INSTITUTE FOR ADVANCED COMPUTING ANDSOFTWARE DEVELOPMENT (IACSD), AKURDI, PUNE

Documentation On

BookMyDoc System

PG-DAC March 2023

Submitted By:

Group No: 87

Roll No.	Name:
233169	Mohan Talavar
233185	Pranay Patil

Mrs. Rupali Thorat
Project Guide

Mr. Rohit Puranik
Centre Coordinator

ABSTRACT

"BookMyDoc" is an Online Doctor Appointment Booking Web Application designed for efficient and user-friendly medical appointment scheduling. It provides a comprehensive list of doctors, their schedules, and availability within specific hospitals, empowering users to make informed decisions and book appointments conveniently. While initially focused on one hospital, the system offers potential expansion to multiple healthcare facilities.

The application includes three core modules: Admin, Doctor, and Patient, enabling seamless management of doctors, patient information, and appointments. Additional features include user registration, specialized doctor listings, and password reset. Non-functional requirements ensure robust performance, security, usability, availability, and regulatory compliance.

"BookMyDoc" simplifies healthcare access, reduces wait times, and enhances the overall healthcare experience for both doctors and patients.

This condensed abstract captures the key aspects of your project while keeping it shorter. If you have any further adjustments or specific points to emphasize, please let me know!

ACKNOWLEDGEMENT

I take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination. I extend my sincere and heartfelt thanks to our esteemed guide, **Mrs. Rupali Thorat** for providing me with the right guidance and advice at the crucial juncture sand for showing me the right way. I extend my sincere thanks to our respected **Centre Co-Ordinator Mr. Rohit Puranik**, for allowing us to use the facilities available. I would like to thank the other faculty members also, at this occasion. Last but not the least, I would like to thank my friends and family for the support and encouragement theyhave given me during the course of our work.

Mohan Talavar (223169)

Pranay Patil (223185)

Table of Contents

LIST OF FIGURES

INTRODUCTION

This document serves as a comprehensive outline of the business requirements and the intended scope for the development of the Book My Doc System. The primary purpose of this document is to establish a clear delineation of both the functional and non-functional prerequisites, encompassing business regulations and other pertinent constraints.

In the wake of the pandemic, a global realization has emerged regarding the substantial influence of the internet. The field of medical services has witnessed a remarkable surge in the generation of information on a daily basis. Relying on manual administrative processes for managing such information can give rise to a myriad of complications, leading to squandered time, financial resources, and human efforts. Moreover, conventional methods often entail prolonged waiting periods and intricate processing procedures before medical information is received. This project stands as a remedy to rectify such errors and eliminate the protracted waiting times. By doing so, it not only optimizes resource allocation but also endeavors to curtail expenditures and enhance overall efficiency.

1.1 Project Objectives:

The objective of the Online Doctor Appointment Booking Web Application is to provide users a convenient, efficient, and user-friendly solution for scheduling medical appointments. Our platform presents a comprehensive list of doctors, including their schedules and availability in specific hospitals. This empowers users to make well-informed decisions and seamlessly book appointments, eliminating the need for unnecessary visits. By minimizing such visits, our system enhances hospital efficiency, reduces wait times, and ensures a smoother experience for both doctors and patient.

1.2 Project Overview:

- Book My Doc System is the public web application used for one specific hospital.
- There are mainly three types of user.
- Admin will be able to register a new (recruited) doctor into the hospital system and will have access to all the details regarding appointment booking and patients taking treatment in the hospital.
- Patients can search and book appointments.
- Already registered patients can view further schedule, records, all related information regarding his/her treatment.
- Doctors can provide prescriptions to the consulting patients regarding medicines, lab tests, operations etc.

Book My Doc System provides such functionalities which connect doctor and patient with each other efficiently and it will reduce crowding in hospital.

2. System Analysis:

System analysis constitutes a vital phase of the development process, encompassing the gathering and interpretation of information, problem diagnosis, and formulation of improvement recommendations. It hinges on robust communication between system users and developers. System analysis represents a holistic view of the entire system, identifying inputs, and conducting an in-depth examination to pinpoint areas of concern. Proposed solutions are presented in the form of a comprehensive proposal, which is subject to review, and adjustments based on user feedback. This iterative process concludes upon user satisfaction with the proposal.

2.1 Existing System:

The current state of the doctor appointment booking system involves manual processes reliant on pen and paper. Adding new patients, scheduling appointments, and notifying users are managed through traditional methods:

User-unfriendly interface.
Secretary's direct involvement for notice dissemination and bill distribution.
Difficulty in congregating all society members in one location.
Paperwork maintenance for flat details and owners.
Time-consuming operations.
Limited accessibility for remote users.

2.2 Proposed System:

The envisaged system revolutionizes the doctor appointment booking process by introducing a user-friendly interface for all stakeholders. The system's features encompass cross-browsing capabilities, multilingual support catering to various demographics, and more. It accommodates the needs of diverse society members, even those residing abroad, ensuring global accessibility.

- Enhanced user interface for all society members.
- Multilingual support for wider inclusion.
- Accessibility for members regardless of their geographical location.
- Reduction of manual documentation.
- Online availability, facilitating 24/7 access.
- User-friendly web platform for smooth interactions.

3. STUDY OF THE SYSTEM

3.1 MODULES:

The system after careful analysis has been identified to be presented with the following modules and roles.

The modules involved are:

Book My Doc System consists of three modules described below.

1. Admin Module
2. Doctor Module
3. Patient Module

3.1.1 Admin Module:

- Admin can login.
- Admin can register a newly recruited doctor.
- Admin can also view the list of all the doctors and patients who are registered in the system.
- Admin can also view all details of doctors and patients and have authority to remove doctor as well as patient.
- Admin can also view all the appointment bookings of all the doctors.
- Admins have authority to give approval on patient's appointment cancellation request.

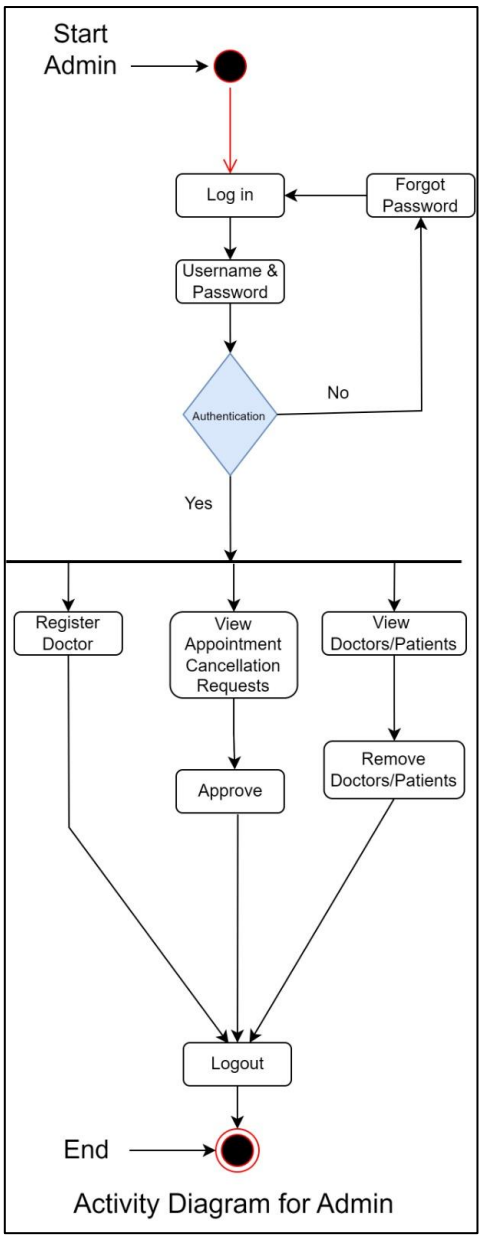

Fig 1. Activity diagram for Admin

3.1.2 **Doctor Module**:

- Doctor can login.
- A Doctor can add/modify his schedule of availability.
- He can view details of a revisiting patient.
- He can view the appointment schedule.

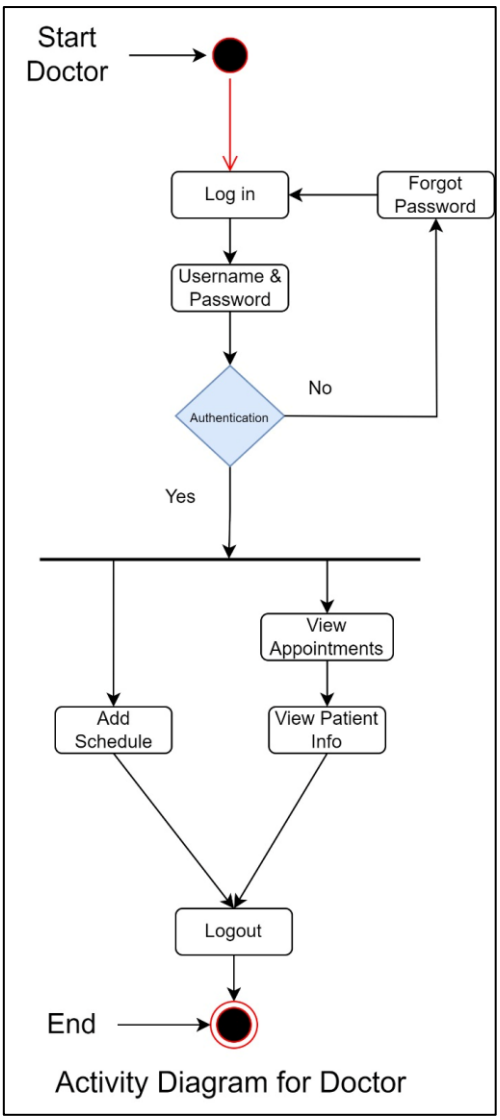

Fig 2. Activity diagram for doctor

3.1.3 Patient Module:

- A Patient can register and book an appointment.
- He can login into his profile once registered.
- He can book/cancel a appointment.
- A Patient can view the prescription given to him by the Doctor.
- He can upload his medical history documents or recent test reports for consultation.

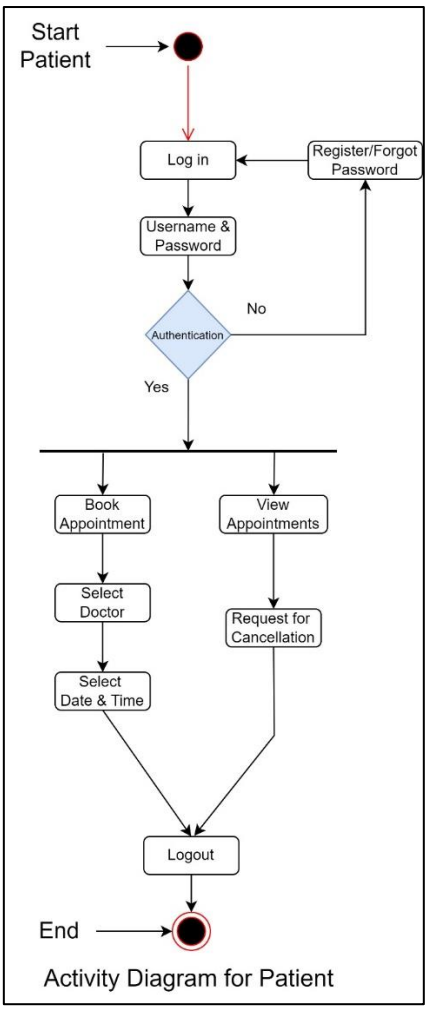

Fig 3. Activity diagram for patient

4. Software Requirement Specification

Document:

System Requirement Specification Document

Title:

System Requirement Specification for Online Doctor Appointment Booking Application

Objective (Purpose):

The objective of the Online Doctor Appointment Booking Web Application is to offer users a convenient, efficient, and user-friendly solution for scheduling medical appointments. Our platform presents a comprehensive list of doctors, including their schedules and availability in specific hospitals. This empowers users to make well-informed decisions and seamlessly book appointments, eliminating the need for unnecessary visits. By minimizing such visits, our system enhances hospital efficiency, reduces wait times, and ensures a smoother experience for both doctors and patients.

Scope:

This System is limited to offering a user-friendly solution for scheduling medical appointments and services exclusively within a single, chosen hospital. The application will empower users with the ability to access doctor information, schedules, and availability within the confines of this hospital, ensuring a seamless and informed booking experience. While presently focused on a specific hospital, the project's potential expansion to encompass multiple healthcare facilities remains a possibility for future iterations.

4.1 Functional Requirements:

The Doctor’s Appointment System is composed of three distinct modules, each designed to serve a specific role within the application:

Admin Module:

- Admin can log in to the system.
- Admin can register a newly recruited doctor.
- Admin can view a comprehensive list of all registered doctors and patients.
- Admin possesses the authority to view detailed information about doctors and patients, as well as the capability to remove them.
- Admin can access and review the complete schedule of appointment bookings for all doctors.
- Admin is empowered to approve or reject patient appointment cancellation requests.

Doctor Module

- Doctors can log in to their accounts.
- Doctors have the ability to add or modify their availability schedules.
- Doctors can access details of revisiting patients to enhance personalized care.
- Doctors can view their own appointment schedules.

Patient Module

- Patients can register within the system and subsequently book appointments.
- Registered patients can securely log in to their accounts.
- Patients can initiate the booking and cancellation of appointments.
- Patients can access their prescribed treatment plans provided by the doctors.
- Patients can request password reset in case of forgotten credentials.

Additional Functionalities:

- User Registration and Login: Users can create accounts, log in securely, and access personalized profiles.
- Specialized Doctor Listing: Users can retrieve a list of doctors categorized by specialization.
- Forgot Password: Users can initiate the process of resetting their password in case it's forgotten.

4.2 Non Functional Requirements:

Performance:

- Response Time: The system should provide near-instantaneous response times for basic interactions such as viewing doctor lists and scheduling appointments.
- Scalability: The application should be able to handle increased user load without significant degradation in performance.

Security:

- User Data Protection: All sensitive user data, including personal and medical information, should be encrypted, and securely stored.
- Authentication and Authorization: User authentication and authorization should be implemented to ensure that only authorized individuals can access specific modules and functionalities.

Usability:

- User-Friendly Interface: The user interface should be intuitive and easy to navigate, catering to users of varying levels of technical expertise.
- Accessibility: The system should be designed to accommodate users with disabilities, adhering to accessibility standards.

Availability:

- Uptime: The system should aim for a high level of availability, minimizing downtime for maintenance or technical issues.
- Backup and Recovery: Regular backups of data should be performed, and mechanisms for quick data recovery in case of failures should be in place.

Compatibility:

- Cross-Platform Compatibility: The application should be compatible with various web browsers and operating systems to ensure a consistent user experience.
- Mobile Responsiveness: The system should be responsive and accessible across a range of mobile devices.

Reliability:

- Data Integrity: The system should ensure the accuracy and integrity of data, preventing data loss or corruption.
- Error Handling: Proper error handling mechanisms should be implemented to provide informative error messages and guide users in resolving issues.

Regulatory Compliance:

- Privacy Regulations: The system should comply with relevant privacy regulations and standards (such as GDPR or HIPAA) to safeguard patient data.

Performance Testing:

- Load Testing: The system should undergo load testing to determine its performance under various user load conditions.
- Stress Testing: Stress testing should be conducted to assess the application's stability under extreme conditions.

