PROJECT REPORT

DocSpot: Seamless Appointment Booking for Health

Team ID : LTVIP2025TMID50339 **Team Size** : 5

Team Members:

- M Vineela – Team Leader
- Rohaz Shaik– Team Member
- Samudram Hemanth – Team Member
- Srilakshmi Avvaru – Team Member
- Vinnakota Chandini Sri – Team Member

Project Overview

Purpose:

Booking a doctor's appointment has never been easier. With our convenient online platform, you can quickly and effortlessly schedule your appointments from the comfort of your own home. No more waiting on hold or playing phone tag with busy receptionists. Our userfriendly interface allows you to browse through a wide range of doctors, making it simple to find the perfect match for your needs.

With our advanced booking system, you can say goodbye to the hassle of traditional appointment booking. Our platform offers real-time availability, allowing you to choose from a range of open slots that fit your schedule. Whether you prefer early morning, evening, or weekend appointments, we have options to accommodate your needs.

The main goal of this platform is to make an easy appointment booking for user without any delay, directly by choosing required doctor as per their need. This platform builds a simple connection with doctor you need, and there is no more waiting time or no middlemen is required for booking an appointment.

Features:

- **User Registration:** User who needs to see a doctor for a routine check-up, visits the Docspot app and signs up as a Patient. He provides his basic details like email and creates a password.
- **Browsing Doctors:** Upon logging in, User is presented with a dashboard displaying a list of available doctors. He filters the list based on his preferences, such as specialty or availability.
- **Booking an Appointment**: User finds a suitable doctor with appointment dates. On clicking the specific date, a form appears where he specifies the issue to visit and selects the desired available timeslots and uploads any necessary documents like previous medical records related to issue. After

submitting the form, User receives a confirmation message indicating that his appointment is conformed.

- **Appointment Management:** As the appointment approaches, User can view and manage his upcoming appointments in the appointment history section of his dashboard. He has the option to cancel or reschedule appointments if needed and can update the status accordingly.
- **Admin Approval (Background Process):** In the background, the admin reviews new doctor registrations and approves legitimate applicants. Approved doctors are then registered in the app and can start managing their appointments.
- **Platform Governance:** The admin oversees the overall operation of the appointment booking system and ensures compliance with platform policies, terms of service, and privacy regulations. The admin addresses any issues or disputes to maintain a smooth user experience.
- **Doctor's Appointment Management:** An approved doctor on the platform, logs into his account and manages his appointments.
- **Appointment Consultation:** On the day of the appointment, User visits the doctor's office for his check-up. Doctor provides medical care and advice during the consultation, fulfilling Patients(user) healthcare needs.
- **Post-Appointment Follow-up:** After the consultation, doctor updates patients medical records and may prescribe medication or recommend further treatment if necessary.

Architecture

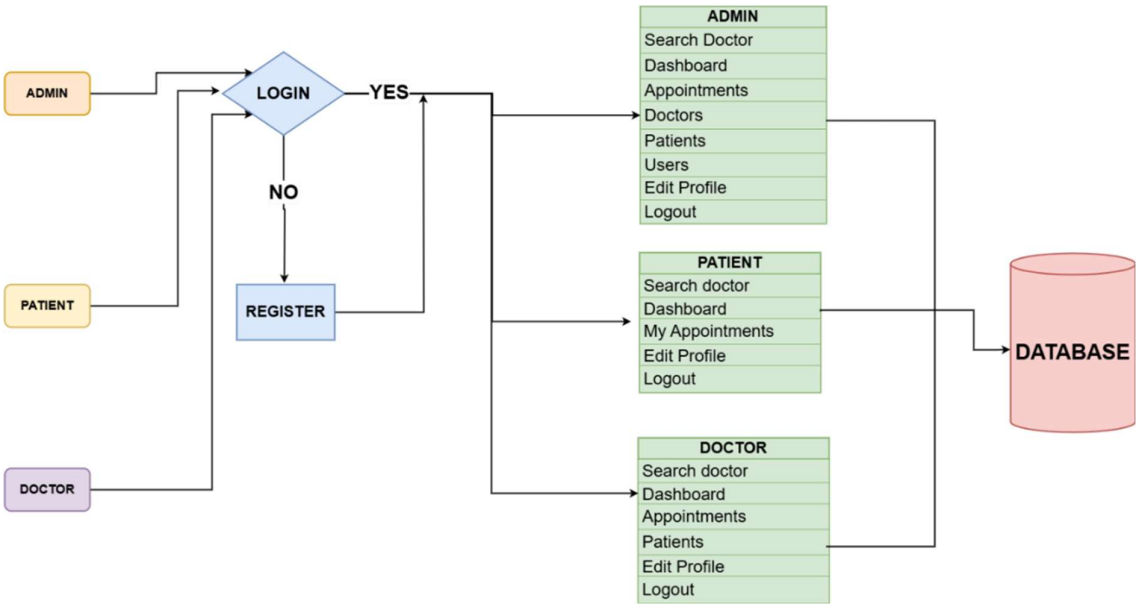


Fig: - Architecture

Frontend (React.js)

The frontend is responsible for the visual representation and interaction of the system. It would include:

- **Login/Registration:** A form for users to enter their credentials or create a new account.

- **Dashboard:** A personalized view for each user role, displaying relevant information and actions.
- **Search Doctor:** A feature to search for doctors based on criteria like specialization or location.
- **Appointments:** A section to view, book, or cancel appointments.
- **Profile Management:** A page to update user information.

Backend (Express.js & Node.js)

The backend handles the server-side operations, including:

- **Authentication:** Verifies user credentials and grants access.
- **Authorization:** Determines the permissions for each user role.
- **Data Storage and Retrieval:** Interacts with the database to store and retrieve user data, doctor information, and appointment details.
- **Business Logic:** Implements the rules and calculations for appointment scheduling, payment processing, and other functionalities.
- **API Endpoints:** Provides endpoints for the frontend to communicate with the backend and exchange data.
- **CRUD Operations**: Use Mongoose to interact with the database for creating, reading, updating, and deleting documents.

Database (MongoDB)

The database stores the persistent data of the system. It would typically include:

- **Users:** Table to store user information (name, email, password, role).
- **Doctors:** Table to store doctor information (name, specialization, availability).
- **Appointments:** Table to store appointment details (date, time, doctor, patient).
- **MongoDB Connection**: Use Mongoose to connect to the MongoDB database and define connection error handling.

Setup Instructions

Prerequisites:

- **Node.js and npm:** Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the server side. It provides a scalable and efficient platform for building network applications.
- **Express.js:** Express.js is a fast and minimalist web application framework for Node.js. It simplifies the process of creating robust APIs and web applications, offering features like routing, middleware support, and modular architecture.

- **MongoDB:** MongoDB is a flexible and scalable NoSQL database that stores data in a JSON-like format. It provides high performance, horizontal scalability, and seamless integration with Node.js, making it ideal for handling large amounts of structured and unstructured data.
- **Moment.js:** Momentjs is a JavaScript package that makes it simple to parse, validate, manipulate, and display date/time in JavaScript. Moment.js allows you to display dates in a human-readable format based on your location. Install React.js, a JavaScript library for building user interfaces
- **React.js:** React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.

Installation:

- React
- Bootstrap
- Material UI
- Axios
- react-bootstrap
- bcryptjs
- express
- dotenv
- mongoose
- Nodemon
- Jsonwebtoken

Folder Structure

•Client:

- **public**: Contains static files like index.html and favicon.ico.
- **src**: Main source folder.
- **assets**: Stores images, styles, and other static assets.
- **components**: Reusable UI components.
- **pages**: Different pages of the application.
- **services**: API calls and other services.
- **App.js**: Main application component.
- **index.js**: Entry point of the React application.
- **package.json**: Lists dependencies and scripts.
- **.env**: Environment variables.

- **README.md**: Project documentation.

Server:

- **config**: Configuration files, such as database and application settings.
- **controllers**: Logic for handling requests and responses.
- **models**: Database schemas and models.
- **routes**: Route definitions for the API endpoints.
- **middlewares**: Custom middleware functions.
- **utils**: Utility functions and helpers.
- **server.js**: Entry point of the Node.js application.
- **package.json**: Lists dependencies and scripts.
- **.env**: Environment variables.
- **README.md**: Project Documentation.

Fig: - Frontend **Fig: - Backend**

Running the Application

- **Frontend:** npm start -in the client directory
- **Backend:** npm run debug- in the server directory.

