Contents

Chapter 1

Introduction:

In Bangladesh, healthcare is a real hassle with all the paperwork, long queues for appointments in both govt and private hospitals and scattered medical records that make it tough for doctors and patients alike. Now time to modernize things. Our web-based system aims to make managing patient appointments and records smoother, cutting down on errors and wait times, especially in busy hospitals.

1.1 Purpose :

The purpose of this project is to design and implement a web-based Patient Appointment and Record Management System that simplifies the process of managing patient appointments, doctor schedules, and medical records in a hospital. Our system will remove manual record-keeping, reduces administrative work, and ensures easy access to patient and doctor information. It creates a platform where patients can book appointments online, doctors can view schedules and prescribe digitally, and admins can handle billing and records securely. It will eliminate the long line of patient in govt hospitals or private hospitals.

1.2 Scope :
We will built this web app from scratch ,It will show a lot of features for patients,doctors and make management easier.
Key benefits :

1 .Registering and managing patients and doctors for hospital by admin.

2. Managing and viewing patient medical records by admin for better service.

3. Online Test reports for patient.

4. Patients can book appointments online from home.

5. Easy access to personal medical records anytime for patient.

6. Doctors can check schedules, update records, and even do text messeging with patient.

Goal :

Modernize the whole medical system and improve patient service in all private and govt medical.

1.3 Intended Audience:

Hospital Administrators

Doctors

Patients

Support Staff

Developers ,Testers.

1.4 Intended Use:

Admin :This is the main team whom will continue to manage patient ,doctors and stuffs.They will manage billing and providing lab reports to patient.

Doctors : Views appointments and updates medical records. Check patient history and consult patient .So SRS will help to understand full functionality.

Patient: Patient will book appoint ,they log in to their own portals and they can book appointment,see their lab reports .

Developers: They can use this SRS to get a clear picture of the project, figure out what to prioritize, spot areas for improvements, or even think about adding cool or modern new features down the line.

Project Testers: This will help testers to test everything systematically—know exactly what to check for bugs or glitches based on the requirements.

1.5 Risk Definition :

1. Unstable internet could interrupt online bookings or record access.

2. High user traffic might overload the server if not handled well.

3. Data privacy issues if security isn't tight, especially with sensitive medical info

Chapter 2

Overall Description :

Medilink is a web-based platform that connects Patients, doctors, and administration in a similar digital environment. It provides appointment time, Different kinds of medical records medical reports to ensure a smooth healthcare service or delivery. This system is designed for removing administrative burdens and automating the process.

2.1. User Classes and Characteristics :

This system is made for four types of people who can access this platform.

Patients: Persons who seek medical appointments and online access to test results.

Doctors: Persons who give medical treatment and want access to the patient. They are registered practitioner who can manage consultation and update medical histories.

Administration : They can maintain medical records , oversee hospital operations, the medical schedule, etc.

Support Staff: They can assist with lab reports, appointment coordination , billing, etc.

2.2 User Need :

1. Patients need a platform where they can book medical appointments, check their schedule. They need a platform where they can access the record. They don't need to visit the hospital physically.

2. Doctors need an efficient way to view their appointment list, upcoming appointments, and access patient history, update patients' history before their upcoming consultation.

3. Administration needs that kind of platform or tool where they can manage hospital data, generate reports, and maintain the records correctly.

4. This system should manage waiting time, ensure data consistency. It should reduce paperwork, too.

2.3 Operating System :

1. Operating System: Any OS supporting modern browsers (Windows, macOS, Linux).

2. Database: MySQL.

3. Platform: Python with Django Framework or Flask .

4. Frontend: HTML, CSS, JavaScript,Tailwind.

5. Browser: Google Chrome, Mozilla Firefox, or Microsoft Edge.

2.4 Constraint :

1. This development must be finished within the academic timeline.

2. It must be implemented in a selected language.

3. This application must function correctly in a Windows client environment.

4. All sensitive information must be comply with privacy rules and be stored securely.

2.5 Assumption :

1. Users can read and understand English.

2. Users have basic knowledge about the internet and a suitable device.

3. Users have internet access.

4. Hospital staff have basic knowledge about computer usage.

5. The hospital will have suitable electricity and network connections.

Chapter 3 : Requirements

This chapter describes both the functional and non functional requirements of the Medilink platform.

Functional requirements specify the exact functionality that the system must provide, whereas non-functional requirements describe the system's performance, security, and usability standards.

3.1 Functional Requirement :

We have nine major functional requirements. Each two/three features will be implemented and tested separately by one team member.

