## 1. Introduction

**Project Title:** DocIT - Doctor Appointment Booking System

**Team Members:**

Shaun Sylvain David
Leo William G
Anto Abisheik J
Prithiviraj A

## 2. Project Overview

**Purpose:**

DocIT is a web application that allows patients to browse through a list of available doctors, view their availability, and book appointments seamlessly. It aims to streamline the appointment scheduling process for both patients and doctors. The application provides three different views: User (Patient), Doctor, and Admin, each with specific features.

**Features:**

- User Profile: Browse doctors, book appointments, view appointment lists, and apply as a doctor.
- Doctor Profile: Manage availability, view appointments, and approve/cancel bookings.
- Admin Profile: Manage lists of doctors and users.

## 3. Architecture

**Frontend:**

The frontend is built using React, leveraging React Router DOM for routing, React Redux for state management, Axios for API requests, and Ant Design for UI components.

**Backend:**

The backend is developed using Node.js with Express.js to handle RESTful APIs. Middleware such as Morgan for logging and Joi for data validation are used. JSON Web Tokens (JWT) manage authentication.

**Database:**
MongoDB serves as the NoSQL database for storing user information, doctor details, appointment schedules, and notifications.

## 4. Setup Instructions

**Prerequisites:**

- Node.js
- MongoDB
- Git

**Installation:**

1. Clone the repository:
   ```
   git clone https://github.com/ShaunSDavid/DocIT_MERN_App.git
   ```
2. Install dependencies for the backend:
   ```
   npm install express joi jsonwebtoken moment mongoose morgan nodemon zxcvbn dotenv colors bcryptjs
   ```
3. Navigate to the client directory and install frontend dependencies:
   ```
   cd client npm i react-router-dom react-redux axios antd @reduxjs/toolkit react-bootstrap moment
   ```
4. Create a .env file in the root directory with the following environment variables:
   ```
   DB_URL=mongodb+srv://<user>:<userdb_password>@cluster0.mongodb.net
   JWT_SECRET=A_Secret_Value
   PORT=4000
   ```

## 5. Folder Structure

- **Client:** Contains the React frontend files, organized into components, pages, Redux store, and styles.
- **Server:** Houses the backend code, including models, controllers, routes, middleware, and configuration files.

## 6. Running the Application

**To start the backend server:**

```
npm start
```

**To start the frontend server:**

```
cd client npm start
```

Access the application at http://localhost:3000.

## 7. API Documentation

| Endpoint | Method | Description |
|---|---|---|
| /api/users/register | POST | Register a new user |
| /api/users/login | POST | Authenticate user and return JWT |
| /api/doctors/apply | POST | Apply as a doctor |
| /api/appointments | GET | Retrieve appointments list |
| /api/appointments/book | POST | Book an appointment |
| /api/admin/users | GET | Get a list of all users (Admin) |

## 8. Authentication

The application uses JWT for authentication. Upon login, users receive a JWT, which is stored in local storage. This token is then used to access protected routes throughout the application.

## 9. User Interface

The user interface is developed with Ant Design for a modern, responsive layout. Key screens include:

- Homepage for patients and doctors
- Appointment booking form
- Doctor's availability management
- Admin dashboard