Maintenance:

- Maintenance Window: Scheduled maintenance should be communicated to users in advance, and the application should be brought down for maintenance during low-usage periods.
- Version Updates: The system should be designed with flexibility to accommodate future updates and enhancements.

5. SYSTEM DESIGN

System design is the solution for the creation of a new system. This phase focuses on the detailed implementation of the feasible system. Its emphasis on translating design. Specifications to performance specification. System design has two phases of development.

➢ Logical Design

➢ Physical Design

During logical design phase the analyst describes inputs (sources), outputs(destinations), databases (data stores) and procedures (data flows) all in a format that meets the user requirements. The analyst also specifies the needs of the user at a level that virtually determines the information flow in and out of the system and the data resources. Here the logical design is done through data flow diagrams and database design. The physical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which specify exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data and produce the required report on a hard copy or display it on the screen.

5.1 INPUT AND OUTPUT DESIGN:

5.1.1 Input Design:

Input design establishes the crucial bridge between the Doctor's Appointment Booking System and its users, particularly in the context of MySQL database integration. This phase involves identifying the necessary input fields, validating the data, streamlining data entry, and accommodating multiple users. The prevalence of erroneous inputs underscores the significance of a robust input design. Such design curbs errors introduced during data entry by incorporating stringent validation mechanisms. User-generated inputs undergo transformation into a MySQL-compatible format, ensuring seamless integration.

Upon data collection, inputs are logically grouped, ensuring coherence. Corresponding MySQL data types are chosen to accommodate various input categories. Validation checks are performed on input data, and in cases where data fail to meet predefined criteria, users are promptly notified through informative messages. Data that successfully satisfy all conditions are subsequently mapped to appropriate tables within the MySQL database. Notably, the registration process for patient and doctor details is implemented in a user-centric manner, promoting usability and error mitigation.

5.1.2 Output Design:

In the context of MySQL integration within the Doctor's Appointment Booking System, output design takes center stage as a pivotal aspect. The generation of efficient and intelligible output becomes paramount. The quality of output significantly impacts user-system interaction and decision-making processes. Ensuring user engagement, the system offers sample output previews, acknowledging the user's role in evaluating output quality.

For this MySQL-integrated system, the output module predominantly encompasses selected notifications. These notifications, as the system's output, are meticulously structured to provide concise and relevant information to users. By leveraging the capabilities of MySQL's querying and data retrieval mechanisms, the system optimizes the generation of notifications, thereby enhancing the user experience and facilitating informed decision-making.

6. DATABASE DESIGN

6.1 DATABASE:

Databases are the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system.

Two essential settings for a database are

- Primary key - the field that is unique for all the record occurrences
- Foreign key - the field used to set relation between tables

Normalization is a technique to avoid redundancy in the tables.

6.2 SYSTEM TOOLS:

The various system tools that have been used in developing both the front end and the back end of the project are being discussed in this chapter.

6.2.1 FRONT END:

React is a library which is developed by Facebook are utilized to implement the frontend. The version of React on which we work is 18.0 . React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

6.2.2 BACKEND:

The back end is implemented using MySQL which is used to design databases.

6.3 MySQL:

MySQL is the world's second most widely used open-source relational database management system (RDBMS). The SQL phrase stands for Structured Query Language. The version of MySQL we work is 8.0.34.

6.4 Spring-Boot:

This is used to connect MYSQL and fetch data from database and store the data in database. The version for spring boot framework we used 2.7.13**.** The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE (Enterprise Edition) platform. Although the framework does not impose any specific programming model.

7. Data Flow Diagram:

7.1 Zero level DFD:

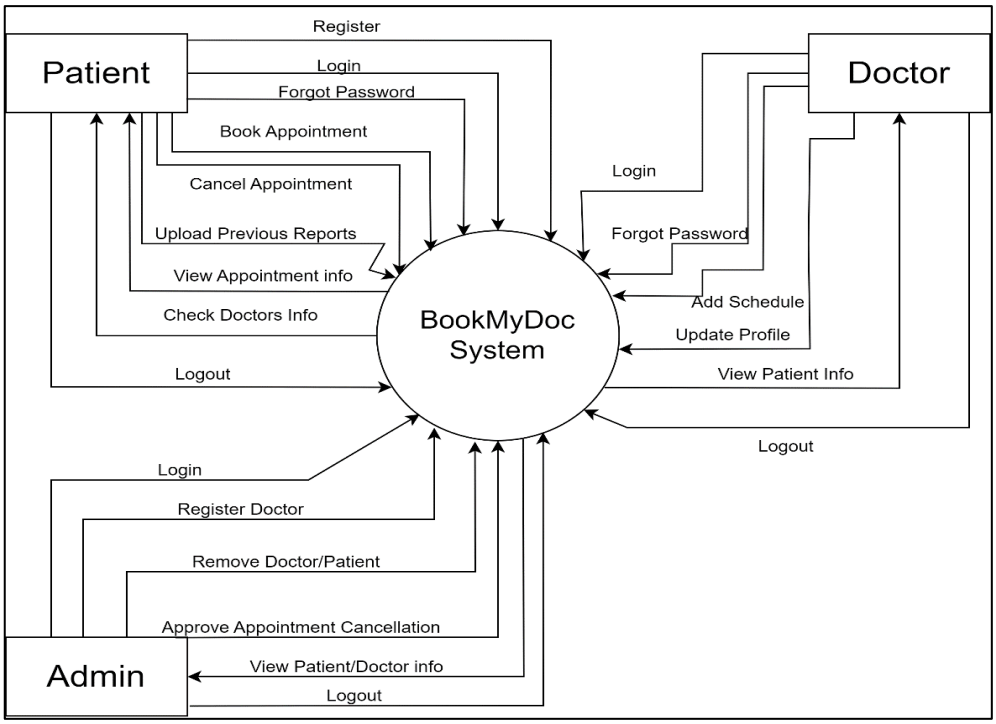

Fig 4.Zero level DFD

7.2 First Level DFD:

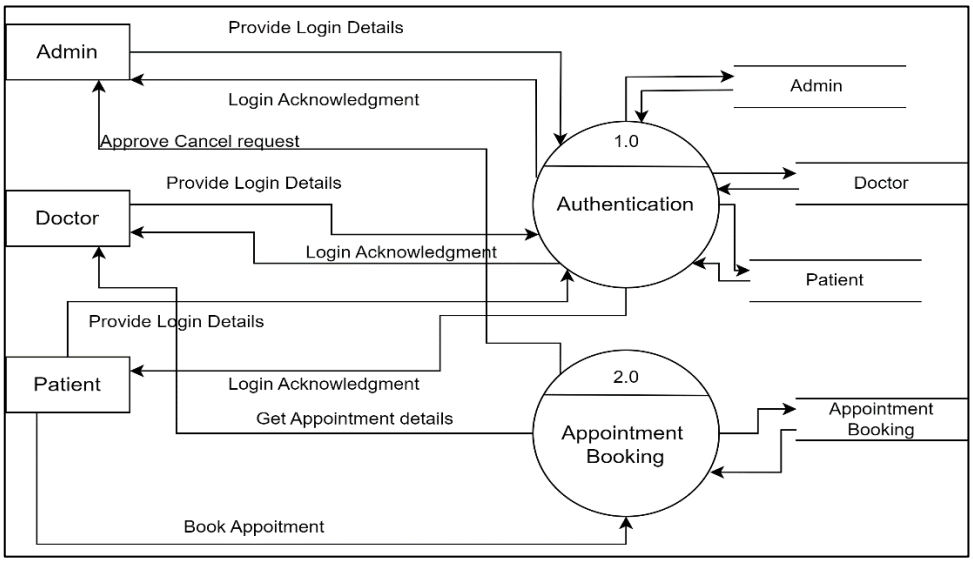

Fig 5.First level DFD

8. Use-Case Diagram

8.1 Admin:

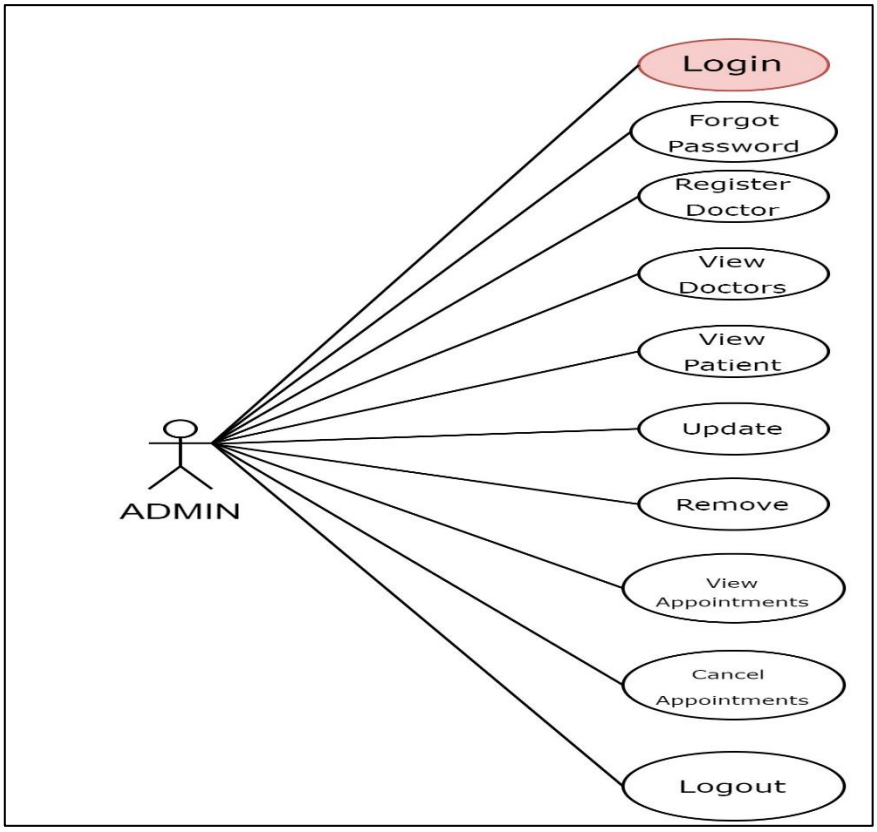

Fig 6. Use-Case Diagram for Admin

1. In Admin use case diagram Admin is the Actor.
2. Admin can handle following use cases:
   a. Login
   b. Forget Password
   c. Register New Doctor/Patient
   d. Remove Doctor/Patient
   e. View Doctor/Patient Details
   f. View Patient Details
   g. Approve for Cancel Appointment
   h. Logou

8.2 Doctor:

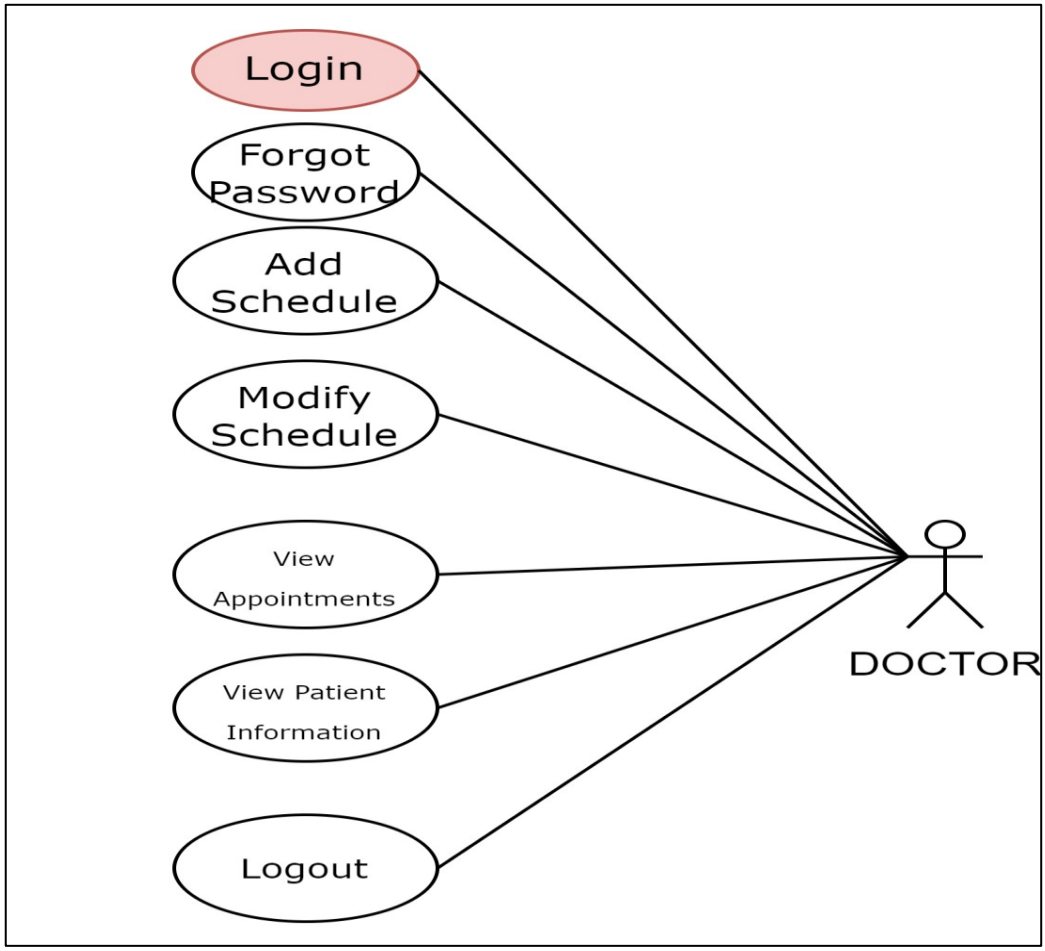

Fig 7. Use-Case Diagram for Doctor

1. In Doctor use case diagram Doctor is the Actor.
2. Doctor can handle following use cases:
    a. Login
    b. Forget Password
    c. Change Password
    d. Update Profile
    e. Add Schedule
    f. View Appointments
    g. View Patients Information
    h. Logout

