# Hospital Management System

# PROJECT REPORT

# Abstract

The Hospital Management System (HMS) is designed to streamline and automate various hospital operations, including appointment booking, doctor scheduling, and the maintenance of patient medical histories. This project aims to illustrate how a database management system (DBMS) can improve the efficiency and accuracy of these tasks, thereby enhancing patient care and administrative operations. This report details the design and implementation of a hospital management database system, developed as part of a DBMS course project.

# Introduction

Hospitals handle numerous interactions and operations daily, involving patients, doctors, and administrative staff. Efficiently managing these activities is critical for the smooth functioning of hospital operations. Traditional manual methods are often inefficient and error-prone, highlighting the need for an integrated system to manage hospital operations effectively. This project aims to develop a Hospital Management System that utilizes databases to manage tasks such as appointment booking, doctor scheduling, patient diagnoses, and maintaining medical histories.

# Methodology

## Technologies Used

VISUAL STUDIO

* **Frontend:** HTML BOOTSTRAP
* ASP.NET
* **Backend:** JavaScript
* **Database:** MySQL

## Query Used

**Patient To Register Account**

CREATE TABLE signup (

    Id INT PRIMARY KEY IDENTITY(1,1),

    Name NVARCHAR(255) NOT NULL,

    Email NVARCHAR(255) NOT NULL,

    Password NVARCHAR(255) NOT NULL

);

**Patient to Admit**

CREATE TABLE Patients\_reg (

    PatientID INT PRIMARY KEY IDENTITY,

    FullName NVARCHAR(100) NOT NULL,

    Gender NVARCHAR(10) NOT NULL,

    MedicalHistory NVARCHAR(MAX) NULL

);

**Doctor To Register**

CREATE TABLE doctors (

    DoctorID INT PRIMARY KEY IDENTITY(1,1),

    Name NVARCHAR(100) NOT NULL,

    Email NVARCHAR(100) NOT NULL UNIQUE,

    Password NVARCHAR(100) NOT NULL,

    Specialization NVARCHAR(100)

);

**Medical History**

CREATE TABLE medical\_history (

    PatientID INT PRIMARY KEY IDENTITY(1,1),

    Name NVARCHAR(100) NOT NULL,

    LastName NVARCHAR(100) NOT NULL,

    Gender NVARCHAR(50) NOT NULL,

    MedicalHistory NVARCHAR(MAX),

    DOB DATE NOT NULL

);