## 10. Testing

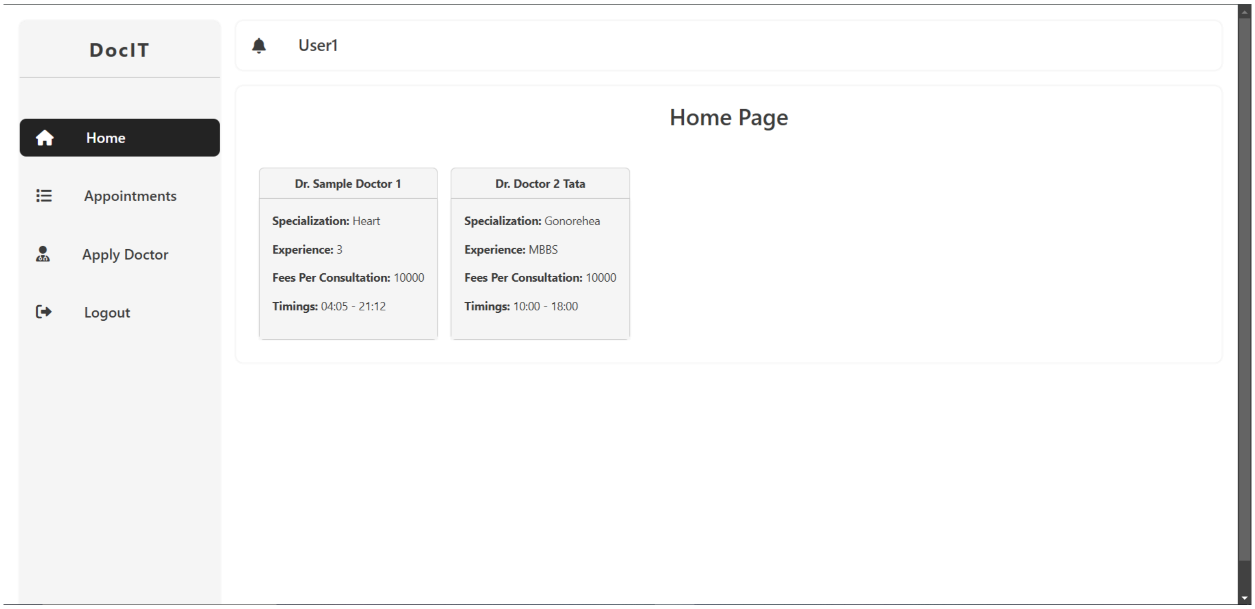
Testing is conducted using Jest and React Testing Library for frontend components, and Mocha with Chai for backend API testing.

## 11. Screenshots or Demo

Screenshots showcasing the homepage, appointment booking form, and admin dashboard will be added.

