Dharmsinh Desai University, Nadiad
Faculty of Technology

Department of Computer Engineering
B.Tech - CE - Semester – V

Subject: (CE – 520) Advanced Technologies

Project Title: FindMeDoc (Doctor Appointment System)

Rinku Rathva CE 118 [21CEUTG110]
Vansh Shah

Guided by: Prof. Jigneshkumar K. Shah

Dharmsinh Desai University, Nadiad

Department of Computer Engineering

Faculty of Technology

CERTIFICATE

This is to certify that the project work carried out

in the subject of

Advanced Technologies

is the bonafide work of

Rinku Rathva CE 118 (21CEUTG110)

Vansh shah

of B. Tech. Semester-V Computer Engineering during the academic year **2023-2024**.

Prof. Jigneshkumar K. Shah
Associate Professor
Computer Engg. Department
Faculty of Technology
D.D.U., Nadiad

Dr. C. K. Bhensdadia
Professor and HoD
Computer Engg. Department
Faculty of Technology
D.D.U., Nadiad

Contents

Abstract

The MERN-based Doctor Appointment System is a modern, efficient, and user-friendly application designed to streamline and enhance the process of scheduling and managing doctor appointments. In an era where digital healthcare solutions are becoming increasingly essential, this system leverages the power of the MERN (MongoDB, Express.js, React, Node.js) stack to provide a comprehensive platform for patients, doctors, and administrators.

Brief Introduction :

- This system harnesses the power of the MERN stack (MongoDB, Express.js, React, Node.js) to revolutionize the way patients and doctors schedule and manage appointments. With the goal of optimizing healthcare processes, enhancing patient-doctor interactions, and providing a seamless experience, this platform caters to the evolving demands of modern healthcare.
- Patients and doctors can benefit from features such as easy appointment scheduling, real-time notifications, secure data management, and personalized dashboards. Additionally, administrators can ensure the smooth operation of the system, making it a comprehensive solution for healthcare facilities.

- **Technology and tools used**

- **Technology:**
  MERN Stack (MongoDB, Express.js, ReactJS, Node.js)

- **Tools**
  Git and GitHub
  Visual studio code
  MongoDB Compass

Software Requirement Specifications (SRS)

1. Introduction

1.1 Purpose

The purpose of developing an appointment booking website is to provide a convenient way to patients for booking an appointment. This platform allows patients to find specialists in respective fields.

1.2 Document Conventions

1.3 Intended Audience and Reading Suggestions

The target audience is patients, current or potential, who need to find a doctor and make an appointment quickly, conveniently, and securely. Also, local doctor's who find using an app convenient to manage their appointments and prescriptions.

1.4 Product Scope

The aim of this project is to create an online doctor appointment system. Patients can be able to book doctor's appointments anywhere, this can be done via web browser. Doctors can login to this system, view appointments and set available time for appointment thereby making it more convenient for them. Administrators also have access to the website and are able to change information of the website and have access to the database.

1.5 References

Fundamentals of Software Engineering by Rajib Mall, PHI Learning

2. Overall Description

2.1 Product Perspective

This program is a stand-alone system that manages appointment scheduling and also the addition of patient examinations. The idea was developed out of people's needs to find the best doctor, to save time, maintain their health history and past prescriptions.

2.2 Product Functions

Functionality of the system

For Admin

1. Admin can log into the system.
2. Admin can manage the data of all the doctors and patients.

For patients:

1. Patients can log into the system.
2. Patients can share their medical history .
3. Patients can review a doctor's profile.
4. Patients can book an appointment with the doctor.
5. Patients can check their appointment schedule.

For Doctor:

1. Doctors can log into the system.
2. Doctors can share their qualifications and experience.
3. Doctors can share their available time slots.
4. Doctors can review patient’s appointments.

2.3 User Classes and Characteristics

- **Admin**: The system should be known to the administrator. The Admin has complete control of the system. His ability to add a doctor is one of his most important functions.
- **Doctors**: The doctor must have a good understanding of how to use the system. The appointments that have been scheduled can be viewed by doctors. Details and information on the patient can also be viewed. The results of patient visits can be included.

- **Patients**: The patient has access to clinic data. He can save, view, and analyze his medical examinations, as well as learn about doctors and appointments and book an appointment online.
- **Casual users**: The clinic's information is available to anyone. He has access to basic information on doctors and the latest news. He can not, however, make an appointment without first logging in to the website.

2.4 Operating Environment

The system will be compatible with all popular web browsers, including Chrome,Firefox, Safari, Edge, and Internet Explorer. The machine should have sufficient ram and speed of internet connection.

2.5 Design and Implementation Constraints

Scalability is one of the most important *advantages* of *MongoDB*. As seen, *MongoDB* uses “sharding”, which expands the storage capacity. For communication HTTPS should be used as it provides secure data transfer.