**Appointment**

CREATE TABLE Appointment (

    AppointmentID INT PRIMARY KEY IDENTITY(1,1),

    PatientID INT FOREIGN KEY REFERENCES medical\_history(PatientID),

    DoctorID INT FOREIGN KEY REFERENCES doctors(DoctorID),

    AppointmentDate DATE NOT NULL,

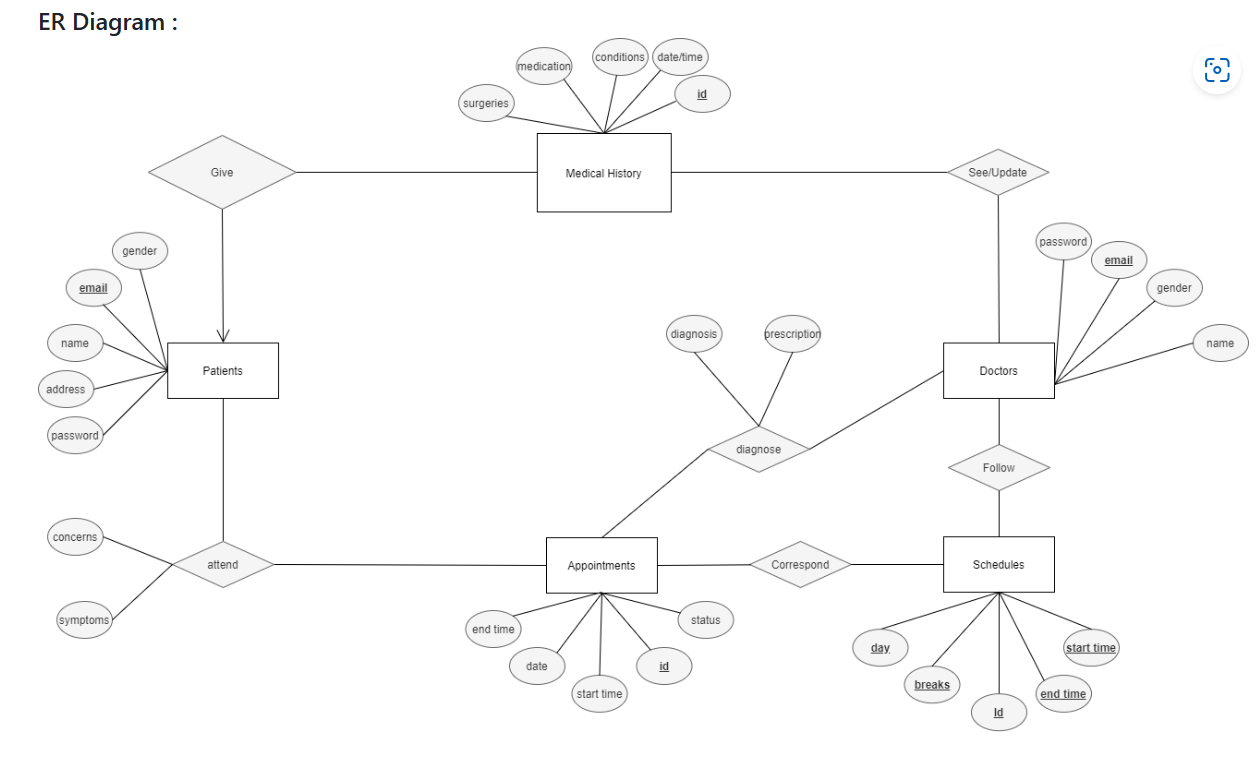
    AppointmentTime TIME NOT NULL

);

# ER Diagram

The Entity-Relationship (ER) diagram represents the entities involved in the Hospital Management System and their relationships. Key entities include:

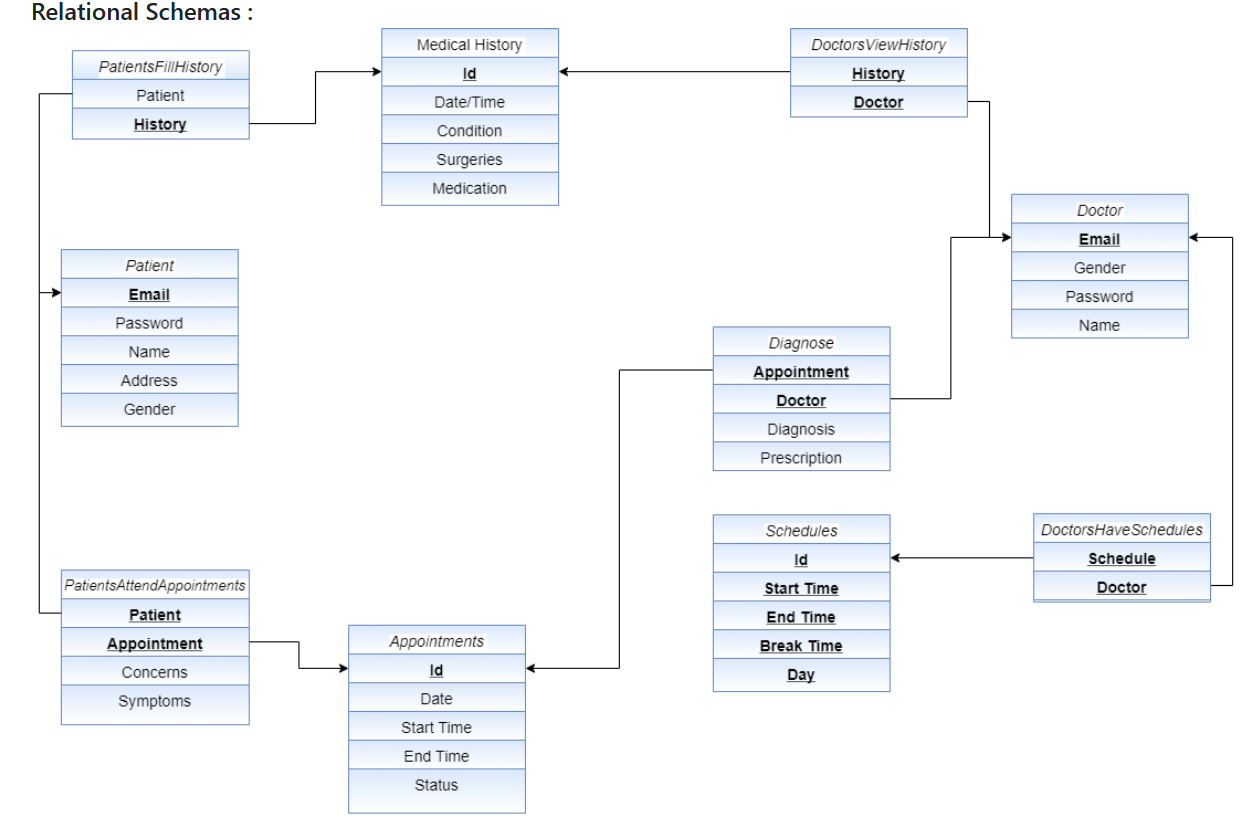
* **Patients:** Representing individual patients.
* **Doctors:** Representing individual doctors.
* **Appointments:** Managing the scheduling of patient visits.
* **Diagnoses:** Recording diagnoses given by doctors.
* **Medical Histories:** Keeping detailed records of patient medical histories.