1. For User Profile

- Homepage



Home Page

Dr. Sample Doctor 1

**Specialization:** Heart

**Experience:** 3

**Fees Per Consultation:** 10000

**Timings:** 04:05 - 21:12

Dr. Doctor 2 Tata

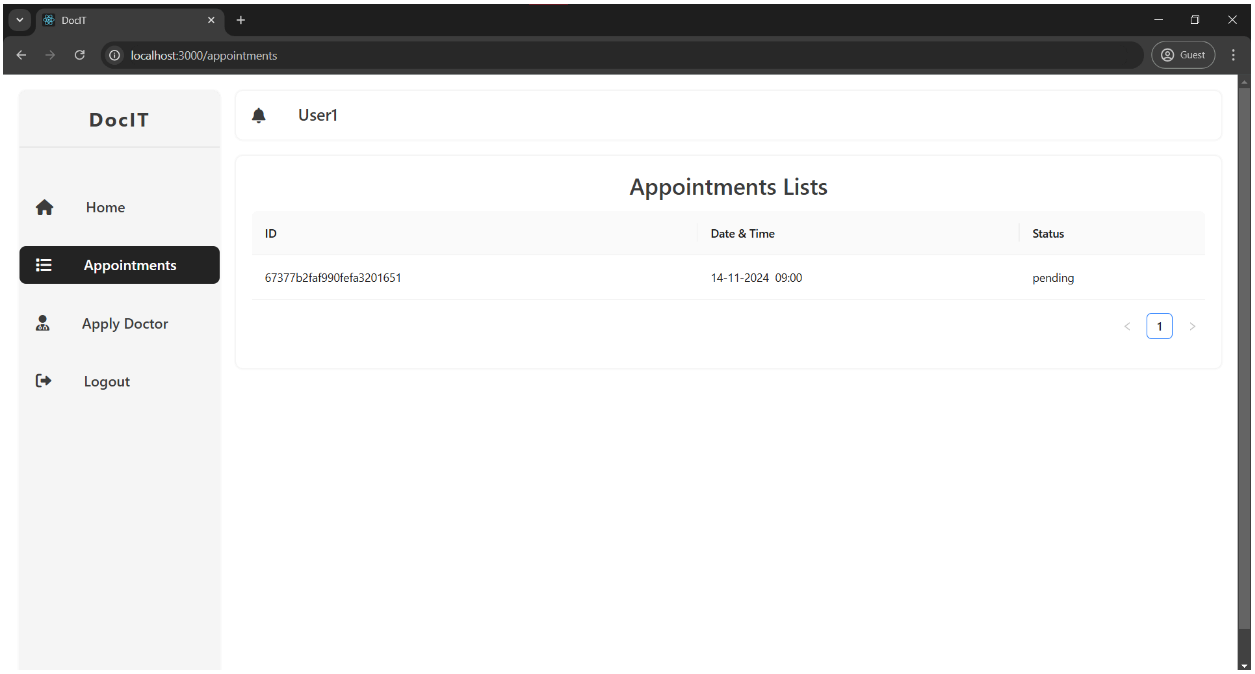
**Specialization:** Gonorehea

**Experience:** MBBS

**Fees Per Consultation:** 10000

**Timings:** 10:00 - 18:00

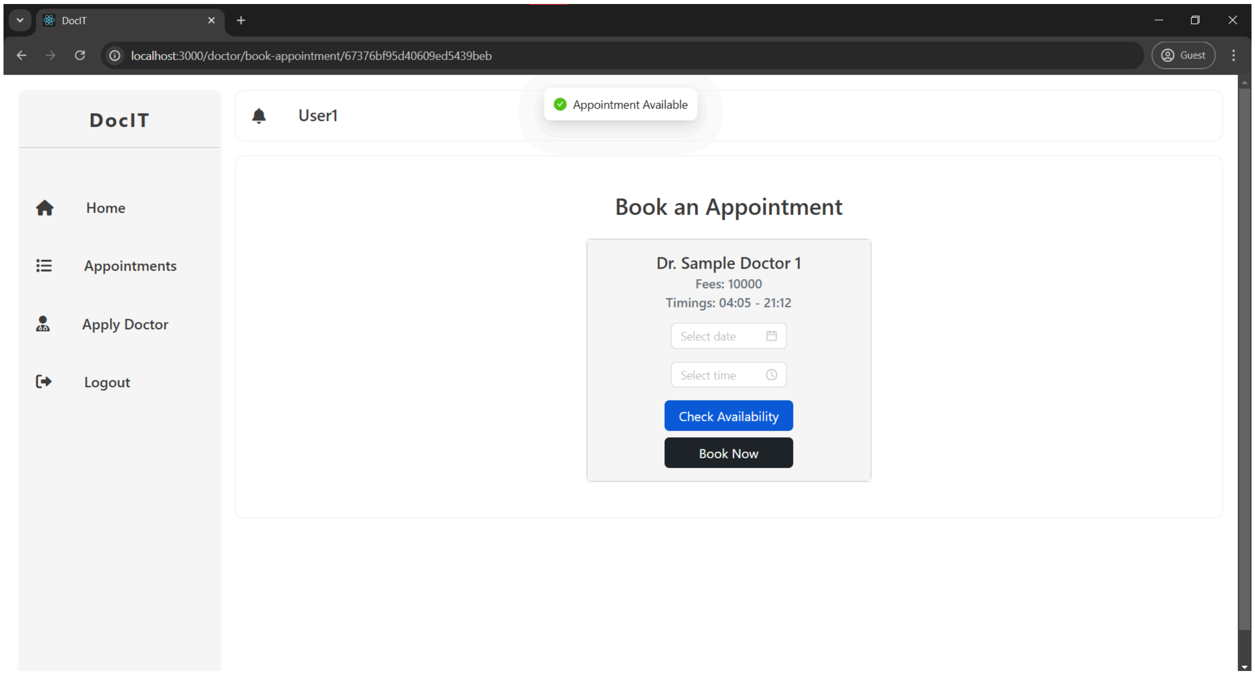
- Appointment Lists



Appointments Lists

| ID | Date & Time | Status |
|---|---|---|
| 67377b2faf990fefa3201651 | 14-11-2024 09:00 | pending |

