EAST WEST UNIVERSITY

Course: Database Systems

SQL Database Project
Project Report

Project Name: Healthcare Management System

Submitted by:

Md. Sajidur Rahman
Dept. of Computer Science & Engineering

Submitted To:
Dr. Mohammad Arifuzzaman

Associate Professor
Department of Computer Science and Engineering
East West University

Table of Contents:

1. Abstract:

Healthcare Management System is a DBMS (Database Management System) website-based software solution designed to various operational tasks within a one-stop solution in DBMS. This system provides an efficient platform for storing Structured Query Language (SQL) to manage patient's data and log in records, appointments, doctor's data and patient's appointment records and giving prescriptions as per analysis. It also saves the records in database.

2. Introduction:

Healthcare Management System is an integrated, high-performance system. today's ever-changing Healthcare landscape, managing patient's data, doctor's data, appointment information, administrative tasks, and more requires a robust and easy to use system. It combines the power of multiple technologies to deliver a seamless, interactive experience for the patient, the Healthcare professional, the administrator. The system ensures data integrity and minimizes redundancy. the Healthcare Management System represents a pivotal tool for Healthcare professionals, administrators, and patients alike. This project has the solution of evolving demands of the Healthcare industry, offering a foundation for efficient, secure, and patient centered Healthcare management. As technology continues to play a central role in Healthcare transformation. The system facilitates complex queries, reporting, and analytics, empowering Healthcare professionals to make informed decisions. The system is developed using a powerful combination of SQL, HTML, CSS, PHP, Bootstrap, and JavaScript, which represents an integrated solution for the efficient administration and organization of Healthcare related data.

3. Project Objectives:

Our objectives are improving Healthcare process, enhance efficiency, and provide better patient care. Here are several ways that it will beneficial for us:

• Utilizing the patient management features to maintain comprehensive records, including medical history, prescriptions, and appointments.

• Optimizing appointment scheduling using the system's capabilities, allowing for efficient allocation of resources and reduced wait times.

• Real-time updates to manage appointment changes or cancellations.

• Improving workflows by automating processes, such as prescription management.

• Generate customized reports for administrators and Healthcare providers.

• Ensure secure access to patient information by implementing user authentication and role-based access controls.

• Limit access to sensitive data based on user roles (Admin, Doctor, Patient).

• Taking advantage of the responsive design with Bootstrap, allowing Healthcare professionals to access the system from various devices.

• Enhancing user experience with a user-friendly interface designed using HTML, CSS, and JavaScript. Use SQL to maintain data integrity and enforce privacy standards.

• Ensure compliance with Healthcare regulations by implementing security measures in the system.

• Using the system to engage with patients through features like online appointment scheduling, secure messaging, and access to their own health records.

• Provide Healthcare professionals with remote access to patient data for timely decision-making. Regularly update and analyze system usage for improvement.

By effectively utilizing the features of the Healthcare Management System, Healthcare facilities can be improved overall with efficiency, data accuracy, and patient care, leading to efficient Healthcare management process.

4. Implementation:

Tools: XAMPP, MySQL, Localhost, PhpMyAdmin, Visual Studio Code, An Internet Browser.

Implementing Process:

To implement this SQL database system, we need some tools and programs, such as -

1. First, starting Apache & MySQL server on XAMPP. Then setting up to run on Browser.

2. Going to XAMPP directory where XAMPP is installed. Example. ("C:\xampp ")

3. Opening the folder named 'htdocs'.

4. Going to address bar and type "localhost/phpMyAdmin/"

5. Creating a database and inside that write SQL command to create a data table.

6. Opening a new folder over there named (Ex: HealthcareProject).

7. Going to web browser and typing on address bar: "localhost/HealthcareProject/"

8. Adding the required Html, CSS, PHP, Javascript, Bootstrap code in Visual Studio Code.

System Architecture:

Step 1: Database Design (SQL)

- Designing the database schema with tables for entities like Patient, Doctor, Admin, Appointments etc.
- Creating the database and tables using SQL commands or a database management tool in phpMyAdmin.

Step 2: Back-End Development (PHP)

- Creating a connection script to connect PHP to the database.
- Implementing PHP scripts for CRUD operations (Create, Read, Update, Delete) for each entity.
- Coding logic for appointment scheduling.

Step 3: Front-End Development (HTML, CSS, Bootstrap)

- Designing HTML pages for different functionalities (patient management, appointment scheduling, etc.).
- Applying CSS for styling and layout.
- Using Bootstrap for a responsive design and pre-built components.

Step 4: Front-End Interactivity (JavaScript)

- Enhancing user experience with JavaScript for form validations, dynamic content updates, and asynchronous requests.

Step 5: User Authentication (PHP, SQL)

- Implementing a secure user authentication system with PHP and SQL.
- Creating user roles (Admin, Doctor, Patient).
- Control access to different sections based on user roles.

Step 6: Integration

- Integrating the front-end and back-end components.
- Ensuring proper communication between the user interface and the server.

Step 7: Testing

- Conducting thorough testing, including functionality and usability testing.
- Identifying for any fix bugs or issues.

5. SQL Operations:

Database: These are our database tables created in “localhost/phpMyAdmin/”.

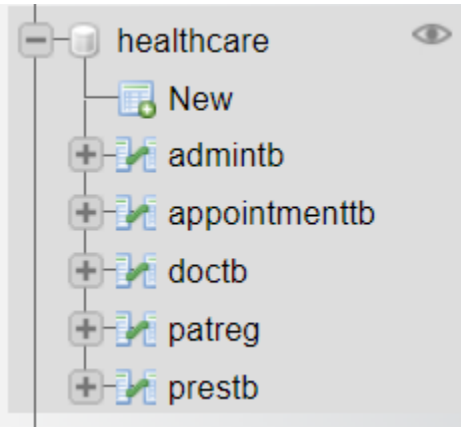


	Table	Action
☐	admintb	Browse Structure Search
☐	appointmenttb	Browse Structure Search
☐	doctb	Browse Structure Search
☐	patreg	Browse Structure Search
☐	prestb	Browse Structure Search
	5 table(s)	Sum

Admin’s Username And Password:

username	password
admin	admin123

Doctors Table:

username	password	doctorname	email	spec	docFees
Dr Sayem	password	Dr Sayem Chowdhury	Sayem@gmail.com	Gastroenterologist	350
Dr Sayeed	pass	Dr Abu Sayeed	Sayeed@gmail.com	General	300
Dr Asad	pass	Dr Asad Ahmed	Asad@gmail.com	Neurologist	400
Dr Shihab	pass	Dr Shihab Hasan	Shihab@gmail.com	Pediatrician	250
Dr Sifat	pass	Dr Sifat Uddin	sifat@gmail.com	Cardiologist	440

Patients Table:

pid	fname	lname	gender	email	contact	password	cpassword
1	Sabbir	Khan	Male	sabbir@gmail.com	012741000	pass	pass
2	Emran	Ahmed	Male	emran@gmail.com	012222222	123456	123456
3	Habib	Aslam	Male	rahmat@gmail.com	012744444	pass	pass
4	Amir	Hossain	Male	amir@gmail.com	012702454	pass	pass
5	Wasim	Al Zubaer	Male	wasim@gmail.com	012789984	pass	pass
6	Zarif	Haydar	Male	zarif@gmail.com	012753000	pass	pass
7	Yamin	Chowdhury	Male	yamin@gmail.com	012785000	pass	pass
8	Deehan	Iftekhar	Male	deehan@gmail.com	012730145	pass	pass
9	Hasibul	Hasan	Male	hasib@gmail.com	012702696	pass	pass
10	Tawhid	Akram	Male	tawhid@gmail.com	012790012	pass	pass

Appointment Table:

pid	ID	fname	lname	gender	email	contact	doctor	docFees	appdate	apptime
1	21	Sabbir	Khan	Male	sabbir@gmail.com	012741000	Dr Shihab	0	2023-12-18	10:00:00

Prescription Table:

doctor	pid	ID	fname	lname	appdate	apptime	disease	allergy	prescription
Dr Shihab	1	21	Sabbir	Khan	2023-12-18	10:00:00	No disease found.	No allergies found.	Eat nutritious food like fruit

6. Frontend and Backend Works:

Language used: SQL, HTML, CSS, PHP, JavaScript, Bootstrap.