2.6 User Documentation

2.7 Assumptions and Dependencies

- The system will assume that users have a reliable internet Connection.
- It is assumed that the user will know how to use the system.
- The system depends on the availability of a stable and secure hosting environment to host the system and store users data.
- It is assumed that patients know what kind of disease they have so they can consult an expert.

3. External Interface Requirements

3.1 User Interfaces

3.2 Hardware Interfaces

The system is compatible with a wide range of hardware, including desktop computers, laptops, and mobile devices. The system will be able to run on popular operating systems such as Windows, MacOS, and Linux.
RAM: 4GB
Hard Drive Storage Needed: 2GB
Other Hardware Requirement: None

3.3 Software Interfaces

The MOngoDB database connection interface is the part of the system that manages the communication and interaction between the system and the underlying database.The frontend is developed using React.js, HTML and CSS.

3.4 Communications Interfaces

This uses standard network protocols, such as HTTPS.The system will comply with industry standards for data encryption and secure communication to ensure that user information and appointments are protected from unauthorized access or attack.

4. System Features

R.1 : Manage Doctor Details

R.1.1 : Update Doctor details

Description : After log in correctly, the data of the Doctor will be shown.
The confirmation message will be displayed to the doctor after editing the data.

Input : Edit the doctor details like Educational Qualifications, available time slots and Address.

Output : Confirmation message to the doctor.

R.2 : Reviewing the patient's appointment

R.2.1 : Check the appointment

Description : After going to the appointment's section, the doctor can view the upcoming appointments.
Input : Select the appointment option.
Output :Appointments will be displayed.

R.2.2 : Confirm the appointment

Description : Doctors can accept or reject an appointment.
Input :Doctor will select accept or decline option.
Output : A mail will be generated and sent to the patient.

R.3 : Schedule Re-Appointment

R.3.1 : Upload the date

input: enter the date.
output : Mail will be generated with date and sent to the patients.

R.4 : Upload the Patient Profile

R.4.1 : Share previous health report

Description : Patients can share their health related information in their profile.
Input : select the 'upload' option.
Output : confirmation message will be displayed after uploading the health report.

R.5 : Book an appointment

R.5.1 : Searching the Doctor for an appointment

Input : select the diseases and time slots option.
Output : Doctor's profile will be displayed according to the patient's search.

R.5.2 : Book the appointment

Input : Select book appointment option

Output : The message will be displayed to patient

5.Non-Functional Requirements

5.1 Reliability

- The system should be available when requested for service by users.
- The system should have a very low failure rate.

5.2 Performance

The system must have a good response time.
The system should be able to achieve a lot in a specified amount of time.
The system must run error free while operating with a huge set of data.

5.3 Security

- All external communications between the system's data server and clients must be encrypted.
- The access permissions for system data may only be changed by the system's data administrator.

5.4 Usability

- The system should include well-structured user manuals.
- The system should have Informative error messages.
- Efficient help facilities.
- The system should have well-formed graphical user interfaces.
- The system should be user friendly.

5.5 Safety

- The system should have a recovery process for handling errors and exceptions.

5.6 Supportability

- The system should be able to be transferred from one environment to another.
- The system should be easy to maintain

- The system should be able to be used on multiple platforms.

Database Design
Entities:

Appoinment :


```
{
  "_id": {
            "$oid": ": A unique identifier for each user"
  },
  "userId": "The user's name or ID(e.g., "xyz") ",
  "doctorId": "The doctor's name or ID(e.g., "xyz").",
  "doctorInfo": "doctor info",
  "userInfo": "user info",
  "date": "2023-09-29T18:30:00.000Z",
  "status": "approved/notapproved",
  "time": "2023-09-22T13:30:00.000Z",
  "createdAt": {
            "$date": "2023-09-22T04:43:02.835Z"
  },
  "updatedAt": {
            "$date": "2023-09-22T04:43:21.005Z"
  },
  "__v": 0
};
```


Doctor : (e.g)


```
{
  "_id": {
            "$oid": "650d1b119e5c40a424f767cf"
  },
  "userId": "650d1ae19e5c40a424f767c4",
  "firstName": "vansh",
  "lastName": "shah",
  "phone": "123456789",
  "email": "v@gmail.com",
  "address": "aa",
  "specialization": "aa",
  "experience": "12",
```

```
  "feesPerCunsaltation": 200000,
  "status": "approved",
  "timings": [
        "18:00",
        "21:00"
  ],
  "createdAt": {
        "$date": "2023-09-22T04:41:53.058Z"
  },
  "updatedAt": {
        "$date": "2023-09-22T04:42:18.243Z"
  },
  "__v": 0
}
```