- Booking Appointment



Appointment Available

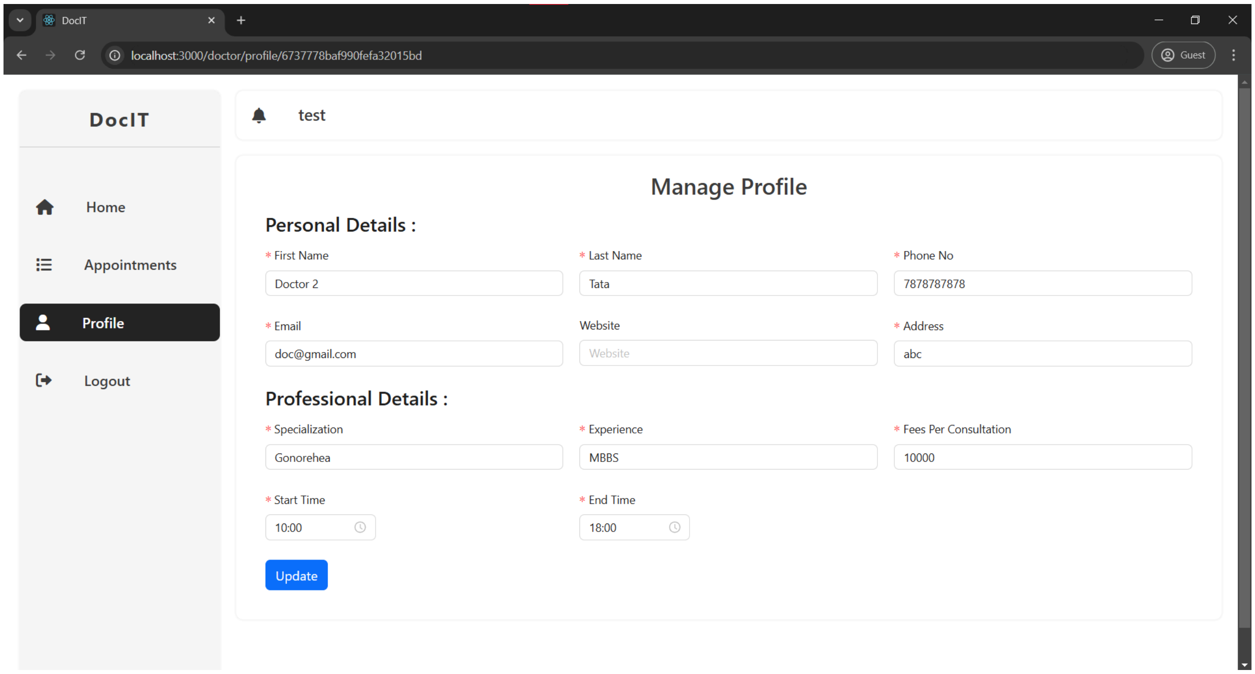
Book an Appointment

Dr. Sample Doctor 1

Fees: 10000

Timings: 04:05 - 21:12

- Apply As Doctor



Manage Profile

Personal Details :

* First Name: Doctor 2

* Last Name: Tata

* Phone No: 7878787878

* Email: doc@gmail.com

Website: Website

* Address: abc

Professional Details :