8.3 Patient:

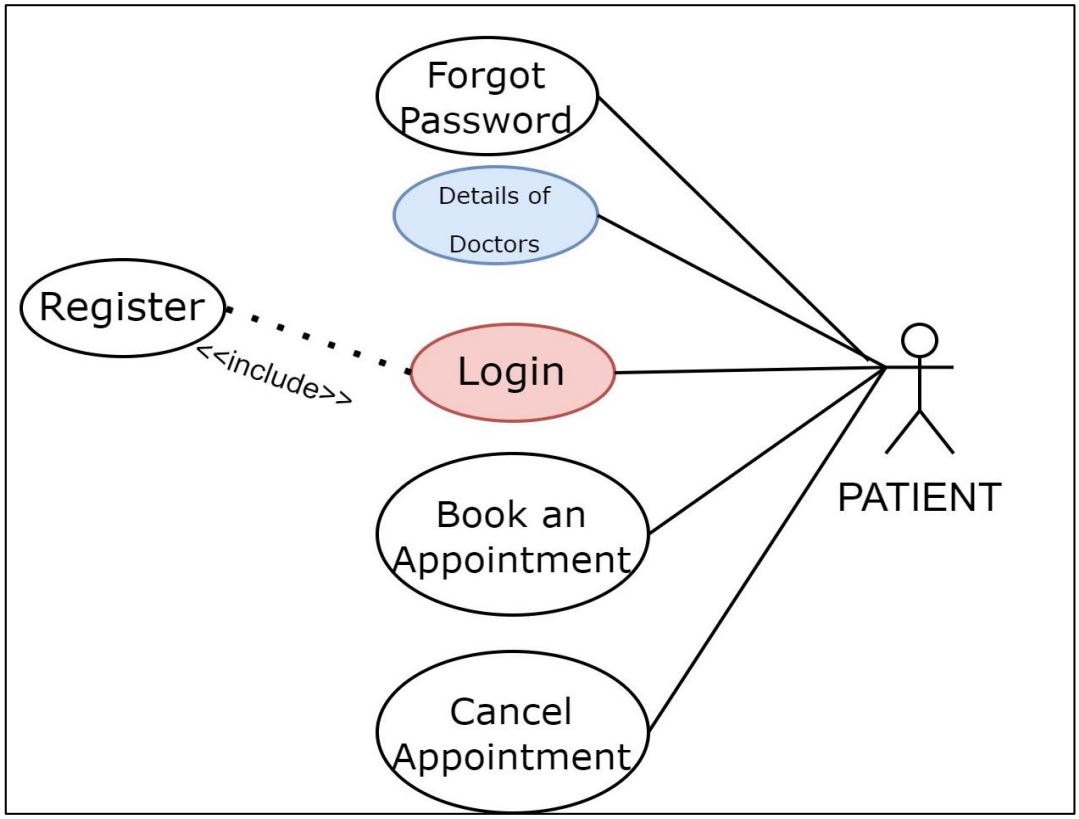

Fig 8. Use-Case Diagram for Patient

1. In Patient use case diagram Patient is the Actor.
2. Patient can handle following use cases:
   a. Register
   b. Login
   c. Forget Password
   d. Update Account
   e. Book an Appointment
   f. Cancel Appointment
   g. View Appointment
   h. Logout

9. ER-Diagram

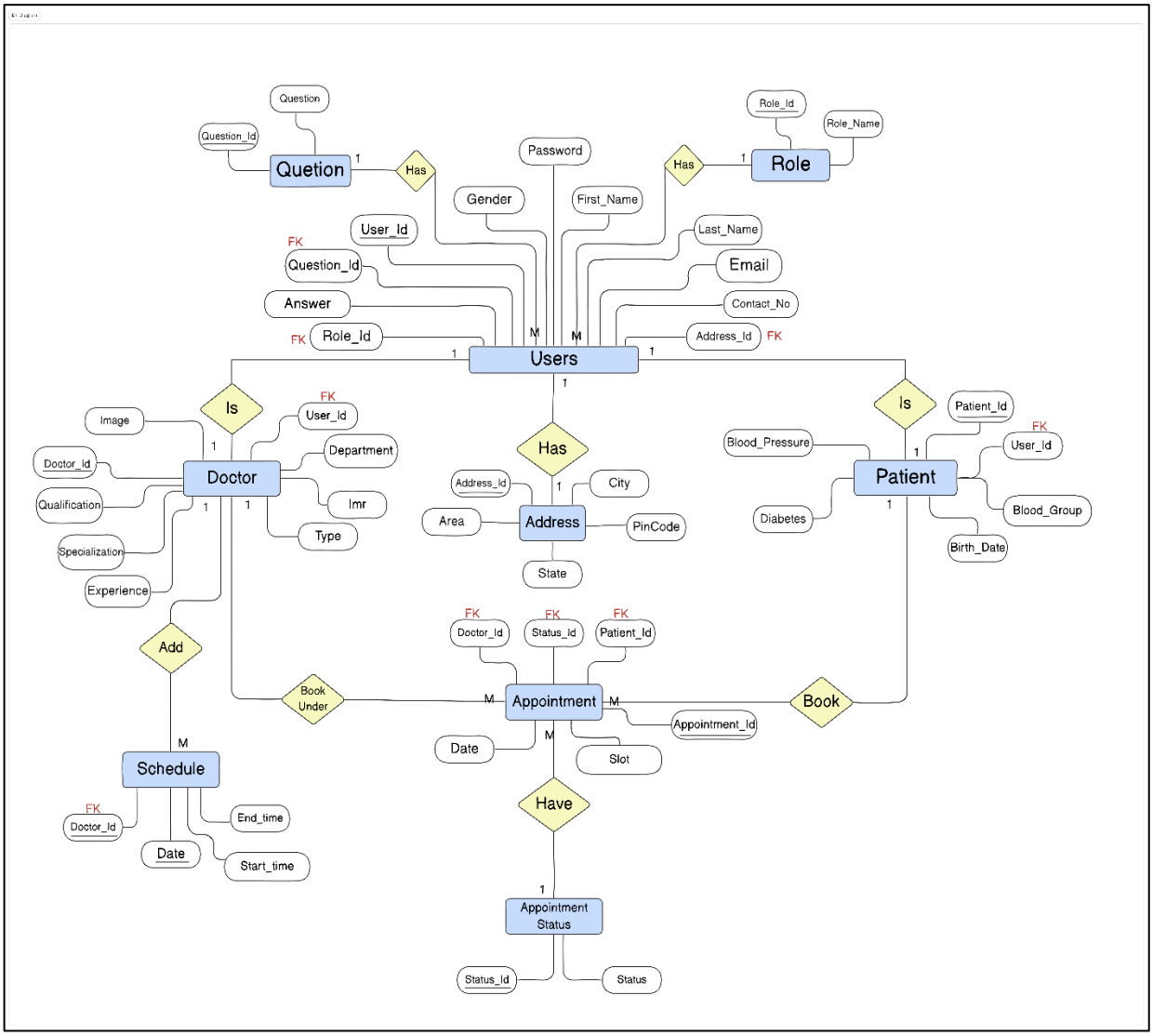

Fig 9. ER-Diagram

9.1 ER-Diagram Mysql

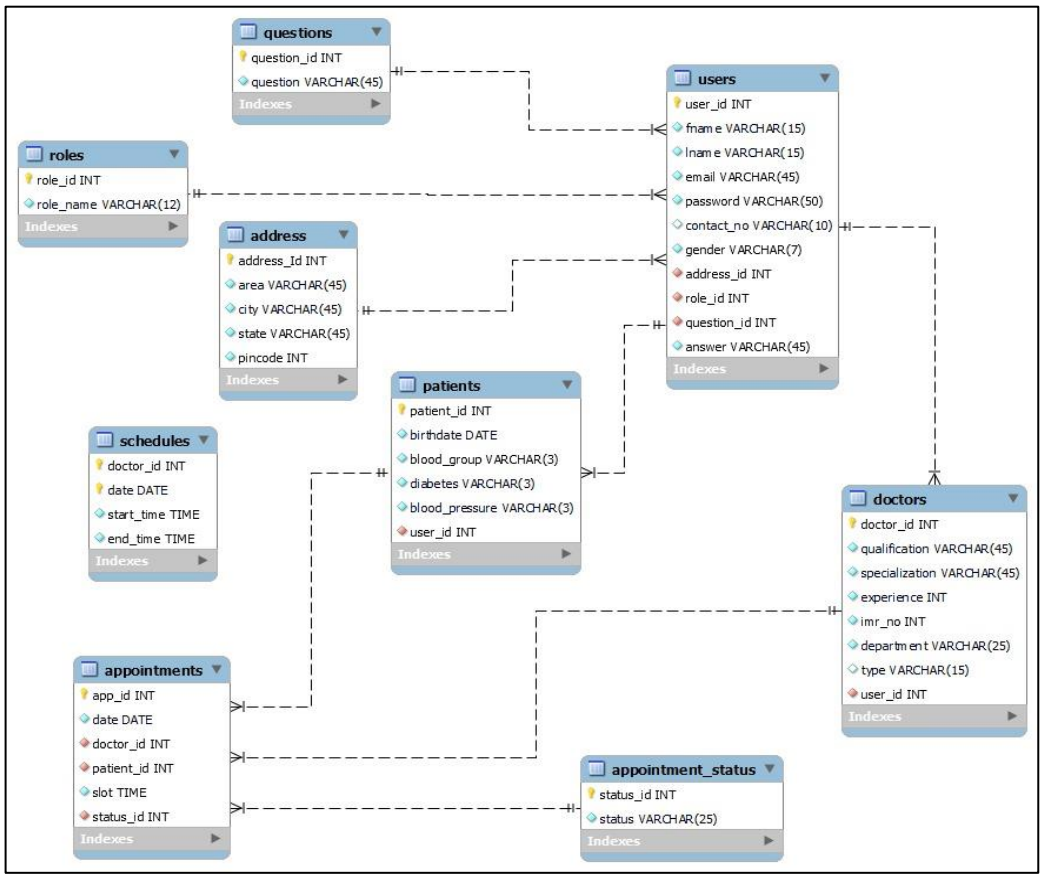

Fig 10. ER-Diagram Mysql

10.Class-Diagram

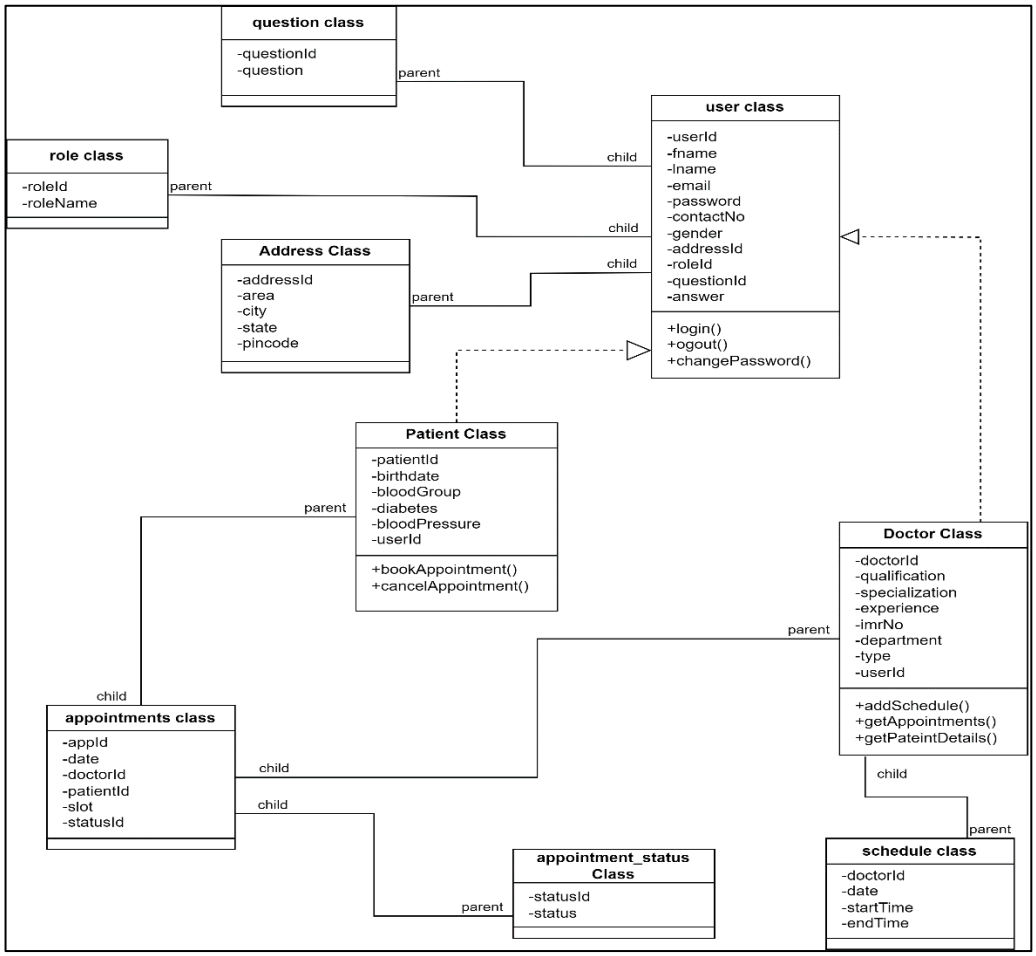

Fig 11 . Class-Diagram

11.Sequence Diagram

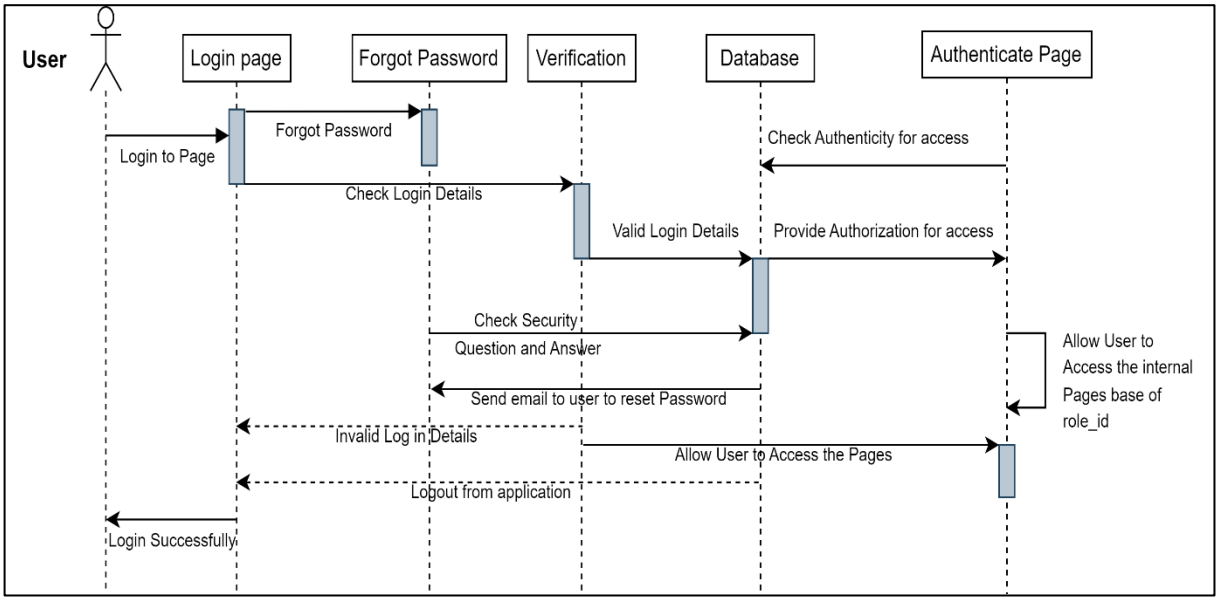

Fig 12. Sequence-Diagram

12.Deployment Diagram

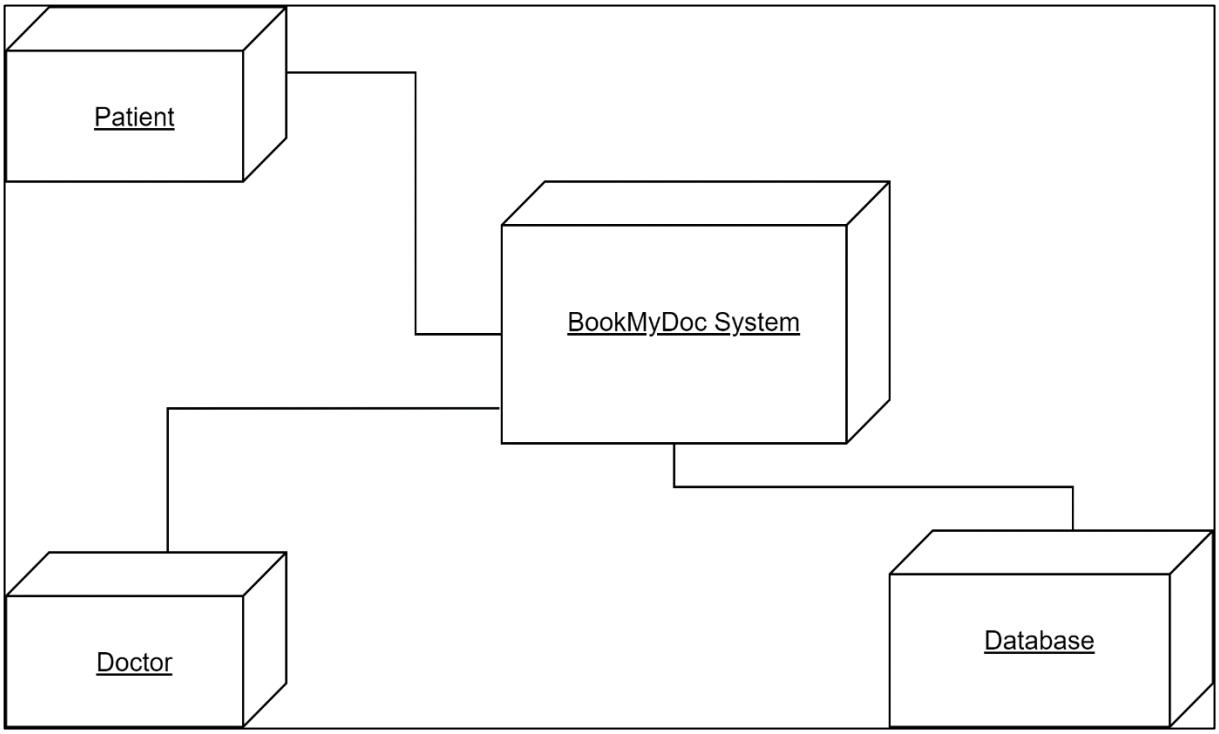

Fig 13. Deployment-Diagram

13.Database Design

```
mysql> show tables;
+----------------------+
| Tables_in_lifelinedb |
+----------------------+
| address              |
| appointment_status   |
| appointments         |
| doctors              |
| patients             |
| questions            |
| roles                |
| schedules            |
| users                |
+----------------------+
9 rows in set (0.06 sec)
```