User : (e.g)


```
{
  "_id": {
    "$oid": "650d1ab89e5c40a424f767b6"
  },
  "name": "a",
  "email": "a@gmail.com",
  "password": "$2a$10$amlRxTx7weSuCl0ahW0BxuiKsw72GJ0L0sQ7sieytm6eR/HCGasv6",
  "isAdmin": true,
  "isDoctor": false,
  "notifcation": [
    {
      "type": "apply-doctor-request",
      "message": "vansh shah Has Applied For A Doctor Account",
      "data": {
        "doctorId": {
          "$oid": "650d1b119e5c40a424f767cf"
        },
        "name": "vansh shah",
        "onClickPath": "/admin/docotrs"
      }
    }
  ],
  "seennotification": [],
  "__v": 0
}
```


Key Relationship :

- One-to-Many Relationship (Users to Appointments): A user (patient) can have multiple appointments, but an appointment is associated with a single user. This relationship is established by having a user ID in the Appointments table.
- One-to-Many Relationship (Doctors to Appointments): A doctor can have multiple appointments, but an appointment is associated with a single doctor. This relationship is established by having a doctor ID in the Appointments table.
- One-to-One Relationship (Users to Patients/Doctors): While a user (patient or doctor) can have multiple appointments, each user corresponds to one record in the Patients or Doctors table. This relationship is established through user IDs in the Patients and Doctors tables.

Testing

Functional Testing

Test Case 1: User Registration
Input: User provides valid registration details (name, email, password).
Expected Output: A new user profile is created, and the user is logged in successfully.

Test Case 2: User Login
Input: User provides valid login credentials (email and password).
Expected Output: The user is authenticated and gains access to their profile

Test Case 3: Apply for a doctor
Input: User provides necessary credentials and sends a request to apply for a doctor.
Output: A request is sent to the admin.

Test Case 4: Appropriate action is taken for doctor request.
Input: All doctors are retrieved and their request is accepted or rejected.
Output: Appropriate notification is sent to the user applying for doctor.

Test Case 5: Book an appointment
Input: User selects a doctor and books an appointment in available time slots.
Output: An appointment request is sent to the respective doctor.

Test Case 6: Take appropriate action for appointment.
Input; All appointments for a doctor are retrieved.
Output: A notification is sent to the user showing whether the appointment is accepted or rejected.

Test Case 7: See all notifications
Input: All notifications are retrieved

Output: User reads all notifications and marks them as read.

Performance Testing

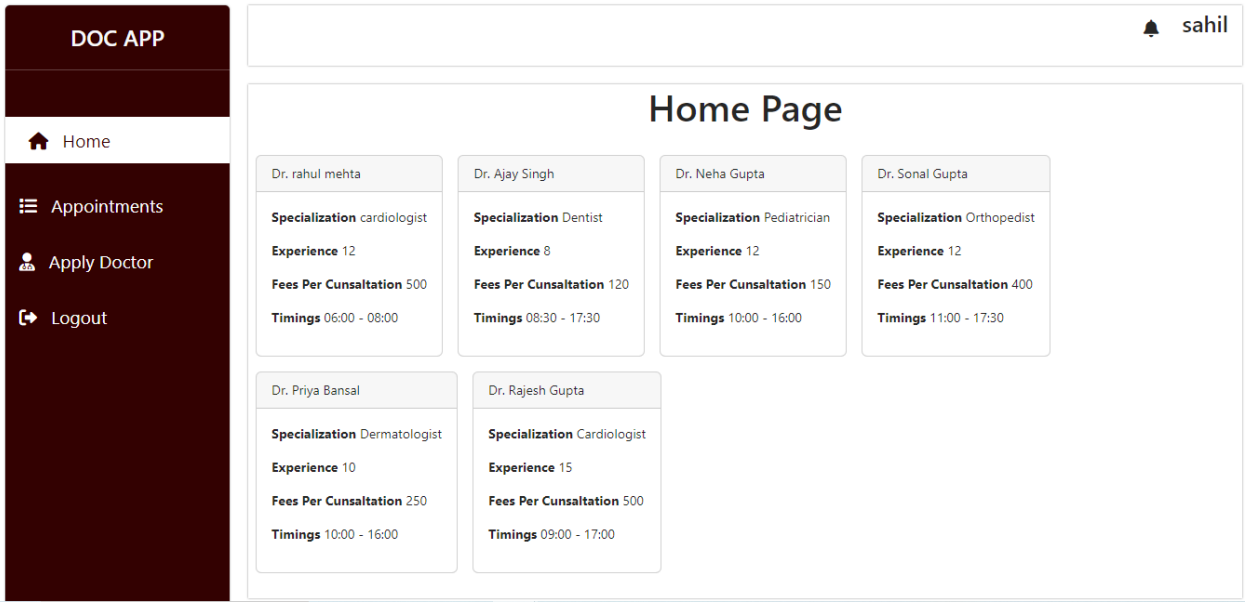
Test Case 8: Loading time
Input: User accesses all available doctors.
Output: Doctors loaded promptly.