API Documentation Authentication

```
  POST(“/api/auth/login”) Authenticate User

- Request Body: {
                      "email": "shannu@gmail.com",
                     "password": "test"
```

}

- Response: {

"success": true,

"data": {

"token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiI2NzE3MDI4Yzk1
ZmU5MmE5NmQ5ZTllNzMiLCJyb2xlIjoiZG9jdG9yIiwiaWF0IjoxNzI5NjcxMz
MwLCJleHAiOjE3Mjk3NTc3MzB9.9lc2PzJqlEFh_RTOyqF4JJd_r4VTn3wRP0A 5vzrn-oc"

},

"message": "Successfully Logged In"

}

POST(“/api/auth/register”) Register New User

- Request Body{

"firstname": "Shanmukha Reddy",

"lastname": "Mettu",

"email": "shannutest@gmail.com",

"password": "test",

"phonenumber": 1230456789,

"gender" : "male",

"dob": "2003-05-10",

"role": "doctor",

"specialization": "general medicine",

"experience": 5,

"fee": 400

}

- Response: {

"success": true,

"data": {

"token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiI2NzE4YjFkM2F
mMDk0MTM5NzI2MjcyYzYiLCJyb2xlIjoiZG9jdG9yIiwiaWF0IjoxNzI5NjcxNjM
1LCJleHAiOjE3Mjk3NTgwMzV9.WLT_tHYZRkWX0fYk69mfy_TTpUuCzP3a94
DVJ4cjnk"

},

"message": "User Succesfully Registered” }

Appointment Management

GET(“/api/appointments/:id”) Getting Patient Appointments

-Request: {patient_id}

-Response: {


```
    "success": true,
    "data": {
        "doctorId": "671702386d319b99176b5a88",
        "patientId": "671702386d319b99176b5a75",
        "appointmentDate": "24/10/2024",
        "timeSlot": "10:00AM",
        "status": "confirmed",
        "period": "Morning",
        "paymentStatus": "unpaid",
        "reasonForVisit": "Fever",
        "files": [],
        "prescriptionFiles": [],
      },
    "message": "Appointment Fetched Successfully!”
}
```


POST(“/api/appointments/book”) Book Appointments

-Request: {


```
    "doctorId": "671702386d319b99176b5a88",
    "patientId": "671702386d319b99176b5a75",
  "appointmentDate": "24/10/2024",
  "timeSlot": "10:00AM",
  "reasonForVisit": "fever"
}
```


-Response: {


```
    "success": true,
    "data": {
```

```
        "doctorId": "671702386d319b99176b5a88",
       "patientId": "671702386d319b99176b5a75",
       "appointmentDate": "24/10/2024",
       "timeSlot": "10:00AM",
      "reasonForVisit": "fever"
    },
    "message": "Appointment booked successfully!"
}

PUT(“/api/appointments/:id”)  Update Appointment
 -Request: {
    "doctorId": "671702386d319b99176b5a88",
    "patientId": "671702386d319b99176b5a75",
    "appointmentDate": "24/10/2024",
    "timeSlot": "10:15AM",
    "status": "confirmed",
    "period": "Morning",
    "paymentStatus": "unpaid",
    "reasonForVisit": "Fever"
}
-Response: {
   "success": true,
    "data": {
        "doctorId": "671702386d319b99176b5a88",
        "patientId": "671702386d319b99176b5a75",
        "appointmentDate": "24/10/2024T00:00:00.000Z",
        "timeSlot": "10:15AM", //updated date
        "status": "confirmed",
        "period": "Morning",
        "paymentStatus": "unpaid",
        "reasonForVisit": "Fever",
        "files": [],
        "prescriptionFiles": [],
```

```
        "createdAt": "2024-10-22T01:39:04.161Z",
        "updatedAt": "2024-10-23T09:06:05.734Z",
        "id": "671702386d319b99176b5c09"
    },
    "message": "Appointment Updated Successfully!"
}
GET(“/api/appointments/” )  All Appointments
-Request: {}
-Response: {
  "success": true,
  "data": [
    {
"doctorId":            "6717028c95fe92a96d9e9e73",
"patientId": "671702386d319b99176b5a8f",
"appointmentDate": "2024-10-23T00:00:00.000Z",
     "timeSlot": "08:00AM",
     "status": "cancelled",
           "period": "Morning",
           "paymentStatus": "unpaid",
           "reasonForVisit": "fever",
           "files": [],
           "prescriptionFiles": [],
           "id": "67176f96a54efd5e1c77a8fb"
       },
      {……}
     ]
 "message": "Appointments Fetched Successfully!"
}

DELETE(“/api/appointments/:id” )  Delete Appointment
-Request:{appointment_id}
-Response: {
    "success": true,
```

```
 "data": {},
 "message": "Appointment Deleted Successfully!"
}
```


Doctor Management:


```
 GET(“/api/doctors/”) All Doctors
–Response: {
    "success": true,
    "data": [
        {
            "doctorId": "671702386d319b99176b5a5c",
            "name": "Theodore Schamberger",
            "specialization": "Cardiology",
            "experience": 12,
            "fee": 800,
            "status": "pending",
            "active": true,
            "availability": [
                {
                    "day": "Monday",
                    "timeSlots": [],
                    "dayOff": true,
                    "_id": "671702386d319b99176b5a93"

},              {
           "day": "Tuesday",
          "timeSlots": [],
         "dayOff": true,
         "_id": "671702386d319b99176b5a94"
       },
       {
          "day": "Wednesday",
           "timeSlots": [],
```

```
    "dayOff": true,
    "_id": "671702386d319b99176b5a95"
        },
        {
            "day": "Thursday",
            "timeSlots": [
                {
                    "startTime": "8:00AM",
                    "endTime": "12:00PM",
                    "period": "Morning",
                    "maxSlots": 20,
                    "interval": 20,
                    "_id": "671702386d319b99176b5a97",
                    "id": "671702386d319b99176b5a97"
                },
                {
                    "startTime": "1:00PM",
          "endTime": "5:00PM",
         "period": "Afternoon",
        "maxSlots": 20,
        "interval": 60,
"_id": "671702386d319b99176b5a98",
"id": "671702386d319b99176b5a98"
     },
       {
           "startTime": "6:00PM",
           "endTime": "9:00PM",
                   "period": "Evening",
                   "maxSlots": 12,
                   "interval": 15,
                   "_id": "671702386d319b99176b5a99",
                   "id": "671702386d319b99176b5a99"
                }
```

```
                ],
                "dayOff": false,
                "_id": "671702386d319b99176b5a96"
            },
            {
                "day": "Friday",
                "timeSlots": [],
                "dayOff": true,
                "_id": "671702386d319b99176b5a9a"
    },
    {
      "day": "Saturday",
      "timeSlots": [],
      "dayOff": true,
      "_id": "671702386d319b99176b5a9b"
    },
    {
        "day": "Sunday",
        "timeSlots": [
                    {
                        "startTime": "7:00AM",
                        "endTime": "12:00PM",
                        "period": "Morning",
                        "maxSlots": 16,
                        "interval": 60,
                        "_id": "671702386d319b99176b5a9d",
                        "id": "671702386d319b99176b5a9d"
                    },
                    {
                        "startTime": "3:00PM",
                        "endTime": "5:00PM",
                        "period": "Afternoon",
                        "maxSlots": 12,"interval": 15,
```

```
                    "_id": "671702386d319b99176b5a9e",
                   "id": "671702386d319b99176b5a9e"
               },
               {
                   "startTime": "6:00PM",
                   "endTime": "9:00PM",
                  "period": "Evening",
                     "maxSlots": 16,
                     "interval": 30,
              "_id": "671702386d319b99176b5a9f",
                              "id": "671702386d319b99176b5a9f"
                          }
                     ],
                     "dayOff": false,
                     "_id": "671702386d319b99176b5a9c"
                  }
              ],
         "message": "Success"
 },
GET(“/api/doctors/:id”) Get Doctor
-Request:{doctor_id}
-Response: {as given in get all doctors
“message”: "Success"
}
```


Patient Management


```
GET(“/api/users/”) Get all users
-Response: {
  "success": true,
  "data": [
      {
```

```
          "firstname": "Theodore",
          "lastname": "Schamberger",
          "email": "Haylee40@yahoo.com",
          "phonenumber": 16192408987,
            "gender": "female",
            "dob": "1997-12-01T06:40:57.895Z",
            "role": "patient",
            "active": true,
            "createdAt": "2024-10-22T01:39:04.023Z",
            "updatedAt": "2024-10-22T01:39:04.023Z",
            "doctorInfo": null,
            "id": "671702386d319b99176b5a5c"
        },
       {…more users…}
 ]
"message": "Fetched All Users"
}
GET(“/api/users/:id”) get patient
-Request: {patient_id}
-Response {
    "success": true,
    "data": {
      "firstname": "patient",
      "lastname": "last",
      "email": "patient@gmail.com",      "phonenumber": 19949405238,
      "gender": "female",
      "dob": "1992-02-19T20:11:29.659Z",
      "role": "patient",
        "active": true,
        "createdAt": "2024-10-22T01:39:04.027Z",
        "updatedAt": "2024-10-22T01:39:04.027Z",
        "doctorInfo": null,
        "id": "671702386d319b99176b5a8f"
```

```
    },
    "message": "Fetched User"
}
```

API Request/Response Formats

- Request: JSON
- Response: JSON

API Error Handling

- Error Codes: 400, 404, 500
- Error Messages: Detailed error messages

Authentication

Authentication:

- User Registration: Users register with email and password.
- Password Hashing: Passwords are hashed using BCrypt.
- Login: Users login with email and password.
- JSON Web Tokens (JWT): Upon successful login, a JWT is generated and sent to the client.

