Figure 5.10	Doctor Management Page	32
Figure 5.11	Emergency Search Page	32
Figure 5.12	Emergency Ambulance Management Page	33
Figure 5.13	Emergency Blood Bank Management Page	33
Figure 5.14	Emergency Pharmacy Management Page	33
Figure 5.15	Emergency Center Management Page	34
Figure 5.16	Ambulance Add Page	34
Figure 5.17	MediguideBD Backend Components	35

Figure 5.18	MediguideBD Login API Sequence Diagram	37
Figure 5.19	MediguideBD Backend Route Handler Components	37
Figure 6.1	MediGuideBD Testing Lifecycle	41
Figure 6.2	MediGuideBD User Journey	43
Figure 6.3	Symptom-Based Specialist and Doctor Selection	44
Figure 6.4	Doctor Availability and Next-Available Logic	45
Figure 6.5	Disease-Based Specialized Hospital Search	45

Chapter 1

Introduction

1.1 Background

Bangladesh's healthcare ecosystem faces critical systemic challenges, notably the scarcity of reliable medical information, difficulties in identifying appropriate specialists, and unequal access to clinical facilities [1], [2]. A significant portion of the population lacks the requisite health literacy to identify specialized services aligned with their specific medical conditions. This information asymmetry is particularly acute for rural patients migrating to urban medical centers for tertiary care. Consequently, vulnerable patients frequently resort to informal intermediaries, such as hospital brokers. These actors often exploit the knowledge gap by offering biased or misleading referrals driven by financial incentives rather than clinical necessity. Such practices result in inflated medical expenditures, unnecessary delays in treatment, and compromised health outcomes. To mitigate this information deficit, a centralized, transparent digital solution is required. MediGuideBD was developed as a direct-to-user healthcare guidance platform designed to bridge this gap, enabling patients to navigate specialist selection and facility location seamlessly without reliance on third-party intermediaries [1], [2].

1.2 Problem Statement

Patients in Bangladesh face critical obstacles in navigating the national healthcare ecosystem in an efficient, cost-effective, and transparent manner. Specifically, the core problem consists of three interconnected operational deficits:

- **Information & Specialist Misalignment:** Patients lack access to verified, centralized guidance to accurately map their symptoms to appropriate medical specialists, resulting in improper consultations and prolonged diagnostic delays.
- **Lack of Direct Facility & Emergency Guidance:** Patients encounter severe difficulties in identifying nearby specialized clinical facilities for specific conditions and securing real-time emergency support during critical situations.
- **Exploitation via Information Asymmetry:** As detailed in the Rational Discussion (Section 3.1), the absence of a direct-to-user information channel forces reliance on informal intermediaries (brokers), leading to inflated healthcare costs and biased referrals.

Therefore, there is an urgent need for a reliable, centralized digital platform like **MediGuideBD** to eliminate intermediary dependency, streamline healthcare navigation, and directly connect patients to appropriate medical resources.

1.3 Objectives

The project aims to establish **MediGuideBD** as a centralized, reliable healthcare navigation portal tailored to the Bangladeshi context. To fulfill the goals outlined in Section 2, the specific objectives are organized below:

➢ 1.3.1 Core Guidance Features

- **Symptom-Based Recommendation:** To recommend relevant medical specialists dynamically based on user-reported symptoms, affected anatomical areas, and geographic location.
- **Specialized Hospital Discovery:** To connect users with appropriate specialized medical centers when a specific disease or condition is already diagnosed.
- **Availability Ranking:** To display doctor profiles ordered by real-time and upcoming availability slots, minimizing patient waiting time.

Doctor Management

- **Admin Verification:** To maintain a trusted medical directory where all doctor profiles undergo mandatory administrative verification.
- **Autonomous Schedule Management:** To enable verified practitioners to update their active practice locations, consultation hours, and available booking slots directly.
- **Provider Incentive:** To encourage active participation by providing healthcare professionals free platform visibility and patient referrals.

Emergency Support

- **Immediate First-Aid Protocols:** To deliver standardized, easily actionable first-aid instructions for common emergency scenarios.
- **Location-Aware Ambulance Services:** To display localized, direct contact details for emergency ambulance providers according to the user's current area.
- **Blood Bank Information:** To help users quickly find nearby blood banks and access their contact information during emergency situations.
- **Emergency Center Information:** To provide information and contact details of nearby emergency centers for urgent medical assistance.
- **Nearby Pharmacy Information:** To help users locate nearby pharmacies and access their contact details when emergency medicines or basic medical supplies are required.

Transparent Healthcare Access

To make healthcare navigation more transparent, efficient, and universally accessible, this project aims to eliminate the core systemic barriers patients' face when seeking medical care in Bangladesh. The primary objective under this dimension is to significantly reduce patient dependency on informal hospital brokers and intermediaries who often misguide patients for financial gain. By providing direct, automated healthcare guidance through a centralized web

application, the platform ensures that users can discover appropriate medical specialists and facilities without incurring additional consultation, referral, or brokerage fees, thereby fostering a fair, direct, and cost-effective healthcare discovery process.

1.4 Project Scope

The scope of **MediGuideBD** defines the boundaries and capabilities of the platform. It outlines the specific functionalities, technical architecture, and geographic limits supported by the system, along with explicit exclusions to maintain focus as a healthcare navigation tool rather than a clinical provider.

➢ 1.4.1 In Scope (System Features & Capabilities)

- **Symptom-to-Specialty Mapping:** A rule-based guidance tool that recommends appropriate medical specialists based on user symptoms, affected body parts, and current location.
- **Disease-to-Hospital Search:** A lookup directory enabling users with pre-diagnosed conditions to find relevant specialized hospitals.
- **Division-Level Location Searching:** Location filtering and categorization restricted to the division level across Bangladesh.
- **Real-Time Doctor Availability & Ranking:** Dynamic sorting of verified doctor profiles based on active schedules and upcoming consultation slots.
- **Admin-Verified Directory:** A secure doctor management system where doctor registrations undergo strict administrator verification before publication.
- **Emergency Support Services:** Quick access to standard first-aid instructions and region-specific emergency ambulance,blood banks, pharmacies and emergency center location and contact numbers.
- **Web-Based Platform:** Web application architecture built using standard full-stack web technologies (HTML, CSS, JavaScript, Node.js, and MySQL).

➢ 1.4.2 Out of Scope (System Exclusions)

- **Medical Diagnosis & Prescription:** The system does not provide automated clinical diagnosis, medication prescriptions, or medical treatment plans.
- **District or Upazila-Level Search:** Detailed location filtering below the division level (e.g., district, upazila, or local area) is excluded in the current implementation.
- **Dedicated Mobile Applications:** Native iOS or Android mobile applications are excluded; access is strictly browser-based.

- **AI/Machine Learning Models:** The symptom mapping logic relies on predefined relational database rules rather than machine learning algorithms.'
- **Telemedicine & Real-Time Consultation:** Live video calling, chat messaging, or direct online consultations between doctors and patients are not supported.

1.5 Expected Outcome

The core deliverable of this project is **MediGuideBD**, a fully functional, responsive, and user-centric web application built using HTML, CSS, JavaScript, Node.js, and MySQL. It directly targets key inefficiencies in Bangladesh's healthcare system by offering a structured digital platform for medical guidance.

The application delivers the following key outcomes:

- **Symptom-Based Navigation:** A rule-based search module that correlates user-reported symptoms and body parts to the correct medical specialist, preventing misdirection and wasted consultation fees.
- **Specialized Hospital Finder:** A directory mapping specific medical conditions directly to appropriate specialized hospitals across all division-level regions.
- **Availability-Driven Ranking:** A verified doctor directory that ranks healthcare professionals based on real-time and upcoming schedule availability, minimizing patient wait times.
- **Emergency Assistance:** Integrated emergency resources including essential first-aid response guides and direct contact details for local ambulance services.

By providing centralized, verified information, **MediGuideBD** removes reliance on informal intermediaries and hospital brokers, empowering patients with a clear, transparent, and efficient way to access necessary healthcare services.

Chapter 2

Literature Review

2.1 Introduction

This chapter provides a comprehensive review of the existing literature, theoretical frameworks, and current platforms relevant to **MediGuideBD**. It establishes the baseline context for web-based healthcare navigation in developing nations, focusing on symptom-to-specialty guidance, hospital discovery directories, and emergency resource management. Furthermore, this chapter critically evaluates existing local and international healthcare applications to identify functional limitations—such as middleman exploitation, inefficient doctor search algorithms, and the lack of verified schedule data. By establishing this theoretical and comparative foundation, the chapter outlines the core rationale behind the system architecture, design choices, and technical methodology adopted for this web application.

2.2 Existing Healthcare Systems

Several healthcare platforms currently provide health information, telemedicine, diagnostics, wellness tracking, and healthcare services. Global platforms such as Healthline [12] and WebMD [13] offer medically reviewed information and symptom-checking tools, while local platforms including Maya, Doctorola [14], Arogga [15], MedEasy [16], and Amar Doctor provide services such as doctor discovery, teleconsultation, appointment booking, and healthcare support. Integrated healthcare providers like Praava Health [17] and Amar Lab focus on consultations and diagnostic services, whereas wellness platforms such as HealthifyMe, Google Health/Fitbit, and Samsung Health [18] emphasize fitness, lifestyle, and personal health monitoring. Additionally, medication management apps, personal health record systems, online appointment platforms, digital healthcare service platforms, and provider-specific hospital portals support healthcare management and service access.

Despite their strengths, these platforms generally focus on information delivery, consultations, diagnostics, wellness tracking, or appointment management. Most do not provide a unified, Bangladesh-focused system that maps symptoms to the appropriate medical specialty, recommends suitable hospitals, ranks doctors based on availability, and offers location-based healthcare navigation. This gap highlights the need for MediGuideBD, which aims to guide users toward the most appropriate healthcare services through structured symptom-to-specialty mapping and localized healthcare recommendations.

➢ 2.2.1 Comparative Analysis

To understand the current healthcare technology landscape, a review of existing healthcare platforms was conducted. Global platforms such as Healthline [12] and WebMD [13] provide health information, educational resources, and symptom-related guidance, while local platforms including Doctorola [14], Arogga [15], MedEasy [16], Amar Doctor, and Jotno offer services such as doctor discovery, consultations, diagnostics, medicine delivery, and appointment management. The comparison table below summarizes the key features, strengths, and limitations of these systems and highlights the gap that MediGuideBD aims to address.

Feature	Healthline	WebMD	Doctorola	Arogga	MedEasy	Amar Doctor	Jotno	Praava	MediGuideBD
Health Information	✓	✓	✓	✓	✓	✓	✓	✓	✓
Symptom Guidance	✓	✓	Limited	Limited	Limited	✓	Limited	Limited	✓
Symptom → Specialist	Limited	Limited	Limited	—	—	✓	Limited	Limited	✓
Disease → Specialized Hospital	—	Limited	—	—	Limited	Limited	Limited	Limited	✓
Doctor Directory	—	Limited	✓	✓	✓	✓	✓	✓	✓
Doctor Availability	—	—	Limited	Limited	Limited	Limited	Limited	Limited	✓
Next Available Doctor	—	—	—	—	—	—	—	—	✓
Emergency Support	Limited	✓	—	Limited	Limited	Limited	✓	✓	✓
Ambulance Information	—	—	—	—	—	—	✓	✓	✓
Bangladesh Focused	—	—	✓	✓	✓	✓	✓	✓	✓
Division-Level Search	—	—	Limited	Limited	Limited	Limited	Limited	Limited	✓
Broker Problem Addressed	—	—	—	—	—	—	—	—	✓
Diagnosis / Treatment	No	No	Consult.	Consult.	Consult.	Consult.	Consult.	Treatment	No

Note: ✓ = Available, Limited = Partially available, — = Not available or not a primary feature.

Table 2.1: Comparison of Existing Healthcare Systems with MediGuideBD

However, these capabilities are generally distributed across different platforms. There is an opportunity for a Bangladesh-focused healthcare guidance system that combines symptom-to-specialty mapping, disease-to-specialized-hospital search, division-level location, doctor availability ranking, next-available-doctor identification, verified doctor information, and emergency resources within a single platform.

➢ 2.2.2 Research Gap

The analysis indicates that existing healthcare platforms provide valuable individual services, but there remains a functional gap in combining healthcare-navigation features specifically for the Bangladeshi context. Most platforms focus on health information, consultation, appointments, medicine delivery, diagnostics, wellness tracking, or provider-specific services.

MediGuideBD addresses this gap by focusing on guidance rather thandiagnosisor treatment. The system combines symptom-to-specialty mapping, specialized-hospital search, division-level location, verified doctor information, availability-based doctor ranking, next-available-doctor identification, and emergency resources in one platform. This approach is intended to reduce confusion, improve access to appropriate healthcare services, and reduce patients' dependence on informal intermediaries or hospital brokers.

2.3 Overview of the Healthcare Landscape in Bangladesh

Bangladesh faces healthcare access challenges due to reliance on informal pharmacy-based providers, unequal distribution of medical specialists, and limited healthcare guidance [1], [2]. Many patients struggle to identify the appropriate specialist and may depend on intermediaries for information. The lack of a centralized source for doctor availability, hospital services, and emergency resources further complicates healthcare navigation, creating a need for accessible and reliable healthcare guidance systems [1], [2].

Chapter 3

Feasibility Analysis and Requirement Gathering

Project: MediGuideBD – A Web-Based Healthcare Guidance Platform for Bangladesh

3.1 Introduction

The success of a software project depends not only on implementation but also on careful planning, feasibility assessment, and systematic requirement analysis. Before developing MediGuideBD, it is necessary to determine whether the proposed system is technically achievable, operationally useful, and economically reasonable within the scope of an academic project.

MediGuideBD is proposed as a web-based healthcare guidance platform for Bangladesh. The system is intended to help users identify appropriate medical specialists using symptoms, affected body parts, and location. It also provides disease-based hospital search, emergency first-aid guidance, ambulance contact information, verified doctor profiles, and doctor-availability information.

The system is designed as an information and referral platform, not as an automated diagnostic or treatment system. Its purpose is to help users find relevant healthcare resources while leaving diagnosis and treatment decisions to qualified healthcare professionals.

3.2 Purpose of Feasibility Analysis

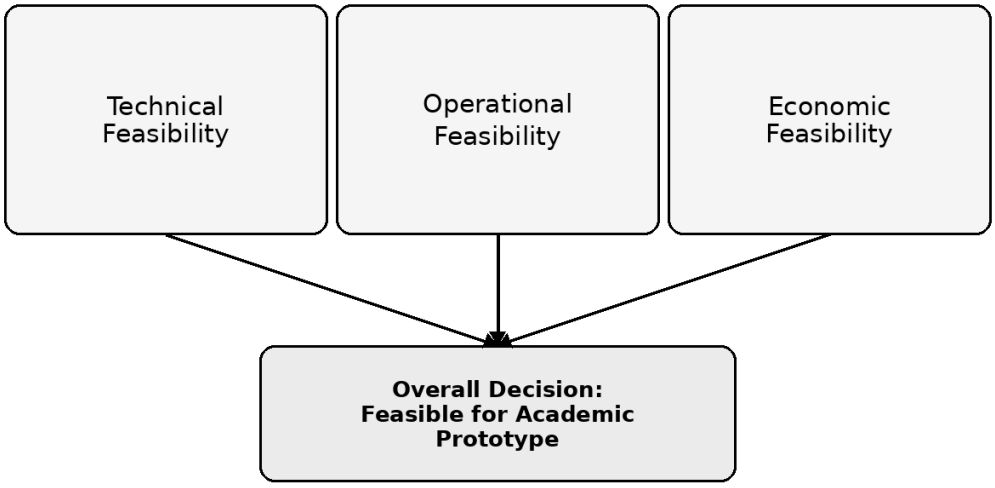


Figure 3.1: Feasibility Analysis of MediGuideBD

Feasibility analysis determines whether the proposed system can be developed and operated within the available technical, operational, and financial resources. For MediGuideBD, the analysis considers the planned technology stack, user roles, data requirements, project scope, and known limitations.