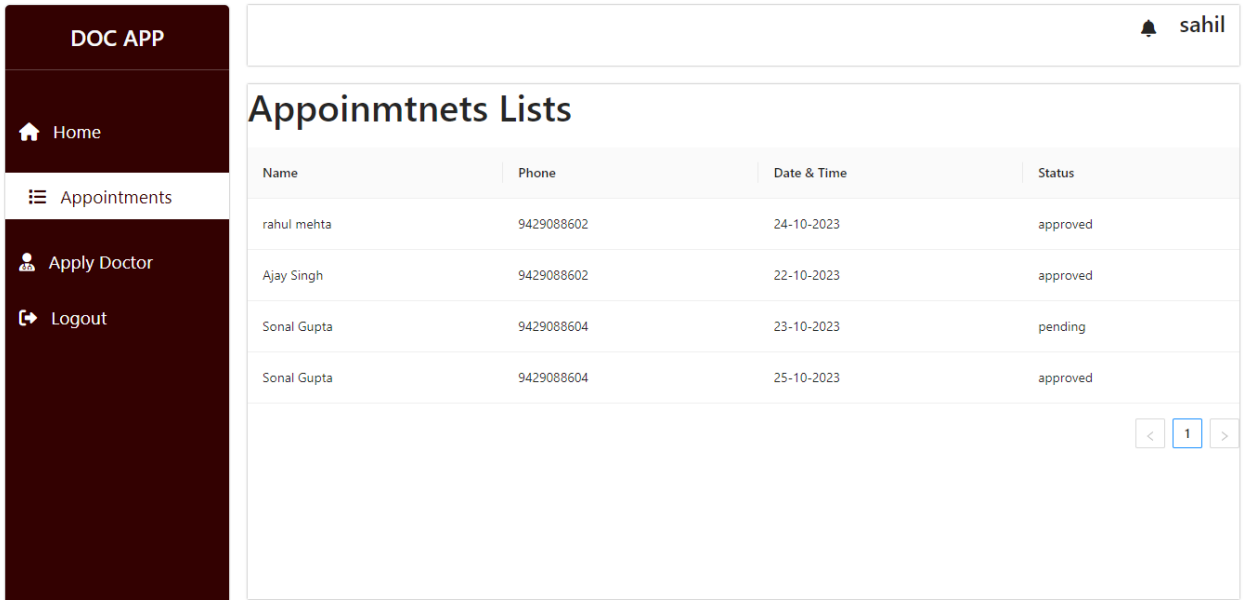
Security Testing

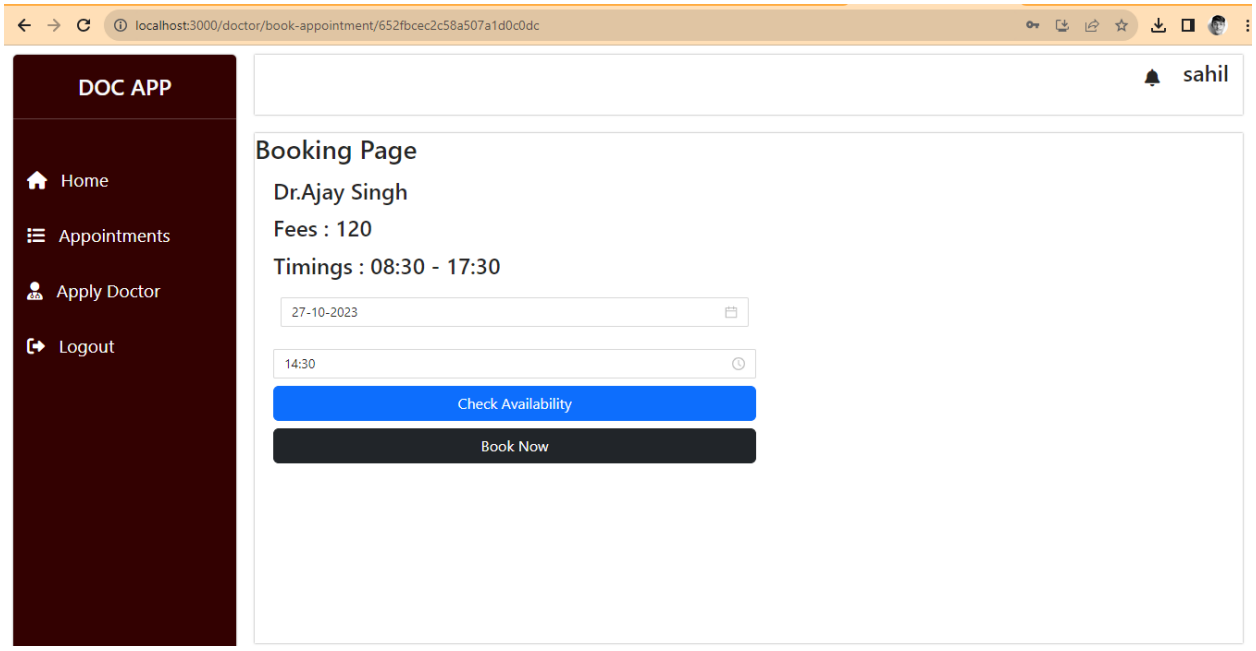
Test Case 9: Data Privacy
Input: User submits personal information.
Expected Output: Data privacy measures protect user information, and the application complies with data privacy regulations.