Authorization:

- Role-Based Access Control (RBAC): Users are assigned roles (admin, doctor, patient).
- Token Verification: JWT is verified on each request to ensure authenticity.
- Route Protection: Routes are protected based on user roles.

Token Management:

- Token Generation: JWT is generated using user ID, role, and expiration time.
- Token Storage: JWT is stored in local storage on the client-side.
- Token Validation: JWT is validated on each request using middle

User Interface

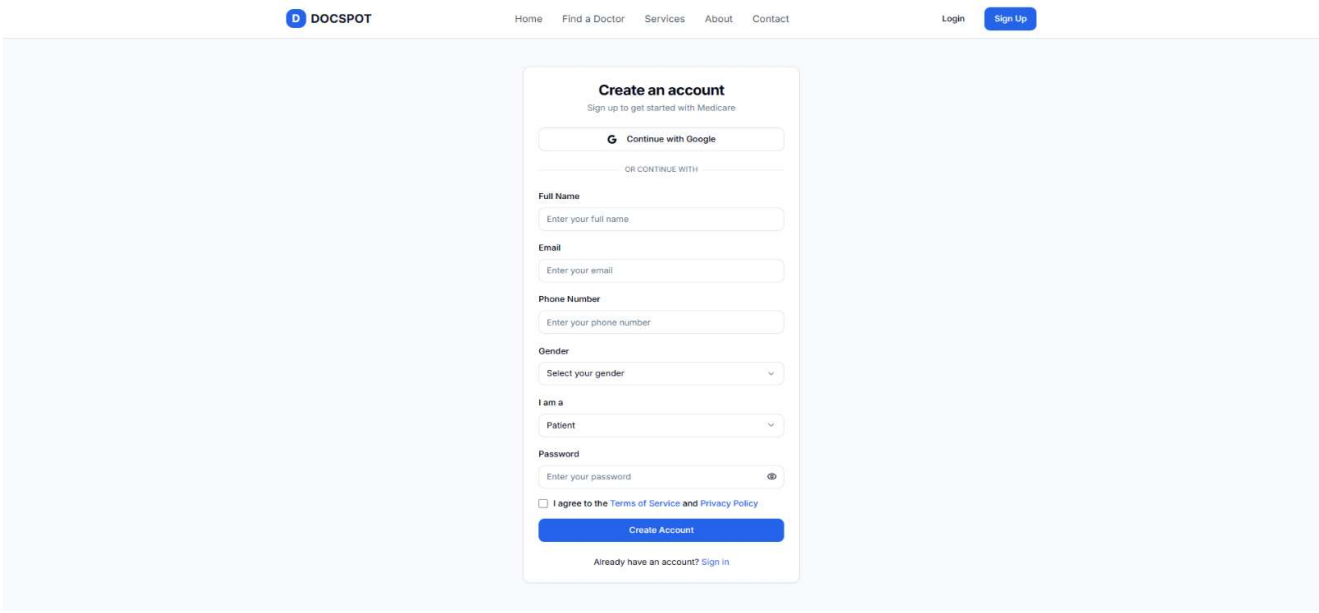


Fig: - Register Form