Feasibility Area	Main Concern
Technical Feasibility	Whether the selected technologies, infrastructure, and skills can support the proposed system.
Operational Feasibility	Whether intended users and administrators can use and maintain the system effectively.
Economic Feasibility	Whether the system can be developed and maintained within the resources available for an academic project.

Table 3.1 Feasibility Areas and Main Concern

3.3 Technical Feasibility

Technical feasibility evaluates the suitability of the proposed hardware, software, programming technologies, database system, and development skills required to implement MediGuideBD.

The proposed technology stack consists of HTML [3], CSS [4], and JavaScript [5] for the frontend, Node.js [6] for backend processing, and MySQL [8] for relational data management.

System Layer	Technology	Purpose
Frontend	HTML	Defines the structure and semantic content of web pages.
Frontend	CSS	Controls presentation, layout, and responsive styling.
Frontend	JavaScript	Provides client-side interaction and asynchronous communication.
Backend	Node.js	Handles server-side application logic and API processing.
Database	MySQL	Stores and manages structured relational data.

Table 3.2: Technology Stack and Purpose

3.4 Operational Feasibility

Operational feasibility evaluates whether the proposed system can address its intended problems and whether stakeholders can use and maintain it effectively.

MediGuideBD is intended to reduce the difficulty of locating relevant specialists, specialized hospitals, emergency information, and ambulance contacts by bringing these resources into a single web-based platform.

➢ 3.4.1 Operational Workflow

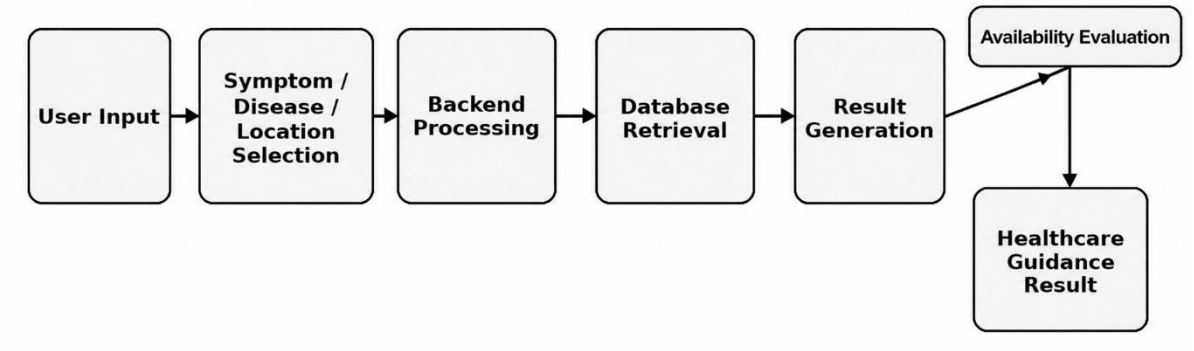


Figure 3.2: MediGuideBD Operational Workflow

The high-level workflow is:

User Input → Selection of Symptom/Disease/Location → Backend Processing → Database Retrieval → Result Generation → Availability Evaluation → Healthcare Guidance Result.

Operational Feasibility Decision: Feasible. The defined user roles provide clear responsibilities and support practical operation of the proposed platform.

3.5 Economic Feasibility

Economic feasibility evaluates whether the system can be developed within the project's financial constraints. Since MediGuideBD uses open-source technologies, development costs remain low, making it feasible for academic implementation. Future production deployment would require additional funding for hosting, security, maintenance, and operational support.

Cost Component	Academic Prototype	Future Production System
Development Hardware	Standard development computer	Dedicated or scalable infrastructure
Network	Standard internet connection	Production-grade connectivity
Software Tools	HTML/CSS/JavaScript, Node.js, MySQL	May include additional commercial services where required
Database	Local MySQL server	Managed or scalable cloud database
Web Hosting	Localhost or suitable low-cost/free-tier hosting where available	Dedicated/cloud hosting
Domain & SSL	Optional during local testing	Production domain and HTTPS configuration
Maintenance	Student project team	Dedicated maintenance/data-management personnel
Operational Support	Internal project administration	Formal support and data-verification process

Table 3.3: Economic Feasibility

3.6 Feasibility Summary

Feasibility Area	Key Finding	Decision
Technical Feasibility	Selected technologies support the planned web functionality.	Feasible
Operational Feasibility	Defined user roles provide clear system responsibilities.	Feasible
Economic Feasibility	Open-source technologies reduce initial academic development cost.	Feasible
Scope & Boundaries	Defined limitations keep the academic development target achievable.	Feasible

Table 3.4: Feasibility Summary

3.7 Requirement Gathering

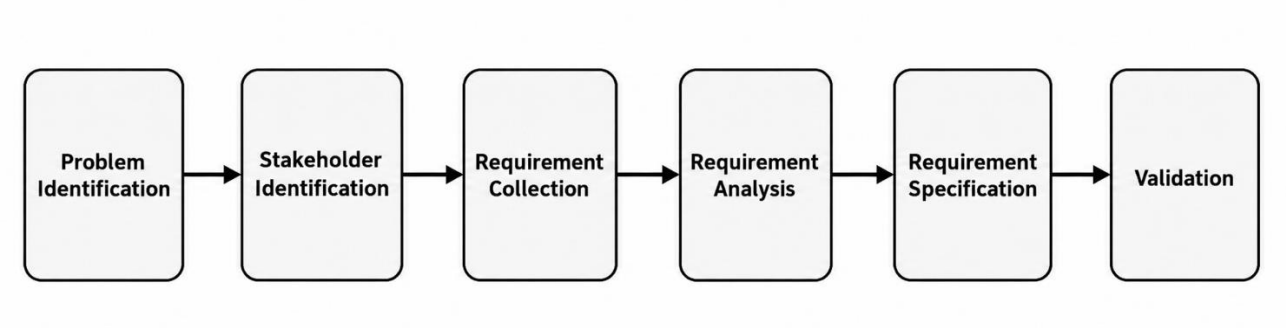


Figure 3.3: Requirement Gathering Process

Requirement gathering identifies what the system must provide and the quality standards it should satisfy. For MediGuideBD, requirements are derived from the project's stated objectives, intended stakeholder needs, proposed modules, and defined technical boundaries. The requirements are organized into functional and non-functional categories.

3.8 Stakeholder Identification

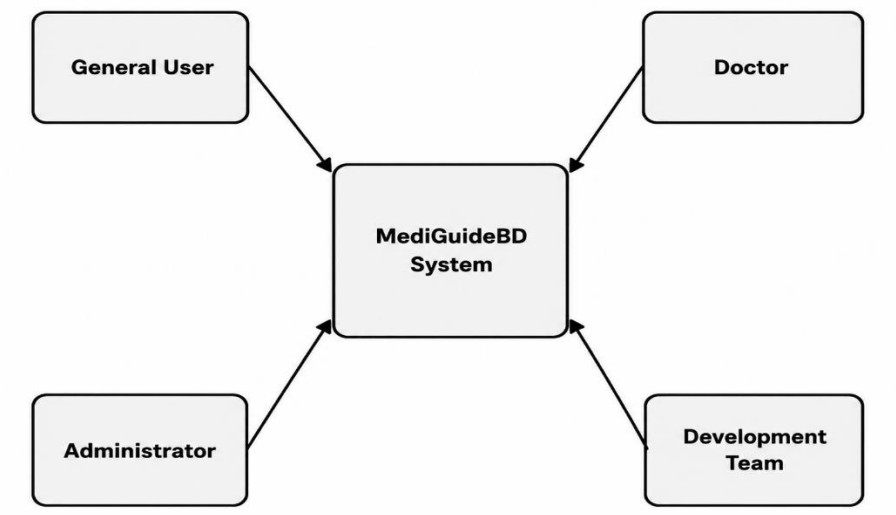


Figure 3.4: Stakeholder Interaction with MediGuideBD

Stakeholder	Role in System	Core Interest
General User	Consumes healthcare guidance services.	Quick identification of relevant doctors and healthcare facilities.
Doctor	Maintains permitted profile and schedule information.	Profile visibility and schedule management.
System Administrator	Verifies users and maintains system data.	Data accuracy, access control, and platform integrity.
Development Team	Designs, develops, tests, and deploys the system.	Successful delivery of the academic project.

Table 3.5: Stakeholder and their role in system with core interest

3.9 Functional Requirements

ID	Requirement	Specification
FR-01	Specialist Search	The system shall allow users to search for appropriate medical specialists using symptoms, body parts, and location.
FR-02	Symptom-Based Guidance	The system shall use predefined symptom-to-specialty mappings to suggest relevant medical specialties.
FR-03	Disease-Based Hospital Search	The system shall allow users to search for hospitals or healthcare facilities associated with selected diseases or conditions.
FR-04	Location-Based Filtering	The system shall filter relevant doctors and hospitals by selected administrative division.
FR-05	Doctor Availability Display	The system shall display verified doctor schedules and availability information.
FR-06	Next Available Doctor Suggestion	When a selected doctor is unavailable, the system shall suggest an available doctor matching the search criteria, where data permits.
FR-07	Doctor Verification	The system shall require authorized administrators to create and verify doctor accounts before public listing.
FR-08	Profile and Schedule Management	Verified doctors shall be able to update permitted schedule and contact information.
FR-09	Emergency and First-Aid Support	The system shall provide emergency first-aid guidance and ambulance contact information.
FR-10	Administrative Control	The system shall provide authorized administrators with CRUD operations for core system records.
FR-11	Frictionless Access	The system shall allow general users to perform public searches without mandatory account registration.
FR-12	Guidance Boundary	The system shall present results as healthcare guidance and shall not claim to provide clinical diagnosis or prescriptions.

Table 3.6: Functional Requirements

3.10 Non-Functional Requirements

ID	Attribute	Specification
NFR-01	Usability	The interface shall provide clear and intuitive navigation for non-technical users.
NFR-02	Performance	Search requests should target a response time of under 2 seconds under defined normal-load test conditions.
NFR-03	Reliability	The system shall return consistent results for valid requests based on current database records.
NFR-04	Security	The system shall enforce role-based access control for protected functions [11].
NFR-05	Data Accuracy	Authorized administrators shall maintain doctor, schedule, hospital, and emergency-contact data.
NFR-06	Maintainability	Application code and database structures shall be modular enough to support maintenance and future updates.
NFR-07	Availability	The application shall be accessible when the hosting environment and network services are operational.
NFR-08	Compatibility	The responsive interface shall support commonly used modern desktop and mobile browsers.
NFR-09	Scalability	The database design shall permit future expansion to district and upazila-level location data.
NFR-10	Privacy	Credentials and protected administrative information shall be handled using appropriate security controls [11].

Table 3.7: Non-Functional Requirements

Chapter 4

System Analysis & Design

4.1 Introduction

MediGuideBD is a comprehensive healthcare guidance web application designed to address a critical problem in Bangladesh's healthcare system: patients often do not know which specialist to consult, leading to wasted time, unnecessary expenses, and delayed treatment. The platform bridges this gap by intelligently guiding users from their symptoms to the right medical specialist, or from their diagnosed disease to the best specialized hospital nearby. The system provides three core services: symptom-based specialist search using weighted mapping to match symptoms with appropriate specialties, disease-based hospital search that ranks hospitals by specialization level, and emergency services connecting users to ambulance services, blood banks, pharmacies, and emergency centers along with first-aid guides.

The system is built as a client-server web application with a three-tier architecture. The frontend uses HTML [3], CSS [4], and JavaScript [5] to provide a responsive, bilingual interface, while the backend is developed with Node.js [6] and Express.js [7] to handle application logic and API communication. The MySQL database [8] stores all healthcare-related information. Including doctors, hospitals, diseases, symptoms, specialties, divisions, time slots, emergency services, and symptom-specialty mappings. This chapter describes the overall architecture, system modules, user interaction flows, database design, search mechanisms, and interface design of MediGuideBD.

4.2 System Architecture

MediGuideBD follows a client-server architecture [9], [10] where the frontend, backend server, and database work together to provide healthcare-related services to users. The major components of the system are the User Interface/Frontend, the Backend Server with Application Routes and Business Logic, and the MySQL Database.

The frontend provides the visual interface through which users interact with the system. HTML is used for the structure of web pages, CSS is used for styling and responsive presentation, and JavaScript is used for client-side interaction and communication with the backend. The backend server processes user requests, handles application routes, performs required operations, and communicates with the MySQL database. The database stores the information required by different modules of the application.

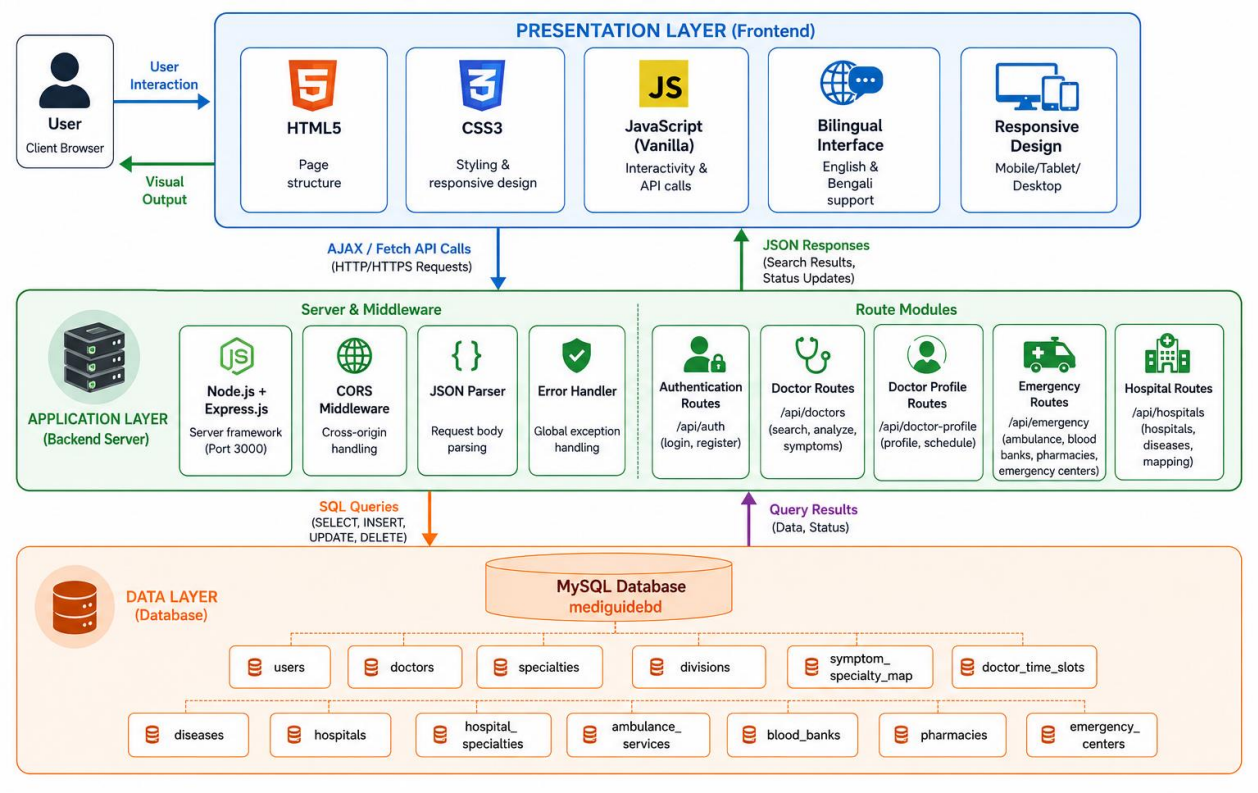


Figure 4.1: Overall System Architecture

The architecture allows the frontend to request information from the backend, while the backend retrieves or modifies the required information in the MySQL database. This separation of concerns [9], [10] ensures that the system remains maintainable, scalable, and secure.