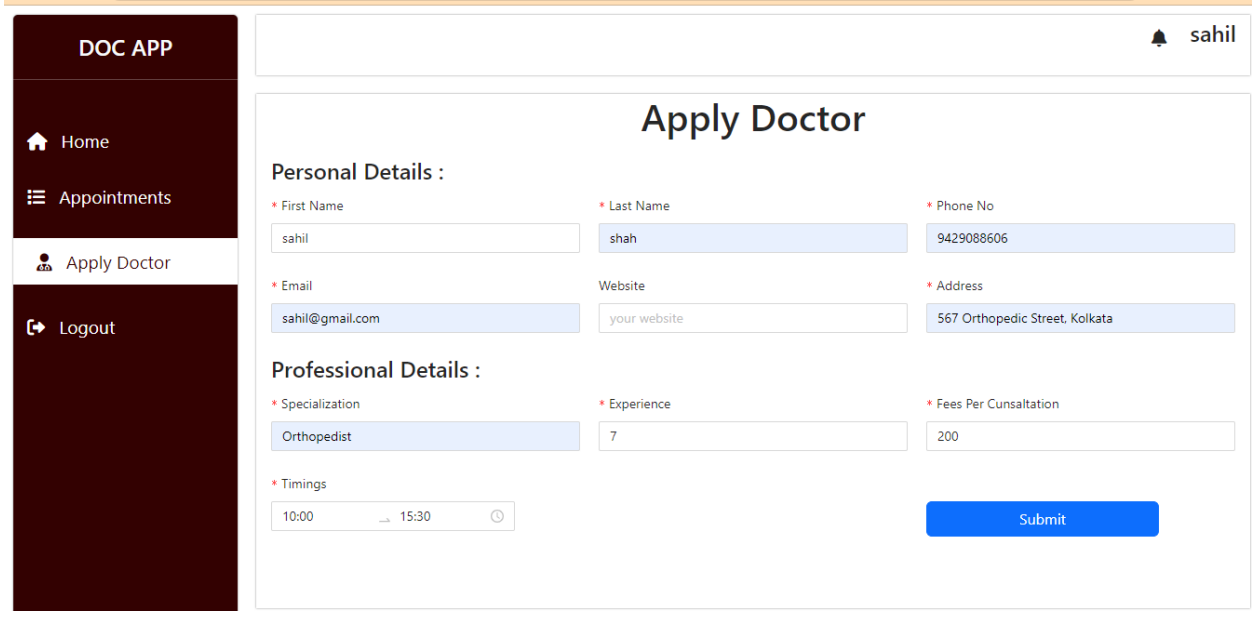
Test Case 10: Secure Authentication
Input: User attempts unauthorized access.
Expected Output: Secure authentication mechanisms prevent unauthorized access.