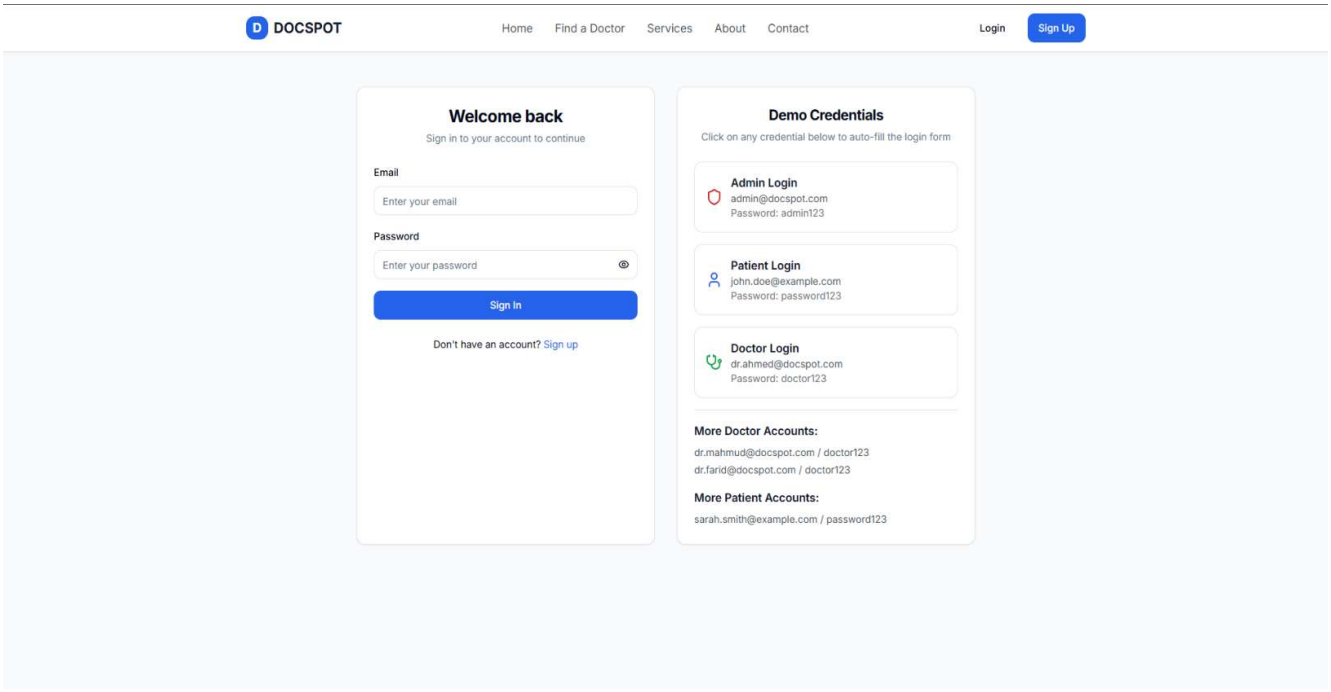


Fig: - Login Page

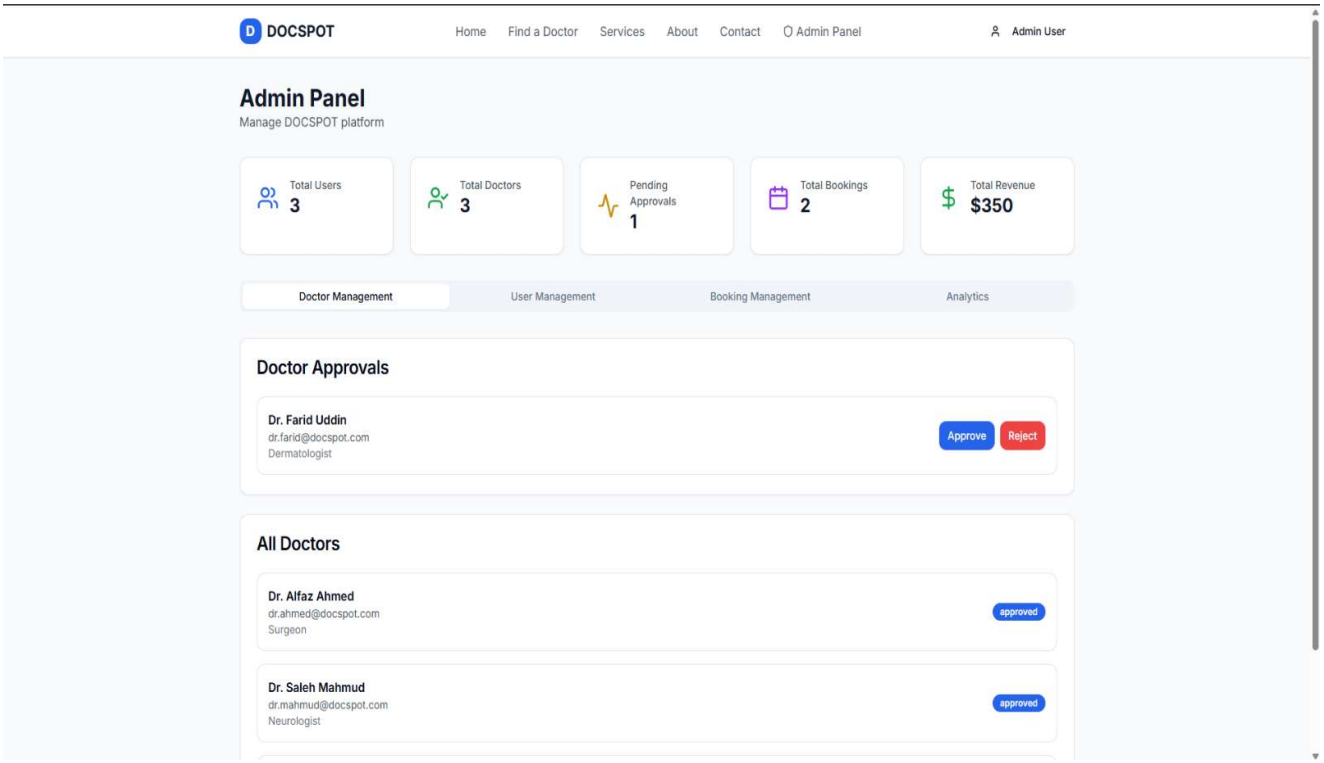


Fig: - Doctor Interface

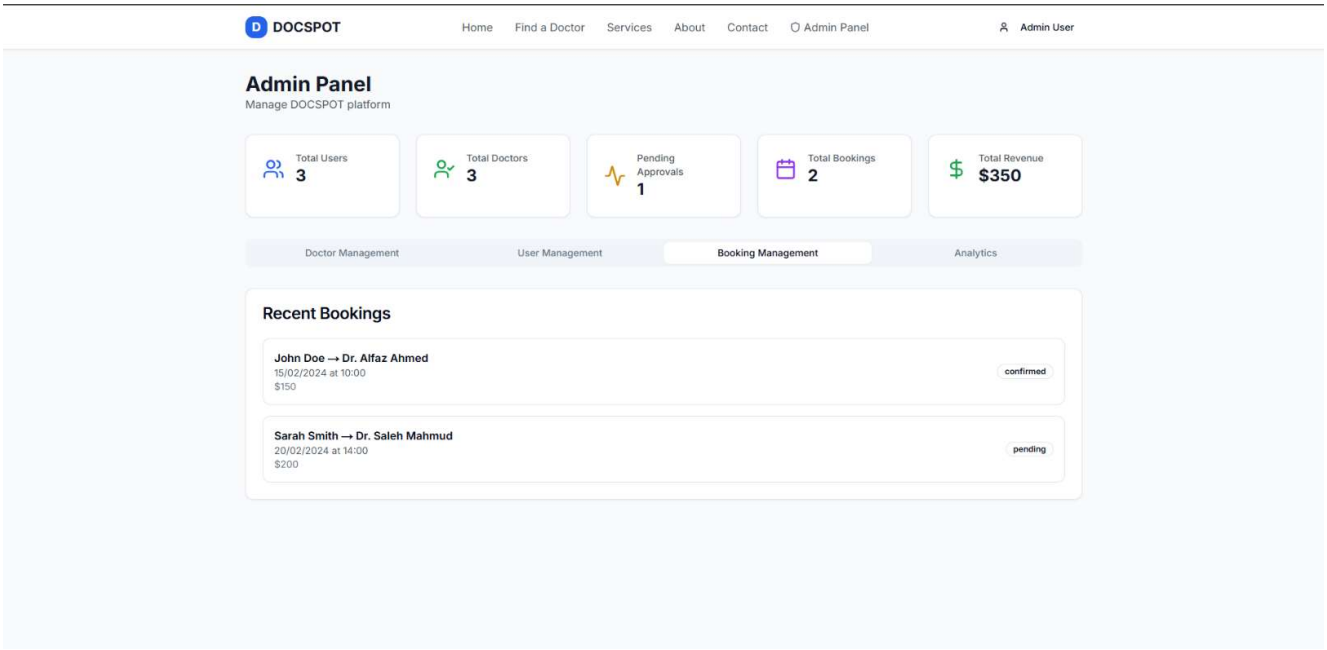


Fig: - Patient Interface

Testing

- **Unit Testing:** Test individual components (functions, classes) to ensure correct behaviour.
- **Integration Testing:** Verify interactions between components.
- **End-to-End (E2E) Testing:** Simulate user interactions to test entire workflows.
- **Acceptance Testing:** Validate app meets requirements and user expectations.