Fig 14. Tables in Lifelinedb

13.1 Users Table:

```
mysql> desc users;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| user_id     | int         | NO   | PRI | NULL    | auto_increment |
| fname       | varchar(15) | NO   |     | NULL    |                |
| lname       | varchar(15) | NO   |     | NULL    |                |
| email       | varchar(45) | NO   | UNI | NULL    |                |
| password    | varchar(50) | NO   |     | NULL    |                |
| contact_no  | varchar(10) | YES  | UNI | NULL    |                |
| gender      | varchar(7)  | NO   |     | NULL    |                |
| address_id  | int         | NO   | MUL | NULL    |                |
| role_id     | int         | NO   | MUL | NULL    |                |
| question_id | int         | NO   | MUL | NULL    |                |
| answer      | varchar(45) | NO   |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
11 rows in set (0.00 sec)
```

Fig 15. User Table in Lifelinedb

13.2 Roles Table:

```
mysql> desc roles;
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| role_id   | int         | NO   | PRI | NULL    | auto_increment |
| role_name | varchar(12) | NO   |     | NULL    |                |
+-----------+-------------+------+-----+---------+----------------+
2 rows in set (0.00 sec)
```

Fig 16. Role Table in Lifelinedb

13.3 Doctors Table:

```
mysql> desc doctors;
+----------------+-------------+------+-----+---------+----------------+
| Field          | Type        | Null | Key | Default | Extra          |
+----------------+-------------+------+-----+---------+----------------+
| doctor_id      | int         | NO   | PRI | NULL    | auto_increment |
| qualification  | varchar(45) | NO   |     | NULL    |                |
| specialization | varchar(45) | NO   |     | NULL    |                |
| experience     | int         | NO   |     | NULL    |                |
| imr_no         | int         | NO   |     | NULL    |                |
| department     | varchar(25) | NO   |     | NULL    |                |
| type           | varchar(15) | YES  |     | NULL    |                |
| user_id        | int         | NO   | MUL | NULL    |                |
+----------------+-------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)
```

Fig 17. Doctor Table in Lifelinedb

13.4 Patients Table:

```
mysql> desc patients;
+----------------+------------+------+-----+---------+----------------+
| Field          | Type       | Null | Key | Default | Extra          |
+----------------+------------+------+-----+---------+----------------+
| patient_id     | int        | NO   | PRI | NULL    | auto_increment |
| birthdate      | date       | NO   |     | NULL    |                |
| blood_group    | varchar(3) | NO   |     | NULL    |                |
| diabetes       | varchar(3) | NO   |     | NULL    |                |
| blood_pressure | varchar(3) | NO   |     | NULL    |                |
| user_id        | int        | NO   | MUL | NULL    |                |
+----------------+------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

Fig 18. Patient Table in Lifelinedb

13.5 Address Table:

```
mysql> desc address;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| address_Id | int         | NO   | PRI | NULL    | auto_increment |
| area       | varchar(45) | NO   |     | NULL    |                |
| city       | varchar(45) | NO   |     | NULL    |                |
| state      | varchar(45) | NO   |     | NULL    |                |
| pincode    | int         | NO   |     | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Fig 19. Address Table in Lifelinedb

13.6 Questions Table:

```
mysql> desc questions;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| question_id | int         | NO   | PRI | NULL    | auto_increment |
| question    | varchar(45) | NO   |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
2 rows in set (0.00 sec)
```

Fig 20. Questions Table in Lifelinedb

13.7 Appointments Table:

```
mysql> desc appointments;
+------------+------+------+-----+---------+----------------+
| Field      | Type | Null | Key | Default | Extra          |
+------------+------+------+-----+---------+----------------+
| app_id     | int  | NO   | PRI | NULL    | auto_increment |
| date       | date | NO   |     | NULL    |                |
| doctor_id  | int  | NO   | MUL | NULL    |                |
| patient_id | int  | NO   | MUL | NULL    |                |
| slot       | time | NO   |     | NULL    |                |
| status_id  | int  | NO   | MUL | NULL    |                |
+------------+------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

Fig 21. Appointments Table in Lifelinedb

13.8 Appointment_status Table:

```
mysql> desc appointment_status;
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| status_id | int         | NO   | PRI | NULL    | auto_increment |
| status    | varchar(25) | NO   |     | NULL    |                |
+-----------+-------------+------+-----+---------+----------------+
2 rows in set (0.00 sec)
```

Fig 22. Appointment Table in Lifelinedb

13.9 Schedules Table:

```
mysql> desc schedules;
+------------+------+------+-----+---------+-------+
| Field      | Type | Null | Key | Default | Extra |
+------------+------+------+-----+---------+-------+
| doctor_id  | int  | NO   | PRI | NULL    |       |
| date       | date | NO   | PRI | NULL    |       |
| start_time | time | NO   |     | NULL    |       |
| end_time   | time | NO   |     | NULL    |       |
+------------+------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

Fig 23. Schedules Table in Lifelinedb

14.Snapshots:

14.1 Home Page:

Following snapshot shows the home page for Book My Doc System

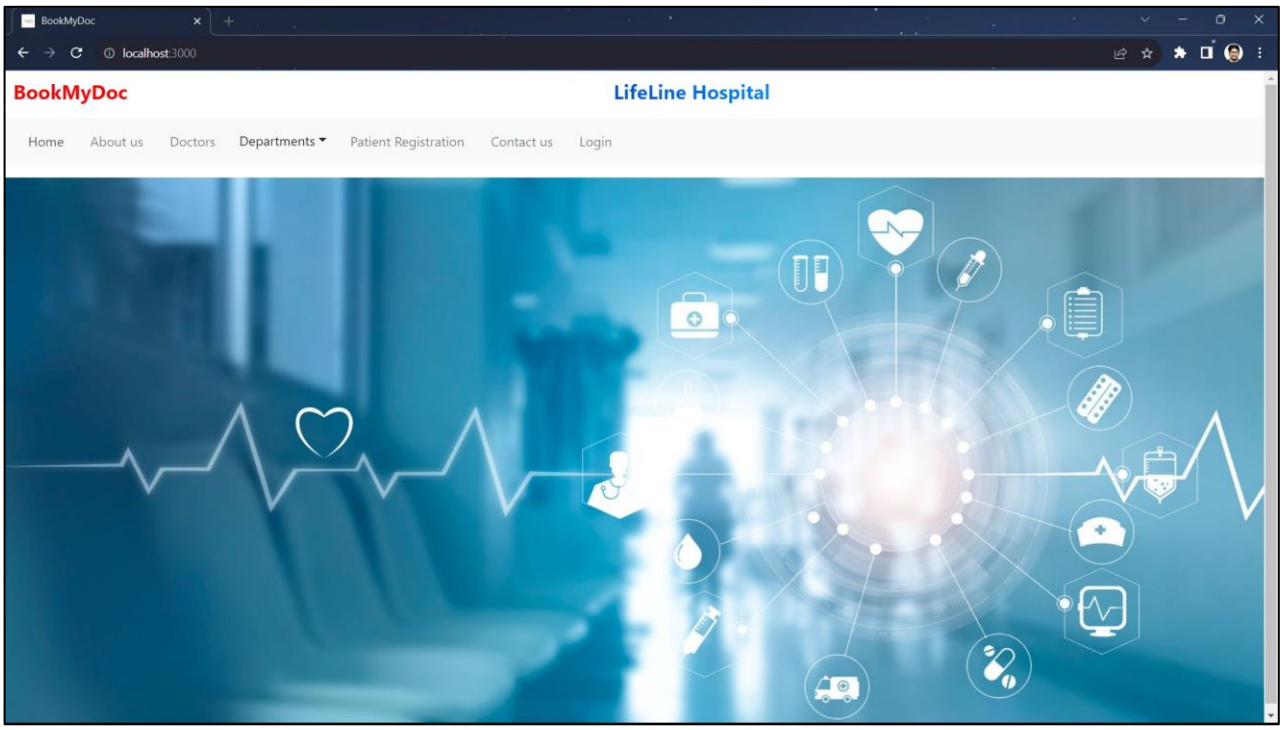

Fig 24. Home Page

This page contains following controls:

- Home
- About us
- Doctor
- Departments
- Patient Registration
- Contact us
- Login

14.2 About us Page:

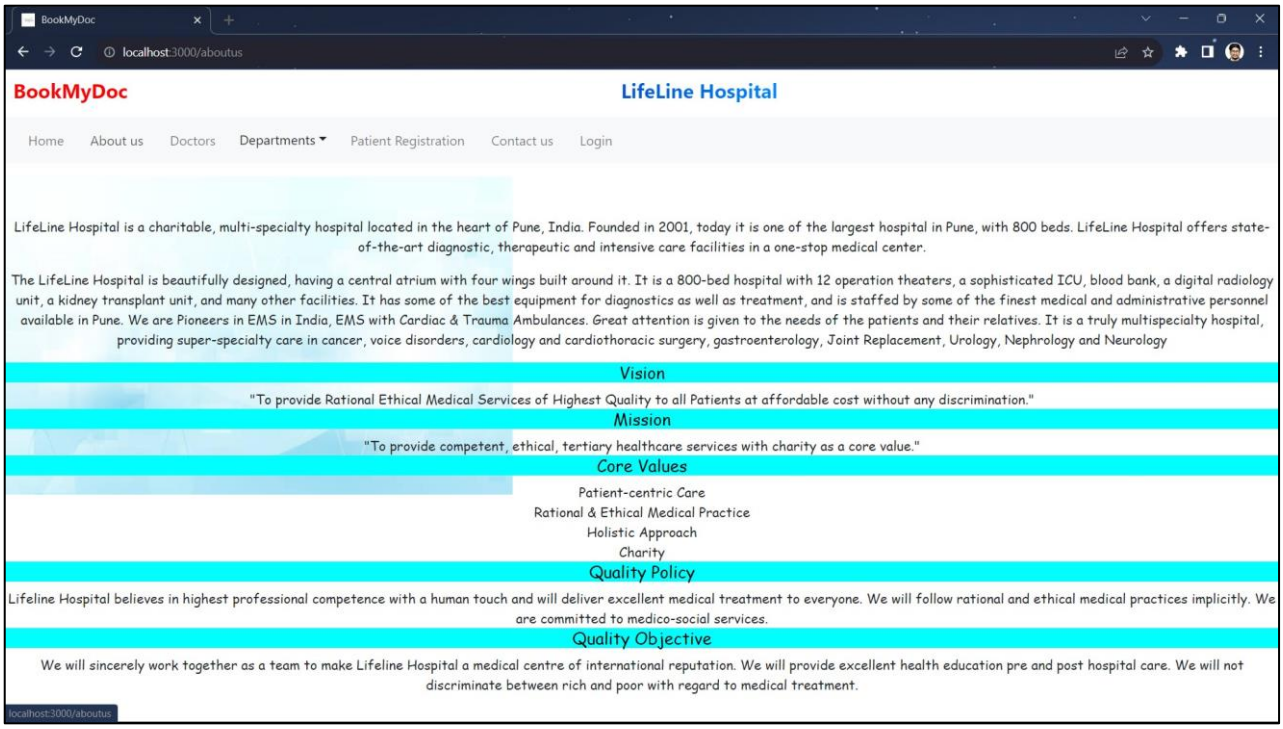

Fig 25. About us Page

14.3 Doctor Page:

Following snapshot shows the list of all doctors in Doctor page for Doctor Appointment System