4.3 System Components

The MediGuideBD system consists of three primary components: the **frontend, backend**, and **database**. The frontend provides the user interface through web pages and interactive features, the backend processes requests and implements business logic, and the database stores healthcare-related information such as doctors, hospitals, symptoms, diseases, emergency services, and user records. These components work together to provide healthcare guidance, information retrieval, and system administration functionalities.

4.4 System Modules

MediGuideBD is divided into several functional modules. Each module is designed to perform a specific healthcare-related task.

➢ 4.4.1 Home Module

The Home page acts as the main entry point of the MediGuideBD system.
The homepage provides access to the major services of the application:

- Search by Disease
- Search by Symptoms
- Emergency

It also introduces the purpose of MediGuideBD and presents the project's main objectives. The homepage additionally contains sections on the motive of the system, medical departments and specialties, covered locations, system statistics, and how the system works. The Home module allows users to quickly understand the purpose of the application and navigate to the required healthcare service.

➢ 4.4.2 Disease-Based Search Module

The disease-based search module allows users to search for hospitals according to a particular disease and location.

The user provides:

1. Disease name
2. Division/location

After submitting the search request, the system identifies relevant hospitals and displays them to the user. For example, when a user searches for **Heart Attack** in **Dhaka**, the system displays information about specialized hospitals related to the selected disease.

The search results may contain:

- Hospital name
- Hospital location
- Hospital ranking/category
- Contact number
- Directions option

This module helps users identify suitable healthcare facilities without manually searching through a large number of hospitals. The disease search page contains a search form where the user enters the disease name and selects the required division. The system then presents specialized hospitals according to the selected criteria.

➢ 4.4.3 Symptom-Based Search Module

The symptom-based search module is one of the main features of MediGuideBD. This module is designed to help users identify an appropriate medical specialist based on their symptoms.

The interface allows the user to provide information such as:

- Body part
- Side of the body
- Duration of the problem
- Location/division
- Description of the problem

After providing the required information, the user selects **Find Specialist**. The system processes the provided information and identifies an appropriate medical specialty and available doctors based on the stored healthcare data.

The module is designed to reduce the difficulty of deciding which type of specialist a user should consult.

➢ 4.4.4 Emergency Search Module

The Emergency module provides users search available private or govt run Amblunace, Blood Banks, Pharmacies and Emergency Centers along with information related to different medical emergencies. The purpose of this module is to make important emergency services and emergency-related information easily accessible from a single page.

The user to provide information such as:

- Selects Ambulances / Blood Banks / Pharmacies or Emergency Centers
- Division or location

After providing the information, the system processes the provided information and provides information about available Ambulances / Blood Banks / Pharmacies or Emergency Centers within that area. This section also give guidance about

- Different emergency conditions and Immediate steps or Emergency precautions
- Emergency precautions
- Relevant medical information

The system presents the information in a structured manner so that users can quickly find the emergency information they require. The Emergency Search module is designed to reduce the difficulty of finding Emergency services when every second matters also intended as an informational guidance feature and does not replace professional medical diagnosis or treatment.

➢ 4.4.5 Admin Module

MediGuideBD contains an administrative component that provides authorized management of system information. The administrative functions include:

- Managing doctor information (add, edit, delete)
- Managing emergency services records (ambulance, blood banks, pharmacies, emergency centers)
- Managing hospital information and disease mapping
- Updating system information

The admin panel features a dashboard with statistics showing total Doctors, Specialties, total Divisions (8), and Hospitals. The doctor management interface displays all doctors in a table also administrators can add new doctors through a dedicated form.

The emergency services management provides tabbed interfaces for each emergency service type such as Ambulance Services, Blood Banks, Pharmacies and Emergency Center. Each tab provides full CRUD operations (Create, Read, Update, Delete) with dedicated forms for adding and editing records.

The hospital management allows administrators to add new hospitals with name, map hospitals to diseases with ranking levels, view all hospitals with their mapped diseases and rankings also edit or delete existing hospital records.

4.5 Use Case Diagram

The main actor of MediGuideBD is the **User**. If the administrative functions are included in the final implementation, **Admin** is also considered an actor.
Main User Use Cases

The user can:

- Access the homepage
- Search by disease
- Search by symptoms
- Search for emergency services
- Find doctors
- Find hospitals
- View doctor information
- View hospital information
- Access emergency information

Admin Use Cases

The administrator can perform authorized data management operations through the administrative interface.

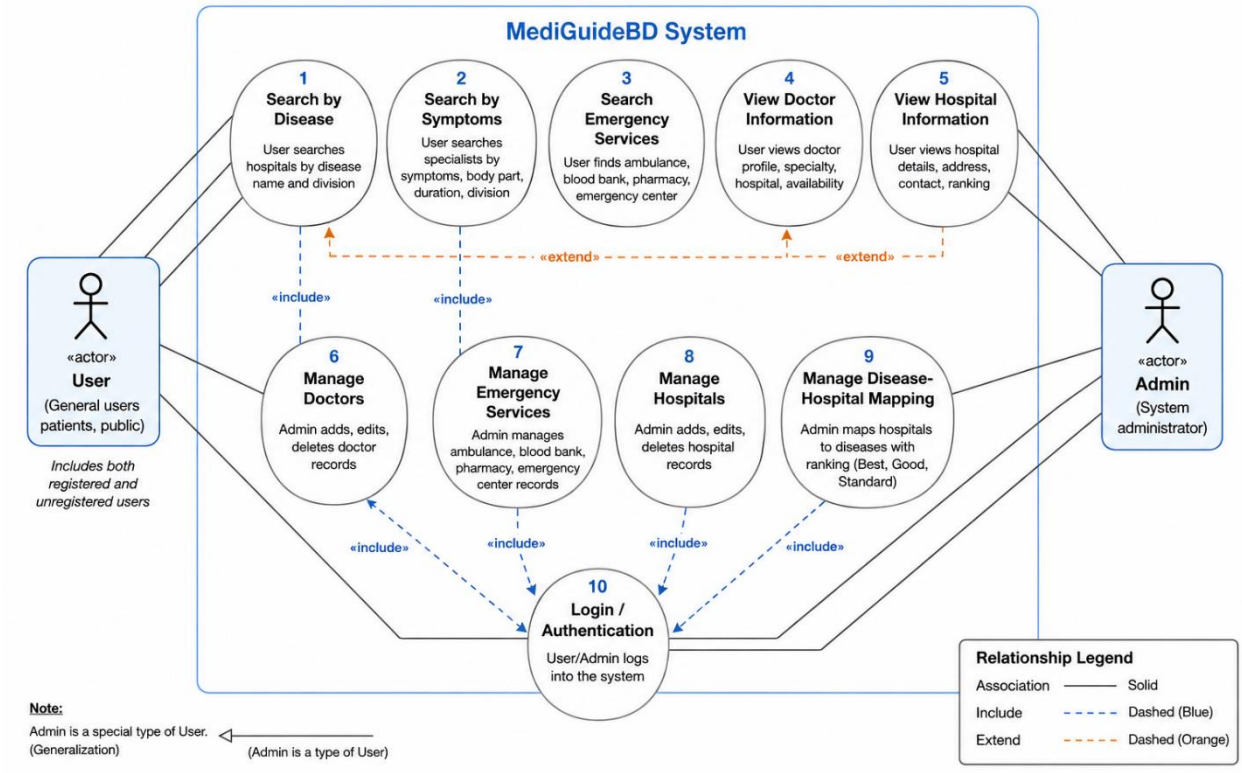


Figure 4.2: Use Case Diagram

4.6 Data Flow Diagram (DFD)

The Data Flow Diagram represents how information moves between the user, application, backend server, and database.

➢ 4.6.1 Level 0 DFD

At Level 0, MediGuideBD is represented as a single system. The user sends a request to MediGuideBD. The system processes the request and communicates with the MySQL database when required. The resulting information is then returned to the user interface.

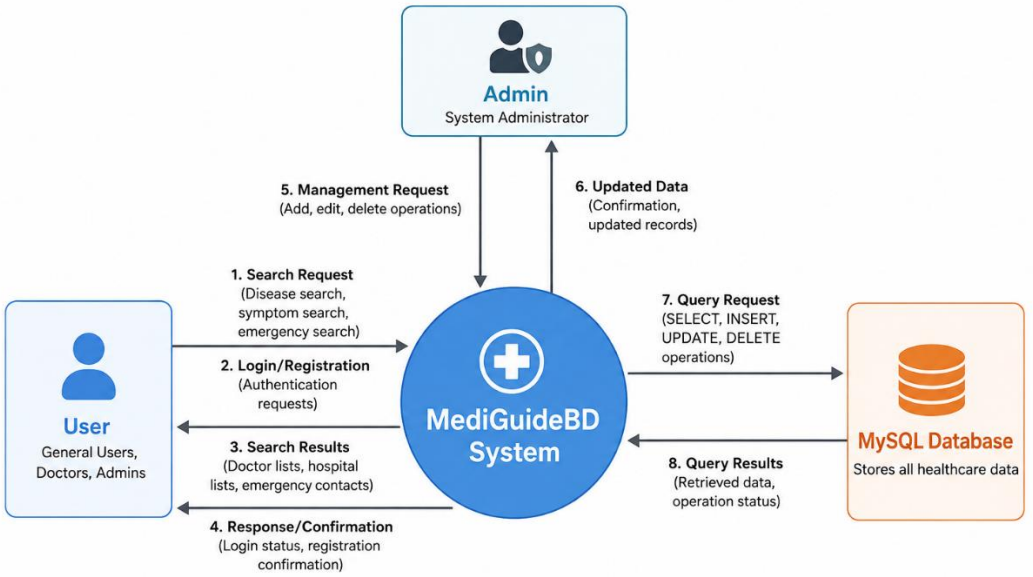


Figure 4.3: Level 0 DFD Diagram

➢ 4.6.2 Level 1 DFD

At Level 1, the main processes are divided into separate functions.

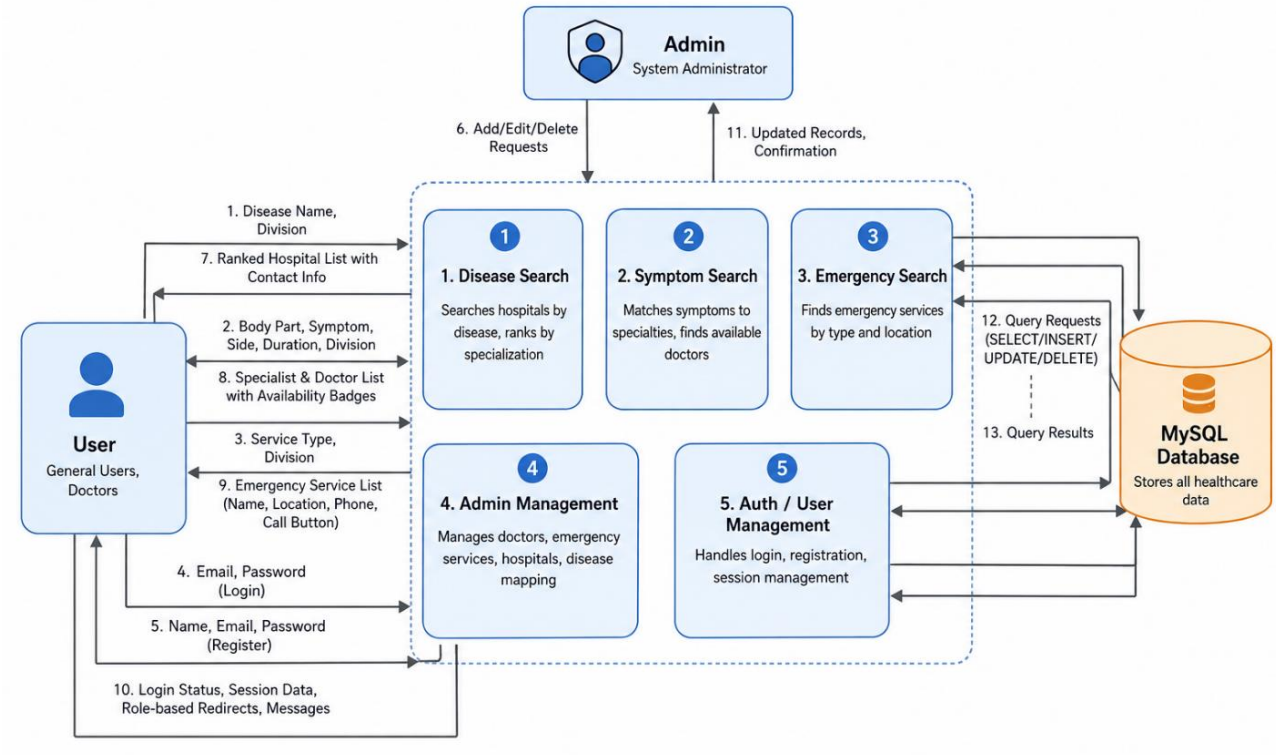


Figure 4.4: Level 1 DFD Diagram

4.7 Database Design

MySQL is used as the relational database management system for MediGuideBD. The database stores structured information required by the system's various modules, while the backend communicates with it through a dedicated database connection module. When the application starts, the database configuration is loaded, a connection to the MySQL server is established and verified, and SQL queries are executed whenever data is needed. The query results are then processed by the application and sent to the frontend.

The database system supports common operations such as:

- SELECT — retrieving information
- INSERT — adding information
- UPDATE — modifying information
- DELETE — removing information

The system also includes error handling for database connection and query-related failures.

The database connection flow developed for MediGuideBD illustrates the complete process from application startup to database connection, query execution, result processing, error handling, and application shutdown.

The database connection mechanism is designed to reduce connection overhead and improve the overall reliability of database operations.

4.8 Entity Relationship Diagram (ERD)

The Entity Relationship Diagram represents the relationships among the major entities of MediGuideBD.

The major conceptual entities include:

- User
- Doctor
- Specialty / Department
- Division
- Symptom
- Disease
- Hospital
- Hospital-Specialty Mapping
- Doctor Time Slot
- Ambulance Service
- Blood Bank
- Pharmacy
- Emergency Center
- Symptom-Specialty Map (Junction Table).