Frontend Implementation:

Step 1: Design UI Layout (HTML, CSS, Bootstrap)

Designing HTML pages for different modules (for patient management, appointment scheduling). Applying CSS styles for layout and aesthetics. Utilizing Bootstrap for responsive design and pre-built components.

Step 2: Patient Management Module

Creating an HTML page for viewing and managing patient information. Implementing dynamic elements using JavaScript for interactive features.

Step 3: Appointment Scheduling Module

Developing an HTML page for appointment scheduling. Using Bootstrap's date and time picker components for a user-friendly interface. Adding JavaScript for dynamic content updates and form validations.

Step 4: Prescription Management Module

Designing an HTML page for prescription management. Using CSS to enhance the visual appeal. Adding JavaScript for real-time updates and interactive features.

Step 5: User Authentication (HTML, CSS)

Creating login and registration pages using HTML and CSS. Applying Bootstrap for consistent styling.

Backend Implementation:

Step 6: Set Up Database (SQL)

Designing the database schema with tables for patients, doctors, appointments, prescriptions, etc. Using SQL commands or a database management tool to create the database and tables.

Step 7: Establish Database Connection (PHP)

Creating a PHP script to connect to the database. Ensuring secure handling of database credentials.

Step 8: Implement CRUD Operations (PHP)

Writing PHP scripts for CRUD operations (Create, Read, Update, Delete) for each entity (patient, doctor, appointment, prescription).

Step 9: User Authentication (PHP, SQL)

Adding PHP scripts for user authentication and authorization. Adding secure password hashing for control access based on user roles (Admin, Doctor, Patient).

Step 10: Server-Side Processing (PHP)

Adding server-side processing for appointment scheduling, prescription management, and other dynamic functionalities.

Step 11: Integrate Frontend and Backend

Linking HTML pages to PHP scripts for server-side processing. Creating asynchronous communication between the frontend and backend.

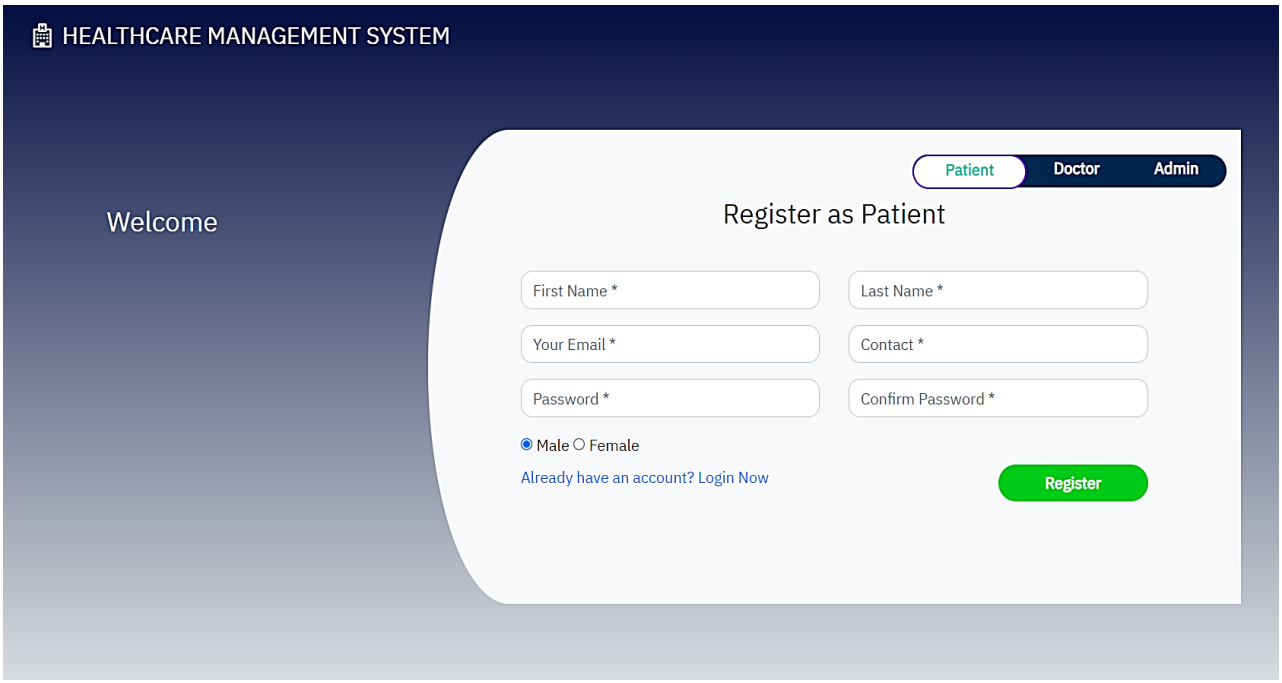
Step 12: Testing

Testing frontend and backend components by hosting the system on a web browser.

This step-by-step process provides a structured guide for implementing the frontend and backend of a Healthcare Management System.

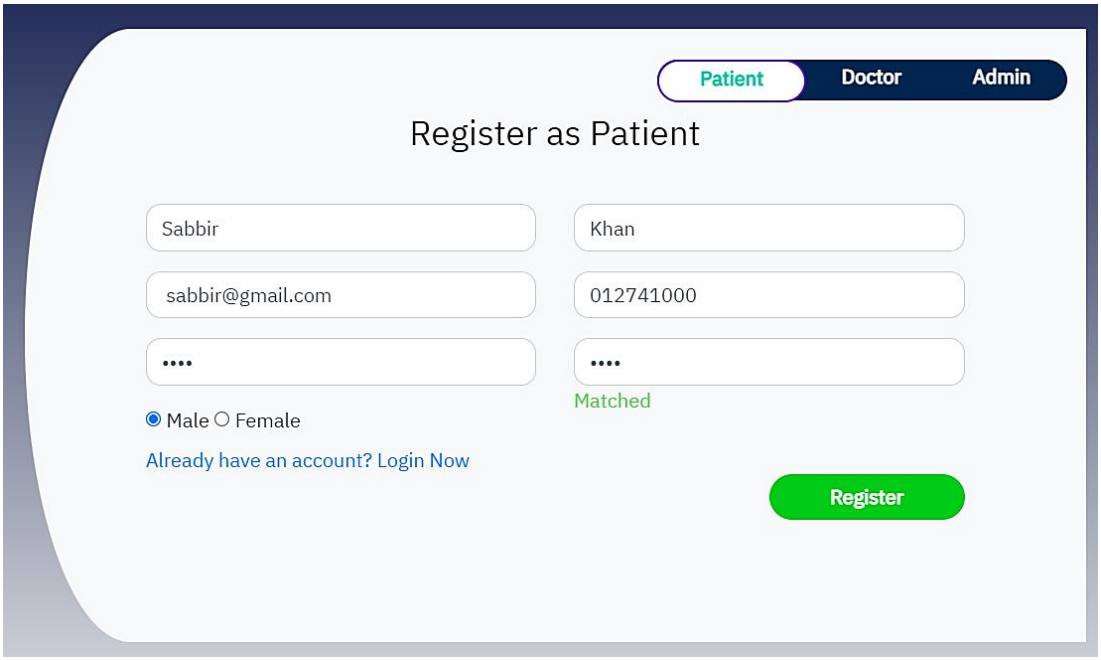
7. User Interface:

Home Page and Patient Registration:



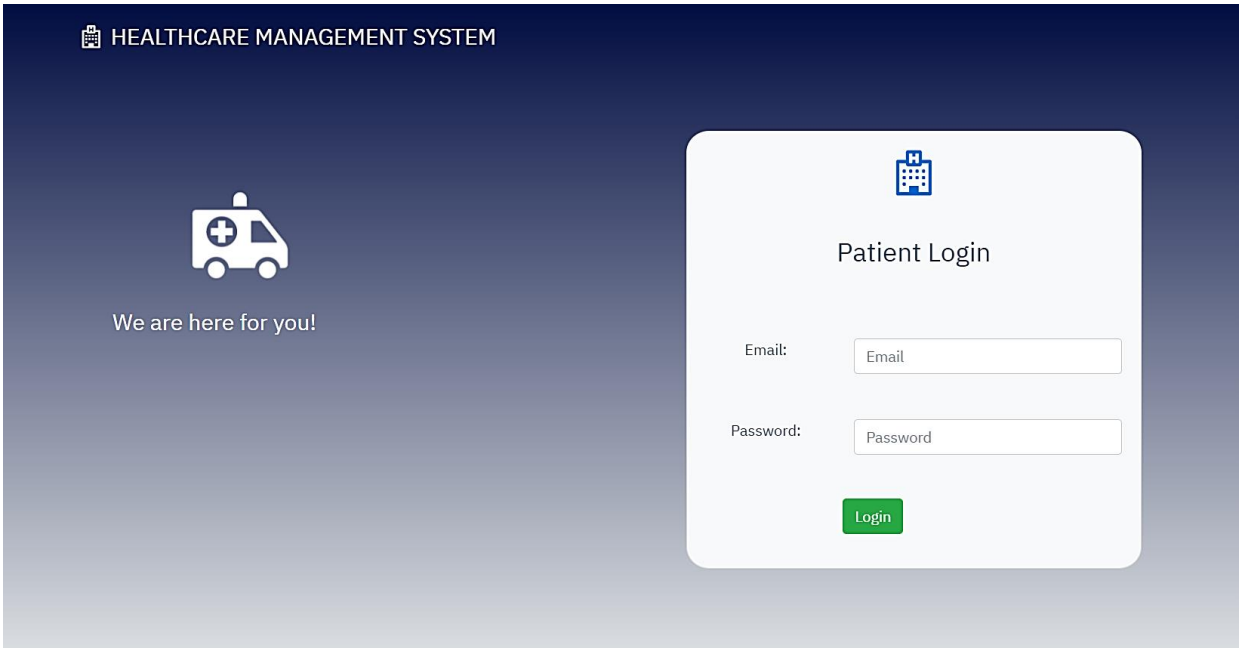
Here, we can see this is the home page and patient registration form.

Filling Patient Data in Patient Registration:



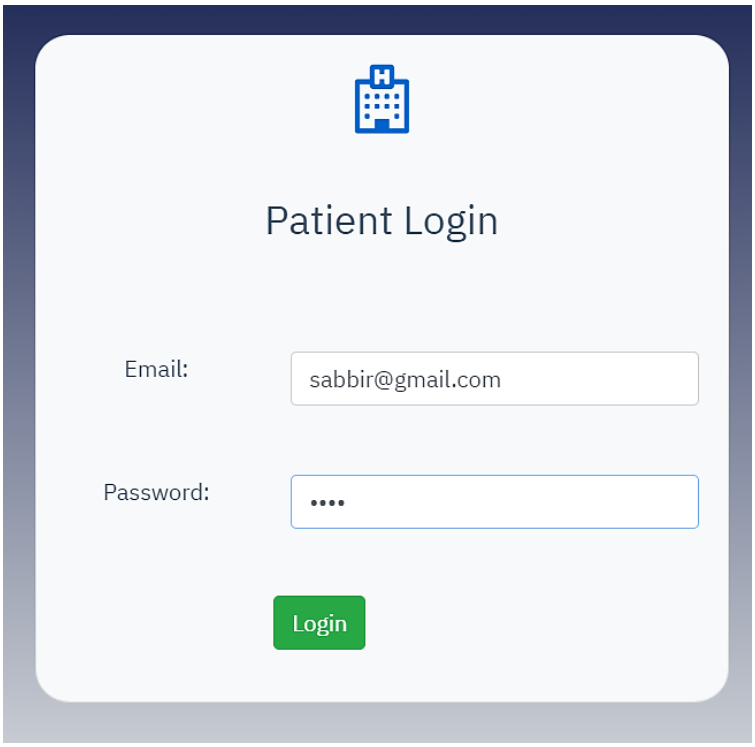
Here, filling the required data as a patient in registration form.

Patient's Log in page:



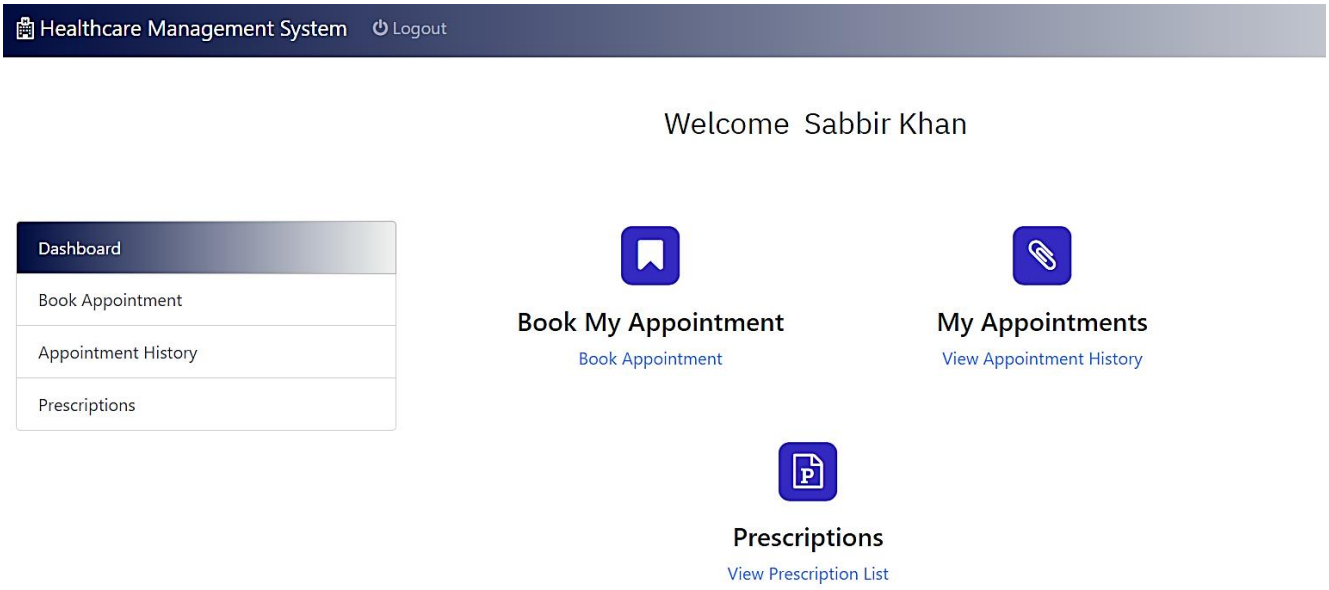
This is patient's log in page. Only Email and Password is required to log in.

Patient Log in with Email and Password:



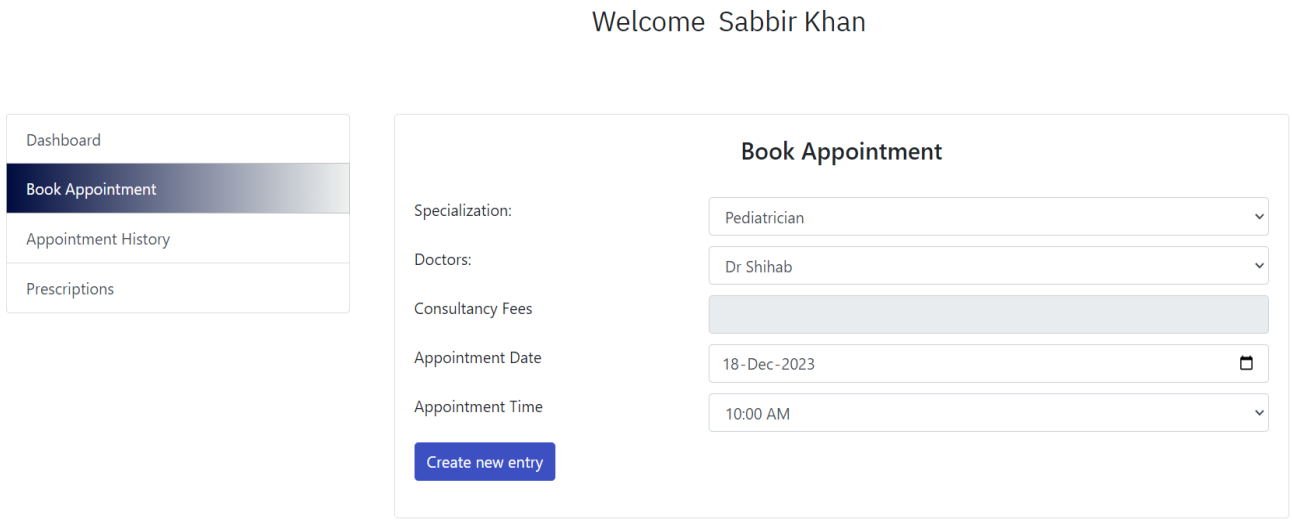
After typing the registered email and password, a patient can log in.

