Building a Doctor Appointment System with the MERN Stack

TEAM MEMBERS

BALEESWARAN.K- A48EA91DCBC00F0B93C409AD940772D3

CHINNAPPA VIGNESH - 8C9C4E279700FF7B9D84FE014C1C3293

GUNA CHANDRAN.C-84C B155D66BD646BD2246C84F1B6AED2

JEBASTIN RAJAN PAUL- 68C39013CC5B2C88C1F42599DFDB6580

Introduction to Healthcare Systems

In today's fast-paced world, **efficient healthcare** systems are crucial. This presentation explores building a **Doctor Appointment System** using the **MERN Stack**, which includes **MongoDB**, **Express**, **React**, and **Node.js**. Our goal is to streamline the appointment process for both patients and healthcare providers.

Understanding the MERN Stack

The **MERN Stack** is a powerful collection of technologies used for web development. It consists of **MongoDB** for database management, **Express** for server-side logic, **React** for user interfaces, and **Node.js** for backend operations. Together, they enable the creation of dynamic and responsive applications.

Key Challenges in Healthcare

Healthcare systems face several challenges, including **long wait times**, **scheduling conflicts**, and **communication gaps**. By addressing these issues through technology, we can enhance patient satisfaction and improve overall healthcare delivery. Streamlining the appointment process is a significant step in this direction.

Defining System Requirements

Before developing the system, it's essential to define clear **requirements**. These include user authentication, appointment scheduling, notifications, and data management. Understanding the needs of both **patients** and **doctors** is crucial for creating a user-friendly and effective application.

Designing the Database Schema

A well-structured **database schema** is vital for managing user data and appointments. In our system, we will create collections for **users**, **appointments**, and **doctors**. This organization allows for efficient data retrieval and management, ensuring a smooth user experience.

Building the Backend with Node.js

Using **Node.js** and **Express**, we will develop the backend of our application. This includes setting up API endpoints for user authentication, appointment creation, and retrieval. The backend serves as the backbone of our system, handling all data processing and business logic.

BACKEND SETUP

```
const express = require('express');

const mongoose = require('mongoose');

const app = express();

app.use(express.json()); // Middleware for JSON parsing

mongoose.connect('mongodb://localhost:27017/appointments')

.then(() => console.log("Connected to MongoDB"))

.catch((err) => console.log(err));
```

MONGO DB SCHEMA FOR DOCTOR APPOINTMENT

```
const appointmentSchema = new mongoose.Schema({

patientId: { type: mongoose.Schema.Types.ObjectId, ref: 'Patient' },

doctorId: { type: mongoose.Schema.Types.ObjectId, ref: 'Doctor' },

date: Date,

time: String,

status: { type: String, default: 'Booked' }

});

const Appointment = mongoose.model('Appointment', appointmentSchema);
```

Creating API routes to handle appointment actions.

```
app.post('/appointments', async (req, res) => {

const appointment = new Appointment(req.body);

await appointment.save();

res.status(201).json(appointment);

});
```

API route to fetch all appointments.

```
app.get('/appointments', async (req, res) => {
const appointments = await
Appointment.find();
res.status(200).json(appointments); });
```

Creating the Frontend with React

The frontend of our application will be built using **React**. This allows us to create a dynamic user interface where patients can easily view available appointments, book them, and receive notifications. A responsive design ensures accessibility across various devices.

React Frontend Setup

bash
npx create-react-app doctor-appointment;
cd doctor-appointment;
npm start

React Component for Booking Appointment

```
import React, { useState } from 'react';
const AppointmentForm = () => { const
[doctorId, setDoctorId] = useState('');
const [patientId, setPatientId] =
useState('');
const [date, setDate] = useState('');
const [time, setTime] = useState('');
```

REACT COMPONENTS FOR BOOKING APPOINTMENTS

```
import React, { useState } from 'react';
const AppointmentForm = () => {    const [doctorId, setDoctorId] = useState('');
const [patientId, setPatientId] = useState('');
const [date, setDate] = useState('');
const [time, setTime] = useState('');
const handleSubmit = async (e) => {e.preventDefault();
const response = await fetch('/appointments',
 {method: 'POST', headers: { 'Content-Type': 'application/json' },
body: JSON.stringify({ doctorId, patientId, date, time }),});
const data = await response.json();      alert('Appointment booked!');};
return (      <form onSubmit={handleSubmit}>
<input type="text" placeholder="Doctor ID" onChange={(e) => setDoctorId(e.target.value)} />
<input type="text" placeholder="Patient ID" onChange={(e) => setPatientId(e.target.value)} />
<input type="date" onChange={(e) => setDate(e.target.value)} />
<input type="time" onChange={(e) => setTime(e.target.value)} />
<button type="submit">Book Appointment</button></form>);};
export default AppointmentForm;
```