# Relational Schemas

The relational schemas define the structure of the database tables, including attributes and their data types. Key tables include:

* **Patients (PatientID, Name, Age, Gender, MedicalHistoryID)**
* **Doctors (DoctorID, Name, Specialty, ScheduleID)**
* **Appointments (AppointmentID, PatientID, DoctorID, Date, Time, Status)**
* **MedicalHistories (MedicalHistoryID, PatientID, RecordDetails)**



# Patient Side Features

1. **User Interface:** Separate login interface for patients.
2. **Appointment Booking:** Patients can book appointments with doctors.
3. **Medical History:** Patients can provide and update their previous medical history.
4. **Appointment Management:** Patients can view, update, or cancel appointments as needed. Cancelled appointments create free slots for other patients.
5. **Medical Records Access:** Patients can view their complete diagnoses, prescriptions, and medical history.
6. **Privacy:** Medical history is accessible only to the doctor with whom the appointment is booked, ensuring patient privacy.

## Doctor Side Features

1. **User Interface:** Separate login interface for doctors.
2. **Schedule Management:** The system considers doctor schedules, preventing appointments when a doctor is busy or on a break.
3. **Patient History Access:** Doctors can access and update patient histories and profiles.
4. **Diagnosis and Prescription:** Doctors can provide diagnoses and prescriptions.
5. **Record Modification:** Doctors can modify diagnoses and prescriptions as needed.

## Discussion

The Hospital Management System addresses various challenges in managing hospital operations:

1. **Frontend:** ASP.NET provides a robust and user-friendly interface for both patients and doctors.
2. **Backend:** JavaScript ensures efficient server-side operations and data processing.
3. **Database:** MySQL offers a reliable and scalable solution for storing and retrieving medical records and other hospital data.

## Enhancements

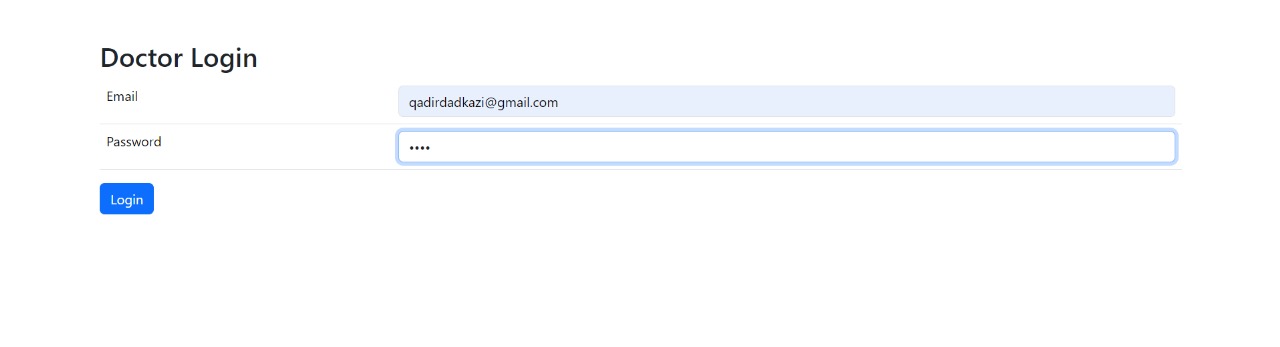
* **Patient Experience:** An easy-to-use platform for booking appointments and accessing medical records.
* **Doctor Efficiency:** Immediate access to patient histories and real-time updates of medical records.
* **Appointment Scheduling:** Automated scheduling minimizes appointment clashes and optimizes doctor availability.