Patient's Logged in Dashboard:

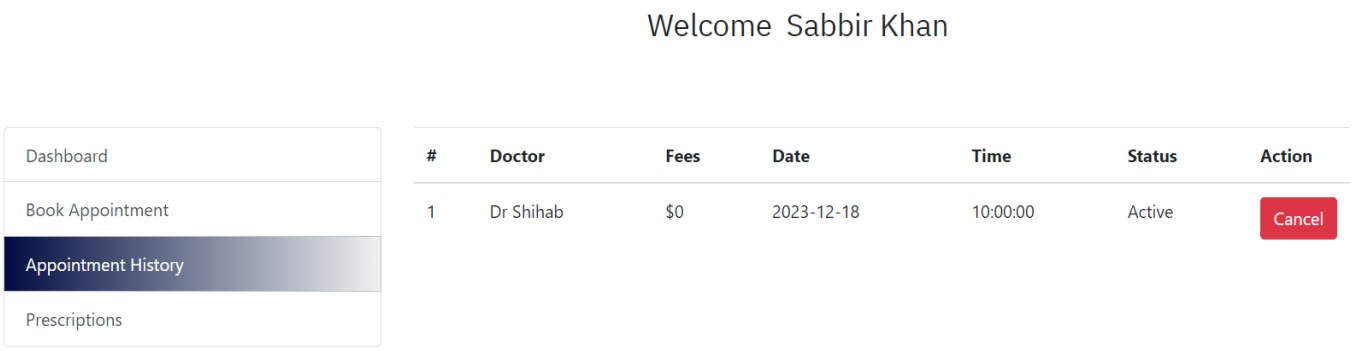


After successfully logging in, patients can fix their appointment timing and selecting a specialized doctor for specific health issues.

Book My Appointment:

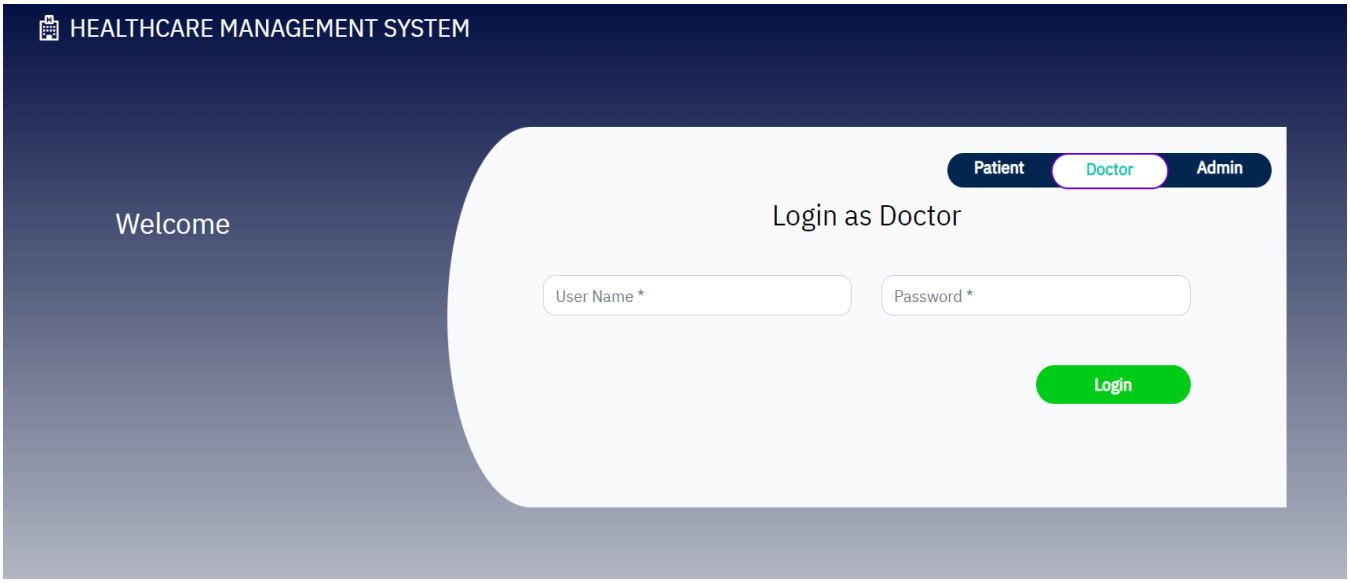


Here, a patient successfully fixed the time of appointment with the specialized doctor.



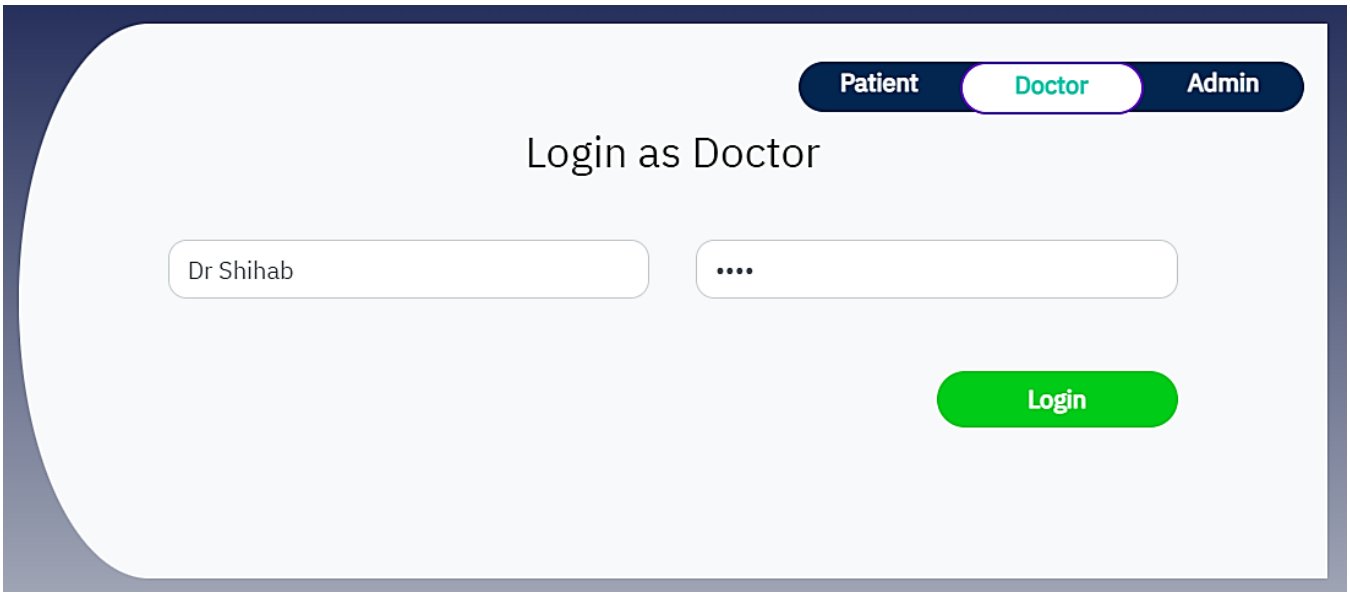
Patients can check their Appointment History and also can cancel the appointment.