React Component for Viewing Appointments

```
import React, { useState, useEffect } from 'react';
const AppointmentList = () => {
const [appointments, setAppointments] = useState([]);
useEffect(() => {
fetch('/appointments')
.then((res) => res.json())
.then((data) =>
setAppointments(data));
}, []);
return(
<ul>
{appointments.map((appointment) => (
<li key={appointment._id}>
{appointment.date} -
{appointment.time} with Doctor {appointment.doctorId}</li>)
)}
</ul>)
;};
export default AppointmentList;
```

React Component for Cancelling Appointment

```
const cancelAppointment = async (id) => {
await fetch(`/appointments/${id}`,
{ method: 'DELETE' });
alert('Appointment canceled!');
};
```

Implementing User Authentication

User authentication is critical for security. We will implement **JWT (JSON Web Tokens)** for secure login and session management. This ensures that patient and doctor data remains confidential while allowing users to access their accounts safely.

Deploying the Application

Once testing is complete, we will deploy the application using platforms like **Heroku** or **AWS**. This step involves setting up the server environment and ensuring the database is properly connected. A successful deployment allows users to access the system online.

Testing the Application

Testing is a crucial phase in the development process. We will conduct both **unit testing** and **integration testing** to ensure every component functions as intended. This helps identify bugs early and guarantees a smooth user experience upon deployment.