A conceptual relationship can be represented as follows:

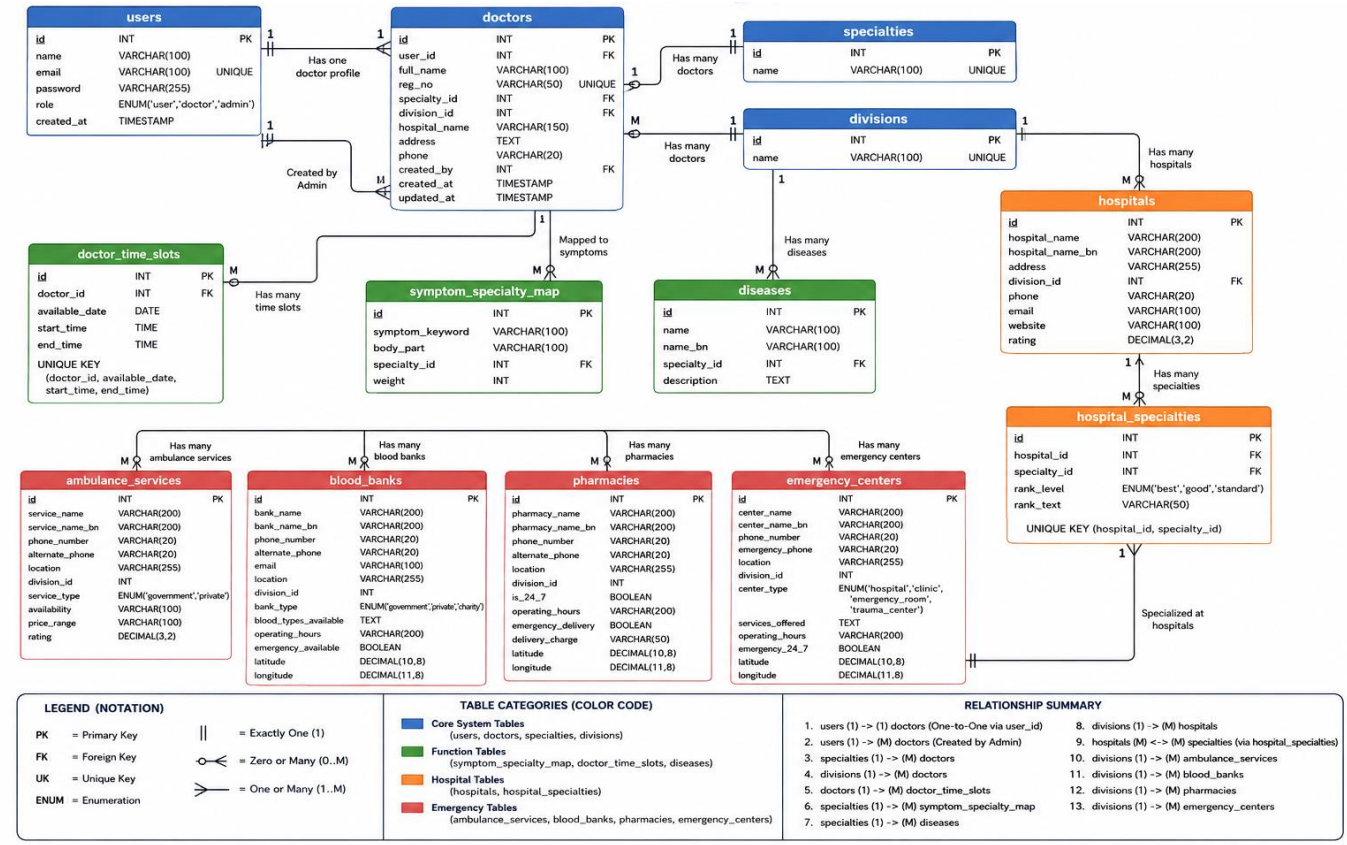


Fig 4.5: MediguideBD Database Entity Relationship Diagram

4.9 Database Table Design

The database tables should be documented using their actual fields, data types, primary keys, and foreign keys.

For example, the documentation format can be:

Field Name	Data Type	Key	Description
id	INT	Primary Key	Unique user identifier
name	VARCHAR(100)	—	User's full name
email	VARCHAR(100)	Unique	User's email address
password	VARCHAR(255)	—	User's password
role	ENUM	—	User role (user, doctor, admin)
created_at	TIMESTAMP	—	Account creation timestamp

Table 4.1: Users Table

Field Name	Data Type	Key	Description
id	INT	Primary Key	Unique doctor identifier
user_id	INT	Foreign Key	Reference to users table
full_name	VARCHAR(100)	—	Doctor's full name
reg_no	VARCHAR(50)	Unique	BMDC registration number
specialty_id	INT	Foreign Key	Reference to specialties table
division_id	INT	Foreign Key	Reference to divisions table
hospital_name	VARCHAR(150)	—	Current hospital/clinic
address	TEXT	—	Full address
phone	VARCHAR(20)	—	Contact number
created_by	INT	Foreign Key	Admin who created the record
created_at	TIMESTAMP	—	Creation timestamp
updated_at	TIMESTAMP	—	Last update timestamp

Table 4.2: Doctors Table

Field Name	Data Type	Key	Description
id	INT	Primary Key	Unique specialty identifier
name	VARCHAR(100)	Unique	Specialty name (e.g., Cardiology)

Table 4.3: Specialties Table

Field Name	Data Type	Key	Description
id	INT	Primary Key	Unique specialty identifier
name	VARCHAR(100)	Unique	Division name (e.g., Dhaka)

Table 4.4: Divisions Table

Field Name	Data Type	Key	Description
id	INT	Primary Key	Unique mapping identifier
symptom_keyword	VARCHAR(100)	—	Symptom/Problem name
body_part	VARCHAR(100)	—	Associated body part
specialty_id	INT	Foreign Key	Reference to specialties table
weight	INT	—	Priority weight (higher = more relevant)

Table 4.5: symptom_specialty_map Table

Field Name	Data Type	Key	Description
id	INT	Primary Key	Unique slot identifier
doctor_id	INT	Foreign Key	Reference to doctors table
available_date	DATE	—	Date of availability
start_time	TIME	—	Start time of slot
end_time	TIME	—	End time of slot

Table 4.6: doctor_time_slots Table

Field Name	Data Type	Key	Description
id	INT	Primary Key	Unique disease identifier
name	VARCHAR(100)	—	Disease name (English)
name_bn	VARCHAR(100)	—	Disease name (Bengali)
specialty_id	INT	Foreign Key	Reference to specialties table
description	TEXT	—	Disease description

Table 4.7: diseases Table

Field Name	Data Type	Key	Description
id	INT	Primary Key	Unique hospital identifier
hospital_name	VARCHAR(200)	—	Hospital name (English)
hospital_name_bn	VARCHAR(200)	—	Hospital name (Bengali)
address	VARCHAR(255)	—	Full address
division_id	INT	Foreign Key	Reference to divisions table
phone	VARCHAR(20)	—	Contact number
email	VARCHAR(100)	—	Email address
website	VARCHAR(100)	—	Website URL
rating	DECIMAL(3,2)	—	User rating (0-5)

Table 4.8: hospitals Table

Field Name	Data Type	Key	Description
id	INT	Primary Key	Unique mapping identifier
hospital_id	INT	Foreign Key	Reference to hospitals table
specialty_id	INT	Foreign Key	Reference to specialties table
rank_level	ENUM	—	Ranking (best, good, standard)
rank_text	VARCHAR(50)	—	Display text for rank

Table 4.9: hospital_specialties Table

Field Name	Data Type	Key	Description
id	INT	Primary Key	Unique service identifier
service_name	VARCHAR(200)	—	Service name (English)
service_name_bn	VARCHAR(200)	—	Service name (Bengali)
phone_number	VARCHAR(20)	—	Primary contact
alternate_phone	VARCHAR(20)	—	Secondary contact
location	VARCHAR(255)	—	Address/Location
division_id	INT	Foreign Key	Reference to divisions table
service_type	ENUM	—	Type (government, private)
availability	VARCHAR(100)	—	Availability information
price_range	VARCHAR(100)	—	Cost range
rating	DECIMAL(3,2)	—	User rating

Table 4.10: ambulance_services Table

Field Name	Data Type	Key	Description
id	INT	Primary Key	Unique bank identifier
bank_name	VARCHAR(200)	—	Bank name (English)
bank_name_bn	VARCHAR(200)	—	Bank name (Bengali)
phone_number	VARCHAR(20)	—	Primary contact
alternate_phone	VARCHAR(20)	—	Secondary contact
email	VARCHAR(100)	—	Email address
location	VARCHAR(255)	—	Address/Location
division_id	INT	Foreign Key	Reference to divisions table
bank_type	ENUM	—	Type (government, private, charity)
blood_types_available	TEXT	—	Available blood types
operating_hours	VARCHAR(200)	—	Working hours
emergency_available	BOOLEAN	—	24/7 emergency service
latitude	DECIMAL(10,8)	—	Map location
longitude	DECIMAL(11,8)	—	Map location

Table 4.11: blood_banks Table

Field Name	Data Type	Key	Description
id	INT	Primary Key	Unique pharmacy identifier
pharmacy_name	VARCHAR(200)	—	Pharmacy name (English)
pharmacy_name_bn	VARCHAR(200)	—	Pharmacy name (Bengali)
phone_number	VARCHAR(20)	—	Primary contact
alternate_phone	VARCHAR(20)	—	Secondary contact
location	VARCHAR(255)	—	Address/Location
division_id	INT	Foreign Key	Reference to divisions table
is_24_7	BOOLEAN	—	Open 24/7
operating_hours	VARCHAR(200)	—	Working hours
emergency_delivery	BOOLEAN	—	Emergency delivery available
delivery_charge	VARCHAR(50)	—	Delivery fee
latitude	DECIMAL(10,8)	—	Map location
longitude	DECIMAL(11,8)	—	Map location

Table 4.12: pharmacies Table

Field Name	Data Type	Key	Description
id	INT	Primary Key	Unique center identifier
center_name	VARCHAR(200)	—	Center name (English)
center_name_bn	VARCHAR(200)	—	Center name (Bengali)
phone_number	VARCHAR(20)	—	Primary contact
emergency_phone	VARCHAR(20)	—	Emergency hotline
location	VARCHAR(255)	—	Address/Location
division_id	INT	Foreign Key	Reference to divisions table
center_type	ENUM	—	Type (hospital, clinic, emergency_room, trauma_center)
services_offered	TEXT	—	Available services
operating_hours	VARCHAR(200)	—	Working hours
emergency_24_7	BOOLEAN	—	24/7 emergency service
latitude	DECIMAL(10,8)	—	Map location
longitude	DECIMAL(11,8)	—	Map location

Table 4.13: emergency_centers Table

4.10 Search Mechanism and System Logic

Search functionality is one of the core components of MediGuideBD.

➢ 4.10.1 Disease Search Logic

The disease search process follows these general steps:

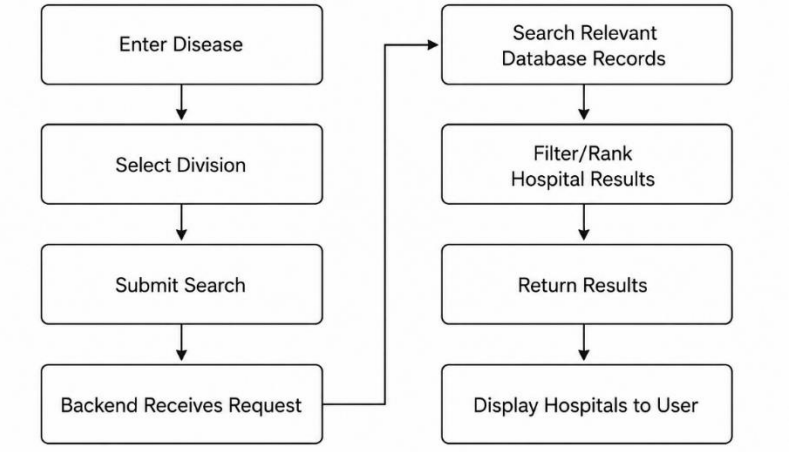


Fig 4.6: Disease Search Diagram

The system uses the disease and location information provided by the user to identify relevant hospital records.

➢ 4.10.2 Symptom Search Logic

The symptom-based search follows the following process:

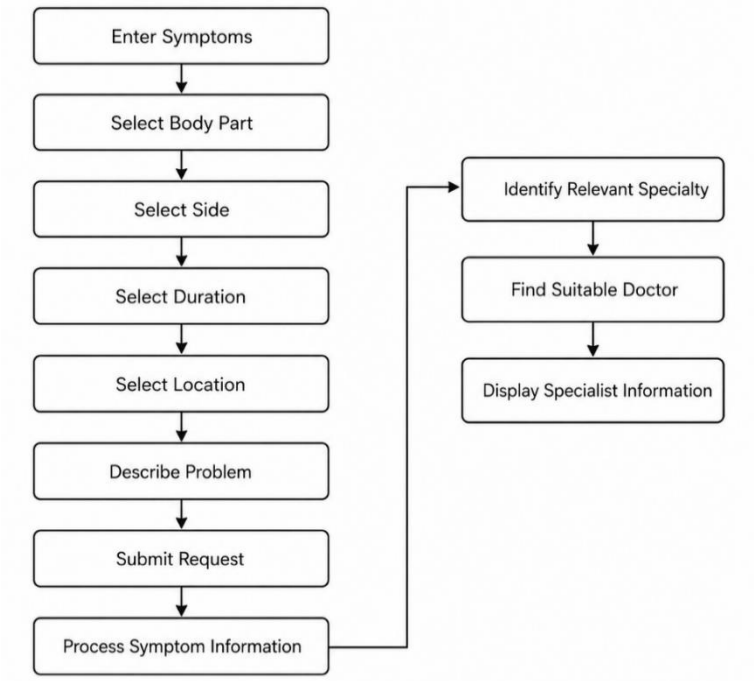


Fig 4.7: Symptom Search Diagram

This mechanism is designed to guide users toward an appropriate medical specialty based on their provided symptoms.

➢ 4.10.3 Emergency Search Logic

The Emergency Ambulance/Blood Bank/Pharmacy/Emergency Center search process can be represented as:

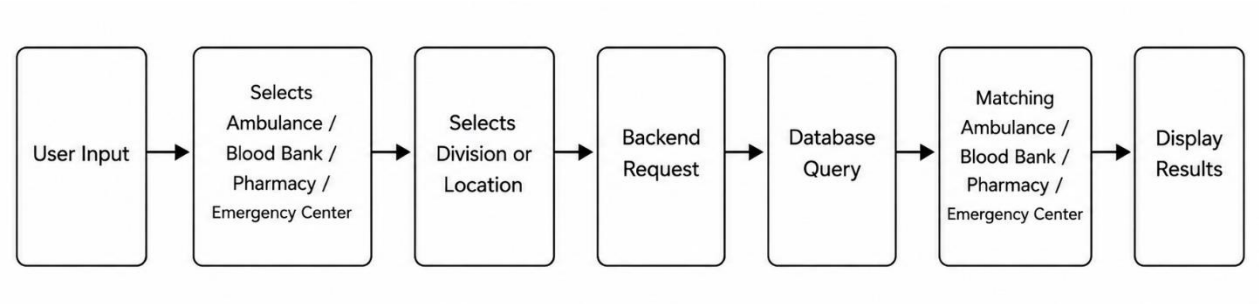


Fig 4.8: Emergency Search Diagram

The result interface provides important information such as hospital name, location, contact number, and directions.

Chapter 5

Implementation

5.1 Frontend Implementation

➢ 5.1.1 Key Page Implementations

A. Homepage (frontpage.html)

The homepage serves as the primary entry point and features:

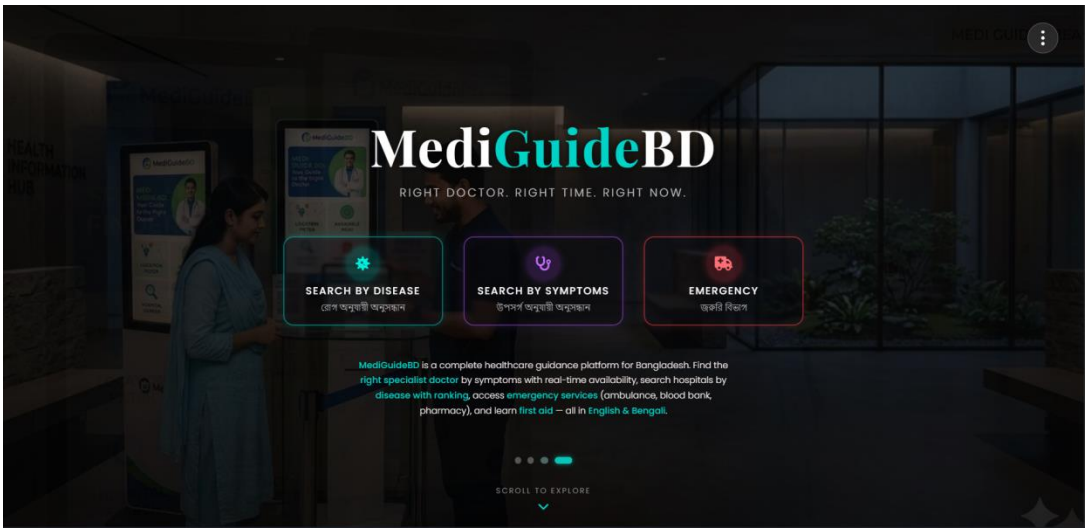


Fig 5.1: Homepage

The homepage serves as the primary entry point for all users. It features an automatic background slideshow that cycles through four healthcare-themed images every five seconds,creating a dynamic and engaging first impression. The centerpiece is the large “MediGuideBD" logo with its tagline "Right doctor. Right time. Right now."