 F1 :

As an Admin

I want to log in securely .so that, I can access the hospital management dashboard and control the system.

Confirmation:

1.As a Admin I must enter a valid username and password.

2.If credentials are correct, the system redirects to the admin dashboard.

3.If credentials are invalid, an error message "Invalid username or password" appears.

4. If the server or database is down, the system shows "Connection error."Or it will redirect me to the login page.

F2:

As an Admin

I want to add, edit, and delete doctor profil`es.I also handle appointment .

so that, I can keep hospital doctor and patient information accurate and up to date.

Confirmation:

1. Admin has to fill in the doctor's name, department, phone, and email,qualities.
2. On submission, the doctor's record is added to the database.
3. If the information is incomplete, the system alerts: "Please fill all required fields."
4. Admin can edit or delete doctor details when needed.
5. A confirmation message appears: "Doctor record updated successfully."

F3:

As a Doctor

I want to log in securely.so that, I can view my appointments and patient information.

Confirmation:

1. Doctor must enter a valid ID and password which will be given by Admin.

2. After successful login, the system displays the doctor's dashboard with scheduled appointments.

3. If login fails, a message "Invalid credentials" appears or failed login

4. If the doctor's account is inactive by admin , the message "Access denied" appears.

F4:

As a Doctor

I want to view all my upcoming appointments so that I can prepare.

so that, I can prepare for patient consultations efficiently.

Confirmation:

1. System shows appointment details including patient name, date, and time.

2. Doctor can filter appointments by date or patient name.

3. If no appointment exists, the message "No appointments found" appears.

4. Data is loaded from the MySQL database.

F5 :

As a Doctor

I want to add and update patient medical records

so that, future treatments can be based on accurate medical history.

Confirm :

1 .Doctor selects a patient and fills in diagnosis, prescription, and visit date.

2. Data is saved into the "medical_records" table.

3. Confirmation message appears: "Record saved successfully."

4. If any required field is empty, the system displays "Please enter mandatory details."

F6

As a Patient

I want to register myself online.

so that, I can create an account to book appointments and check medical history.

Confirmation:

1. Patient has to fill in name, age, gender, phone, email, and password.

2. On submission, data is saved in the "patients" table.

3. If the email already exists, the system shows "User already registered."

4. After successful registration, the patient can log in immediately.

F7

As a Patient

I want to log in to my account

so that, I can access my dashboard and view appointments or medical records.

Confirmation:

1. Patient enters correct email and password.

2. System verifies credentials and redirects to the patient dashboard.

3. If incorrect, "Invalid email or password" appears.

4. If not registered, "User not found" is displayed.

F8

As a Patient

I want to book an appointment with a doctor according to my need .

so that, I can receive treatment at my preferred time.

Confirmation:

1. Patient selects a doctor, date, and time slot from the available list.

2. The system checks for slot availability.

3. If available, appointment is confirmed and saved to the "appointments" table.

4. Confirmation message: "Appointment booked successfully."

5. If the slot is taken, the system shows "Selected time is unavailable."

6. If required fields are empty, the system shows "Please select doctor and date."

F9

As a Patient

I want to view my past appointments and medical records

so that, I can review my visit history and follow my doctor's advice.

Confirmation:

1. Patient dashboard displays all completed and upcoming appointments.

2. Each record includes doctor name, date, and diagnosis summary.

3. If no record exists, message: "No history available."

4. Data is retrieved from "appointments" and "medical_records" tables.

3.2 Non-Functional Requirements

Performance

The system should handle many users at once without crashing. All pages, such as login or appointment booking, must load within 3 seconds. Database searches and updates should respond quickly to give users a smooth experience.

Safety

The system must keep patient data safe from unauthorized access. Regular backups will prevent data loss during technical problems. The use of the system should not cause any harm or confusion to users.

Security

The system will use a secure MySQL database with encrypted passwords. Each user will have specific access rights — patients can only view their own data, doctors can view their patients' records, and admins can manage everything. All data transfers will be protected to keep patient information safe.

Error Handling

The system should handle all errors smoothly without losing data. If something goes wrong, like a failed database connection, users will get a clear error message. All major errors will be recorded for the admin to check later.

Usability

The interface will be simple and easy to use for both admin ,doctor and patients. It will work well on computers and mobile devices, and support both English and Bengali languages for local users.

Availability

The system should be available 24/7 with at least 99% uptime. If the system fails, it should recover within a few minutes, and daily backups will make sure no data is lost.