Screenshots or Demo

ADMIN

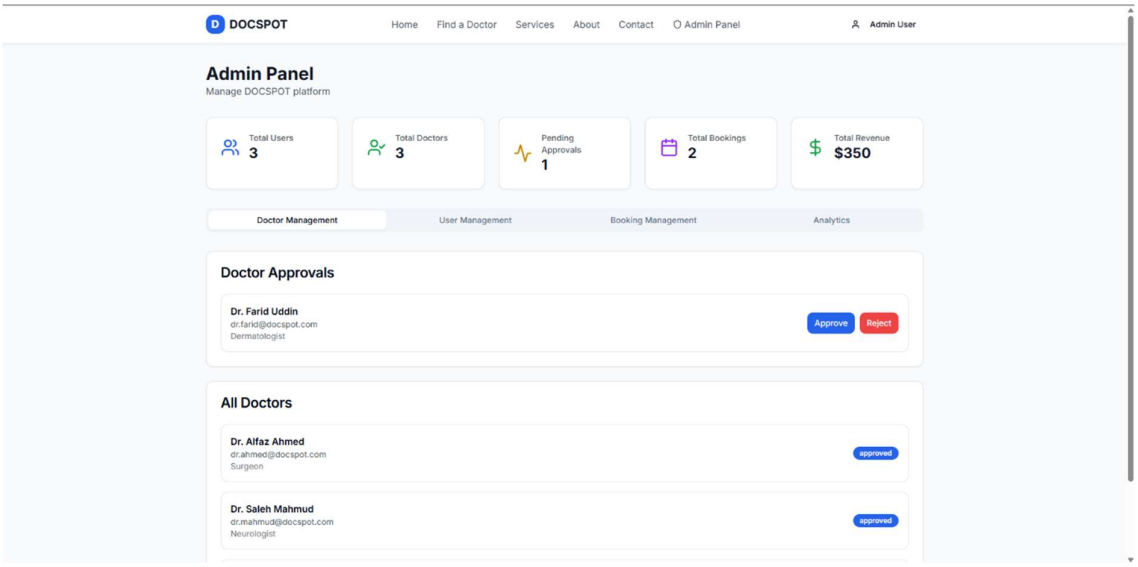


Fig: - Admin Dashboard

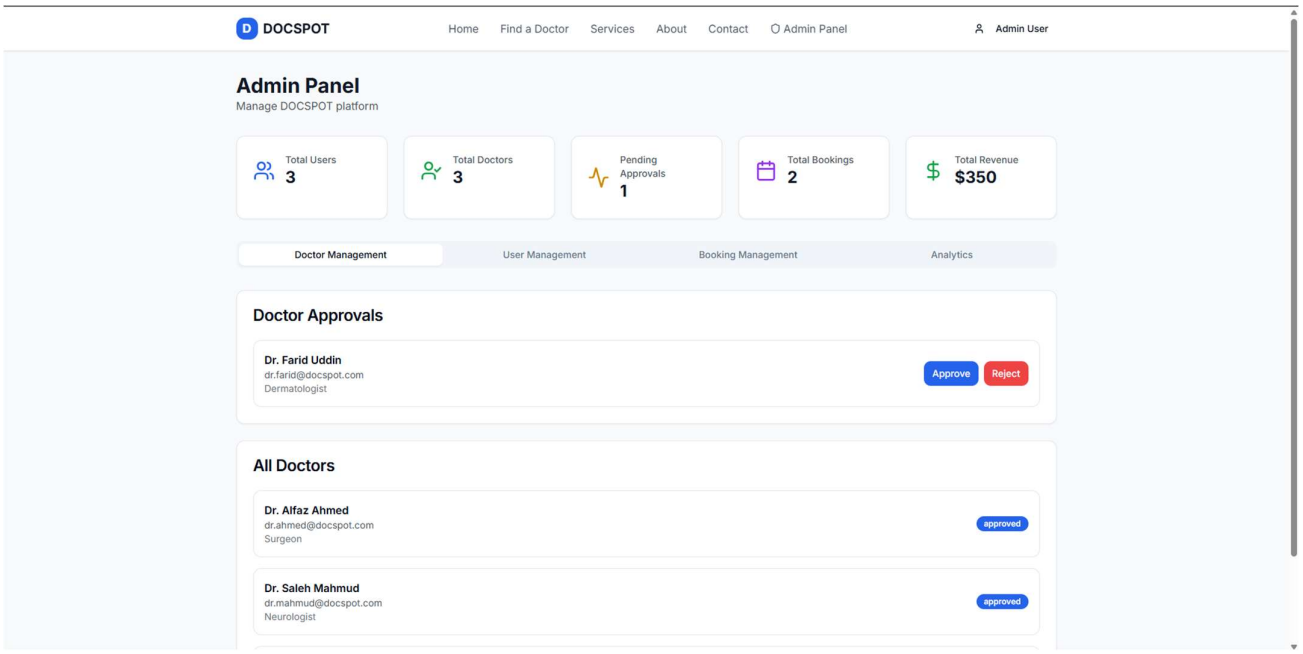


Fig: - Admin Search Doctors

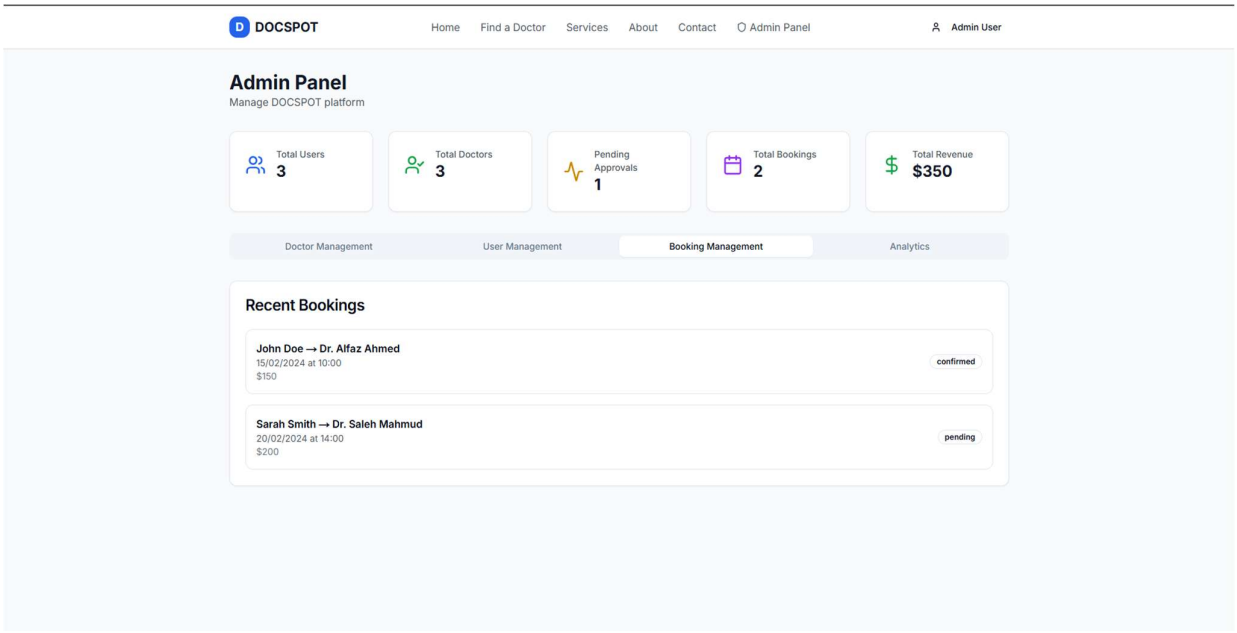


Fig: - Appointments

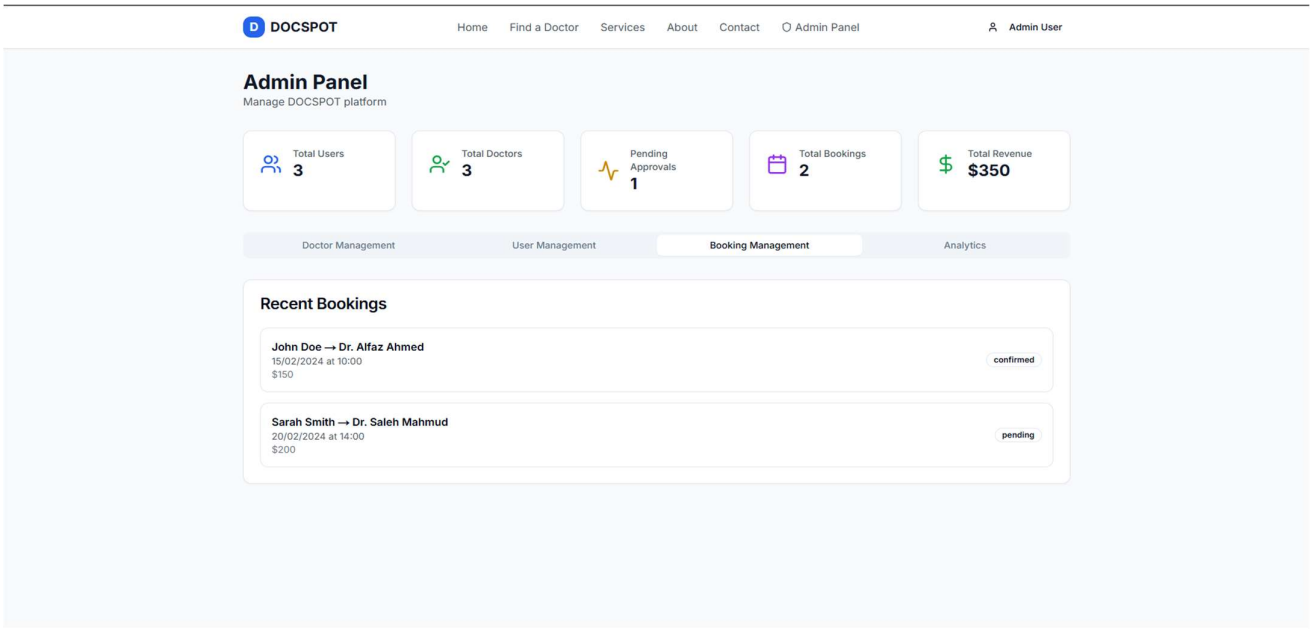


Fig: - Patients

DOCTOR

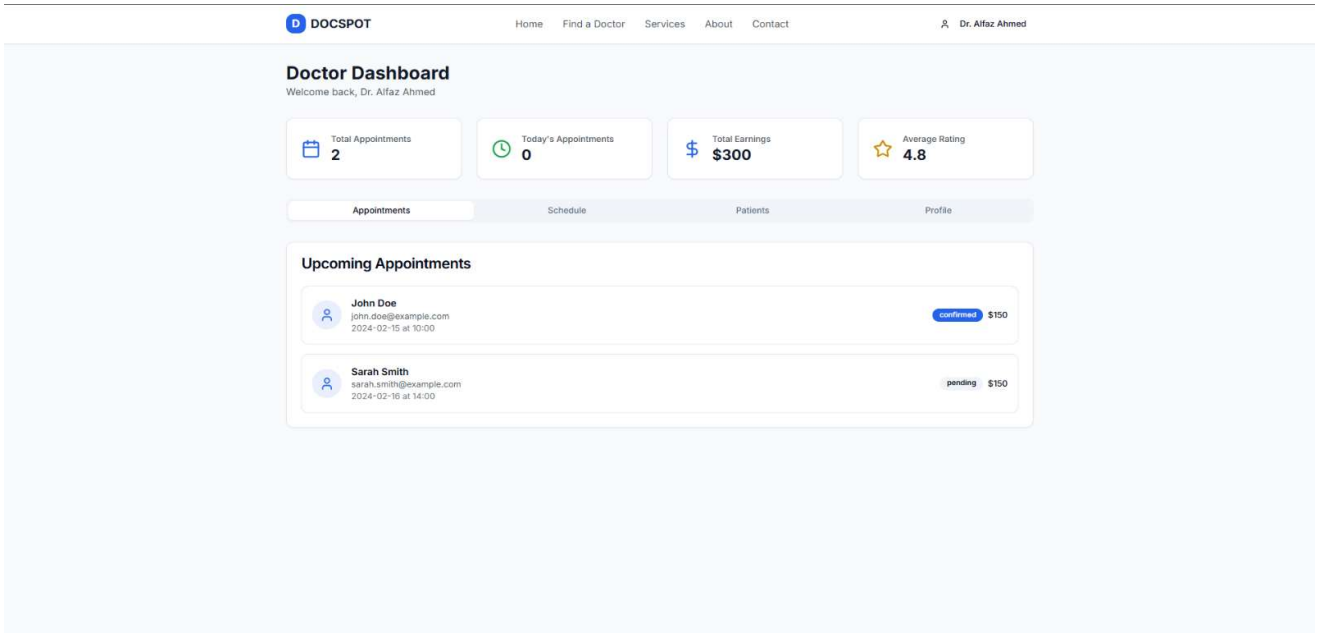


Fig: - Dashboard

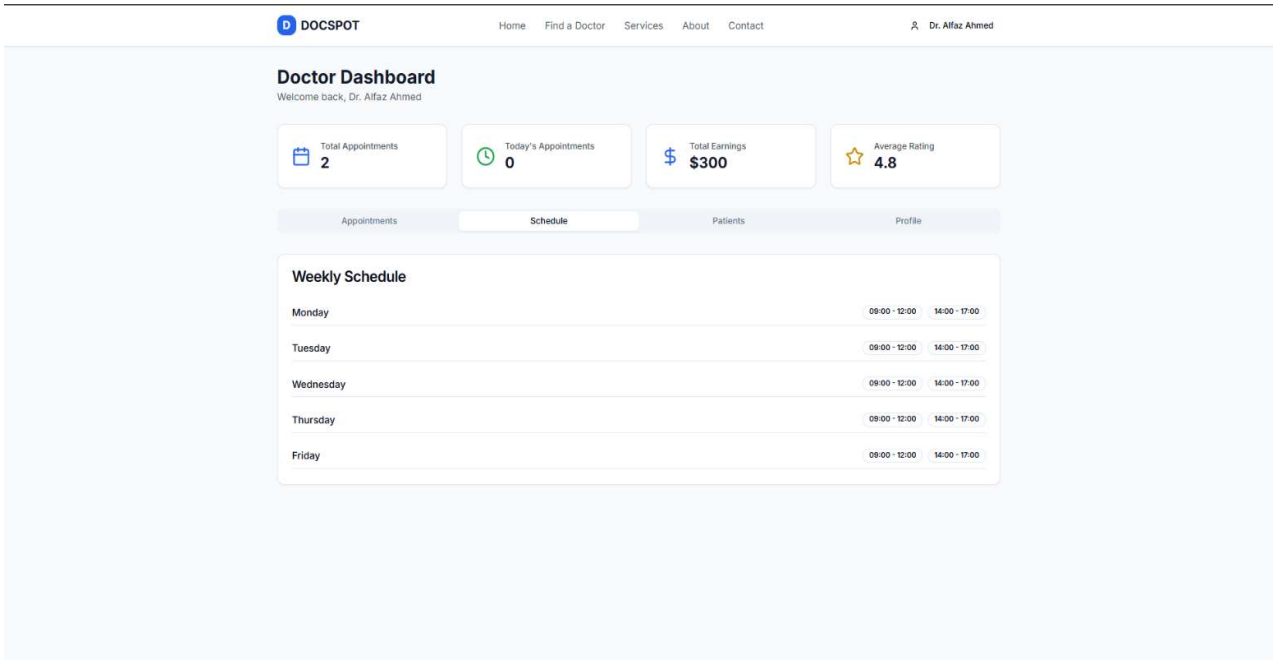


Fig: - Schedule Assignment

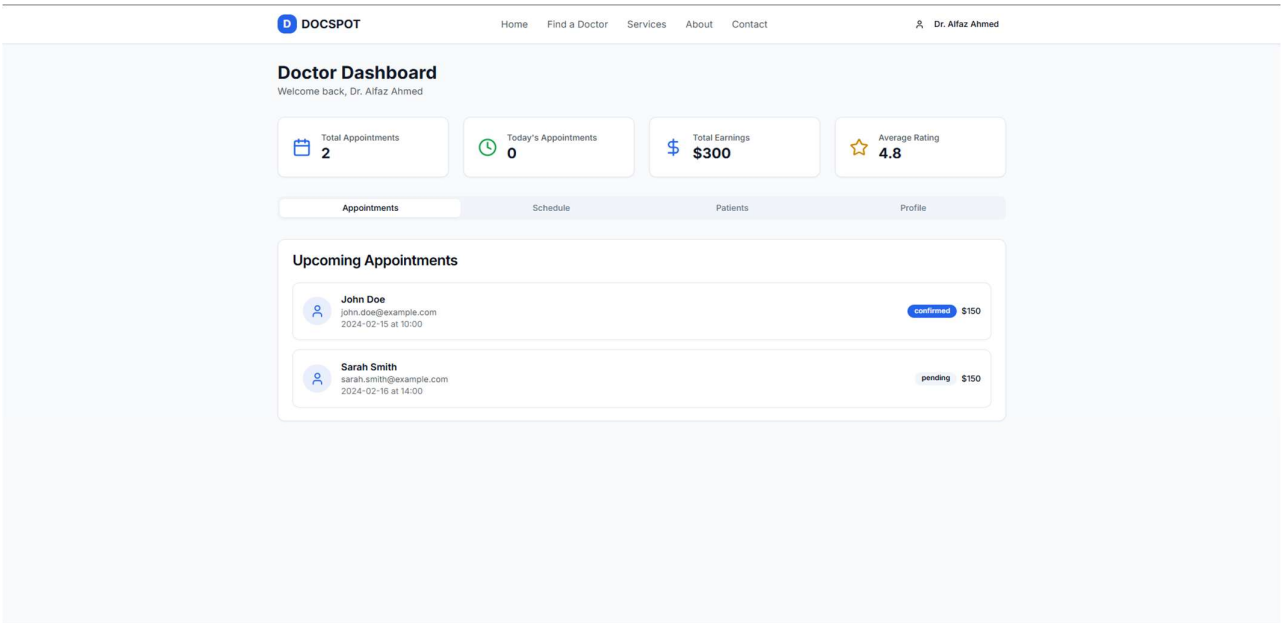


Fig: - Appointments

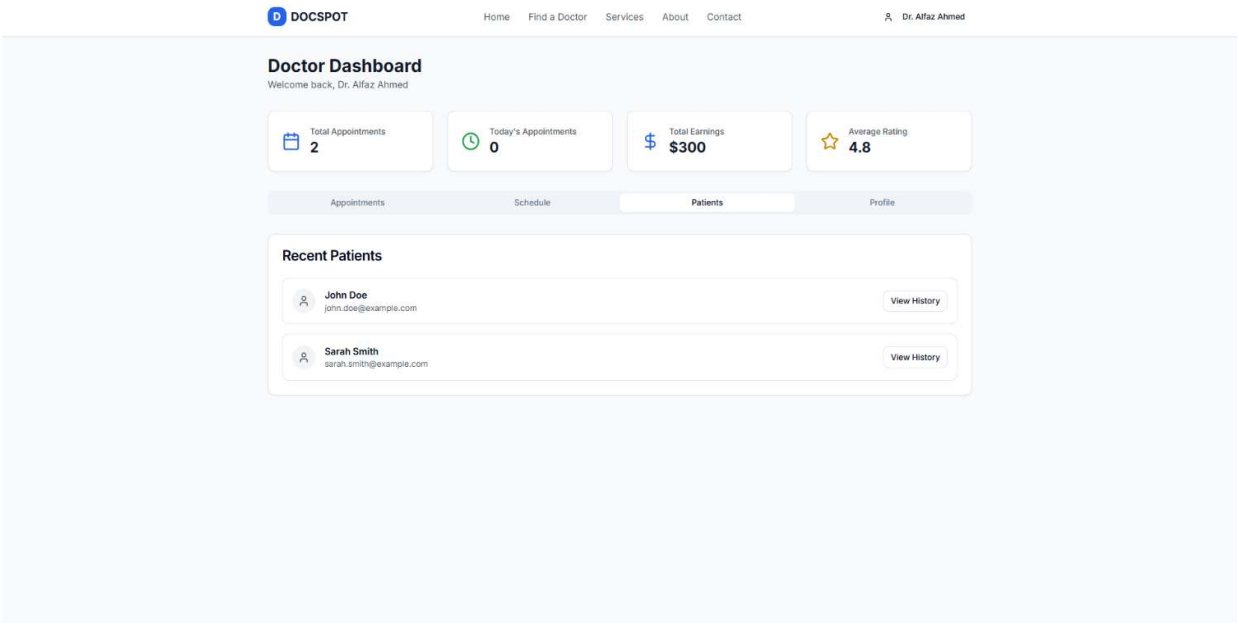


Fig: - Patients

PATIENT

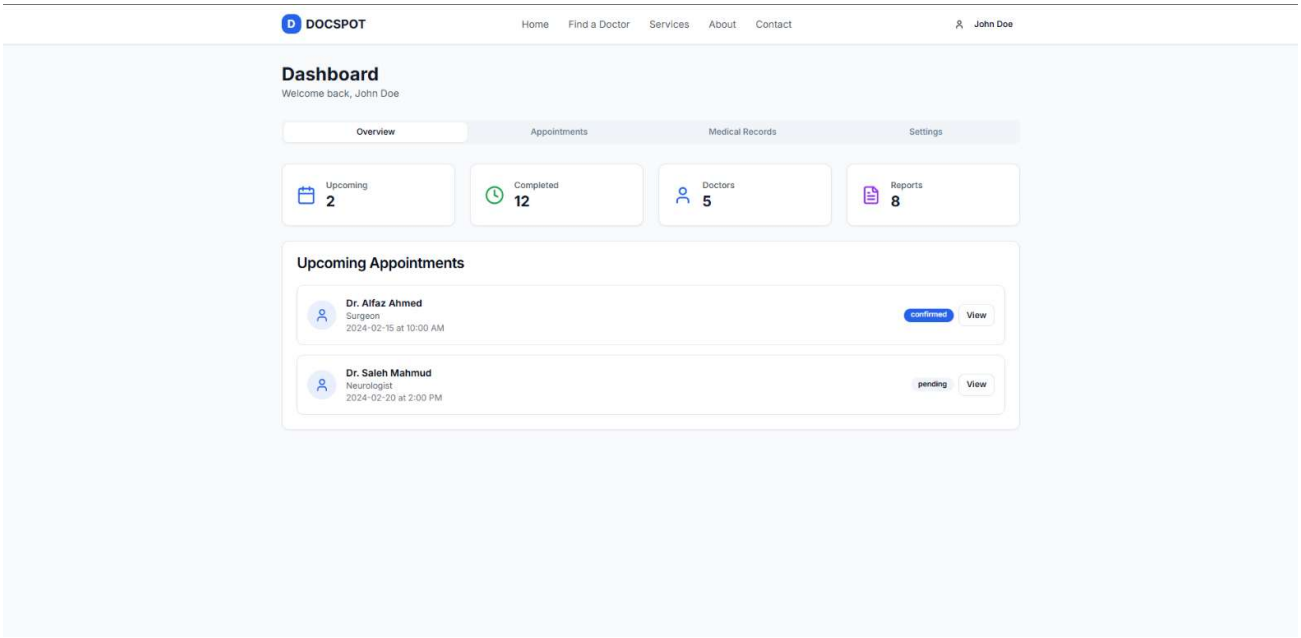


Fig: - Dashboard

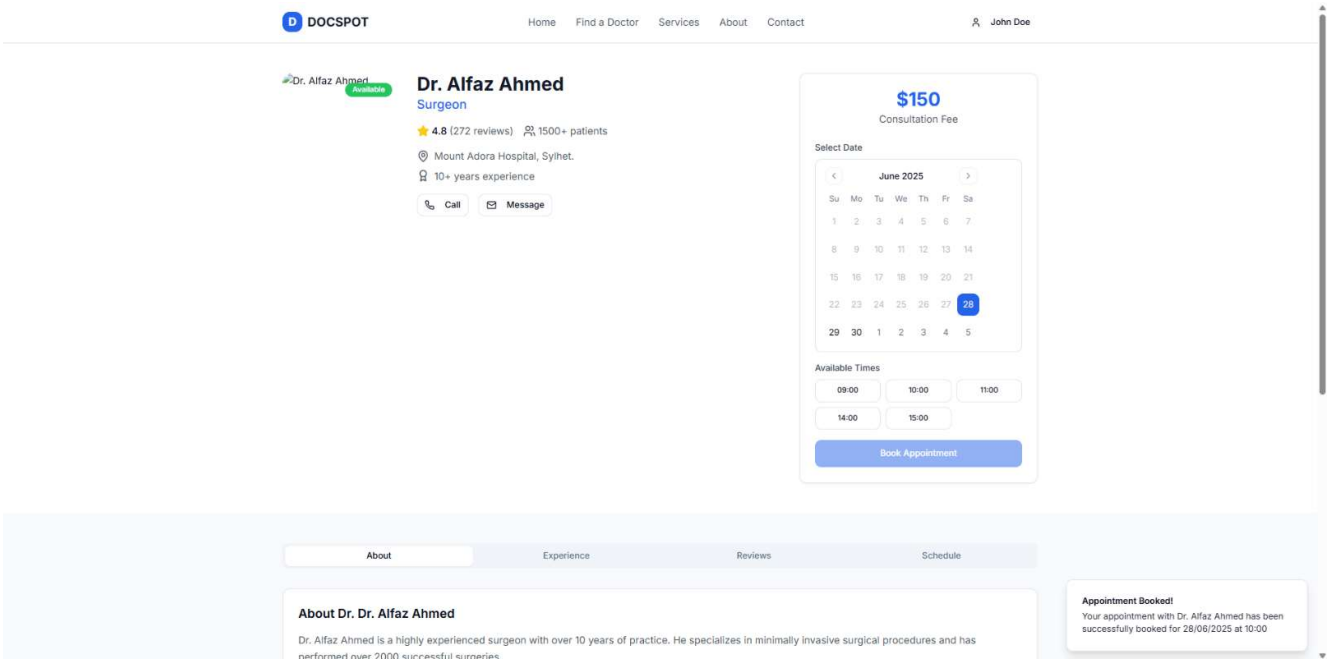


Fig: - Search Doctor & Booking Appointment

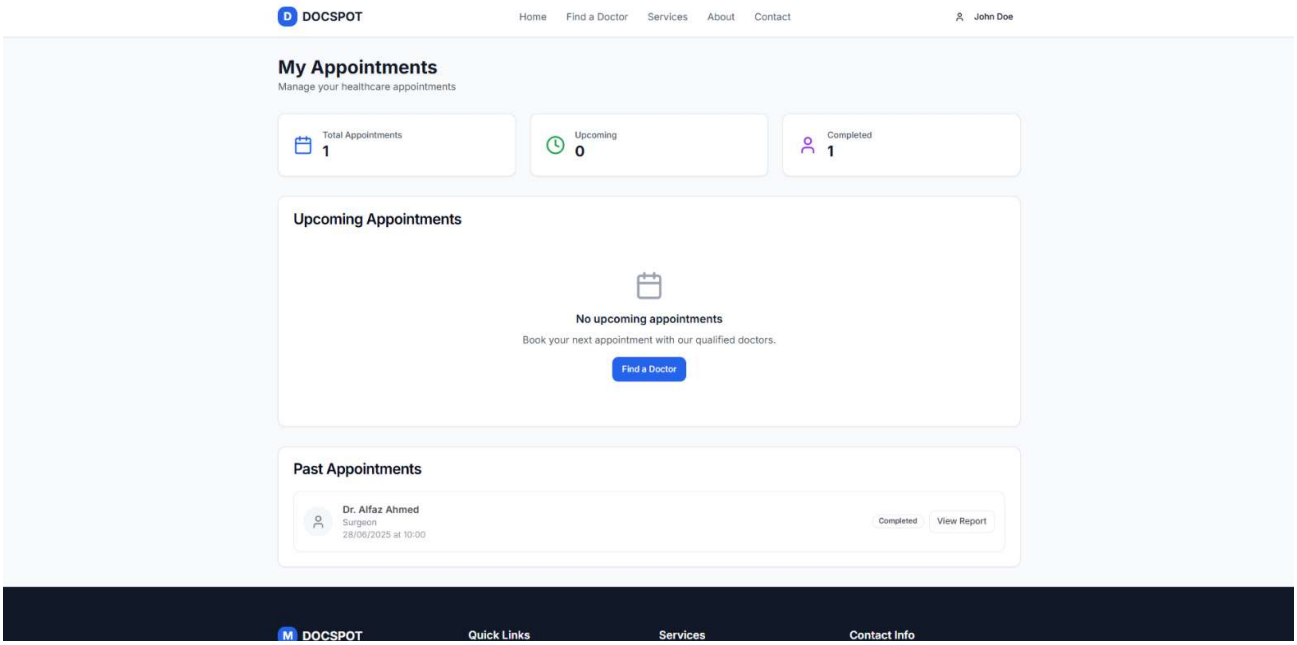
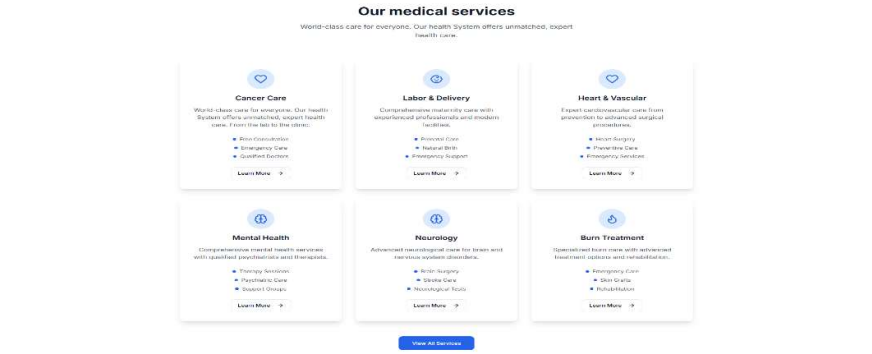
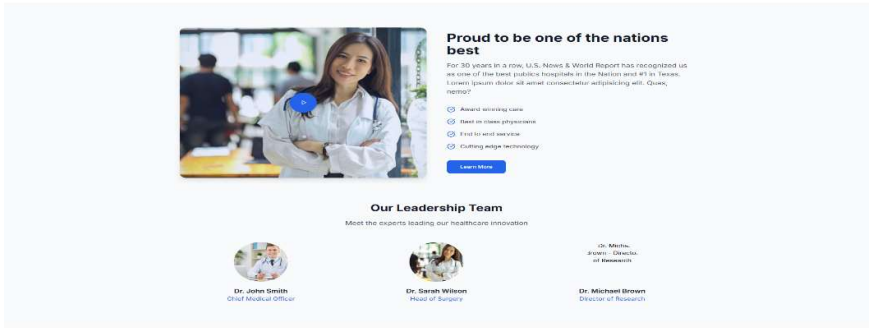
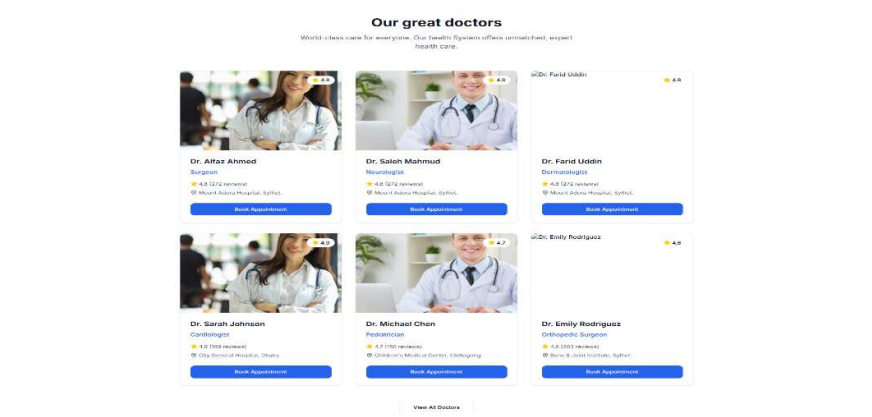


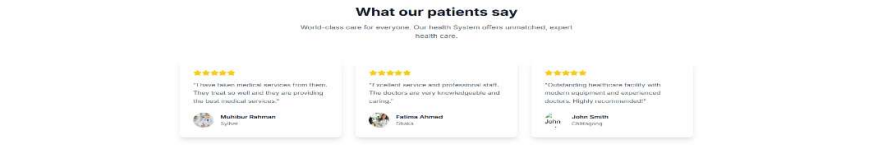
Fig: - Appointments

DOCSPOT
Home
Find a Doctor
Services
About
Contact
Login
Sign Up
We help patients live a healthy, longer life.
Our health system offers unmatched, expert health care. From the lab to the clinic, we provide world-class care for everyone.
Request an Appointment
Watch Video
30+
Years of Experience
15+
Clinic Locations
100%
Patient Satisfaction
Medical equipment


Our medical services
World-class care for everyone. Our health System offers unmatched, expert health care.