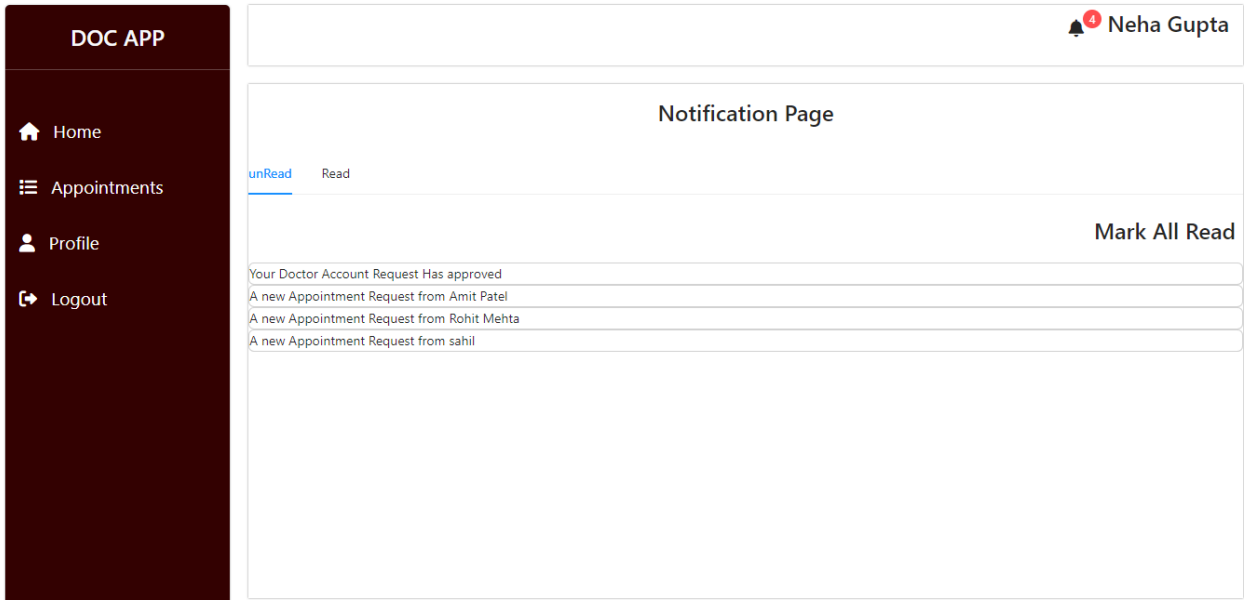
Screenshots

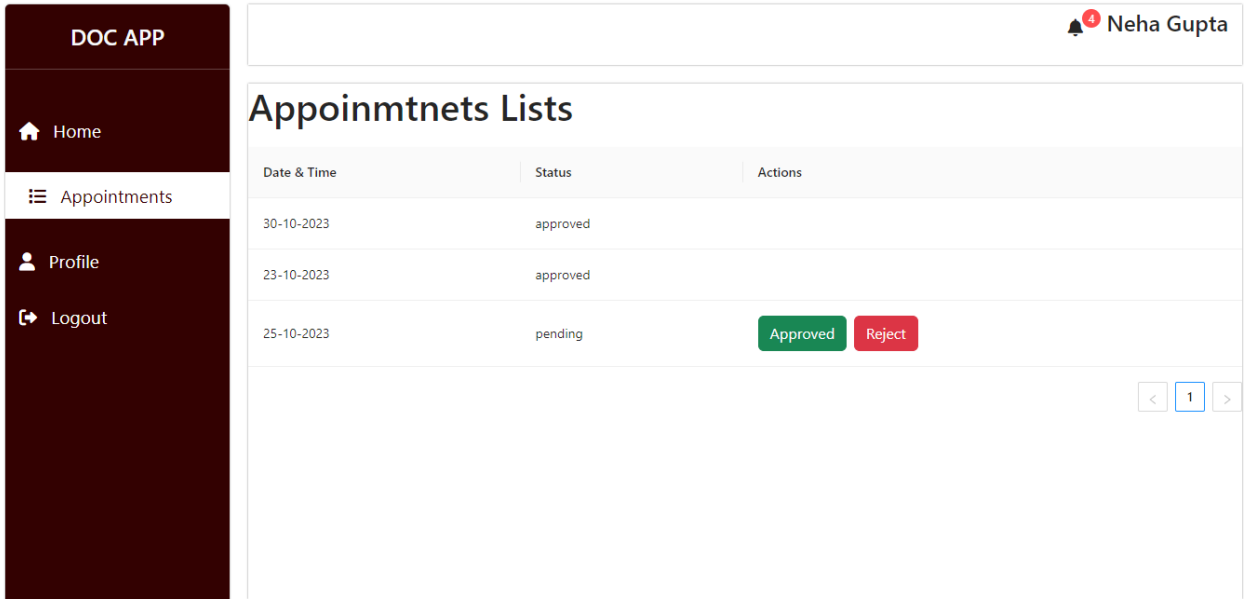

DOC APP
Home
Appointments
Apply Doctor
Logout
sahil
Home Page
Dr. rahul mehta
Specialization cardiologist
Experience 12
Fees Per Cunsaltation 500
Timings 06:00 - 08:00
Dr. Ajay Singh
Specialization Dentist
Experience 8
Fees Per Cunsaltation 120
Timings 08:30 - 17:30
Dr. Neha Gupta
Specialization Pediatrician
Experience 12
Fees Per Cunsaltation 150
Timings 10:00 - 16:00
Dr. Sonal Gupta
Specialization Orthopedist
Experience 12
Fees Per Cunsaltation 400
Timings 11:00 - 17:30
Dr. Priya Bansal
Specialization Dermatologist
Experience 10
Fees Per Cunsaltation 250
Timings 10:00 - 16:00
Dr. Rajesh Gupta
Specialization Cardiologist
Experience 15
Fees Per Cunsaltation 500
Timings 09:00 - 17:00