Neon Box Buttons:

Three visually distinct cards guide users to the main features:

- **Search By Disease** (Teal/Cyan glow)
- **Search By Symptoms** (Purple/Magenta glow)
- **Emergency** (Red/Orange glow with pulse animation)

A three-dot menu in the top-right corner provides access to additional pages including Departments, About Us, Contact, and Profile. The homepage also includes a brief description explaining the platform's purpose and a subtle scroll indicator encouraging users to explore further.

B. Symptom Search Page (index.html)

This is the most interactive page, featuring a five-step form that collects user input:

- **Body Part Selection-** A dropdown with over thirty options ranging from Head to Skin
- **Side Selection-** Context-sensitive options (Left, Right, Both, Center) that change based on theselected body part
- **Duration Selection -** Time period options including less than 3 days, 1 week, or more than 15 days
- **Location Selection-** Division dropdown covering all eight divisions of Bangladesh
- **Problem/Symptom Selection-** A dynamically populated dropdown that loads relevant symptoms based on the selected body part

When the user clicks "Find Specialist," the system sends a request to the backend with all the collected data. The results section displays the recommended specialty name, the matched symptom

keyword,and a list of available doctors with clear availability badges showing who is "Available Now," who has appointments "Tomorrow," or who is "Not Available."



Fig 5.2: Symptom-Based Search Page

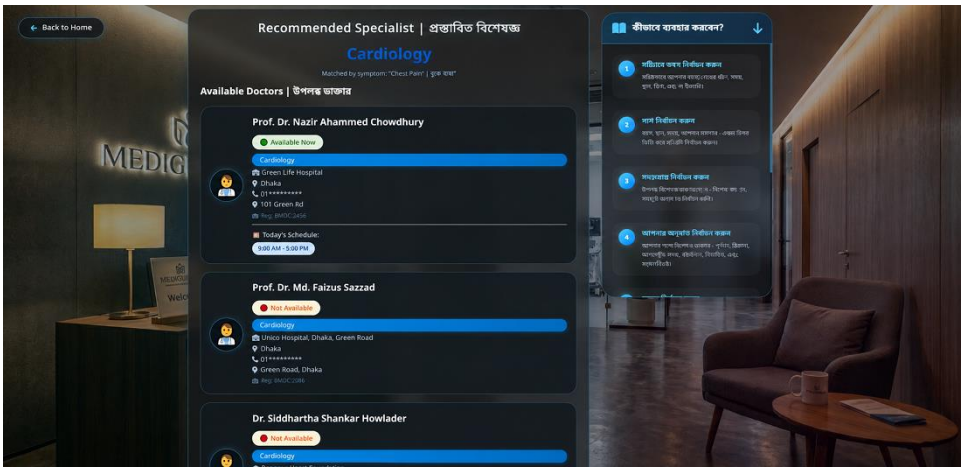


Fig 5.3: Dynamic Dropdown

Real-Time Availability Badges:

Doctors are displayed with clear availability indicators:

```
<span class="badge-available-now">🟢 Available Now</span>
<span class="badge-next-available">🔴 Tomorrow at 9:00 AM</span>
<span class="badge-no-schedule">🔴 Not Available</span>
```

C. Disease Search Page (Index2.html)

Features real-time disease autocomplete:

For users who already know their medical condition, this page offers a streamlined search experience. Users type a disease name into a text field with real-time autocomplete suggestions drawn from the database. As they type, matching diseases appear with their Bengali translations and associated specialties. Selecting a disease auto-fills the input field. After selecting a division from the dropdown and clicking "Search Hospitals," the system returns a list of specialized hospitals.

Each hospital displays a ranking badge indicating its specialization level:

- ☆**Best**- Top specialized hospital for that disease
- 👍**Good** - Well-equipped hospital
- 🏥**Standard**- General hospital

Each hospital card shows the facility name, address, phone number, user rating, and provides action buttons for calling directly or getting directions via Google Maps.

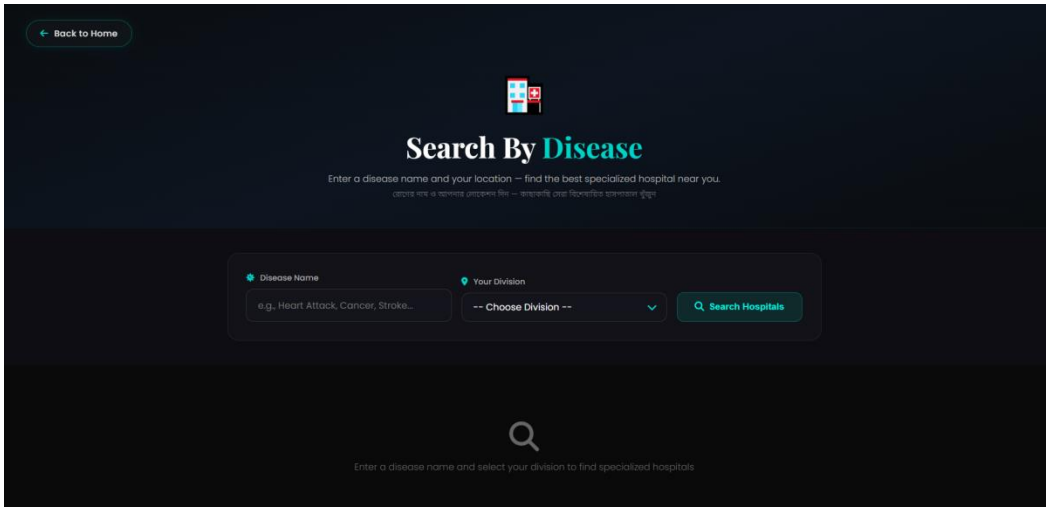


Fig 5.4: Disease-Based Search Page

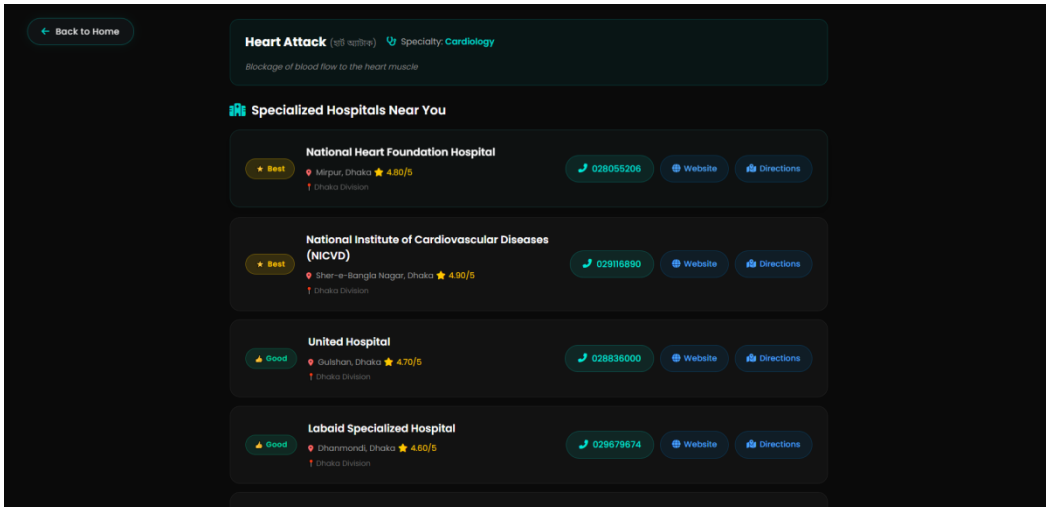


Fig 5.5: Disease-Based Search Dropdown

D. Doctor Dashboard (doctor.html)

The doctor dashboard provides authenticated healthcare professionals with a centralized control panel. The sidebar navigation offers access to:

- **Dashboard** - Overview with statistics
- **My Profile** - View personal information
- **Edit Profile** - Update contact details
- **My Schedule**- Manage availability
- **Logout** - End session

The dashboard shows welcome message with the doctor's name and summary cards displaying their specialty, hospital, and division. The profile view presents complete information including name, specialty badge, registration number, email, hospital, division, address, phone, and last update date.

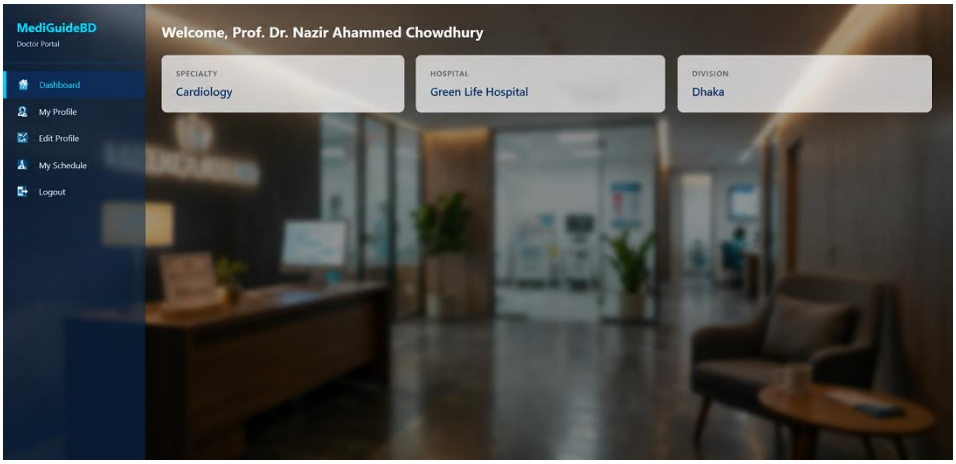


Fig 5.6: Doctor Dashboard

Calendar-Based Schedule Management:

Doctors can manage their availability through a calendar interface that allows navigation by month and year. Days are color-coded to indicate scheduled slots, unscheduled days, and past dates that cannot be modified. By selecting a future date, doctors can add, edit, or remove one or more time slots, with each slot defined by a start and end time.

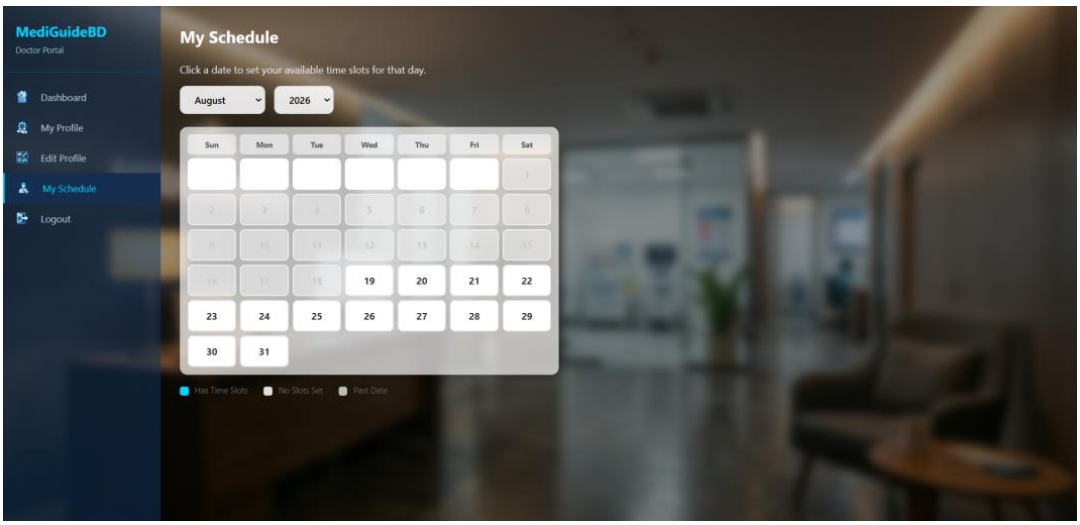


Fig 5.7: Doctor's Schedule Management Page

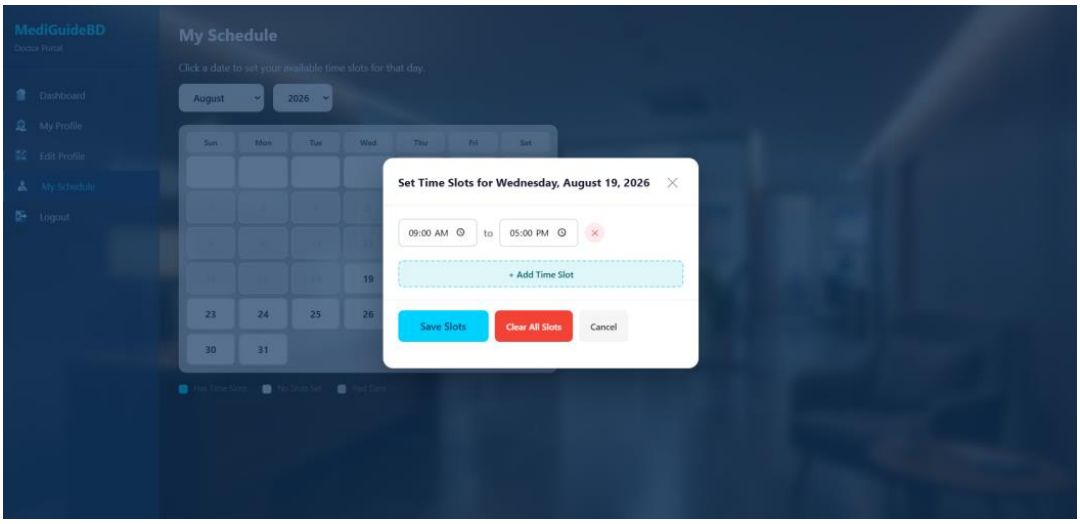


Fig 5.8: Doctor's Time Slot Management page

E. Admin Panel (admin.html)

The Admin Panel serves as the central management interface of MediGuideBD, allowing administrators to monitor and maintain the system's healthcare data. The dashboard provides an overview of key system information and offers management tools for doctors, medical specialties, hospitals, disease-to-hospital mappings, and emergency services. Through this interface, administrators can perform CRUD (Create, Read, Update, Delete) operations to ensure that all healthcare information remains accurate and up to date. The panel is designed to provide efficient system administration through a structured and user-friendly interface.