Get Started

Already have an account ? Login

Create an Account

Name

Name

Email

Email

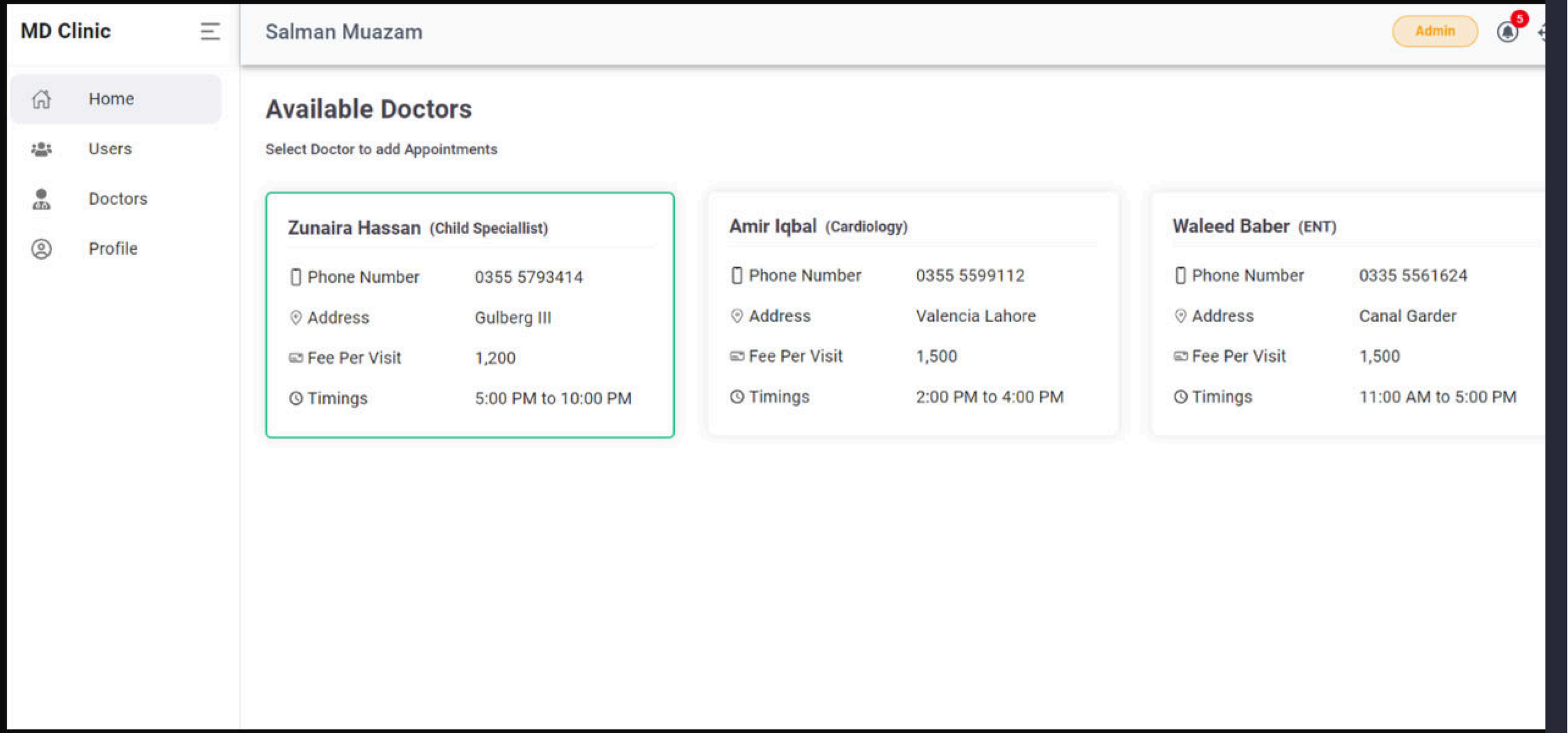
Mobile Number

+92

Password

Password

Sign Up


MD Clinic
Salman Muazam
Admin
5
Home
Users
Doctors
Profile
Available Doctors
Select Doctor to add Appointments
Zunaira Hassan (Child Speciallist)
Phone Number 0355 5793414
Address Gulberg III
Fee Per Visit 1,200
Timings 5:00 PM to 10:00 PM
Amir Iqbal (Cardiology)
Phone Number 0355 5599112
Address Valencia Lahore
Fee Per Visit 1,500
Timings 2:00 PM to 4:00 PM
Waleed Baber (ENT)
Phone Number 0335 5561624
Address Canal Garder
Fee Per Visit 1,500
Timings 11:00 AM to 5:00 PM

Home
Users
Doctors
Profile

Users

Name	Email	Date	Roles	Actions
Salman Muazam	admin@gmail.com	11/22/2023, 12:44 PM	Owner	
Zunaira Hassan	zunaira@gmail.com	11/22/2023, 7:52 PM	Doctor	Delete
Muhammad Kashif	kashif@gmail.com	11/22/2023, 7:53 PM	User	Delete
Ch Faizan	faizan@gmail.com	11/22/2023, 7:55 PM	User	Delete
Amir Iqbal	amir@gmail.com	11/22/2023, 7:56 PM	Doctor	Delete
Waleed Baber	baber@gmail.com	11/22/2023, 10:55 PM	Doctor	Delete
Test User	test@gmail.com	11/22/2023, 11:04 PM	User	Delete
John Doe	john@gmail.com	11/22/2023, 11:09 PM	User	Delete

MD Clinic

Salman Muazam

Admin

- Home
- Users
- Doctors
- Profile

Profile Details

Owner

Admin

SM

Salman Muazam

0323 4910955

Created At: 11/22/2023, 12:44 PM

Home

Appointments

Apply Doctor

Profile

Apply For Doctor

1 Basic Information

Prefix: Dr.