* Specialization: Gonorehea

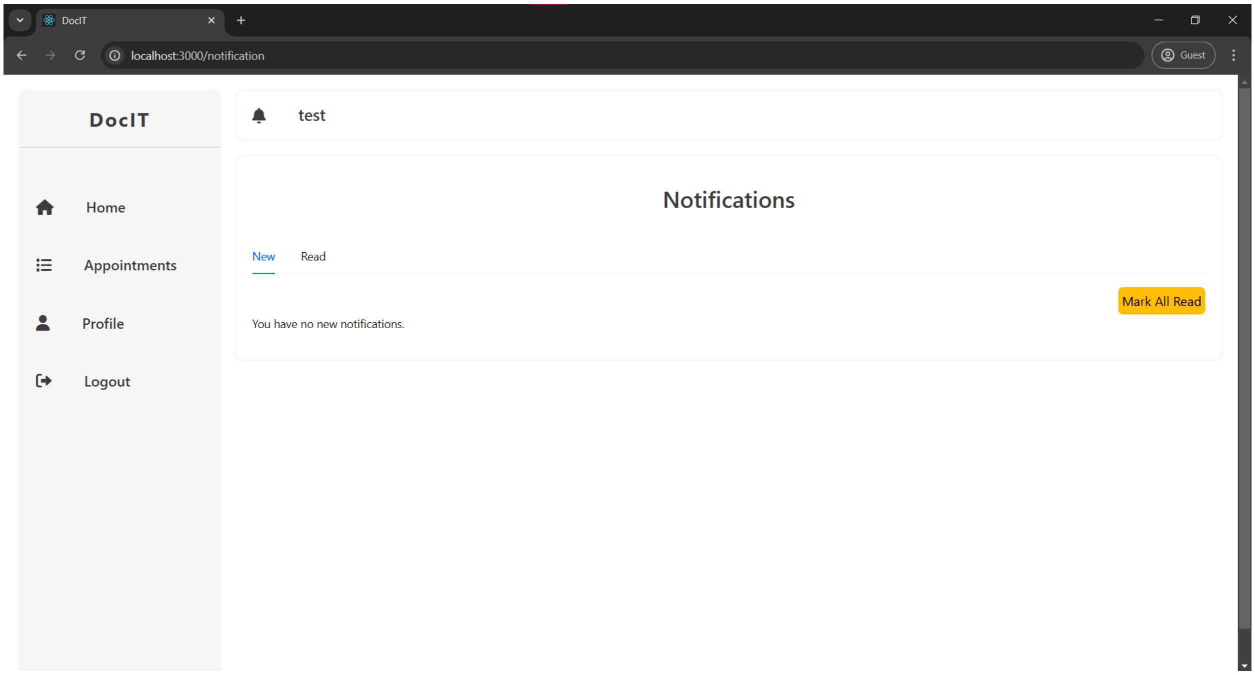
* Experience: MBBS

* Fees Per Consultation: 10000

* Start Time: 10:00

* End Time: 18:00

- Read Notifications

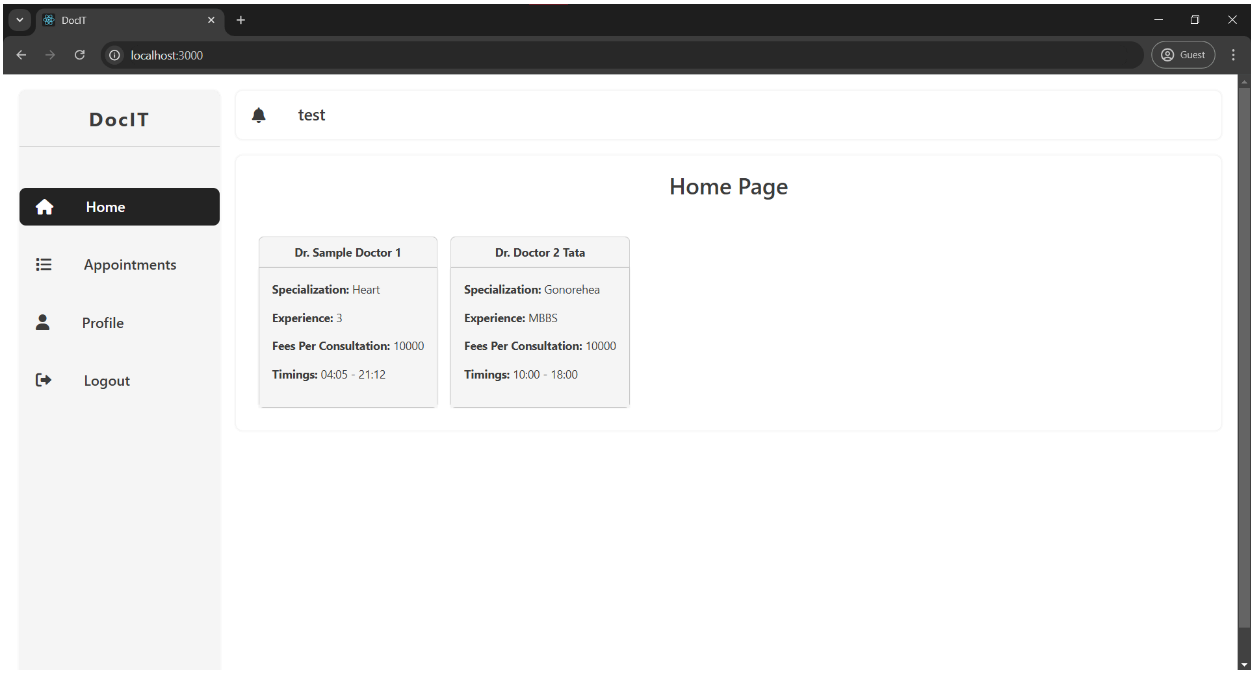


Notifications

You have no new notifications.