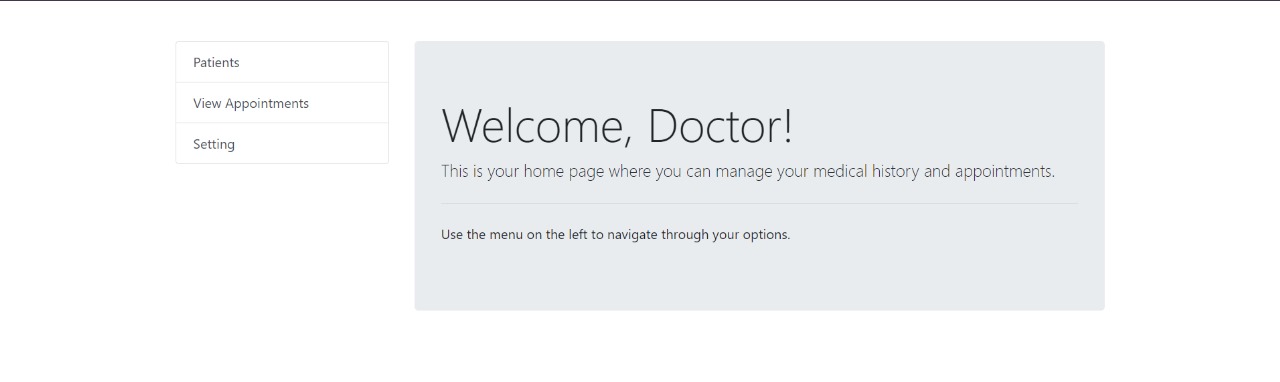
# Results

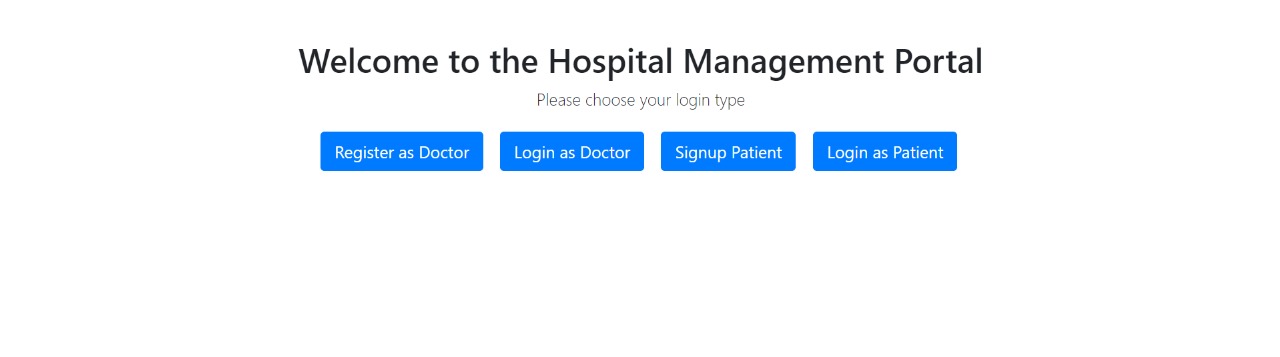
The implementation of the Hospital Management System resulted in significant improvements:

1. **Reduced Waiting Times:** Efficient appointment scheduling reduced patient waiting times.
2. **Enhanced Accuracy:** Improved accuracy in maintaining and retrieving patient medical histories.
3. **Improved Communication:** Better communication between patients and doctors.
4. **Administrative Efficiency:** Increased overall efficiency in hospital administration.