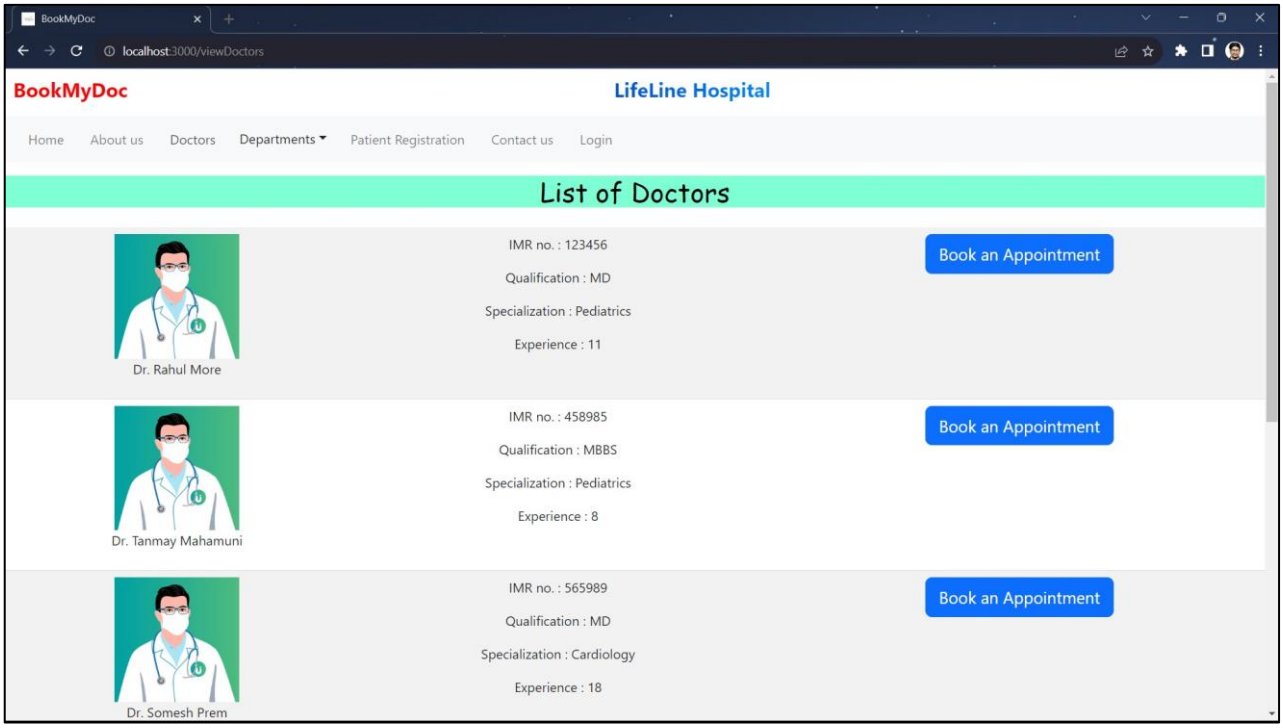

Fig 26. Doctor Home Page

14.4 Department Dropdown Page:

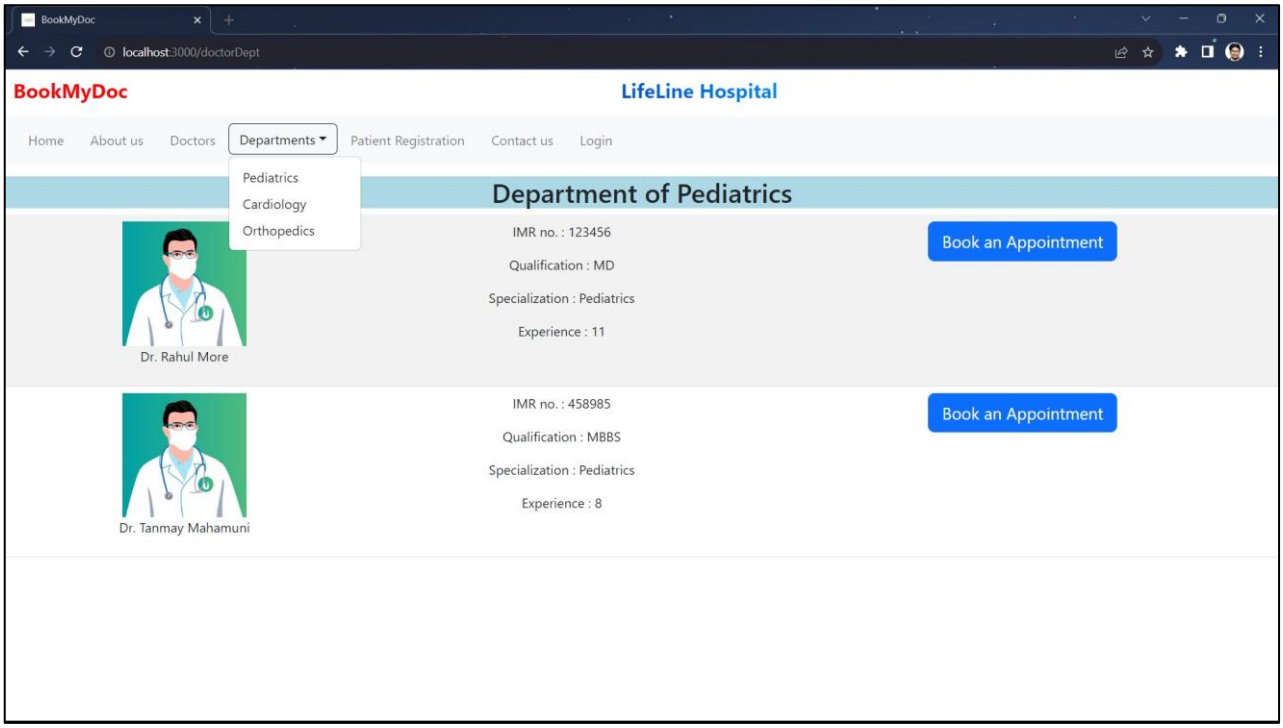

Fig 27. Department Dropdown Home Page

14.4.1 List of All Doctors Department wise:

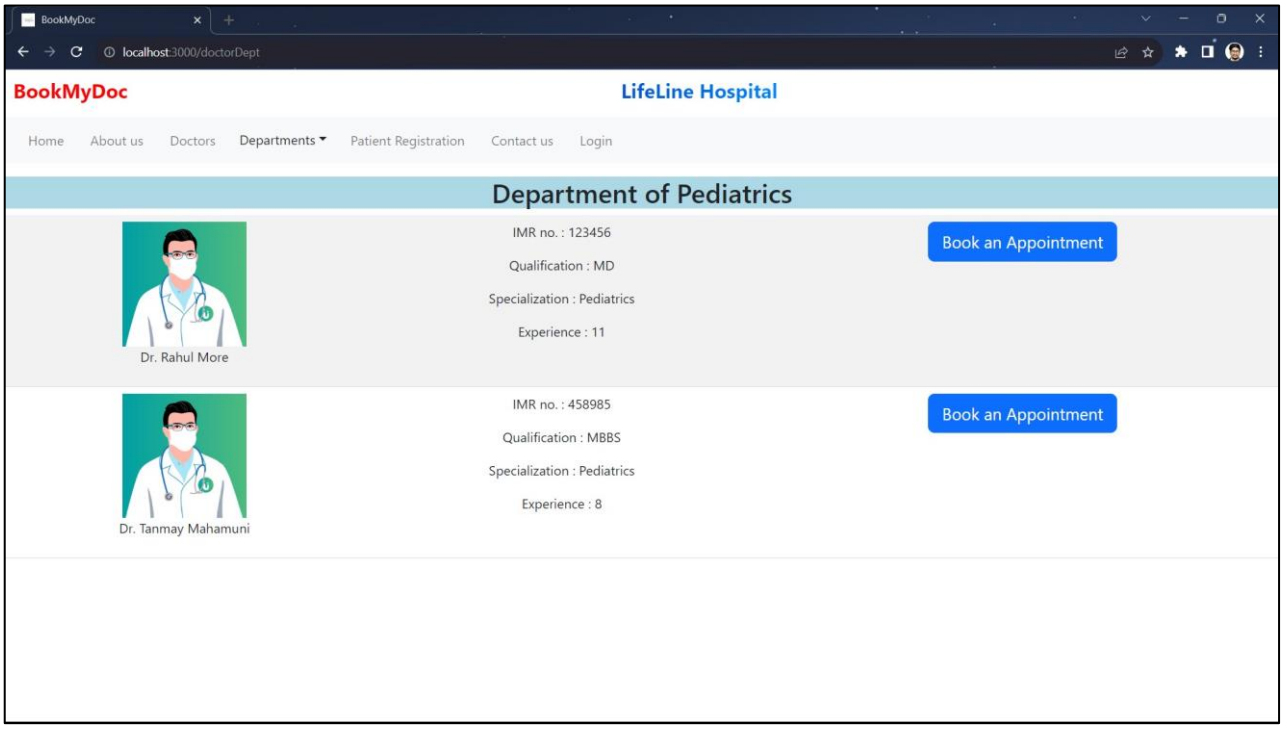

Fig 28. Department wise Doctor Page

14.5 Patient Registration Page:

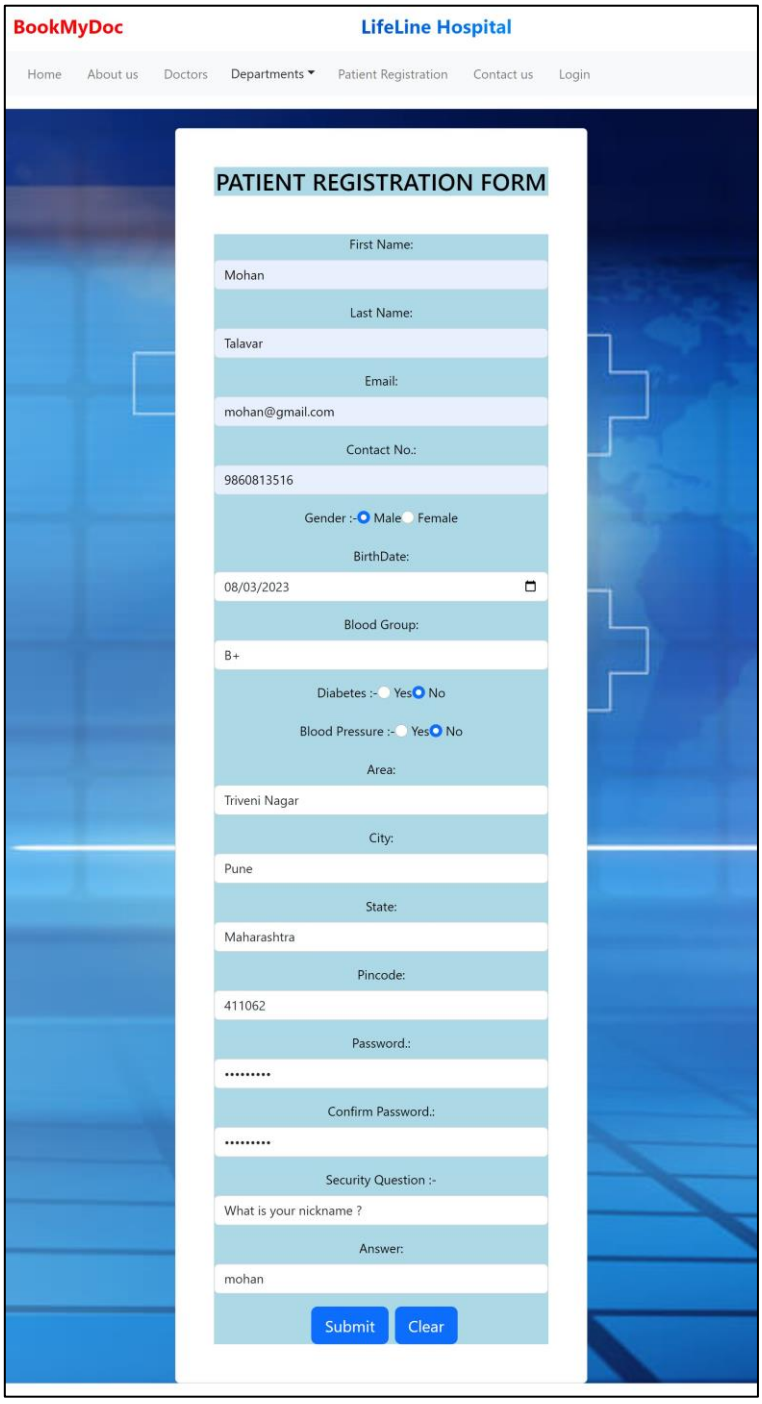

Fig 29. Patient Registration Page

14.6 Contact us Page:

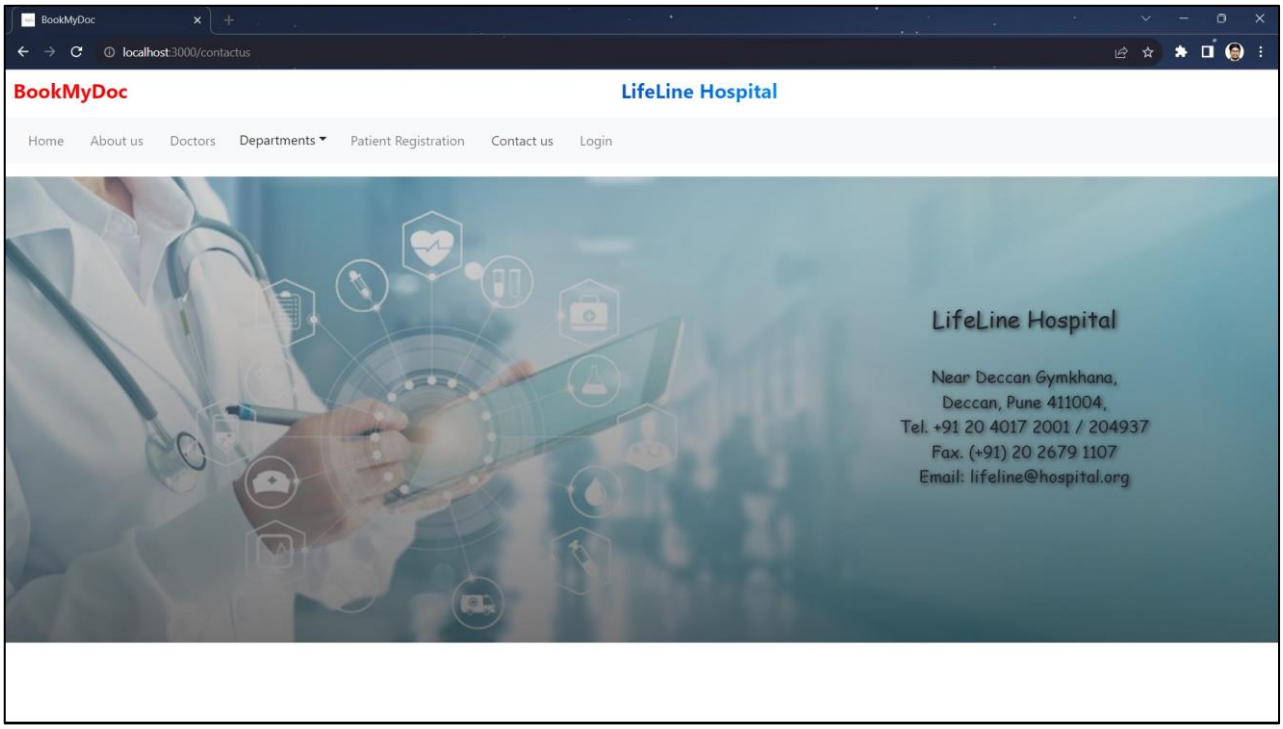

Fig 30 . Contact us Page

14.7 Login Page:

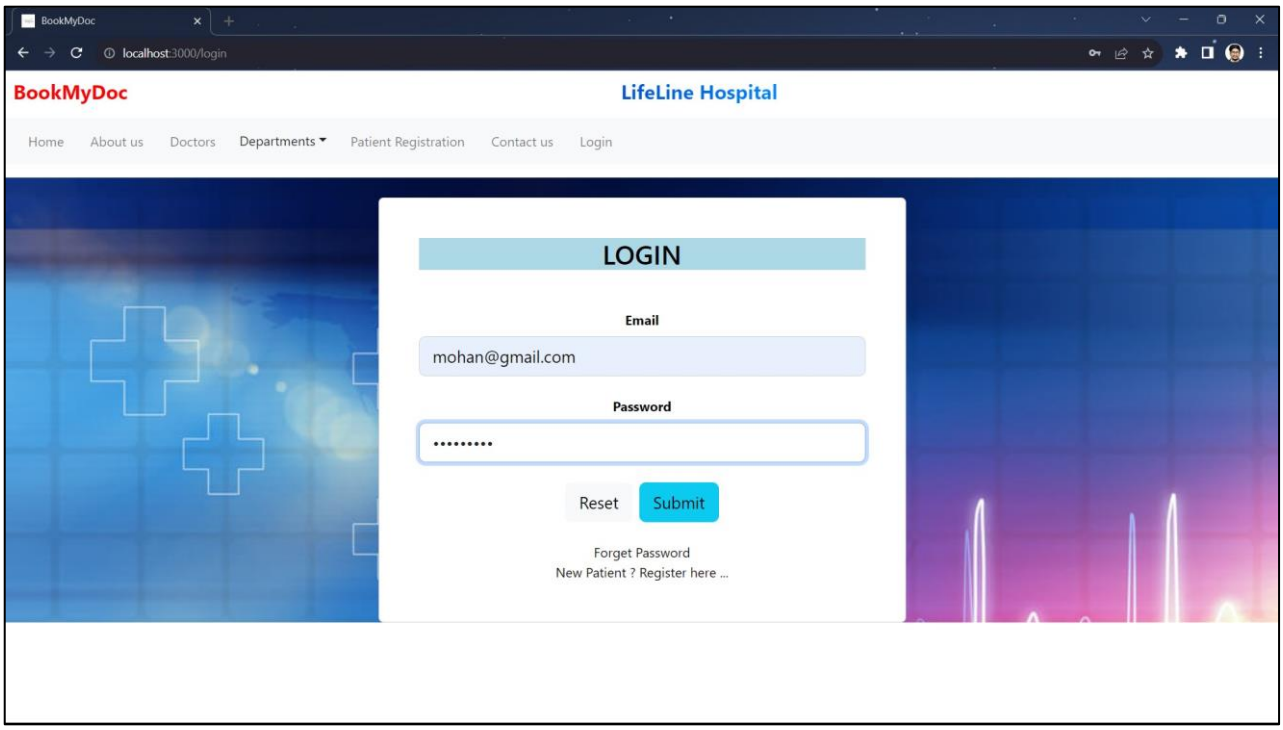

Fig 31. Login Page

14.7.1 Forget Password Page:

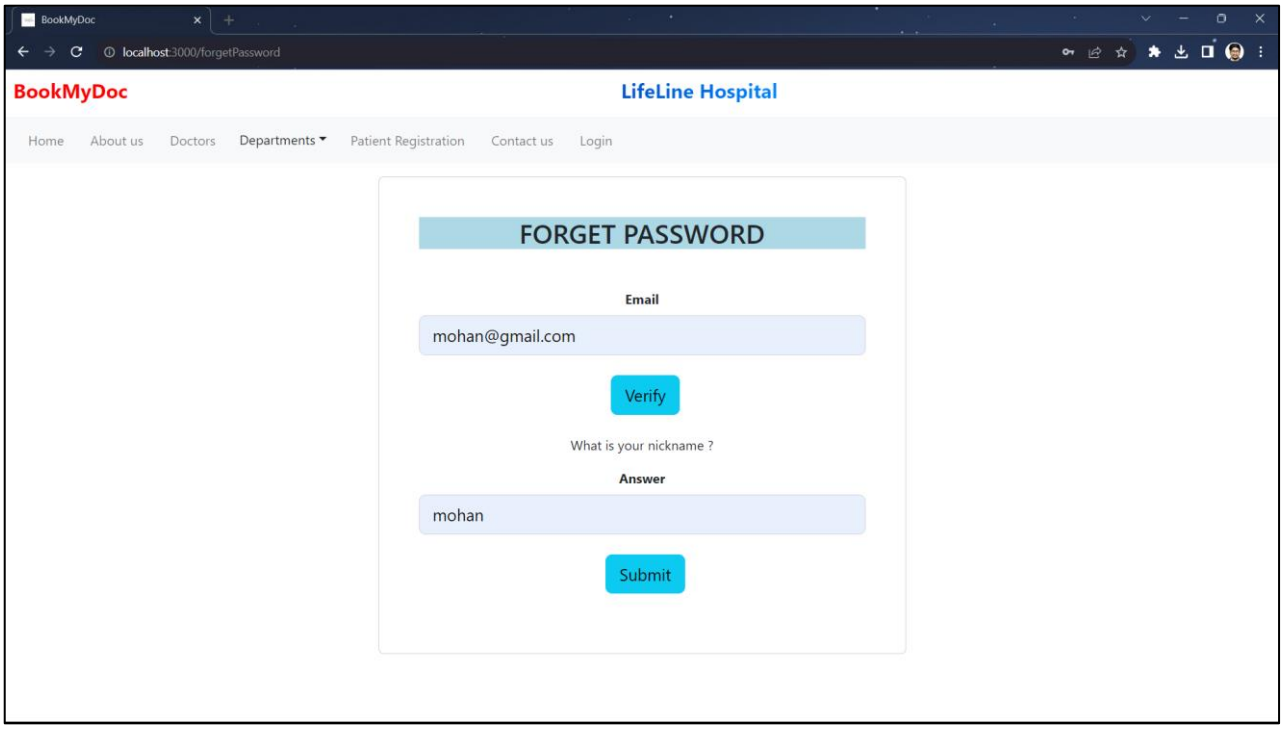

Fig 32. Forgot Password Page

14.7.2 Change Password Page:

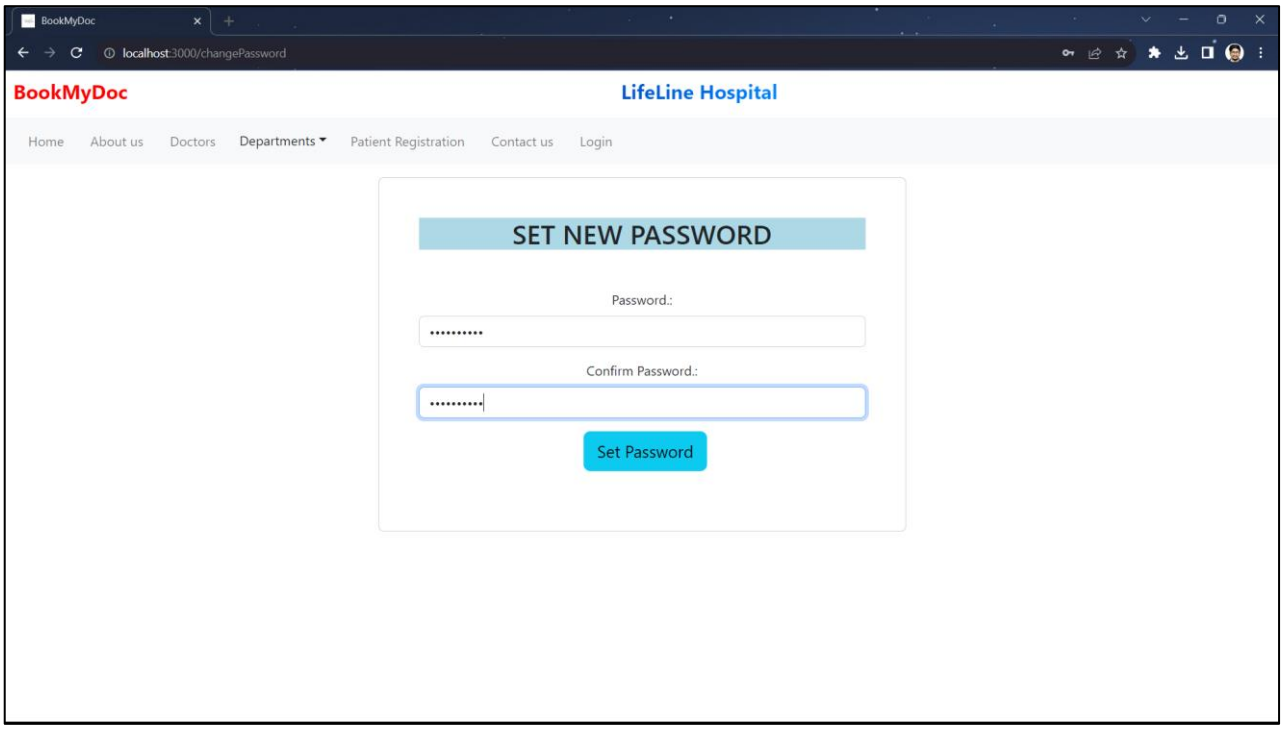

Fig 33 . Change Password Page

14.8 Admin View Doctors Page:

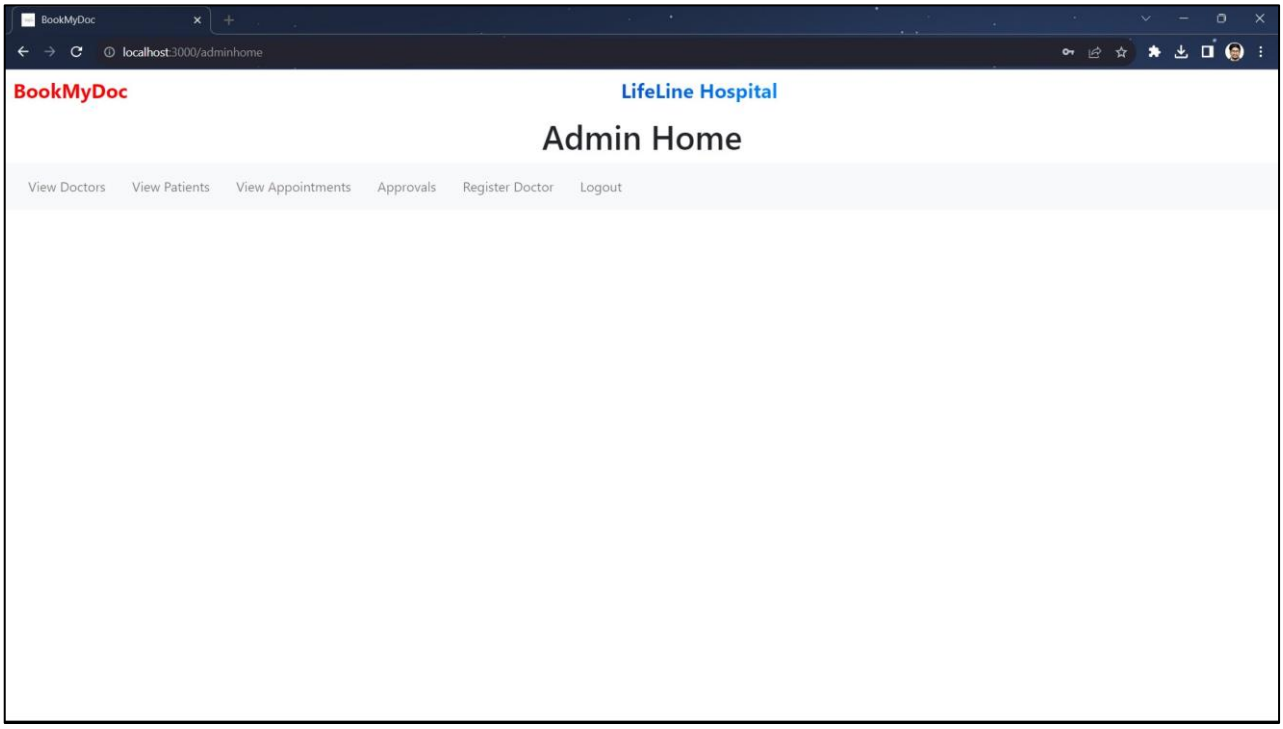

Fig 34. Admin view doctors page

14.8.1 Admin View Patients Page:

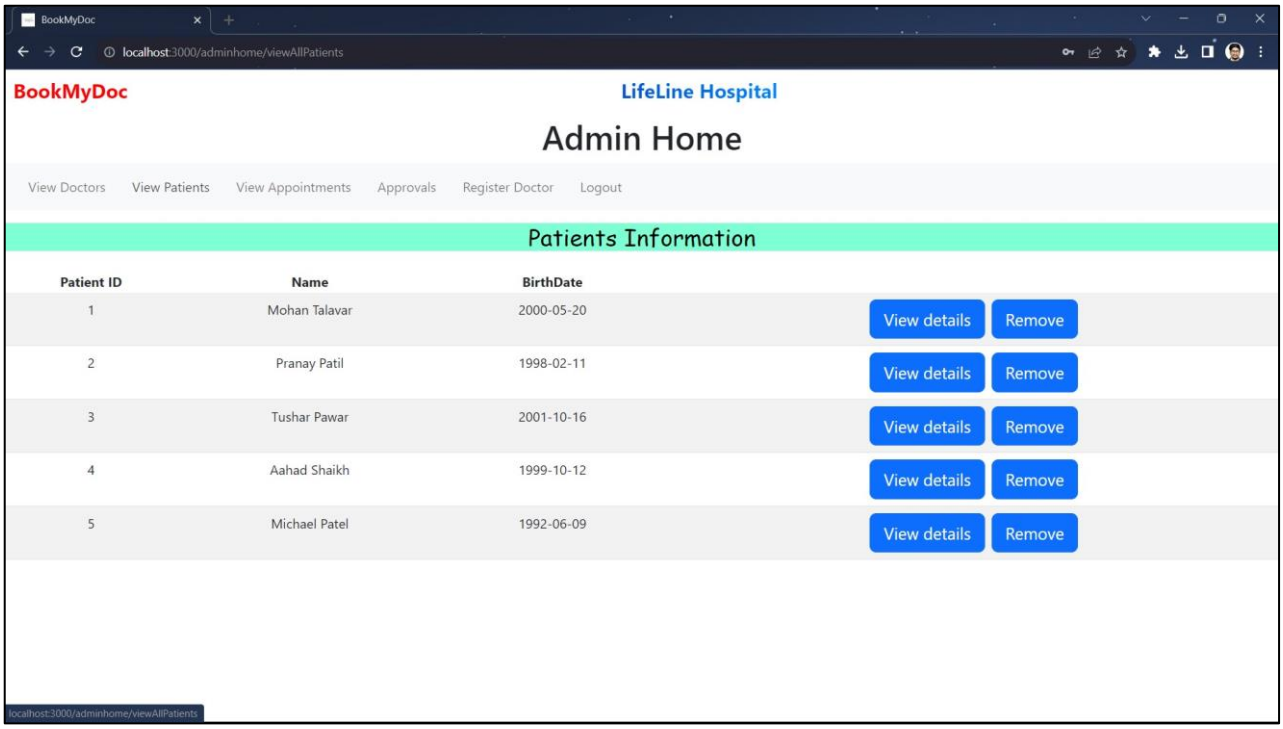

Fig 35. Admin view patients page

14.8.2 View Details of Patients Page:

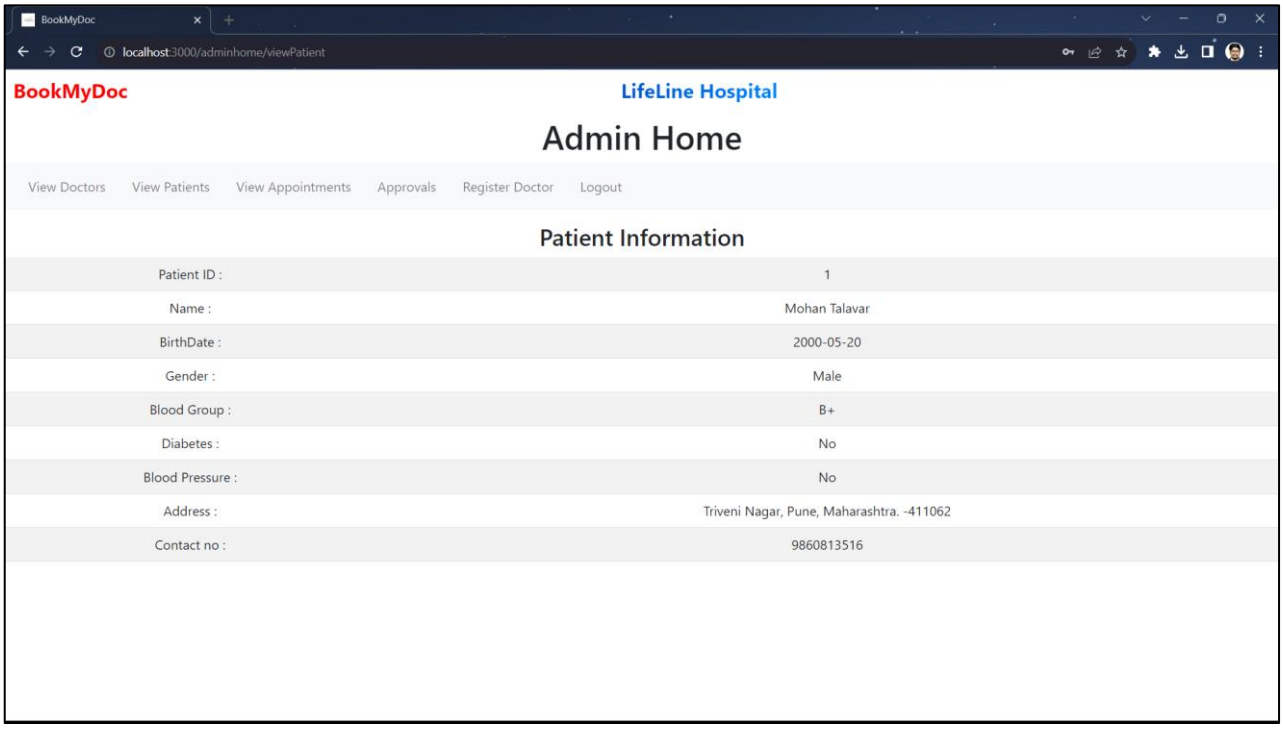

Fig 36. View Details of Patients Page

14.8.3 View Appointments Page:

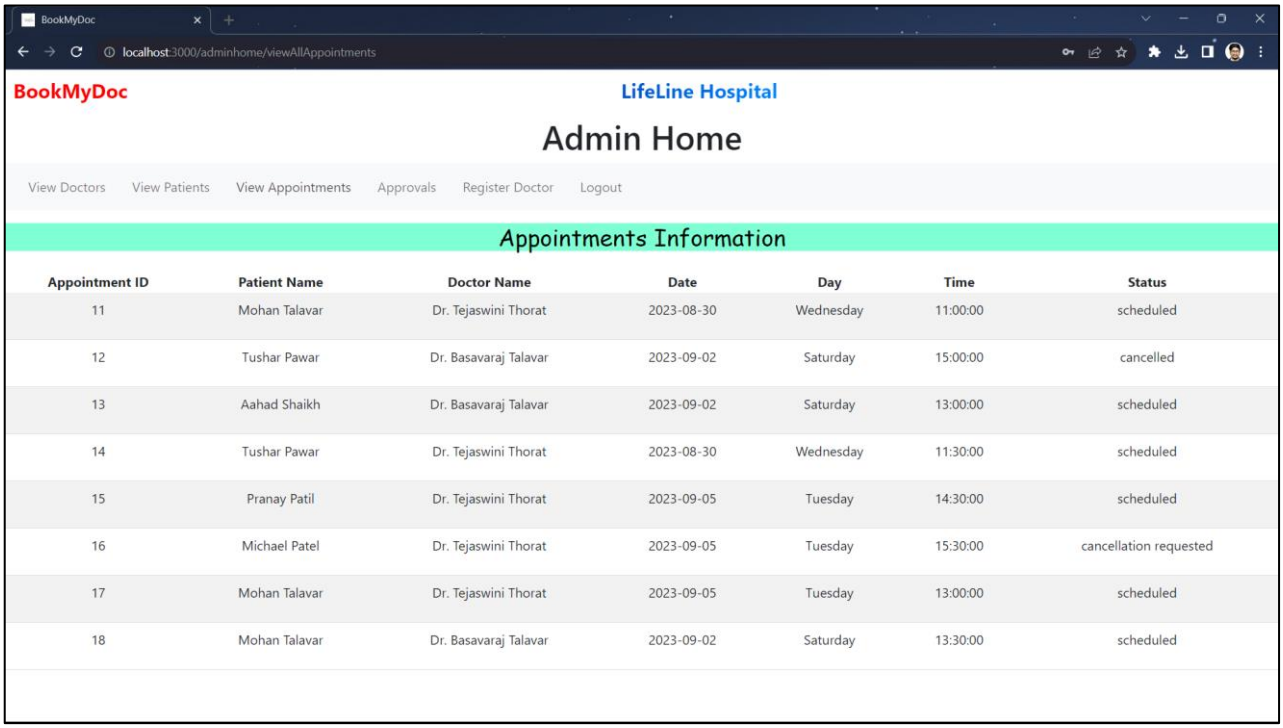

Fig 37. View Appointments Page

14.8.4 Admin Cancel Request Approval Page:

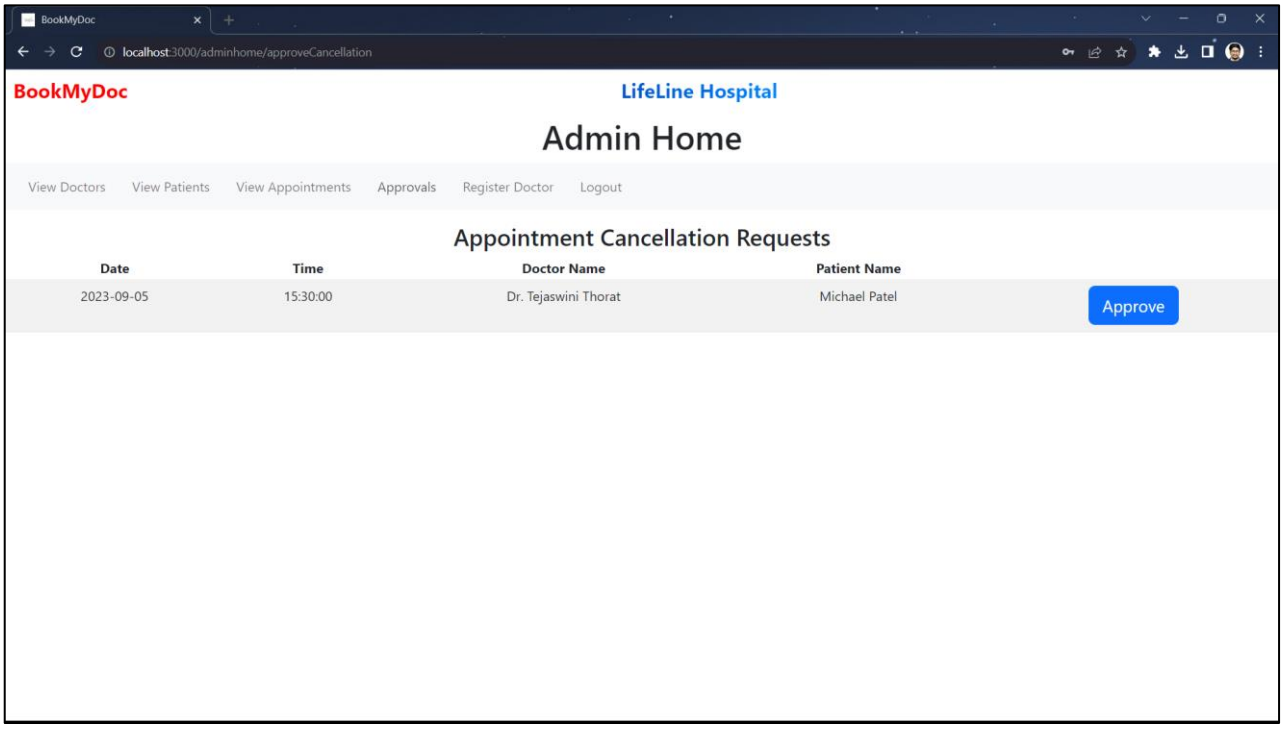

Fig 38. Admin cancel request approve page

15.8.5 Doctor's Registration Page:

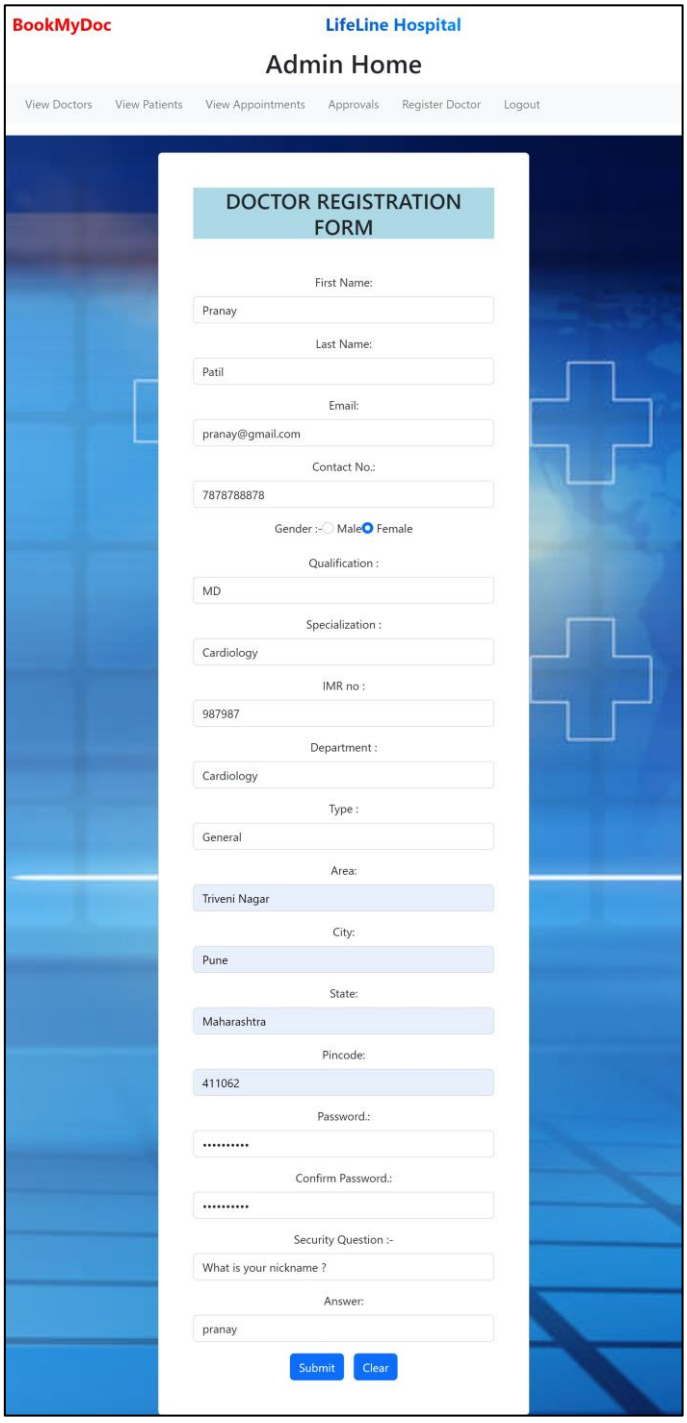

Fig 39. Doctors registration page

14.9 Doctor Home Page:

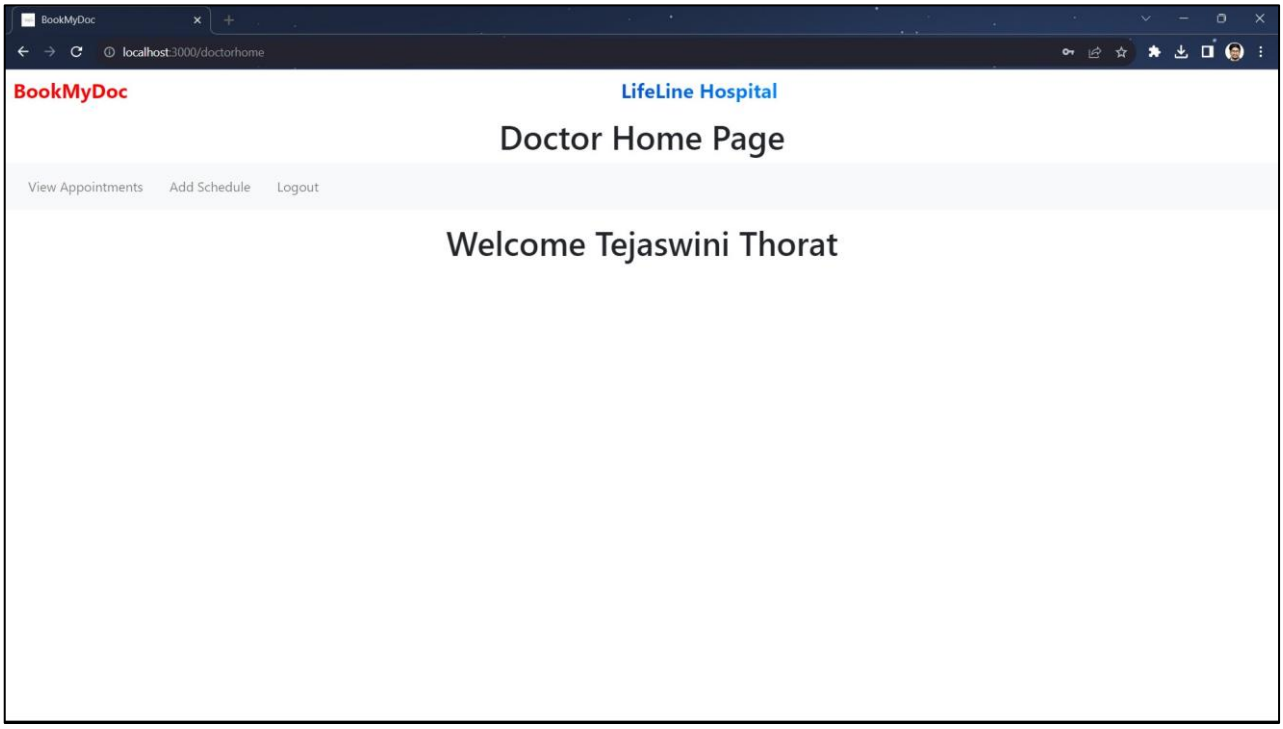

Fig 40. Doctor home page

14.9.1 Add Schedule Page:

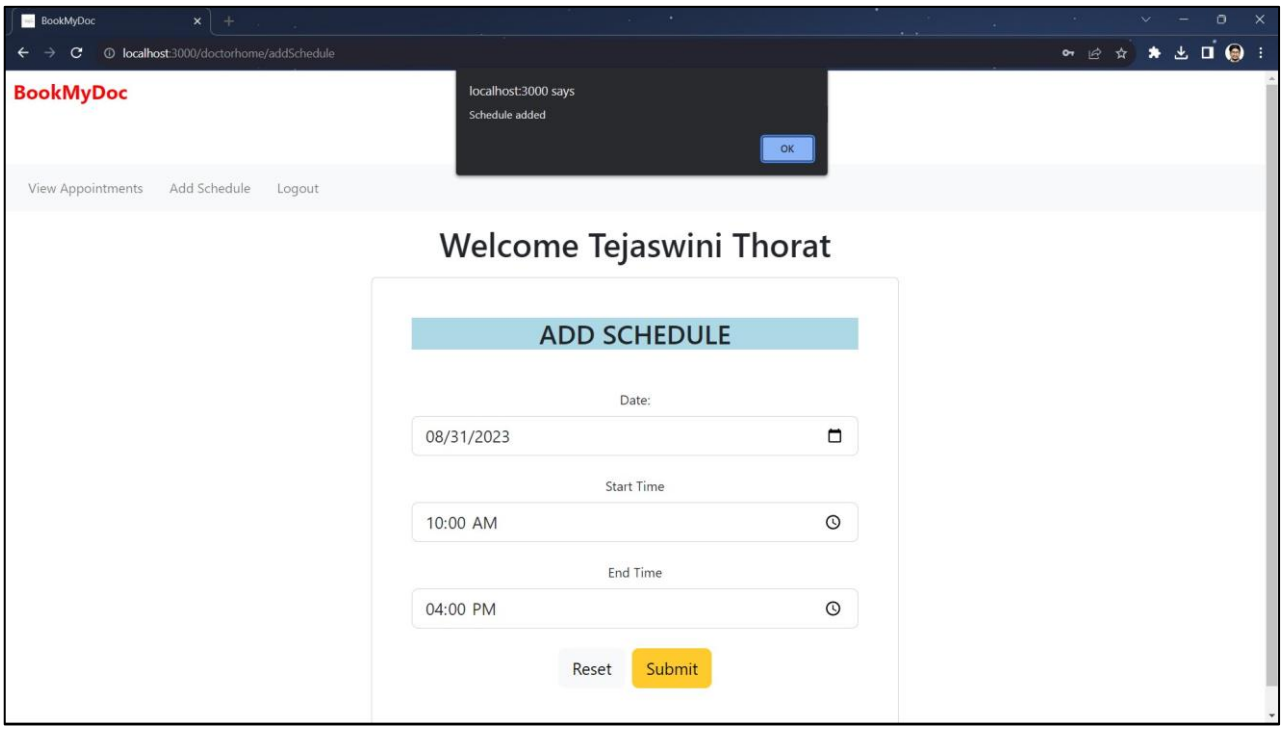

Fig 41. Doctor's add schedule page

14.9.2 View Appointments Page:

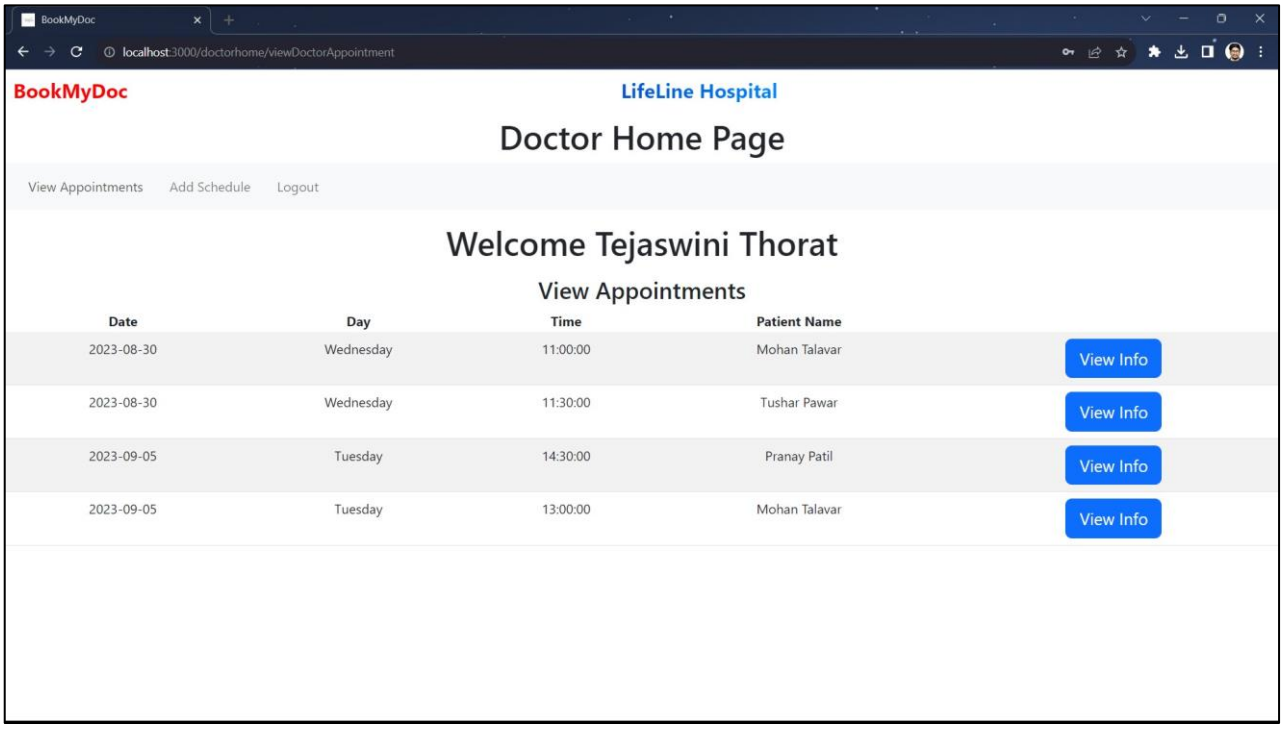

Fig 42. Doctors view appointments page

14.10 Patient Home Page:

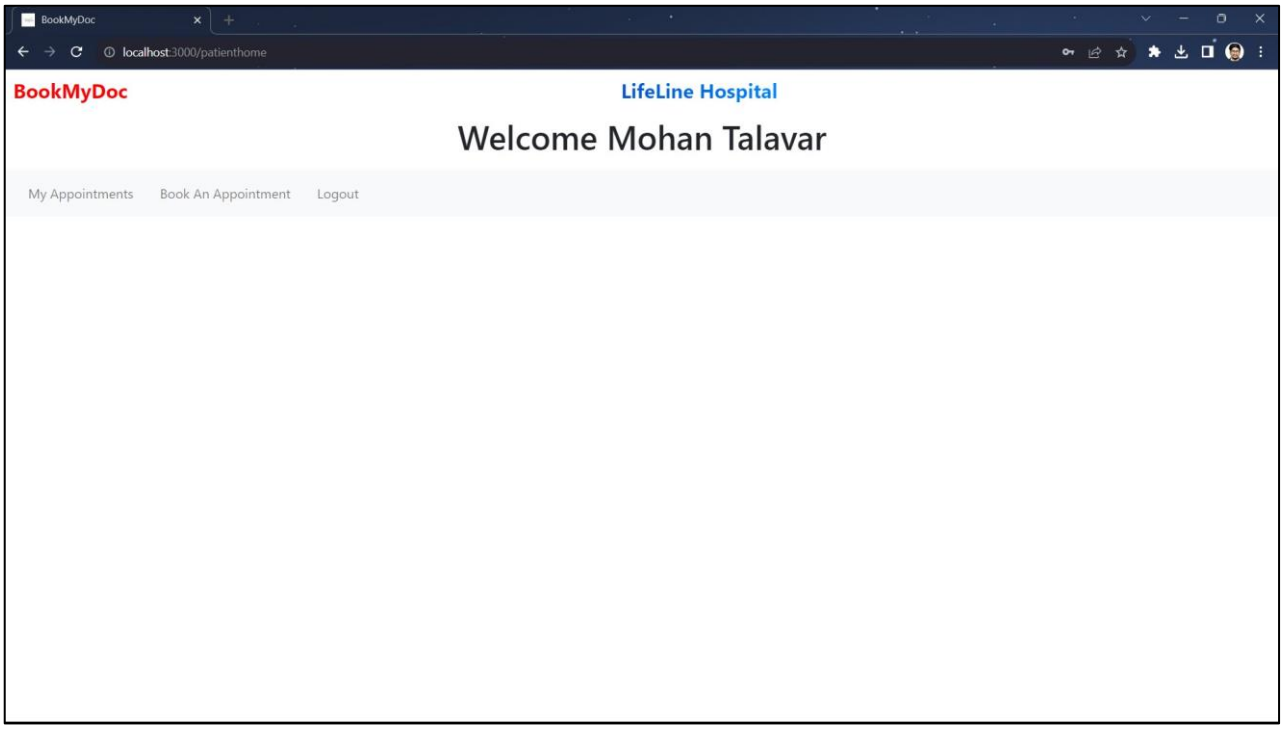

Fig 43. Patient home page

14.10.1 Book An Appointment Page:

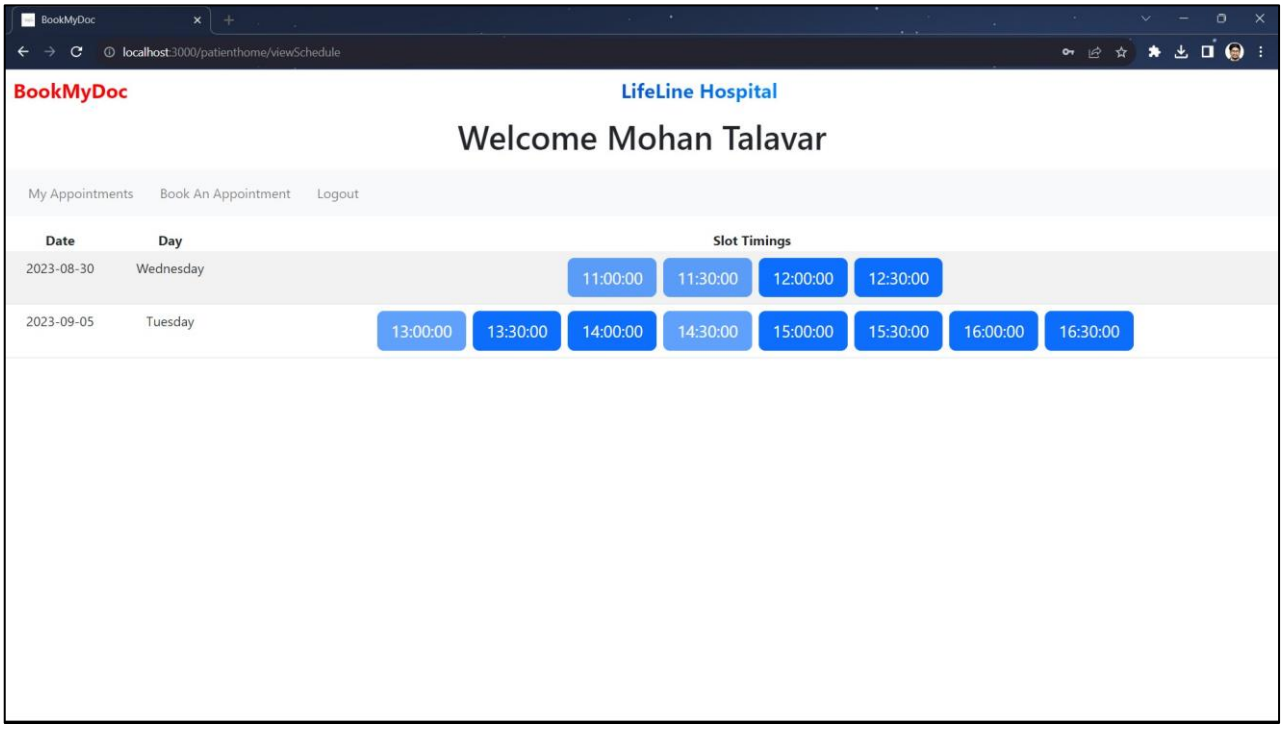

Fig 44. Patient book appointment page

14.10.2 Patient Appointments Page:

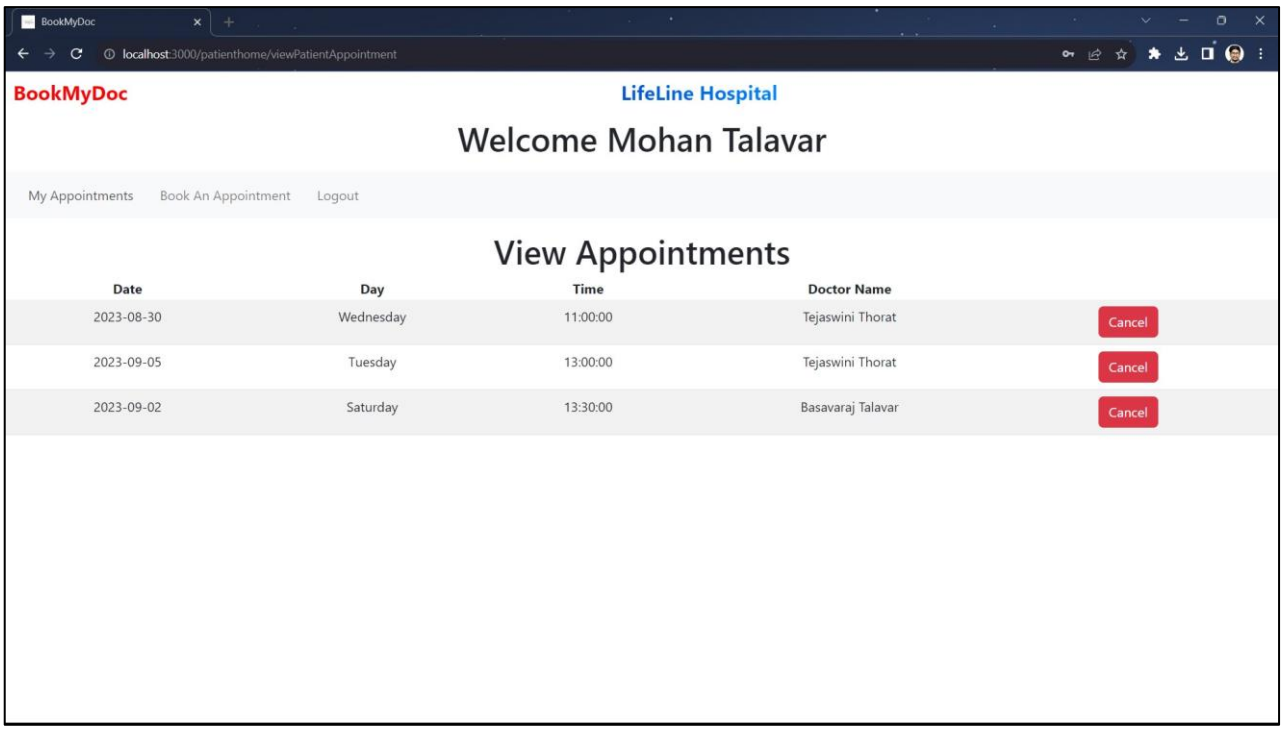

Fig 45. Patient my appointment page

15.CONCLUSION AND FUTURE SCOPE

- Book My Doc System provides a better platform to get connected with doctors and patients efficiently.
- Use of this system will be specific for one hospital or its franchises.
- Patients will be able to book an appointment 24 hrs prior to the schedule.
- Doctors will be able to provide their schedule at least for one week in advance.
- Booking options will be available 24 hours, 7 days. Doctors can modify their schedule at least 48 hours in advance.

16.REFERENCES

a. Spring Boot Official Website https://spring.io/
b. React-Js Official Website https://reactjs.org/
c. Spring Boot Official Website https://www.baeldung.com/
d. www.w3school.com
e. www.javatpoint.com