Doctor's Log in page:

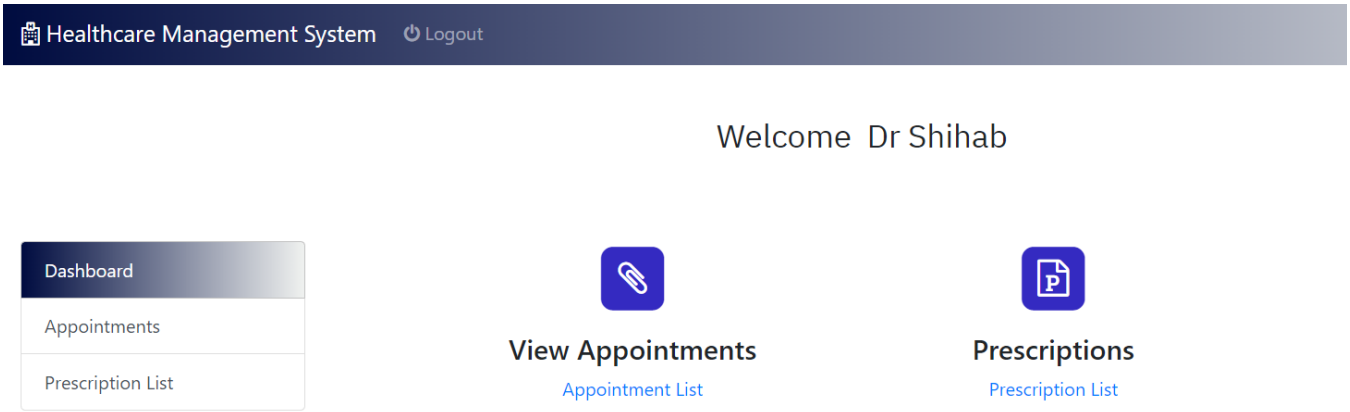


This is the Log in page of Doctor where registered username and password is needed for log in.

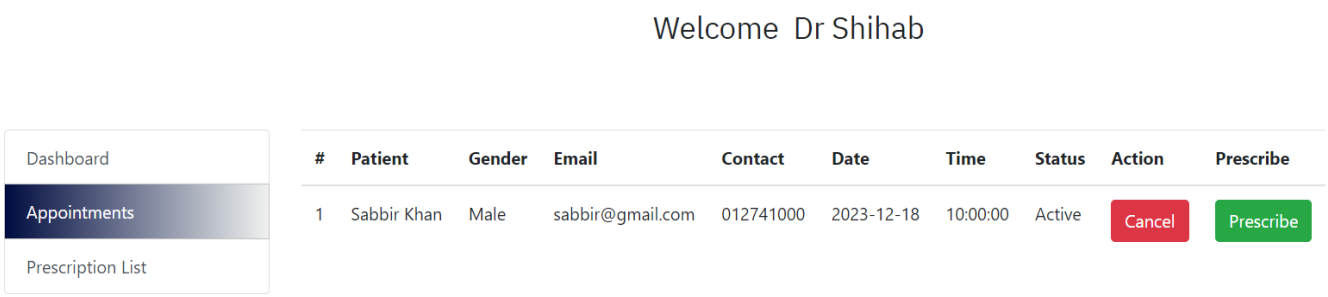
Doctor Log in with Username and Password:



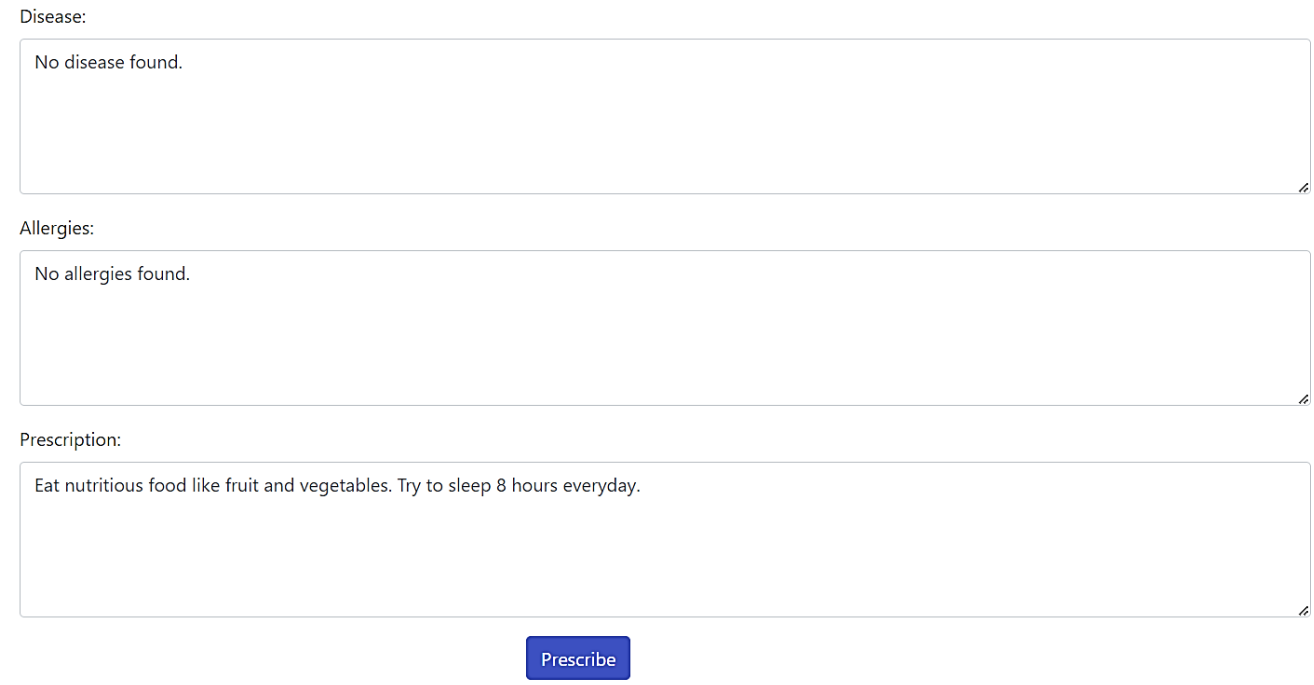
Doctor's Dashboard:



This is Doctor's Dashboard after logging in. Here, Doctors can check the pending appointment requests coming from patients. And also, Doctors can prescribe and can view the given prescriptions.

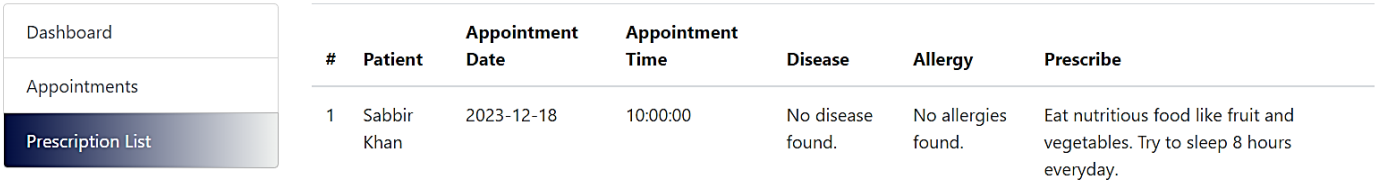


Doctors can check the appointment schedules of patients. There is a green box of Prescribe where doctors can give prescription to patients.



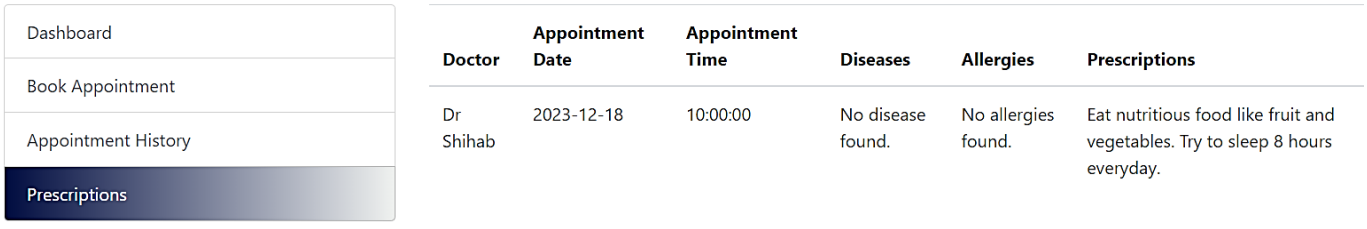
This is the Prescribe box where doctor can write prescriptions for a specific patient.