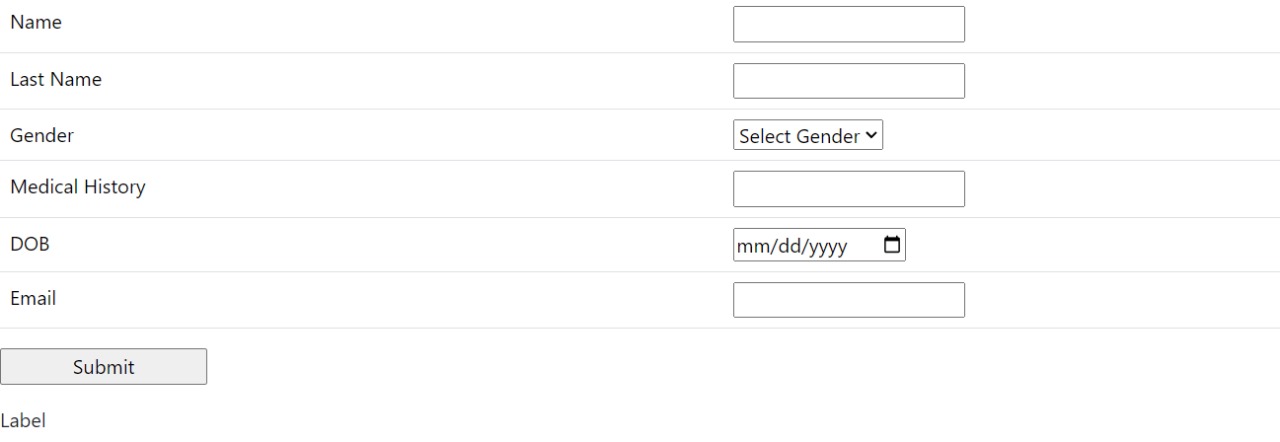
Output Screenshot

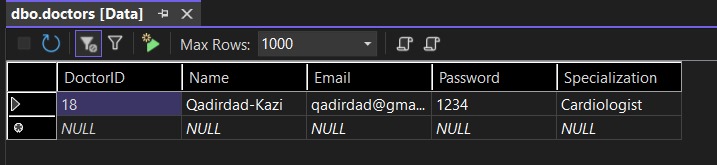


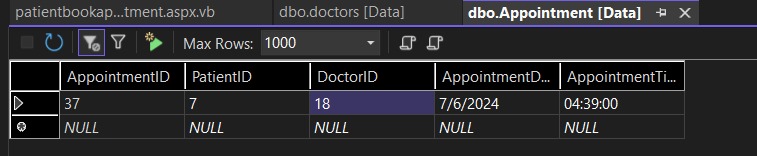


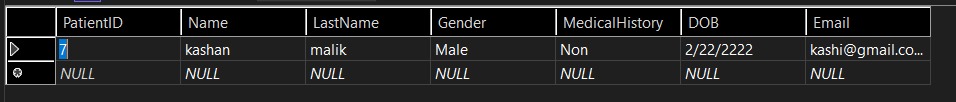


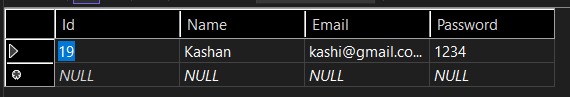


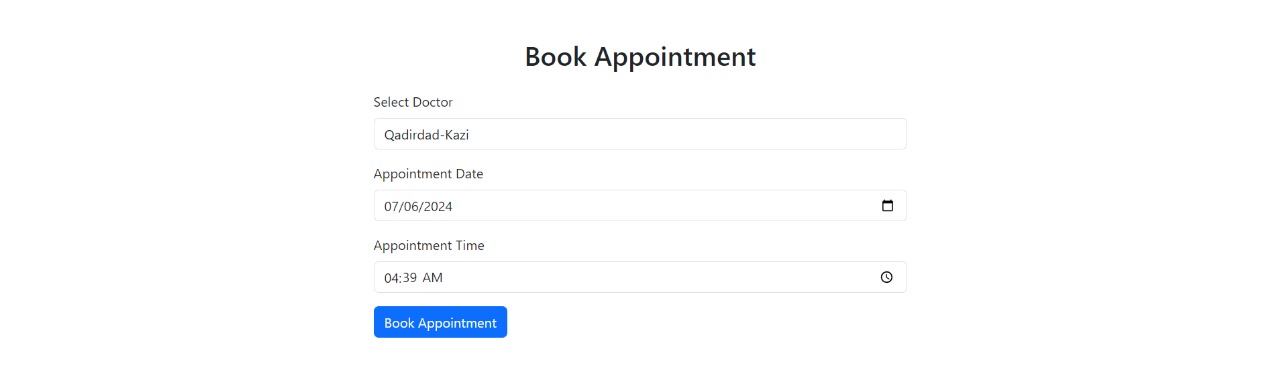


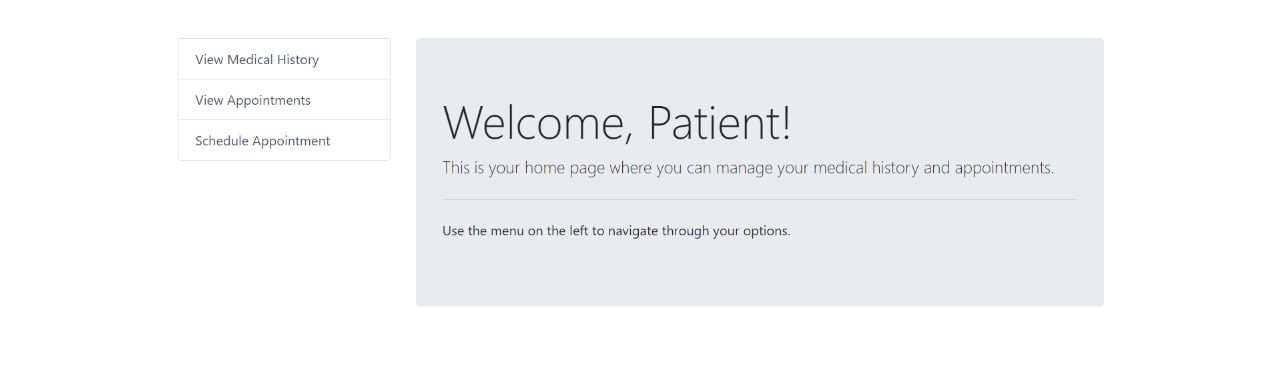


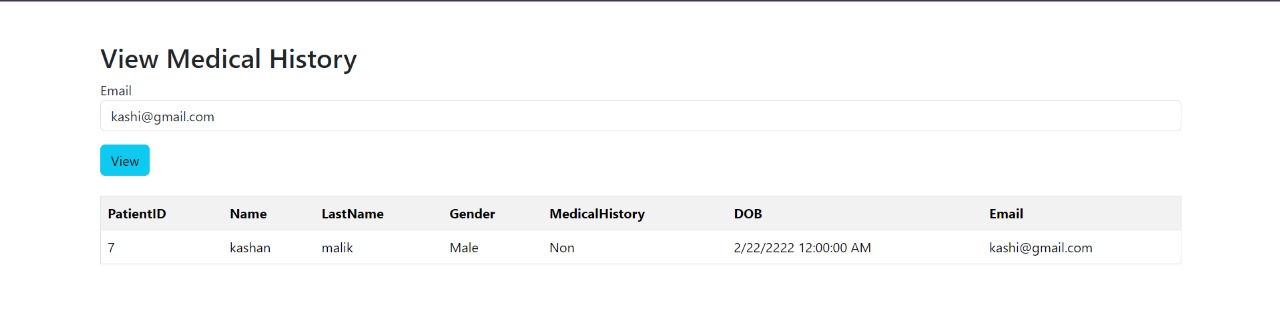