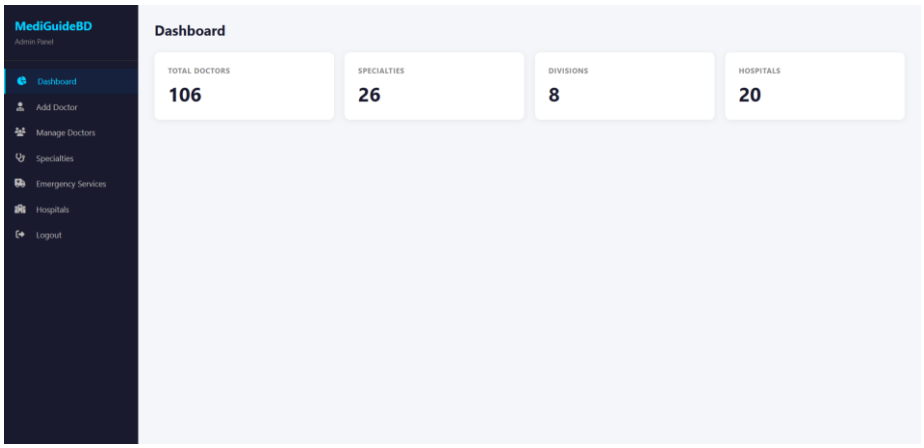


Fig 5.9: Admin Dashboard

Doctor Management:

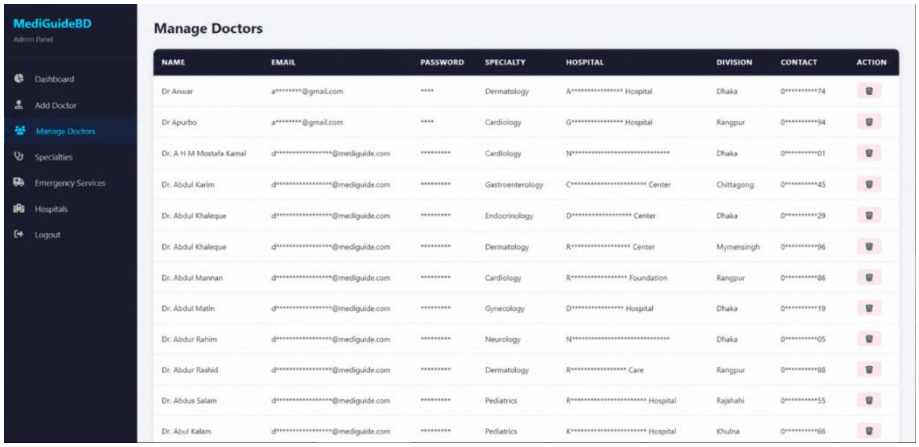


Fig 5.10: Doctor Management Page

Emergency Services Management:

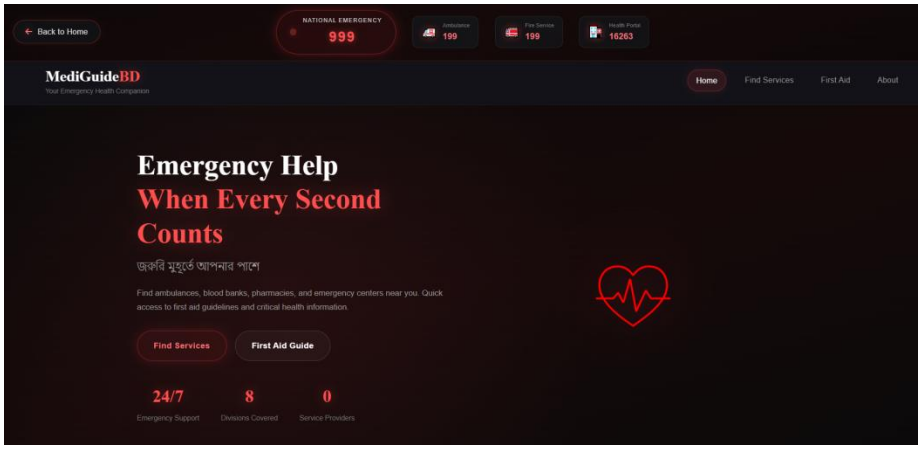


Fig 5.11: Emergency Search Page

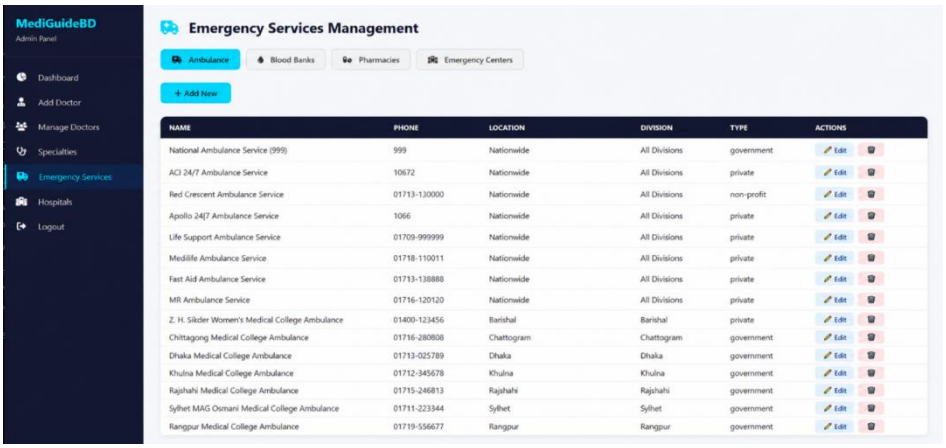


Fig 5.12: Emergency Ambulance Management Page

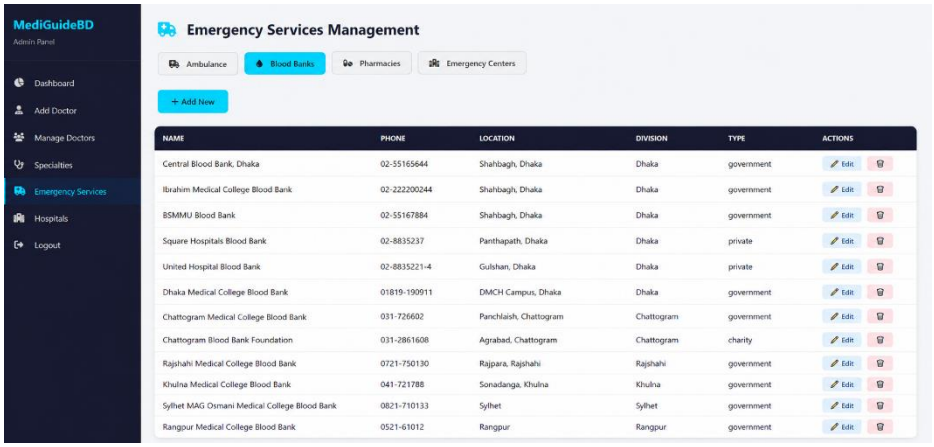


Fig 5.13: Emergency Blood Bank Management Page

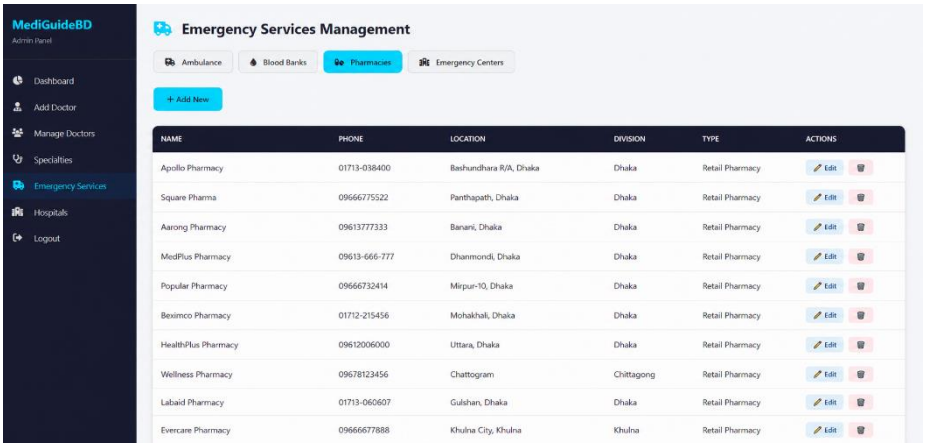


Fig 5.14: Emergency Pharmacy Management Page

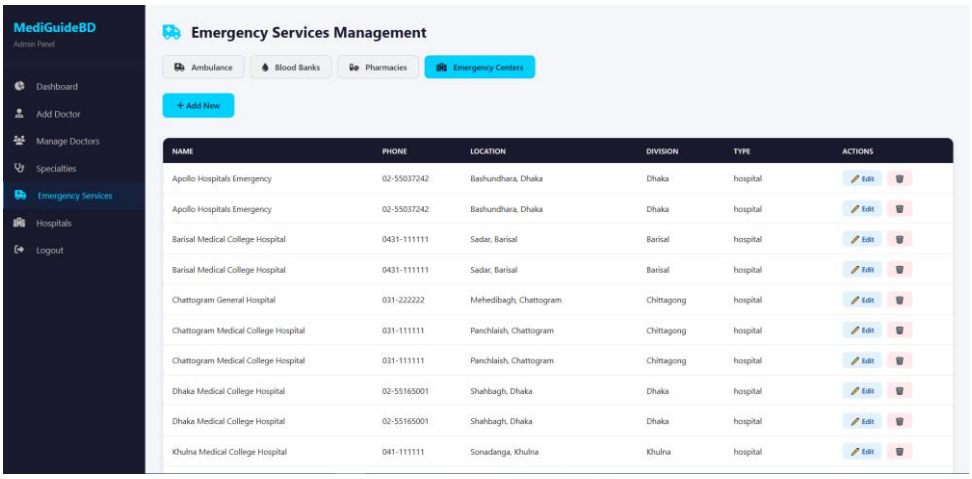


Fig 5.15: Emergency Center Management Page

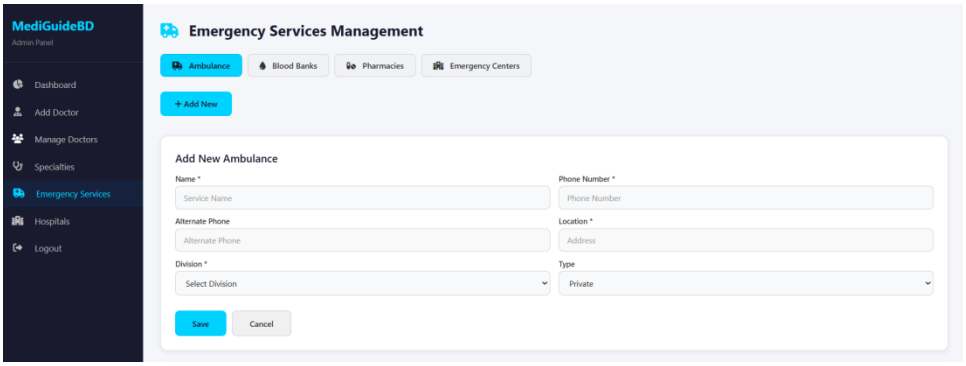


Fig 5.16: Ambulance Add Page

(We can add Blood Bank, Pharmacies and Emergency Centers too)

➢ 5.1.2 Responsive Design Strategy

All pages were built using a mobile-first approach. CSS [4] media queries adjust the layout for different screen sizes—rearranging columns into single-column layouts on mobile, adjusting font sizes, and ensuring touch targets remain accessible. The sidebar collapses and navigation adapts to smaller screens, providing an optimal experience whether users are on a desktop computer, tablet, or smartphone.

5.2 Backend Implementation

➢ 5.2.1 Server Architecture

The backend of MediGuideBD is developed using Node.js [6] and Express.js [7], providing the server-side logic and API services of the system. It uses middleware for request processing, cross-origin communication, and error handling, while routes are organized into separate modules for authentication, doctor management, hospital information, emergency services, and symptom analysis. The server operates on port 3000 and supports efficient communication between the frontend and database. Security controls are implemented following established application security practices [11].

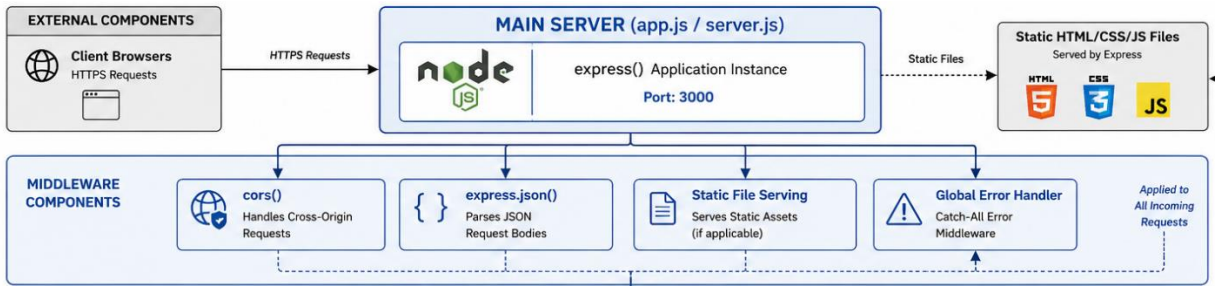


Fig 5.17: MediguideBD Backend Components

➢ 5.2.2 RESTful API Design

All endpoints follow RESTful conventions [9], [10]:

A. Authentication APIs

Method	Endpoint	Description
POST	`/api/auth/register`	Register new user account
	`/api/auth/login`	Authenticate and return user data

Table 5.1 : Authentication APIs

The login process queries the database for the provided email, verifies the password, and returns user information to the frontend. Role detection determines where the user will be redirected after successful login.

B. Doctor Search APIs

Method	Endpoint	Description
GET	`/api/doctors/symptoms`	Get symptoms for a body part
	`/api/doctors/analyze`	Intelligent symptom analysis
	`/api/doctors`	Get all doctors with filters
	`/api/doctors/specialties`	Get all medical specialties
	`/api/doctors/divisions`	Get all divisions of Bangladesh

Table 5.2 : Doctor Search APIs

The symptom analysis algorithm follows a three-step process:

- **Specialty Matching** - Queries the symptom-specialty mapping table using the problem keyword and body part, ordering by weight to find the most relevant specialty
- **Doctor Discovery**- Finds all doctors belonging to that specialty, filtered by the user's division
- **Availability Check** - Examines each doctor's time slots to determine if they are available now, today, or in the future, then sorts results with "Available Now" first, followed by "Today," then "Future"

C. Doctor Profile APIs

Method	Endpoint	Description
GET	`/api/doctor-profile/:userId`	Get full doctor profile
PUT	`/api/doctor-profile/:userId`	Update profile information
GET	`/api/doctor-profile/:userId/schedule-slots`	Get all time slots
POST	`/api/doctor-profile/:userId/schedule-slots`	Save time slots for a date
DELETE	`/api/doctor-profile/:userId/schedule-slots`	Remove all slots for a date

Table 5.3 : Doctor Profile APIs

D. Hospital & Disease APIs

Method	**Endpoint**	**Description**
GET	`/api/hospitals/diseases`	Get all diseases
	`/api/hospitals/disease-suggestions`	Autocomplete disease names
	`/api/hospitals/search-by-disease`	Find hospitals for a disease
	`/api/hospitals/hospitals`	Get all hospitals
POST	`/api/hospitals/hospitals`	Add new hospital
	`/api/hospitals/hospitals/:id/specialties`	Map hospital to specialty

Table 5.4 : Hospital and Disease APIs

E. Emergency Services APIs

Method	**Endpoint**	**Description**
GET	`/api/emergency/ambulance`	Get all ambulance services
POST	`/api/emergency/ambulance`	Add new ambulance service
PUT	`/api/emergency/ambulance/:id`	Update ambulance service
DELETE	`/api/emergency/ambulance/:id`	Delete ambulance service

Table 5.5 : Emergency Services APIs

All emergency modules (blood-banks, pharmacies, emergency-centers) follow the same pattern.