## 2. For Doctor Profile

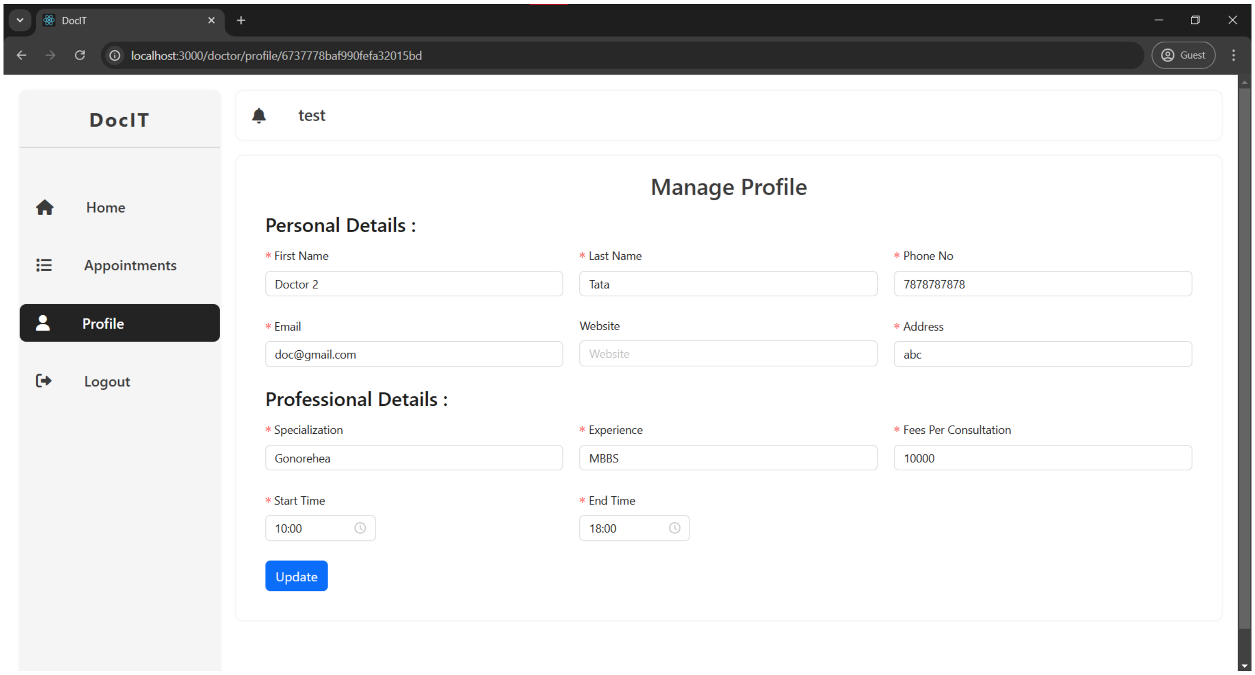
- Homepage



Home Page

| Dr. Sample Doctor 1 | Dr. Doctor 2 Tata |
|---|---|
| **Specialization:** Heart | **Specialization:** Gonorehea |
| **Experience:** 3 | **Experience:** MBBS |
| **Fees Per Consultation:** 10000 | **Fees Per Consultation:** 10000 |
| **Timings:** 04:05 - 21:12 | **Timings:** 10:00 - 18:00 |