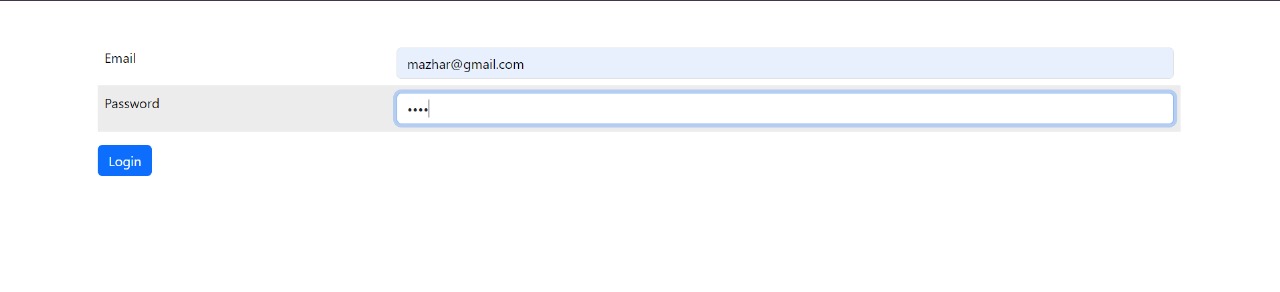


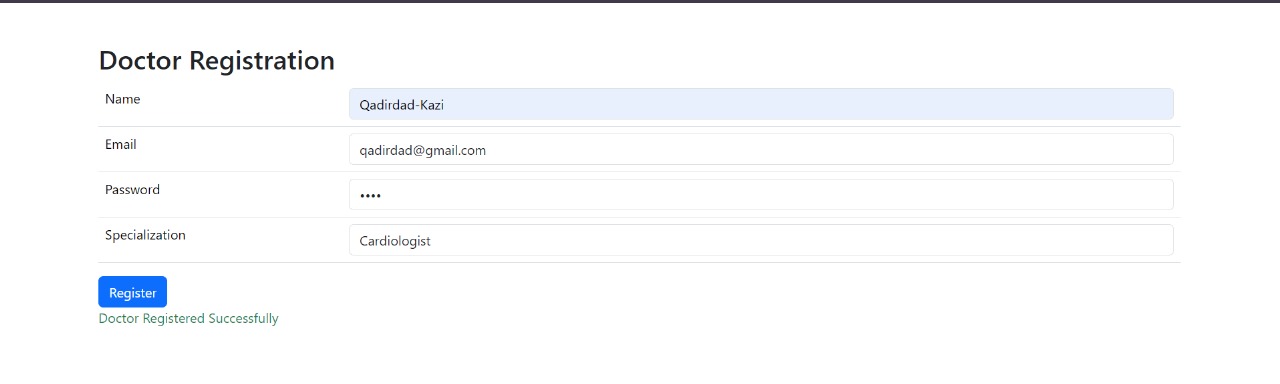


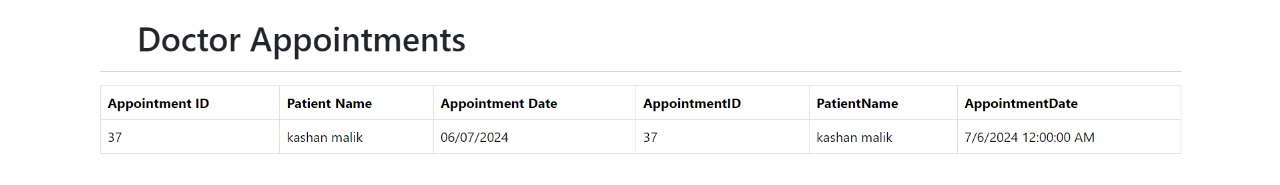


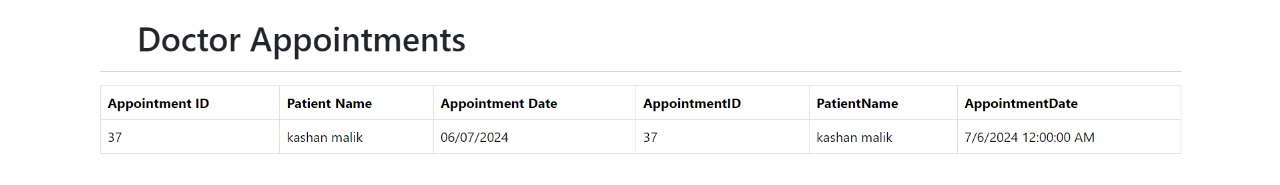












# Conclusion

The Hospital Management System demonstrates how leveraging databases and modern web technologies can significantly improve the efficiency and accuracy of hospital operations. By automating tasks such as appointment booking, doctor scheduling, and medical record management, the system enhances patient care and administrative processes. This project highlights the benefits of integrating database management systems into hospital management operations, serving as a valuable example for similar initiatives in the healthcare sector.