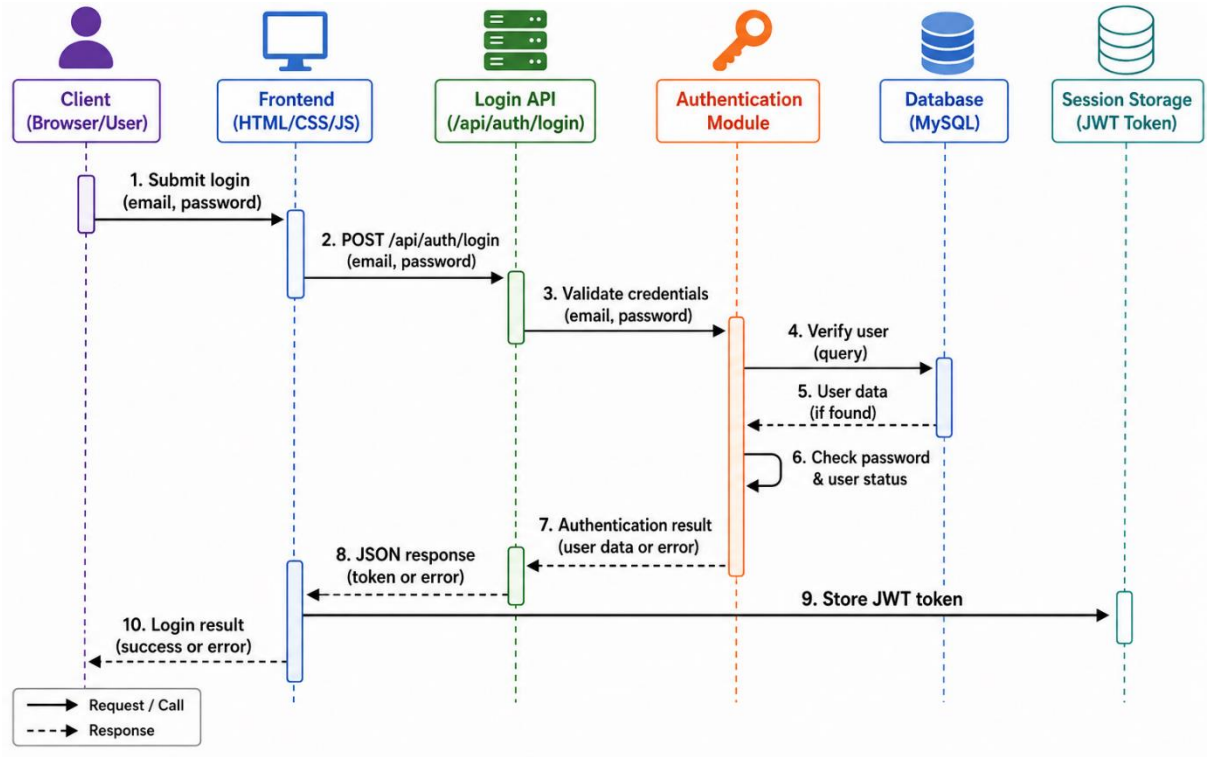


Fig 5.18: MediguideBD Login API Sequence Diagram

➢ 5.2.3 Database Connection

The database connection is established using the MySQL2 driver. The connection configuration specifies the host, user, password, and database name. Upon successful connection, the server logs confirmation. The connection is exported as a module and used by all route handlers to execute queries and retrieve data.

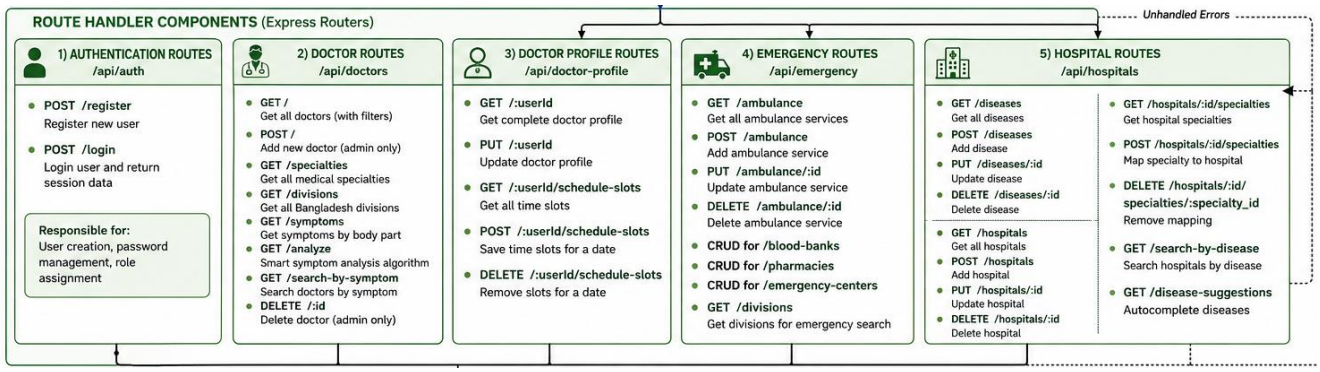


Fig 5.19: MediguideBD Backend Route Handler Components

5.3 Database Implementation

➢ 5.3.1 Database Schema Design

The database was designed using MySQL with a normalized structure. The Entity-Relationship diagram includes:

Core Tables:

1. **users** - Authentication and role management

2.**doctors -** Doctor profiles and contact information

3. **specialties-** Medical specialty reference

4. **divisions**- Bangladesh administrative divisions

5. **symptom_specialty_map -** Weighted symptom-to-specialty mapping

Supporting Tables:

6. **doctor_time_slots** - Availability schedule

7. **diseases** - Disease names and descriptions

8. **hospitals**- Hospital information

9. **hospital_specialties** - Hospital-specialty mapping with ranks

Emergency Tables:

10. **ambulance_services**

11. **blood_banks**

12. **pharmacies**

13. **emergency_centers**

➢ 5.3.2 Table Creation SQL

Each table was designed with appropriate data types and constraints:

The **user** stable uses auto-incrementing IDs as primary keys, with unique email constraints to prevent duplicate accounts. The role field uses the ENUM data type to restrict values to user, doctor, or admin.

The **doctors** table establishes foreign key relationships to users (for account linking), specialties (for medical specialty), divisions (for location), and users again (for tracking who created the record). The registration number is unique to prevent duplicate doctor entries.

The **symptom_specialty_map** table uses weight values to prioritize more relevant matches. Higher weights indicate stronger associations between symptoms and specialties.

The **doctor_time_slots** table uses a unique constraint on the combination of doctor ID, date, start time, and end time to prevent duplicate slots.

The **hospital_specialties** table maps hospitals to specialties with ranking levels. The rank_level field uses ENUM with values 'best', 'good', and 'standard', with a display-friendly rank_text field for the user interface.

➢ 5.3.3 Initial Data Population

The database was populated with reference data to enable immediate functionality:

Specialties: Twenty medical specialties were inserted including Cardiology, Dermatology, Neurology, Orthopedics, Pediatrics, ENT, Ophthalmology, Gastroenterology, Dentistry, Gynecology, Urology, Pulmonology, Psychiatry, Rheumatology, Endocrinology, Nephrology, Hematology, General Surgery, Plastic Surgery, and Vascular Surgery.

Divisions: All eight divisions of Bangladesh were inserted: Dhaka, Chittagong, Rajshahi, Khulna, Barisal, Sylhet, Rangpur, and Mymensingh.

Symptom-Specialty Mapping: Over 200 symptom mappings were created with weights. For example, "Chest Pain" maps to Cardiology with a weight of 3, "Headache" maps to Neurology with a weight of 3, and "Shortness of Breath" maps to Pulmonology with a weight of 3. Each entry includes the symptom keyword, body part, specialty ID, and weight value.

5.4 User Module Implementation

➢ 5.4.1 General User Module

The general user module provides three distinct search functions:

Symptom-Based Search:
Users complete the five-step form with body part, symptom, duration, side, and division. The system sends the data to the analysis endpoint and displays the recommended specialty and available doctors with their basic information and time slots.

Disease-Based Search:
Users enter a disease name using autocomplete and select a division. The system retrieves specialized hospitals and displays their ranking, name, contact details, address, rating, and calling/direction options.

Emergency Search:
Users select an emergency service and division. The system retrieves relevant services such as ambulances, blood banks, pharmacies, and emergency centers, displaying their location and contact information. Searchable first-aid guides are also provided in English and Bengali.

➢ 5.4.2 Doctor Module

Login & Session Management:
When a doctor submits the login form, the system authenticates their credentials through the API. Upon successful authentication, the user's ID, role, and name are stored in the browser's session storage. The user is then redirected to the doctor dashboard. Throughout the session, these stored values are used to identify the user and make authenticated requests. Role-based access control is enforced to protect sensitive functions and data [11].

Profile Management:
Doctors can view their complete profile information including name, specialty, registration number, email, hospital, division, address, and phone. The edit profile form allows updating hospital name, phone number, and address. Submitting the form sends a PUT request to the profile endpoint, which updates the database and confirms success.

Schedule Management:
The schedule page displays a monthly calendar. Each day cell shows the date number and, if slots exist, the slot count. Clicking a future date opens a modal where the doctor can manage time slots for that day. The modal displays existing slots with start and end times, allows adding new slots, and provides buttons to save changes or clear all slots. Saving sends a POST request that replaces all slots for that date, while clearing sends a DELETE request.

➢ 5.4.3 Admin Module

Doctor Verification & Management:
Administrators can view all doctors in a table and perform CRUD operations. Adding a doctor requires entering full name, email, password, registration number, specialty, division, hospital, address, and phone. The admin creates both a user account and a doctor profile in a single operation. Deleting a doctor removes both the doctor profile and the associated user account.

Emergency Service Management:
The admin panel provides tabs for each emergency service type. Administrators can add new services with appropriate fields—for example, ambulance services require service name, phone number, alternate phone, location, division, type, and availability. Blood banks require bank name, phone, email, location, division, type, blood types available, operating hours, and emergency availability. Similar forms exist for pharmacies and emergency centers. All services are displayed in tables with edit and delete options.

Hospital-Disease Mapping:
Administrators can map hospitals to diseases with ranking levels. When viewing a hospital, admins can see all mapped diseases and their rankings. The mapping interface displays a checklist of all diseases; selecting a disease and setting a rank (Best, Good, Standard) creates the mapping in the hospital_specialties table. This ranking system powers the disease-based search feature, ensuring patients see the most specialized hospitals first.

Chapter 6

Testing, Results, and Demonstration

6.1 Testing Methodology

MediGuideBD is an online healthcare guidance platform designed to help people find an appropriate specialist, doctor, hospital, or emergency service without needing to understand medical terminology first. The testing of this project therefore focuses not only on individual functions, but also on whether a person can move from a simple health concern to useful healthcare information in a clear and practical way.

A major testing goal is to verify the complete user journey. A user may know the name of a disease, or may know only where the problem is and what symptom is being experienced. The system should support both situations. In the symptom-based route, the user selects a body part and symptom, the system maps the symptom to a relevant specialty, and doctors in the selected division are presented with their hospital, address, phone number and availability information. In the disease-based route, the user searches for a known disease and receives specialized hospitals in the selected division.

The project source contains the frontend, Node.js/Express-style backend routes, MySQL database structure, doctor profiles, doctor time slots, disease and hospital relationships, and emergency-service records. The testing process is therefore organized into unit, integration, system and user-acceptance levels.

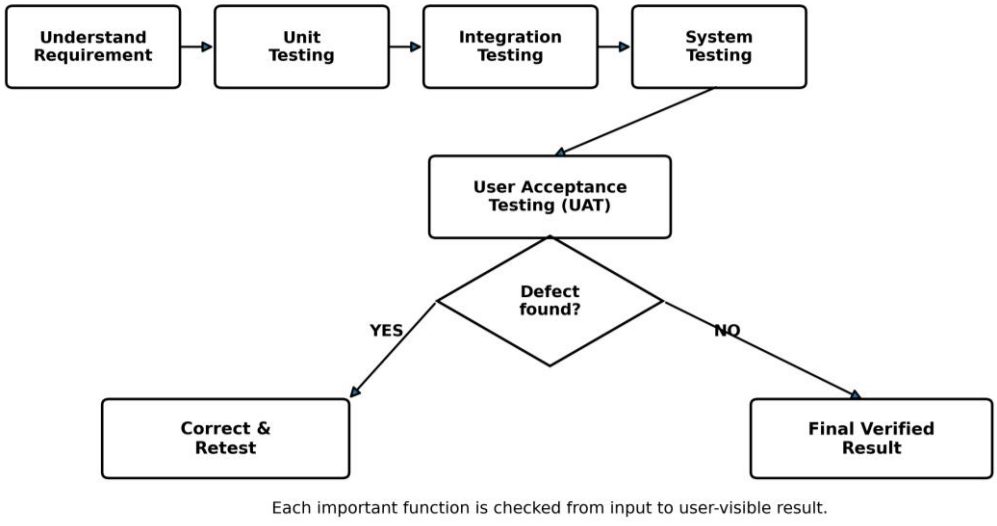


Figure 6.1: MediGuideBD Testing Lifecycle

- **Unit Testing:** Checks individual functions such as symptom retrieval, disease suggestions, doctor profile operations, time-slot operations, authentication and emergency-service retrieval.
- **Integration Testing:** Checks whether related modules work together, especially symptom → specialty → doctor, doctor → schedule → availability, and disease → specialty → hospital.

- **System Testing:** Checks the complete online workflow from the user interface through the backend and database to the final information shown to the user.
- **User Acceptance Testing (UAT):** Checks whether the platform is understandable for ordinary users, including users who may not know the medical name of their condition.

➢ 6.1.1 Test Environment

Environment Item	Project Configuration / Scope
Application	Online web-based healthcare guidance platform
Frontend	HTML, CSS and JavaScript
Backend	Node.js with Express-style routing
Database	MySQL database used by the supplied project
Main data	Users, doctors, specialties, symptoms, diseases, hospitals, divisions, doctor time slots and emergency services
User areas	General user, doctor and administrator interfaces
Location scope	Division-based filtering is implemented in the supplied backend; exact GPS-distance ranking is not established in the source package

Table 6.1: Test Environment

6.2 Test Cases and Results

The following test cases are written around the real purpose of MediGuideBD: helping a person move from a health concern to a suitable healthcare option. The status wording intentionally separates implementation evidence from independently observed runtime results. The supplied project contains the corresponding routes and processing logic, but it does not contain a complete recorded execution log for every case.

Test ID	Scenario / User Action	Expected Result	Implementation Result
TC-01	User does not know the disease; selects body part and symptom.	System identifies a supported matching specialty.	Supported by /doctors/analyze and symptom-specialty mapping.
TC-02	User selects a division after symptom search.	Doctors for the matching specialty are filtered to the selected division.	Supported by specialty and division filtering in doctors.js.
TC-03	User views a doctor result.	Doctor name, specialty, hospital, address, phone and registration information are displayed when available.	Supported by doctor result fields and frontend doctor cards.
TC-04	User checks a doctor at the current date/time.	System identifies whether the doctor is available now and shows the current schedule.	Supported by doctor_time_slots and calculateAvailability().
TC-05	No doctor is available at the searched time.	System looks for a future slot and shows the next available date/time when a future slot exists.	Supported by next-available logic in doctors.js.
TC-06	User wants to contact a doctor for an appointment.	Doctor phone/contact information is available so the user can contact the doctor directly.	Supported by doctor phone field and tel/contact display; appointment booking itself is not a separate booking module in the supplied source.

TC-07	User already knows the disease and selects a division.	Specialized hospitals associated with the disease are listed and ranked.	Supported by /search-by-disease and hospital-specialty/disease relationships.
TC-08	User selects a hospital result.	Hospital name, address, division and map/contact information help the user reach the facility.	Supported by hospital result page and map-link generation.
TC-09	User needs urgent support.	Ambulance, blood bank, pharmacy or emergency-center records can be retrieved by division.	Supported by emergency.js retrieval routes.
TC-10	Doctor logs in and manages profile.	Allowed doctor profile information can be viewed and updated.	Supported by authentication and doctorProfile.js.
TC-11	Doctor creates or deletes availability slots.	Date-specific schedule slots are stored, updated or removed.	Supported by doctor profile schedule routes.
TC-12	Administrator manages directory records.	Core doctor, hospital and emergency records can be maintained through administrative functions.	Backend CRUD routes are present; final role restriction should be runtime-verified.

Table 6.2: Scenario and Test Result

➢ 6.2.1 Functional Test Result Discussion

The testing analysis shows that MediGuideBD supports two main search paths. Users who do not know their disease can enter a body part and symptom to receive a recommended specialty and doctors filtered by division. Users who already know the disease can search for specialized hospitals by disease and division.

Doctor availability is checked using stored time slots, allowing the system to show current or future availability. Users can view doctor schedules and contact information for appointments.

The emergency module provides division-based access to ambulance, blood bank, pharmacy, and emergency-center information.

6.3 Demonstration

The demonstration is organized around a realistic user journey. The strongest demonstration is not a list of isolated screens; it is the story of how a person with little medical knowledge can use the website to move from a problem or symptom to a specialist, doctor, hospital and contact information.