-Manage Profile



Manage Profile

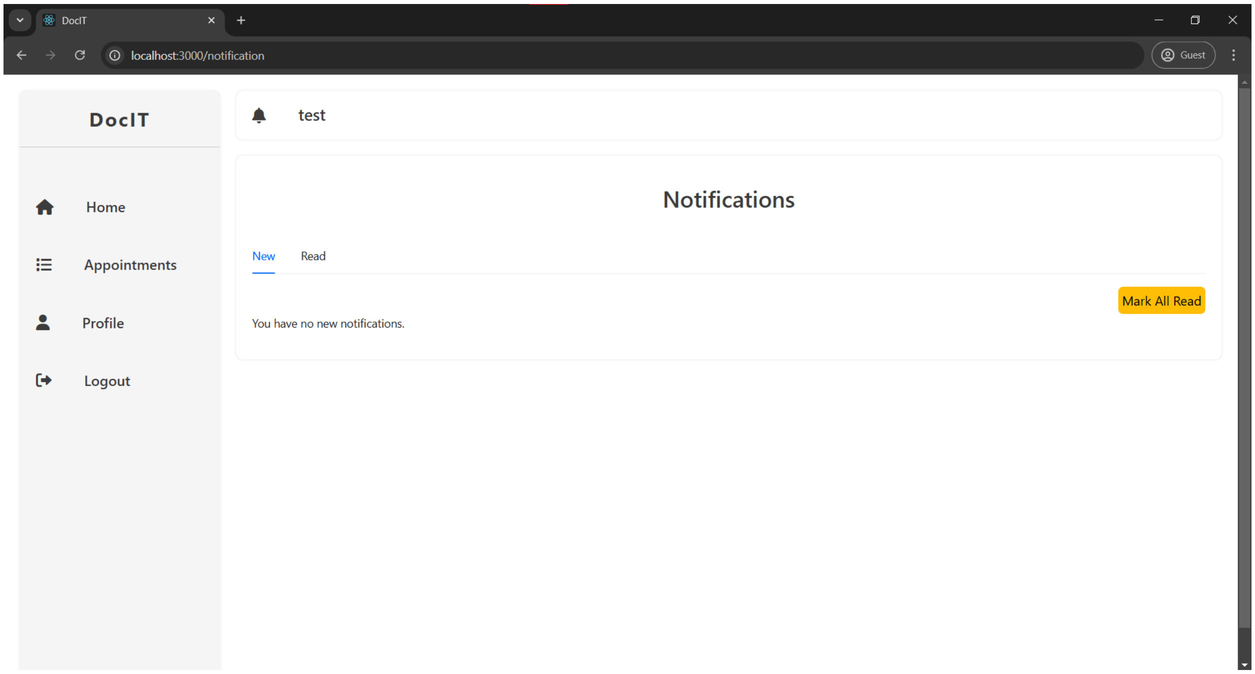
Personal Details :

| | | |
|---|---|---|
| * First Name | * Last Name | * Phone No |
| Doctor 2 | Tata | 7878787878 |
| * Email | Website | * Address |
| doc@gmail.com | Website | abc |

Professional Details :

| | | |
|---|---|---|
| * Specialization | * Experience | * Fees Per Consultation |
| Gonorehea | MBBS | 10000 |
| * Start Time | * End Time | |
| 10:00 | 18:00 | |