Prescription from Doctor's side:



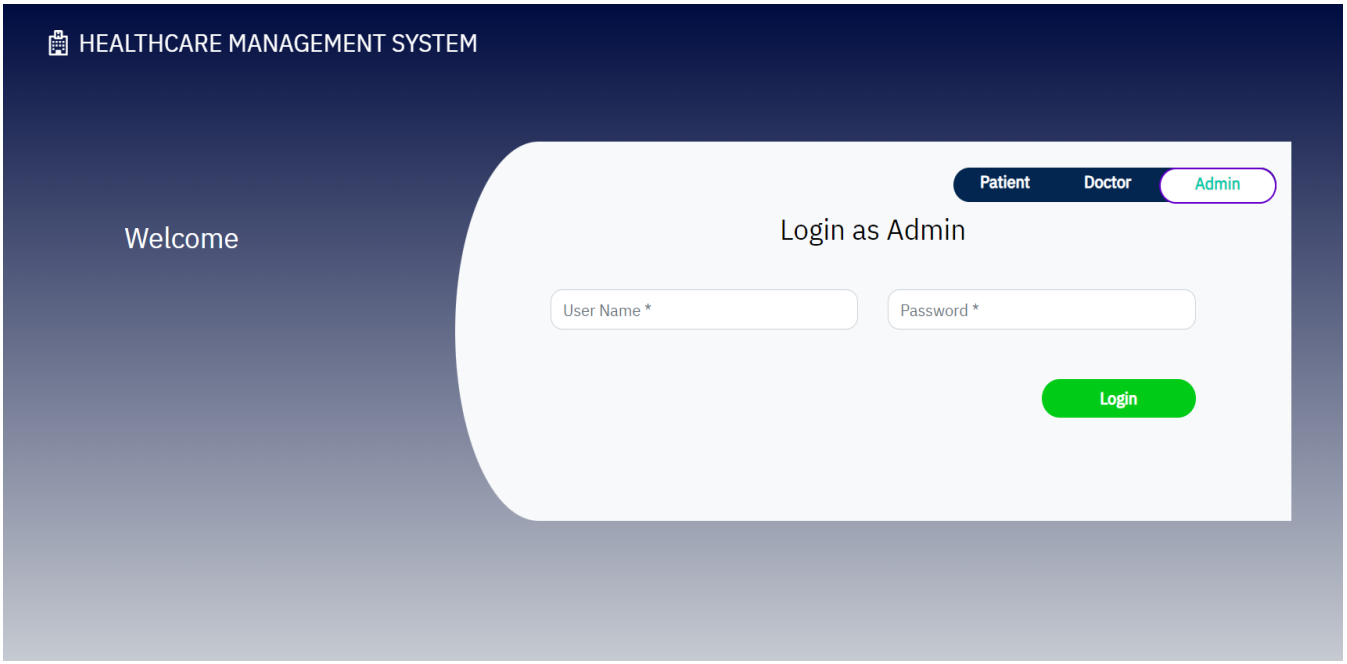
After submitting a prescription, it looks like this.

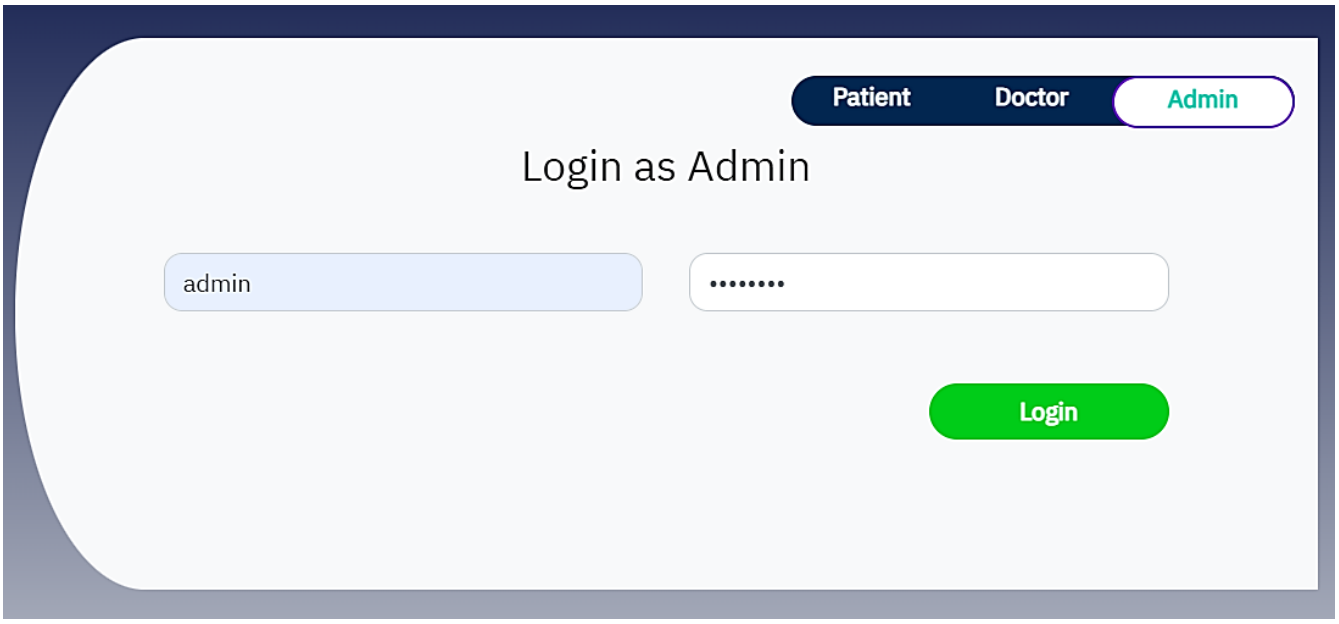
Prescription from Patient's side:



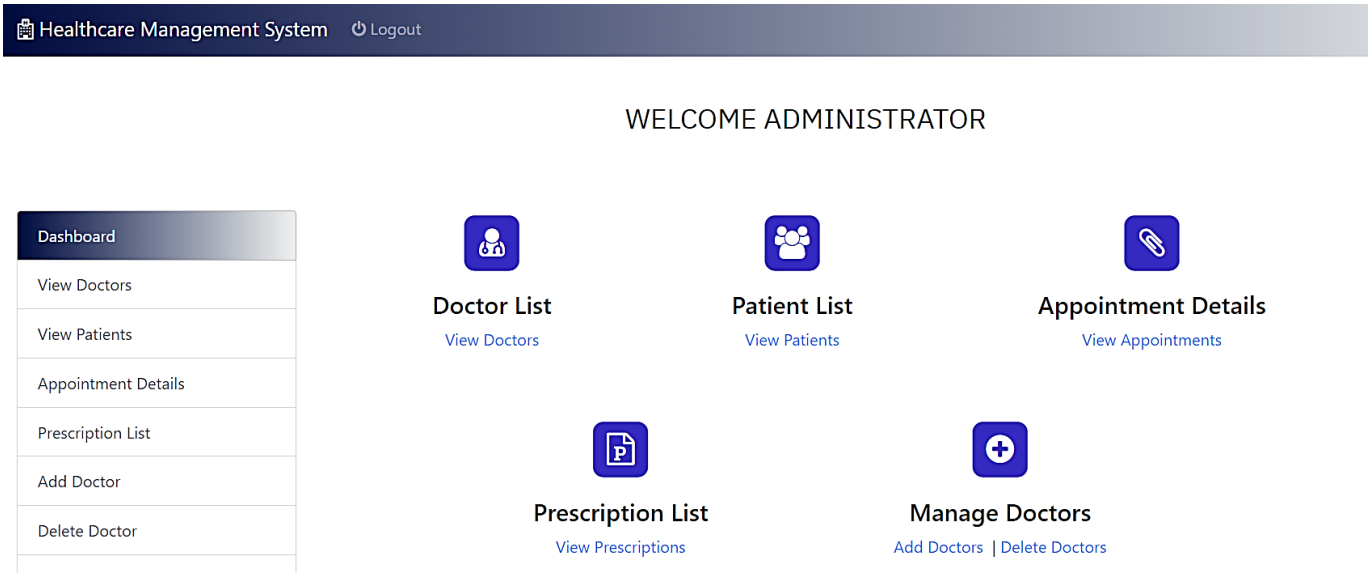
This is the look of prescribed prescription from patient's side.