➢ 6.3.1 System Walkthrough

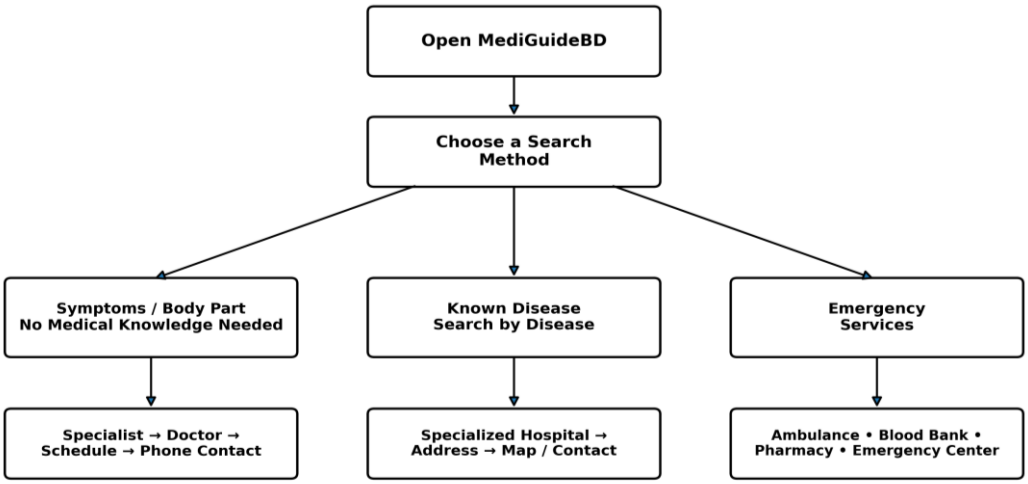


Figure 6.2: MediGuideBD User Journey

Step 1 – Open the online platform. The user opens MediGuideBD and chooses the required healthcare guidance option.

Step 2 – Start without knowing the disease name. If the user does not know the disease, the user can select the affected body part and then select a supported symptom. This reduces the need for prior medical knowledge.

Step 3 – Select the user area. The user selects a division so that the returned doctor or hospital information is relevant to the chosen area.

Step 4 – Get the relevant specialist. The system matches the symptom to a medical specialty using the predefined symptom-specialty relationship and weight.

Step 5 – Find a suitable doctor. Doctors belonging to that specialty and selected division are displayed with hospital, address, phone and registration information when available.

Step 6 – Check the doctor schedule. The system reads stored time slots and shows whether a doctor is available now or when the next available slot is expected.

Step 7 – Contact the doctor. The user can use the displayed phone/contact information to communicate with the doctor and arrange an appointment. The supplied project does not claim a separate online booking transaction.

Step 8 – Search by disease when the disease is already known. The user can instead search for a known disease and receive specialized hospitals in the selected division, with address and map/contact support.

Step 9 – Use emergency services when necessary. The emergency area provides access to ambulance, blood bank, pharmacy and emergency-center information.

Step 10 – Doctor and administrator functions. Doctors can manage permitted profile and schedule information, while administrators maintain core directory records.

➢ 6.3.2 Key Demonstration: From Symptom to Doctor

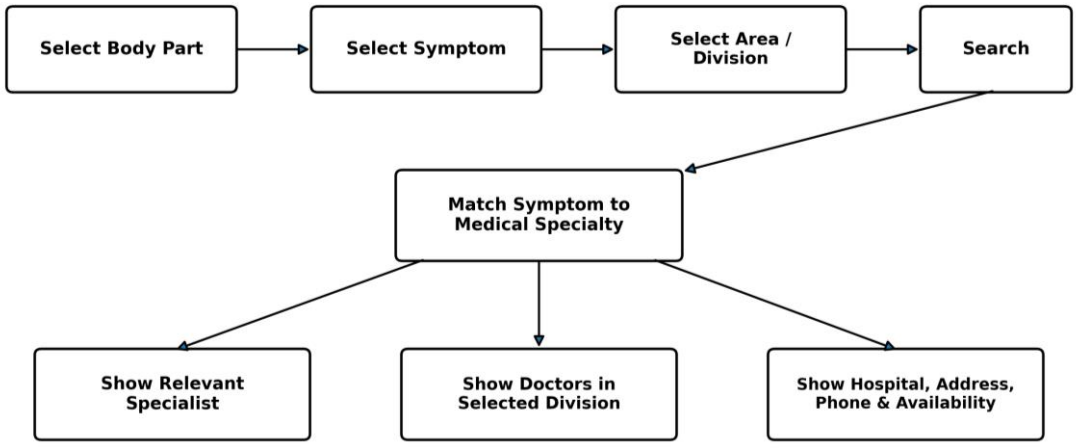


Figure 6.3: Symptom-Based Specialist and Doctor Selection

This is a key demonstration path of MediGuideBD. Users can select a body part and symptom without knowing the disease name. The system identifies a suitable specialty, filters doctors by division, and displays their workplace, contact information, and availability.

➢ 6.3.3 Doctor Availability and Appointment Contact

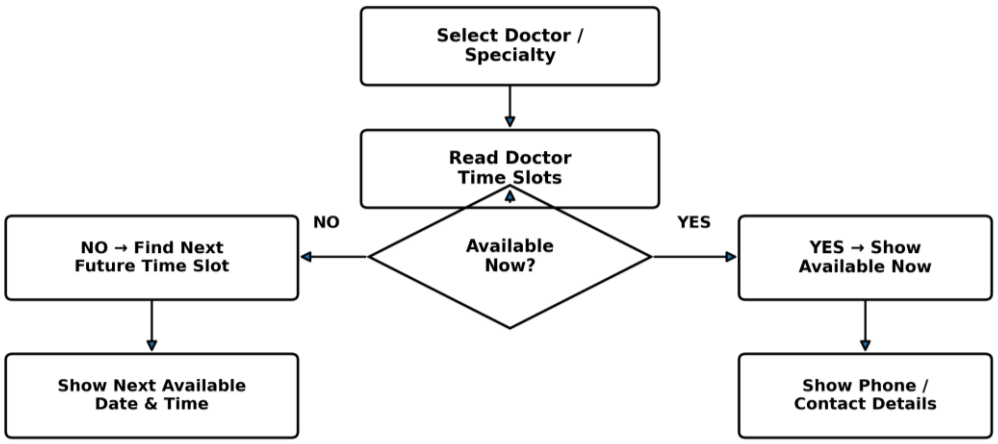


Figure 6.4: Doctor Availability and Next-Available Logic

Doctor availability is checked against stored time slots. The system shows whether a doctor is available now or identifies the next available slot. Users can also view the schedule and contact the doctor using the provided phone number.

➢ 6.3.4 Disease-Based Hospital Search

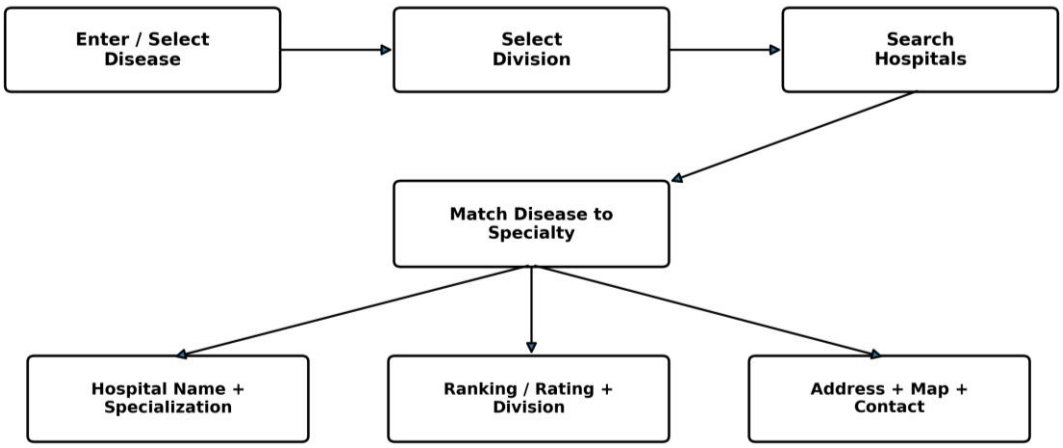


Figure 6.5: Disease-Based Specialized Hospital Search

The disease-search route is designed for users who already know the disease or condition they are looking for. The user enters or selects the disease and chooses a division. The backend connects the disease with its specialty and the available hospital-specialty relationships. The result can show the hospital name, specialization/ranking information, division, address and map/contact support. This allows a patient or family member to identify an appropriate hospital before travelling.

➢ 6.3.5 Emergency Services Demonstration

For urgent situations, MediGuideBD provides a separate emergency information path. The supplied emergency module contains structured records for ambulance services, blood banks, pharmacies and emergency centers. The frontend provides direct phone actions for service records. These records should be kept current and verified by the administrator because incorrect emergency information could be harmful.

6.4 Results Summary

The testing review confirms that the project source fully supports the platform's central healthcare-guidance workflow, including the symptom-to-specialty analyzer, division-based doctor filtering, time-slot availability and next-available logic, disease-to-hospital search, doctor schedule management, authentication routes, and division-filtered emergency-service retrieval.

From a user perspective, the key value is the ability to start with a symptom rather than a diagnosis, which benefits users unfamiliar with medical terminology, and then receive guidance toward a relevant specialist and doctor. For users who know their disease, the disease-search route offers a separate path to specialized hospitals, while doctor schedules and contact details help inform care decisions. The project is a guidance and referral platform only—it does not diagnose or treat disease, so symptom-module results should be interpreted as specialty guidance, not medical diagnosis.

6.5 Testing Considerations and Limitations

- The supplied implementation uses division-level location filtering. It does not establish exact GPS-distance or nearest-hospital ranking from the source package.
- The project source does not include a complete automated test suite or a complete recorded runtime execution report for every test case.
- The supplied source supports doctor phone/contact information, but a separate online appointment-booking transaction is not established in the source package.
- Symptom-to-specialty matching is based on predefined database relationships and weights; unsupported or ambiguous symptoms may not produce a useful match.
- Doctor availability depends on the accuracy of stored time-slot data and the date/time supplied to the search process.
- Healthcare directory usefulness depends on the accuracy and freshness of doctor, hospital and emergency-service records.
- Authentication and role restrictions should be verified again in the final deployed environment before production use.
- Emergency contact information should be verified and maintained carefully because incorrect information can be harmful during urgent situations.

Chapter 7

Conclusion & Future Scope

7.1 Conclusion

MediGuideBD is a unified web platform designed to simplify access to healthcare services in Bangladesh by enabling users to find doctors, hospitals, and emergency support in one place. It offers symptom-based doctor search—users select body area, duration, location, and symptoms, and the system suggests the relevant specialist along with real-time availability. Additionally, a disease-based hospital search ranks nearby facilities as Best, Good, or Standard, helping users compare options. The platform also includes an emergency section with ambulances, blood banks, pharmacies, and hotlines for rapid assistance. Supporting both English and Bengali, the system ensures accessibility for a wide range of users.

The platform is built with HTML, CSS, and JavaScript for the frontend, while Node.js, Express.js, and MySQL power the backend and data storage. Key supporting features include a Doctor Dashboard for schedule management, an Admin Panel for system oversight, and user registration with profile and favorites management. Throughout development, we learned how frontend, backend, APIs, and databases integrate to form a functional system, and we gained practical insights into design, authentication, and data relationships. Most importantly, we recognized that maintaining accurate and up-to-date healthcare information is essential for user trust and safety. MediGuideBD thus demonstrates how a well-structured digital tool can reduce confusion, save time, and improve healthcare decision-making for patients across Bangladesh.

7.2 Future Scope

Although MediGuideBD provides several healthcare guidance features, the system can be further improved to make it more convenient, reliable, and useful for users. The following features can be added in the future:

- Online Doctor Appointment
- GPS-Based Nearby Search
- Mobile Application
- Improved Symptom Matching and AI Assistance
- Healthcare Data and Verification
- Emergency Service Improvement
- User Reviews and Notifications.
- Improved Language Support and Security
- Advanced Admin Panel

REFERENCES

[1] World Health Organization, "Advancing Digital Health Strategy in Bangladesh," WHO Bangladesh, Jan. 3, 2021. [Online]. Available: https://www.who.int/bangladesh/news/detail/03-01-2021-advancing-digital-health-strategy-in-bangladesh. [Accessed: Sep. 01, 2026]..

[2] World Health Organization, "Bangladesh initiates Digital Health Strategy," WHO Bangladesh, Oct. 17, 2019. [Online]. Available: https://www.who.int/bangladesh/news/detail/17-10-2019-bangladesh-initiates-digital-health-strategy. [Accessed: Sep. 01, 2026]..

[3] World Wide Web Consortium (W3C), "HTML: The Living Standard," WHATWG, 2026. [Online]. Available: https://html.spec.whatwg.org/. [Accessed: Sep. 01, 2026]..

[4] Mozilla Developer Network, "CSS: Cascading Style Sheets," MDN Web Docs, 2026. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/CSS. [Accessed: Sep. 01, 2026]..

[5] Mozilla Developer Network, "JavaScript," MDN Web Docs, 2026. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript. [Accessed: Sep. 01, 2026]..

[6] OpenJS Foundation, "Node.js Documentation," Node.js, 2026. [Online]. Available: https://nodejs.org/en/docs/. [Accessed: Sep. 01, 2026]..

[7] OpenJS Foundation, "Express.js Documentation," Express.js, 2026. [Online]. Available: https://expressjs.com/. [Accessed: Sep. 01, 2026]..

[8] Oracle Corporation, "MySQL 8.4 Reference Manual," MySQL Documentation, 2026. [Online]. Available: https://dev.mysql.com/doc/refman/8.4/en/. [Accessed: Sep. 01, 2026]..

[9] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Ph.D. dissertation, Dept. Inf. Comput. Sci., Univ. California, Irvine, CA, USA, 2000. [Online]. Available: https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm. [Accessed: Sep. 01, 2026]..

[10] R. T. Fielding and R. N. Taylor, "Principled design of the modern Web architecture," ACM Trans. Internet Technol., vol. 2, no. 2, pp. 115–150, May 2002. doi: 10.1145/514183.514185

[11] OWASP Foundation, "Application Security Verification Standard 3.0.1," OWASP, 2013. [Online]. Available: https://owasp.org/www-pdf-archive/OWASP_Application_Security_Verification_Standard_3.0.1.pdf. [Accessed: Sep. 01, 2026]..

[12] Healthline Media, "Healthline: Medical Information and Health Advice," Healthline, 2026. [Online]. Available: https://www.healthline.com/. [Accessed: Sep. 01, 2026]..

[13] WebMD LLC, "WebMD: Health Information and Medical Advice," WebMD, 2026. [Online]. Available: https://www.webmd.com/. [Accessed: Sep. 01, 2026]..

[14] Doctorola, "Doctorola: Online Doctor Appointment and Healthcare Services," Doctorola, 2026. [Online]. Available: https://www.doctorola.com/. [Accessed: Sep. 01, 2026]..

[15] Arogga, "Arogga: Online Pharmacy and Healthcare Platform," Arogga, 2026. [Online]. Available: https://www.arogga.com/. [Accessed: Sep. 01, 2026]..

[16] MedEasy, "MedEasy: Healthcare and Medicine Platform," MedEasy, 2026. [Online]. Available: https://www.medeasy.health/. [Accessed: Sep. 01, 2026]..

[17] Praava Health, "Praava Health: Healthcare and Diagnostic Services," Praava Health, 2026. [Online]. Available: https://www.praavahealth.com/. [Accessed: Sep. 01, 2026]..

[18] Samsung Electronics, "Samsung Health," Samsung Health, 2026. [Online]. Available: https://www.samsung.com/us/apps/samsung-health/. [Accessed: Sep. 01, 2026]..