- Appointment Notifications

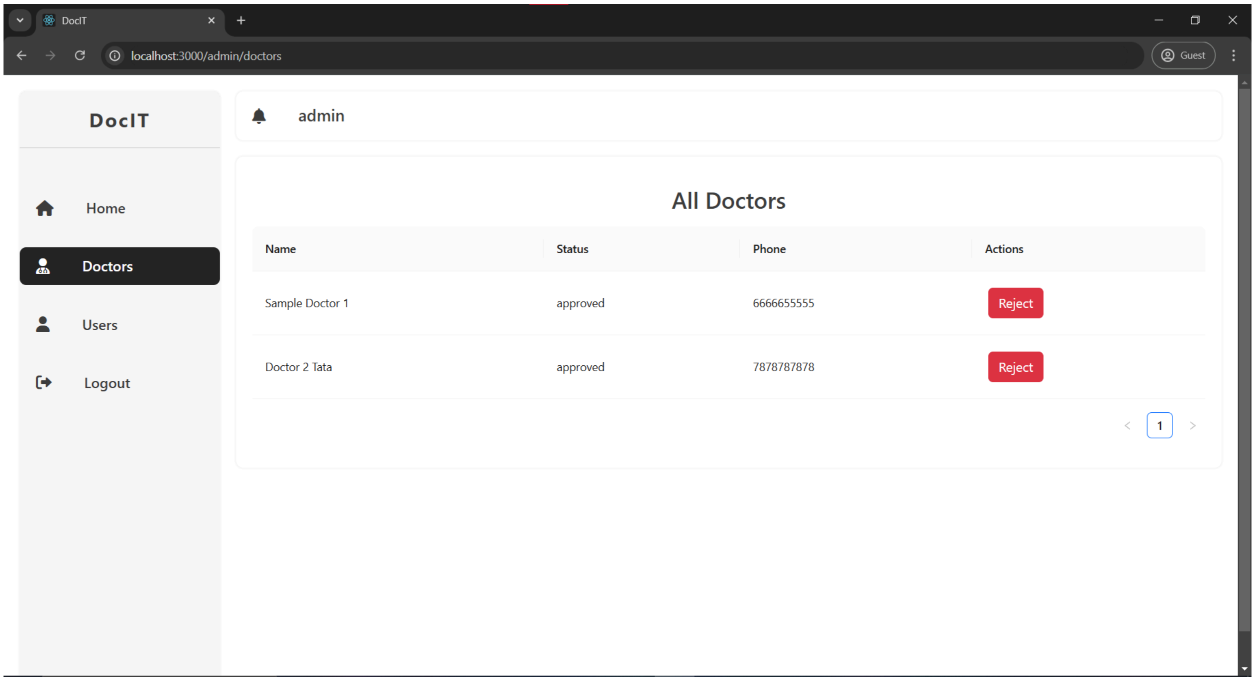


Notifications

You have no new notifications.

### 3. For Admin Profile

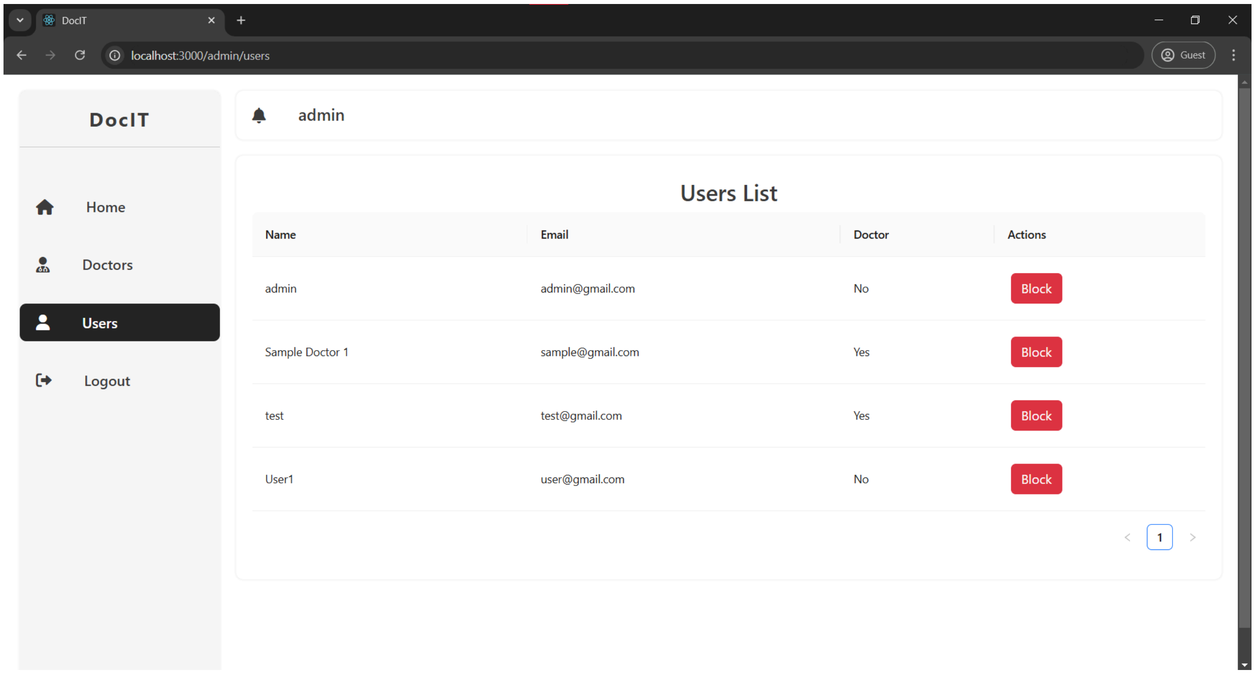
- Doctors List



All Doctors

| Name | Status | Phone | Actions |
|---|---|---|---|
| Sample Doctor 1 | approved | 6666655555 | Reject |
| Doctor 2 Tata | approved | 7878787878 | Reject |

- Users List



Users List

| Name | Email | Doctor | Actions |
|---|---|---|---|
| admin | admin@gmail.com | No | Block |
| Sample Doctor 1 | sample@gmail.com | Yes | Block |
| test | test@gmail.com | Yes | Block |
| User1 | user@gmail.com | No | Block |

## 12. Known Issues

- Inconsistent date formatting across different regions
- Performance optimization required for large user data sets

## 13. Future Enhancements

- Implement video call feature for virtual appointments
- Add payment integration for appointment booking
- Introduce multilingual support for a broader user base