Cancer Care
Labor & Delivery
Heart & Vascular
Mental Health
Neurology
Burn Treatment
Learn More
View All Services


Proud to be one of the nations best
For 30 years in a row, U.S. News & World Report has recognized us as one of the best publics hospitals in the Nation and #1 in Texas. Lorem ipsum dolor sit amet consectetur adipisicing elit. Quae, nemo?
Award winning care
Best in class physicians
End to end service
Cutting edge technology
Learn More
Our Leadership Team
Meet the experts leading our healthcare innovation
Dr. John Smith
Chief Medical Officer
Dr. Sarah Wilson
Head of Surgery
Dr. Michael Brown
Director of Research


Our great doctors
World-class care for everyone. Our health System offers unmatched, expert health care.
Dr. Alfaz Ahmed
Surgeon
Dr. Saleh Mahmud
Neurologist
Dr. Farid Uddin
Dermatologist
Dr. Sarah Johnson
Cardiologist
Dr. Michael Chen
Pediatrician
Dr. Emily Rodriguez
Orthopedic Surgeon
Book Appointment
View All Doctors


What our patients say
World-class care for everyone. Our health System offers unmatched, expert health care.
Mutibur Rahman
Fatima Ahmed
John Smith

Most questions by our beloved patients
Find answers to commonly asked questions about our healthcare services and facilities.
What is your medical care?
What happens if I need to go to a hospital?
What happens if I need to go to a hospital?