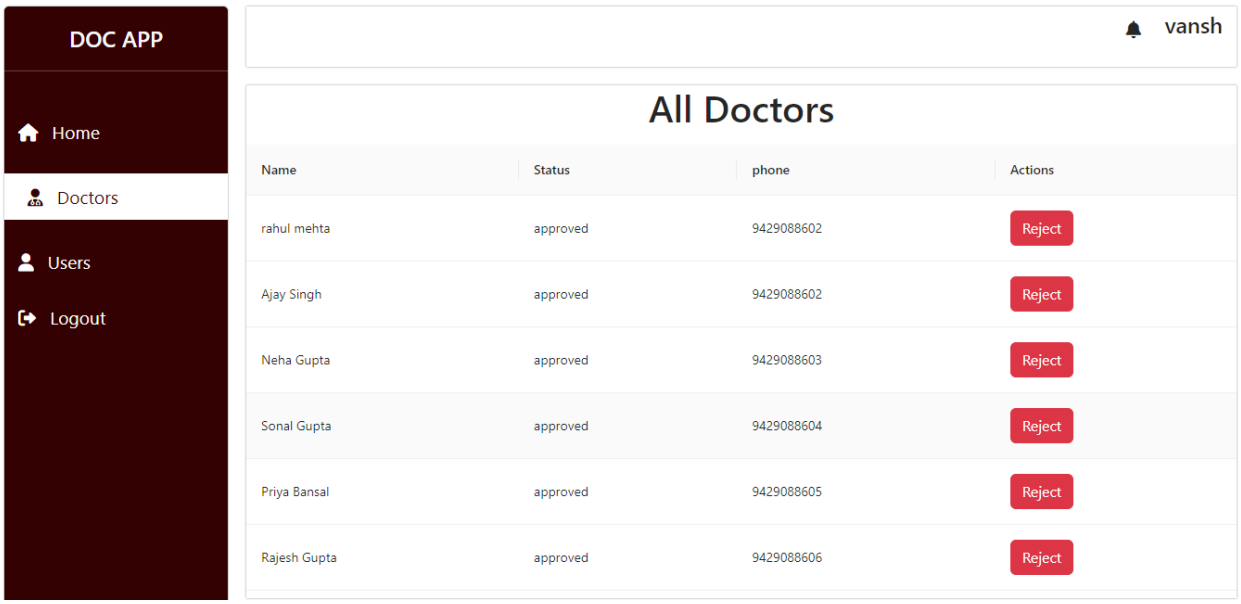

DOC APP
Home
Appointments
Apply Doctor
Logout
sahil
Appoinmtnets Lists
Name
Phone
Date & Time
Status
rahul mehta
9429088602
24-10-2023
approved
Ajay Singh
9429088602
22-10-2023
approved
Sonal Gupta
9429088604
23-10-2023
pending
Sonal Gupta
9429088604
25-10-2023
approved
1