Admin's Log in page:

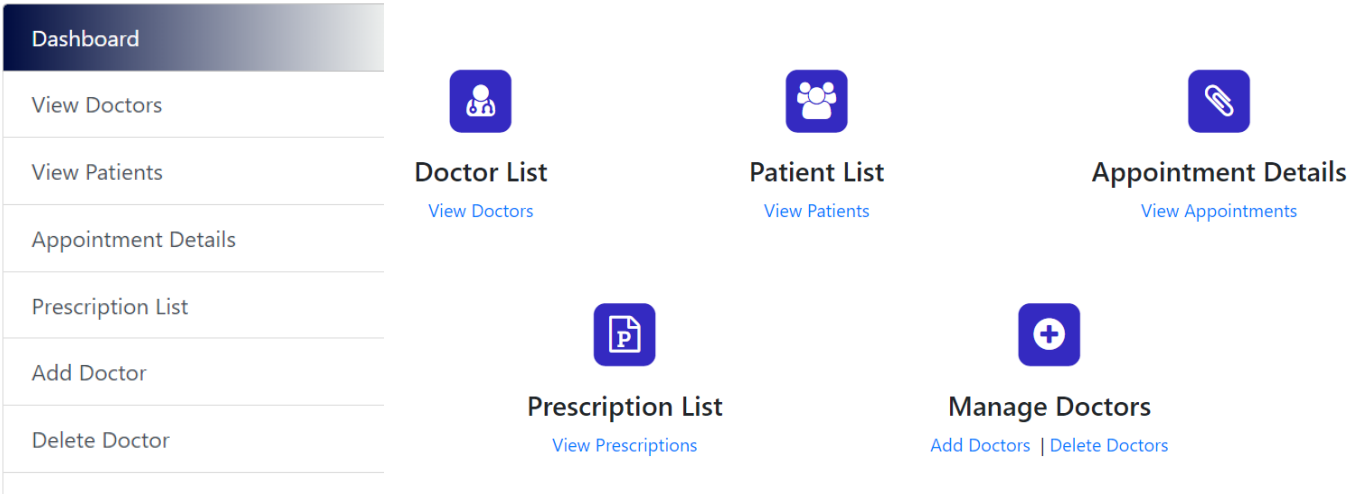




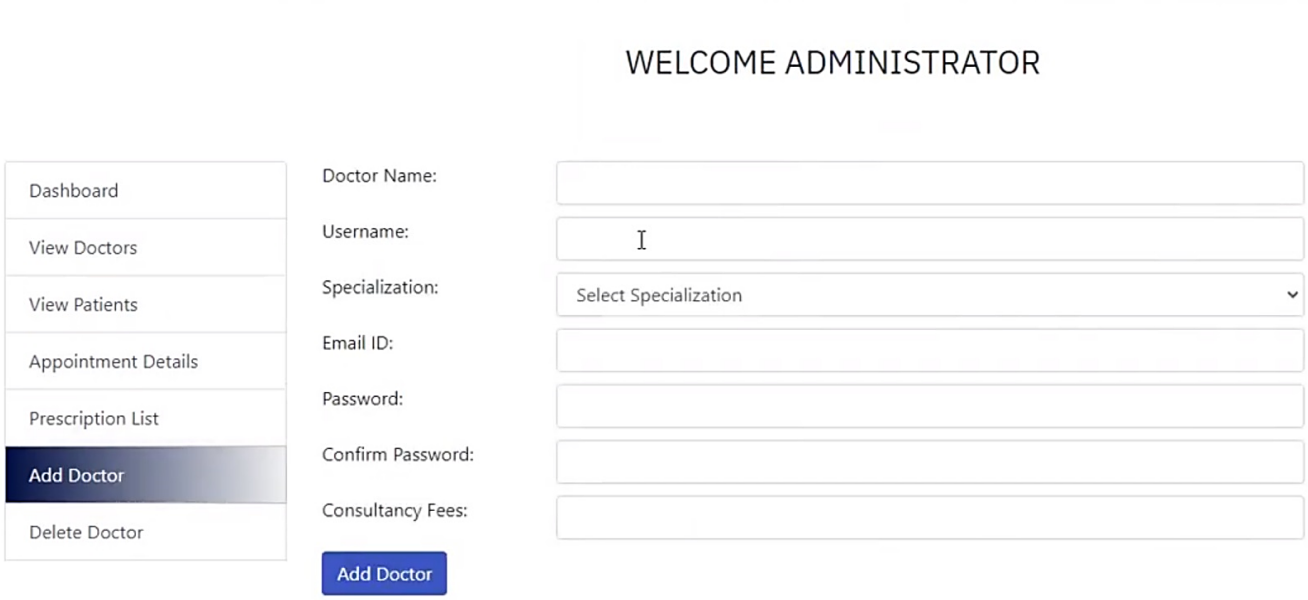
Admin have to log in the dashboard by using username and password.



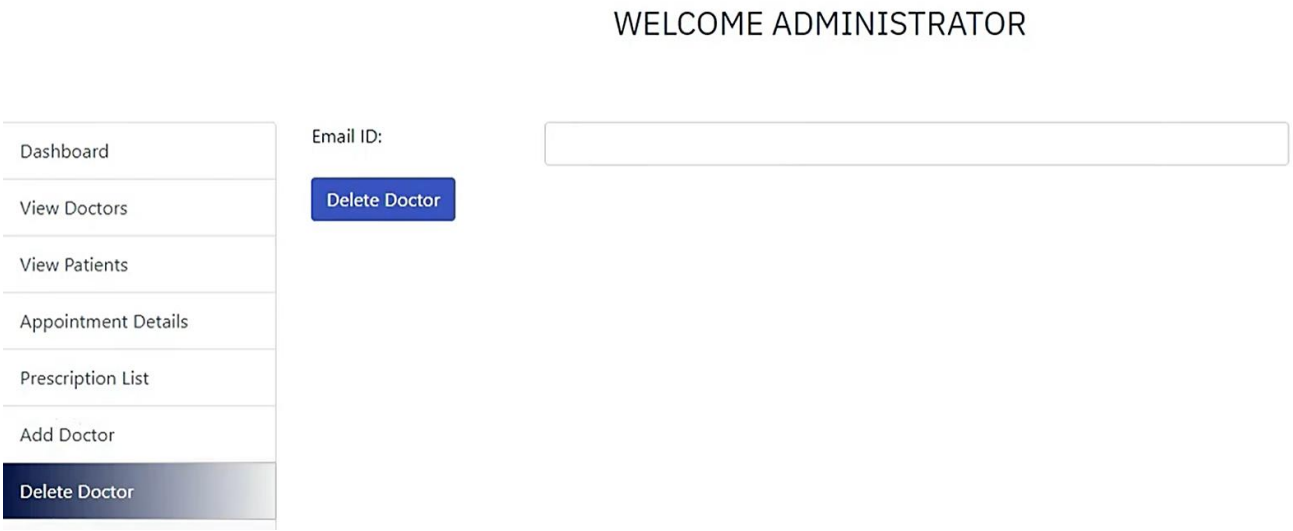
This is the dashboard's look of Admin.