Can I visit your medical office?
Does you provide urgent care?

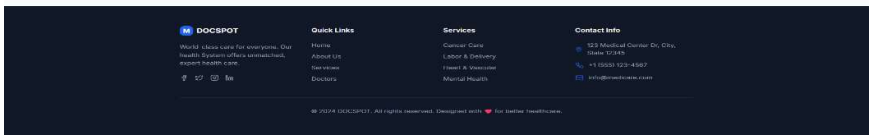

DOCSPOT
Quick Links
Home
About Us
Services
Doctors
Services
Cancer Care
Labor & Delivery
Heart & Vascular
Mental Health
Contact Info
123 Medical Center Dr, City, State 12345
+1 (555) 123-4567
info@medicare.com

Fig: - Interactive Web Design

Known Issues

Backend (Node.js):

- MongoDB connection issues: Occasional disconnections require manual restart.
- API rate limiting: No rate limiting implemented, vulnerable to abuse.

Frontend (React):

- Browser compatibility: Issues with older browser versions (IE, Safari). Mobile responsiveness: Layout issues on smaller screens.
- State management: Complex state updates may cause rendering errors.

Database (MongoDB):

- Data consistency: Lack of transactions may lead to inconsistent data.
- Query optimization: Slow queries impact performance.
- Indexing: Missing indexes cause slow query execution.

Security:

- Authentication: Weak password hashing (BCrypt).
- Authorization: Incomplete role-based access control.
- XSS protection: Insufficient input sanitization.

Future Enhancements

Outline potential future features or improvements that could be made to the project.

Mobile Optimization:

- Native mobile apps (iOS, Android)
- Mobile-friendly design and layout
- Integration with mobile payment methods

Telemedicine Integration:

- Video consultation capabilities
- Remote monitoring and health tracking

Insurance Integration

Insurance verification and eligibility checks

Conclusion

- DOCSPOT represents a comprehensive solution to modern healthcare accessibility challenges. By leveraging cutting-edge web technologies and user-centered design principles, the platform successfully bridges the gap between patients and healthcare providers.

- The project demonstrates proficiency in full-stack development, showcasing skills in React/Next.js frontend development, Node.js/Express backend architecture, MongoDB database design, and modern DevOps practices. The implementation of features like real-time booking, role-based authentication, and responsive design reflects industry best practices and scalable architecture patterns.

- Through this project, we have gained valuable experience in:

  Modern web development frameworks and libraries
  - Database design and optimization
  - API development and documentation
  - User experience design and accessibility
  - Security implementation and best practices
  - Testing strategies and quality assurance

- The platform is positioned for future growth with planned enhancements including mobile applications, telemedicine integration, and AI-powered features that will further revolutionize the healthcare booking experience.