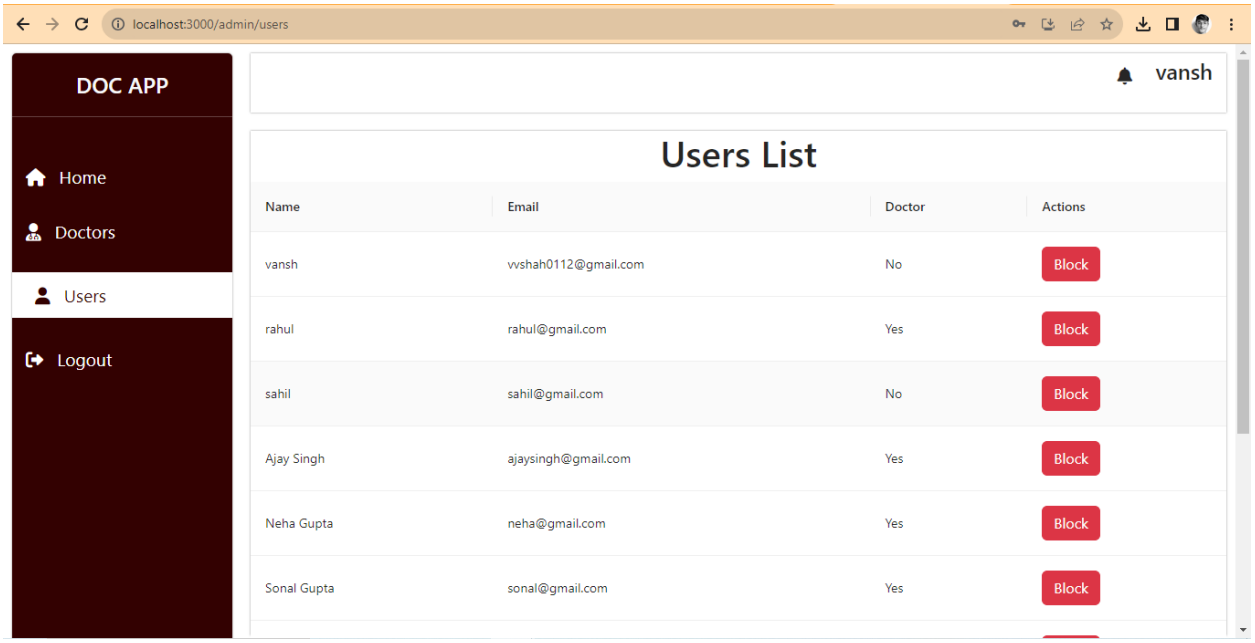

localhost:3000/doctor/book-appointment/652fbcec2c58a507a1d0c0dc
DOC APP
Home
Appointments
Apply Doctor
Logout
sahil
Booking Page
Dr.Ajay Singh
Fees : 120
Timings : 08:30 - 17:30
27-10-2023
14:30
Check Availability
Book Now


DOC APP
Home
Appointments
Apply Doctor
Logout
sahil
Apply Doctor
Personal Details :
* First Name
sahil
* Last Name
shah
* Phone No
9429088606
* Email
sahil@gmail.com
Website
your website
* Address
567 Orthopedic Street, Kolkata
Professional Details :
* Specialization
Orthopedist
* Experience
7
* Fees Per Cunsaltation
200
* Timings
10:00
15:30
Submit


DOC APP
Home
Appointments
Profile
Logout
4
Neha Gupta
Notification Page
unRead
Read
Mark All Read
Your Doctor Account Request Has approved
A new Appointment Request from Amit Patel
A new Appointment Request from Rohit Mehta
A new Appointment Request from sahil


DOC APP
Home
Appointments
Profile
Logout
4
Neha Gupta
Appoinmtnets Lists
Date & Time
Status
Actions
30-10-2023
approved
23-10-2023
approved
25-10-2023
pending
Approved
Reject
1


DOC APP
Home
Doctors
Users
Logout
vansh
All Doctors
Name
Status
phone
Actions
rahul mehta
approved
9429088602
Reject
Ajay Singh
approved
9429088602
Reject
Neha Gupta
approved
9429088603
Reject
Sonal Gupta
approved
9429088604
Reject
Priya Bansal
approved
9429088605
Reject
Rajesh Gupta
approved
9429088606
Reject


localhost:3000/admin/users
DOC APP
Home
Doctors
Users
Logout
vansh
Users List
Name
Email
Doctor
Actions
vansh
vvshah0112@gmail.com
No
Block
rahul
rahul@gmail.com
Yes
Block
sahil
sahil@gmail.com
No
Block
Ajay Singh
ajaysingh@gmail.com
Yes
Block
Neha Gupta
neha@gmail.com
Yes
Block
Sonal Gupta
sonal@gmail.com
Yes
Block

Conclusion

- User & Doctor Profile:
  Users can create and manage their profiles.
  Profiles include basic information.
- Search :
  Users can search for a doctor. Also the doctor will be notified regarding the patient's appointment.
- Efficient Appointment Management: With clear relationships between users and appointments, the system streamlines the process of scheduling, tracking, and managing appointments, benefiting both patients and healthcare providers.
- Enhanced User Experience: Ultimately, the structured database schema contributes to an improved user experience by providing a foundation for efficient and secure healthcare appointment management, which is crucial in the digital era.

In the ever-evolving healthcare landscape, a well-structured database system like this is a fundamental component of the MERN-based Doctor Appointment System, contributing to the optimization of healthcare processes and the satisfaction of both patients and healthcare providers.

Limitation and Future Extension :

Limited Scope: The MERN-based Doctor Appointment System, as described, primarily focuses on appointment scheduling and management. It does not encompass advanced medical features like telemedicine, electronic health records (EHR), or billing.
User Adoption: Ensuring that patients and doctors use the system effectively might face resistance due to a reluctance to shift from traditional appointment booking methods.

Future Extension :

Telemedicine Integration: To meet the growing demand for remote healthcare services, integrating telemedicine features, like video consultations and secure messaging, would be a valuable addition

Electronic Health Records (EHR) Integration: Incorporating EHR capabilities to securely store and manage patients' medical records, histories, and test results can enhance the system's utility and benefit healthcare providers.
Payment and Billing: Adding payment and billing features, integrating with insurance systems, and enabling online payment for medical services can streamline the entire patient journey.

Bibliography:

Websites:

- https://react.dev/
- https://nodejs.org/en
- https://www.mongodb.com/cloud/atlas/lp/try4?utm_ad_campaign_id=6501677905&adgroup=150907552714&cq_cmp=6501677905&gad=1
- https://mongoosejs.com/
- https://expressjs.com/