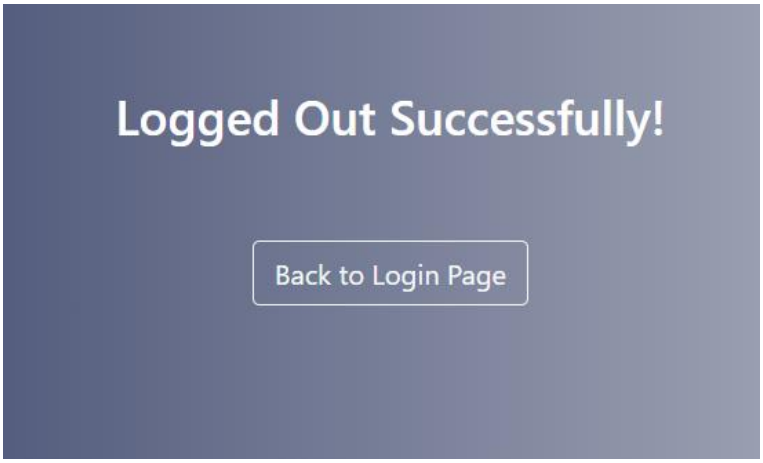
As we can see the Admin's dashboard has many segments such as Doctor List is for viewing how many doctors work there, Patient List is for watching the serial of registered patients, Appointment Details is for viewing how appointments are issued. Prescription List is for viewing every given prescription from the doctor's side. Add Doctor is a segment for adding new doctor in the system. And Delete Doctor is for deleting a doctor from the system.



This is the Add Doctor segment where a new doctor can be added in the system by admin. Several details of a Doctor have to add on those boxes of Add Doctor section.



This is Delete Doctor option where only Email ID of a doctor is needed to remove a doctor.



After clicking Log Out, this shows up as a Log Out confirmation from Admin's side. Admin, Patient, Doctor has the Log Out segment. When any data added in the four sides (Admin, Patient, Doctor, Prescription) from this website, then all of those data are also interlinked and stored in SQL's database (localhost/phpMyAdmin/) as well.

After seeing the work process and screenshots from our database project, we can say that we have successfully visualized this project and fulfilled the required proposal, prerequisites, objectives, implementations correctly. Our Healthcare Management System project is successfully working.

8. Healthcare Management System E-R diagram:

Entity:

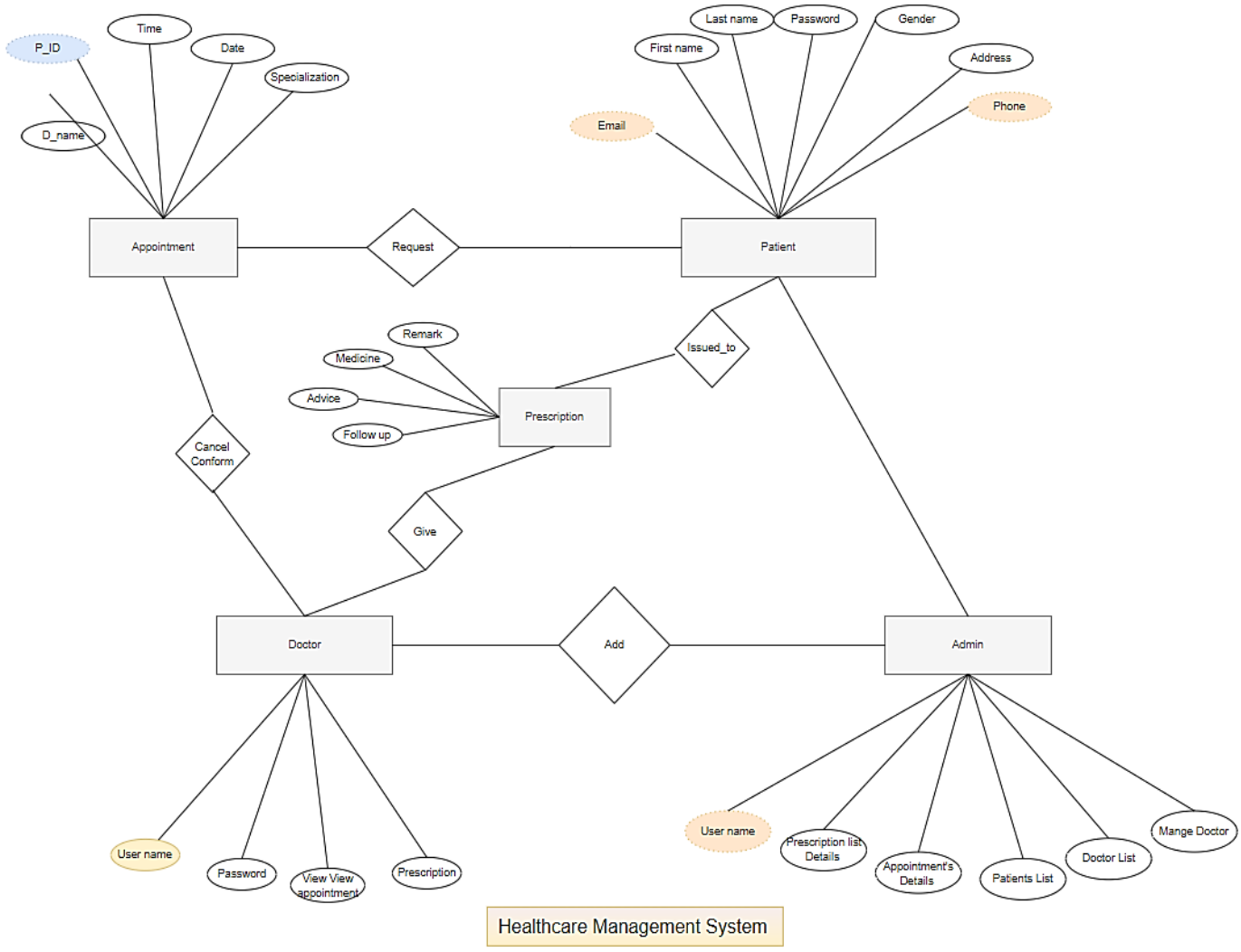
1. Patient: i. Patient Registration:- First Name, Last Name, Email, Contact, Password, Confirm Password, Gender.

ii. Patient Log In:- Email, Password, Book My Appointments, My Appointments, Prescriptions.

2. Doctor: i. Doctor Log in:- User Name, Password, View Appointment, Prescriptions.

3. Admin: i. Admin Log in:- User Name, Password, Doctors List, Patient List, Appointment Details, Prescription List, Manage Doctor (Add Doctor & Delete Doctor)

4. Appointment: Patient's Name, Specialization, Doctor's Name, Time, Date.



9. Relationships:

i. One-to-Many relationship between Patient and Appointment (One patient can have multiple appointments, but each appointment is associated with one patient).

ii. One-to-Many relationship between Doctor and Appointment (One doctor can have multiple appointments, but each appointment is associated with one doctor).

iii. One-to-Many relationship between Patient and Prescription (One patient can have multiple prescriptions, but each prescription is associated with one patient).

iv. One-to-Many relationship between Doctor and Prescription (One doctor can issue multiple prescriptions, but each prescription is associated with one doctor).

v. Many-to-Many relationship between Admin and Patient (An admin can manage multiple patients, and a patient can be managed by multiple admins. This relationship could represent activities like registration, data updates, etc.).

vi. Many-to-Many relationship between Admin and Doctor (An admin can manage multiple doctors, and a doctor can be managed by multiple admins)

10. Conclusion:

We have successfully developed the Healthcare Management System project by the help of SQL, HTML, CSS, PHP, Bootstrap and JavaScript. This project represents a transformative solution for Healthcare facilities for patients and it has operational efficiency, data management, and patient care. This user-friendly platform that addresses various aspects of Healthcare services. The use of SQL organized the database structure. HTML, CSS, and Bootstrap contributes the creation of responsive user interface. PHP enables dynamic server, logic and communication between the front-end and back-end components. JavaScript enhances interactivity, real-time updates. The advantages of this project defined patient management, appointment scheduling, and responsive workflows,

reducing administrative burdens. The system's ability to generate insightful reports for improvement in Healthcare sector. User authentication and role-based access controls ensures data security with privacy standards. The responsive design of the system accommodates the dynamic needs of Healthcare facilities. This system can save time and promotes patient engagement through features like easy to access the system. In future this project can be updated and new features can be added for the betterment for Healthcare system. The Healthcare Management System represents itself as an essential tool for doctors, Healthcare professionals, administrators and patients. This project proves that, it is a successful implementation of Healthcare Management System in DBMS which can fulfill the evolving demands of the Healthcare industry.