Full Name: Ch Faizan

Mobile Number: +92 315-5516847

Website: Website

Address: DHA Phase IV

2 Professional Information

Specialization: ENT

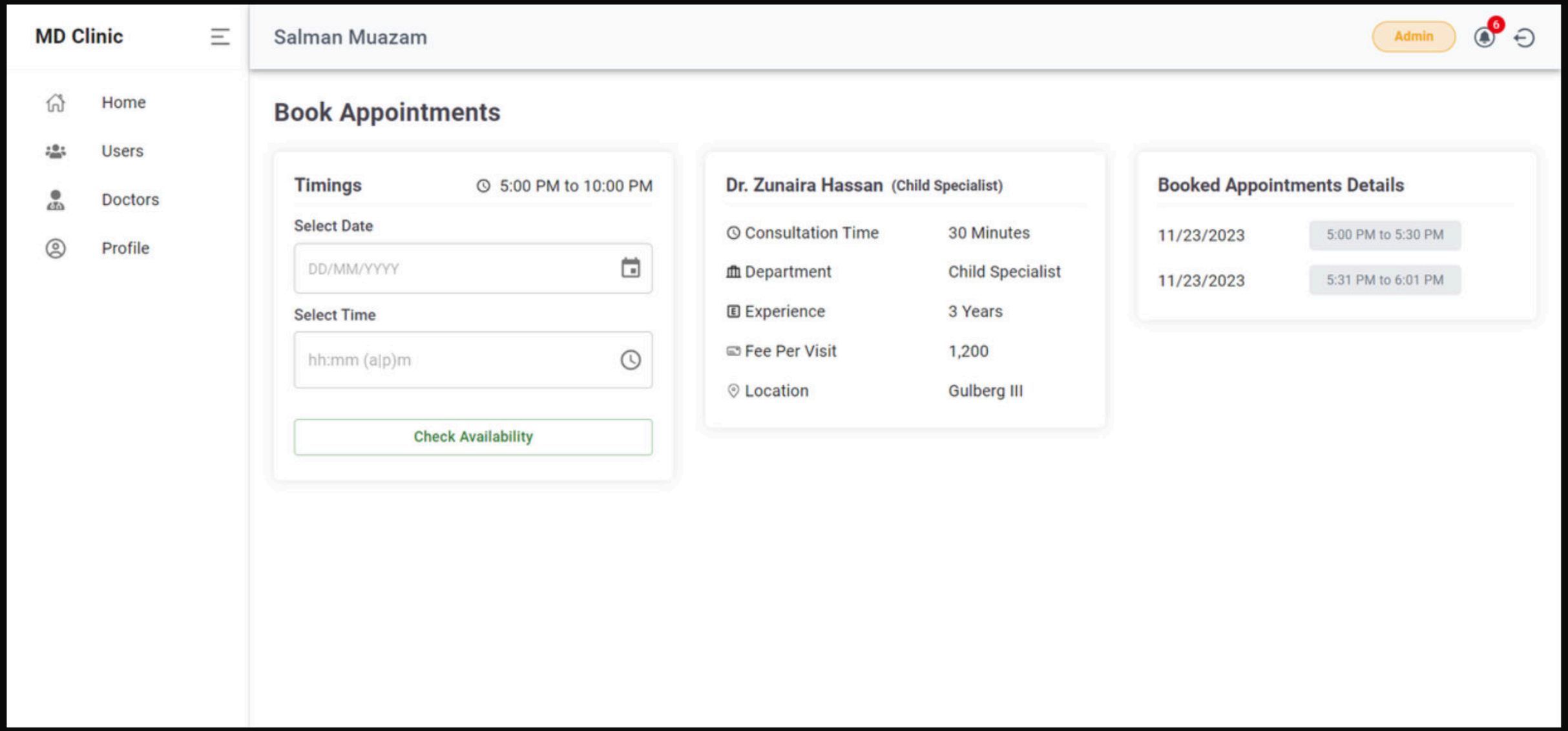
Experience: 5

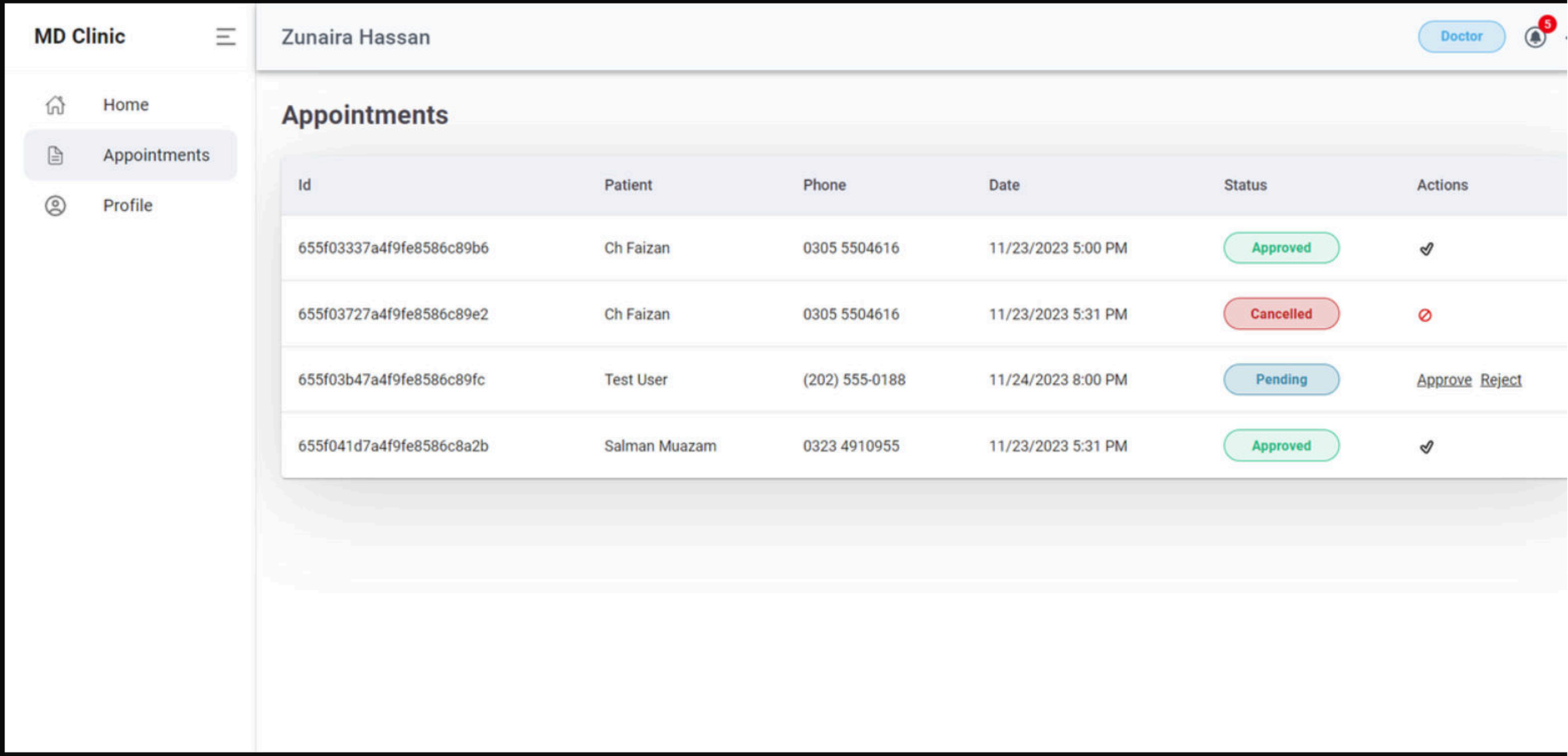
Fee Per Consultation: 1200

Start Time: 07:00 PM

End Time: 10:00 PM

Apply


MD Clinic
Salman Muazam
Admin
6
Home
Users
Doctors
Profile
Book Appointments
Timings
5:00 PM to 10:00 PM
Select Date
DD/MM/YYYY
Select Time
hh:mm (a|p)m
Check Availability
Dr. Zunaira Hassan (Child Specialist)
Consultation Time
30 Minutes
Department
Child Specialist
Experience
3 Years
Fee Per Visit
1,200
Location
Gulberg III
Booked Appointments Details
11/23/2023
5:00 PM to 5:30 PM
11/23/2023
5:31 PM to 6:01 PM


MD Clinic
Zunaira Hassan
Doctor
5
Home
Appointments
Profile
Appointments
Id
Patient
Phone
Date
Status
Actions
655f03337a4f9fe8586c89b6
Ch Faizan
0305 5504616
11/23/2023 5:00 PM
Approved
655f03727a4f9fe8586c89e2
Ch Faizan
0305 5504616
11/23/2023 5:31 PM
Cancelled
655f03b47a4f9fe8586c89fc
Test User
(202) 555-0188
11/24/2023 8:00 PM
Pending
Approve
Reject
655f041d7a4f9fe8586c8a2b
Salman Muazam
0323 4910955
11/23/2023 5:31 PM
Approved

Doctors

Profile

Unseen | Seen

MARK ALL AS READ

Name: Zunaira Hassan
Title: New Doctor Request
Message: Zunaira Hassan has requested to join as a doctor.

Name: Amir Iqbal
Title: New Doctor Request
Message: Amir Iqbal has requested to join as a doctor.

Name: Waleed Baber
Title: New Doctor Request
Message: Waleed Baber has requested to join as a doctor.

Name: Test User
Title: New Doctor Request
Message: Test User has requested to join as a doctor.

Name: John Doe
Title: New Doctor Request
Message: John Doe has requested to join as a doctor.

MD Clinic

Zunaira Hassan

Doctor

5

Home

Appointments

Profile

To exit full screen, press and hold Esc

Notifications

Unseen | Seen

MARK ALL AS READ

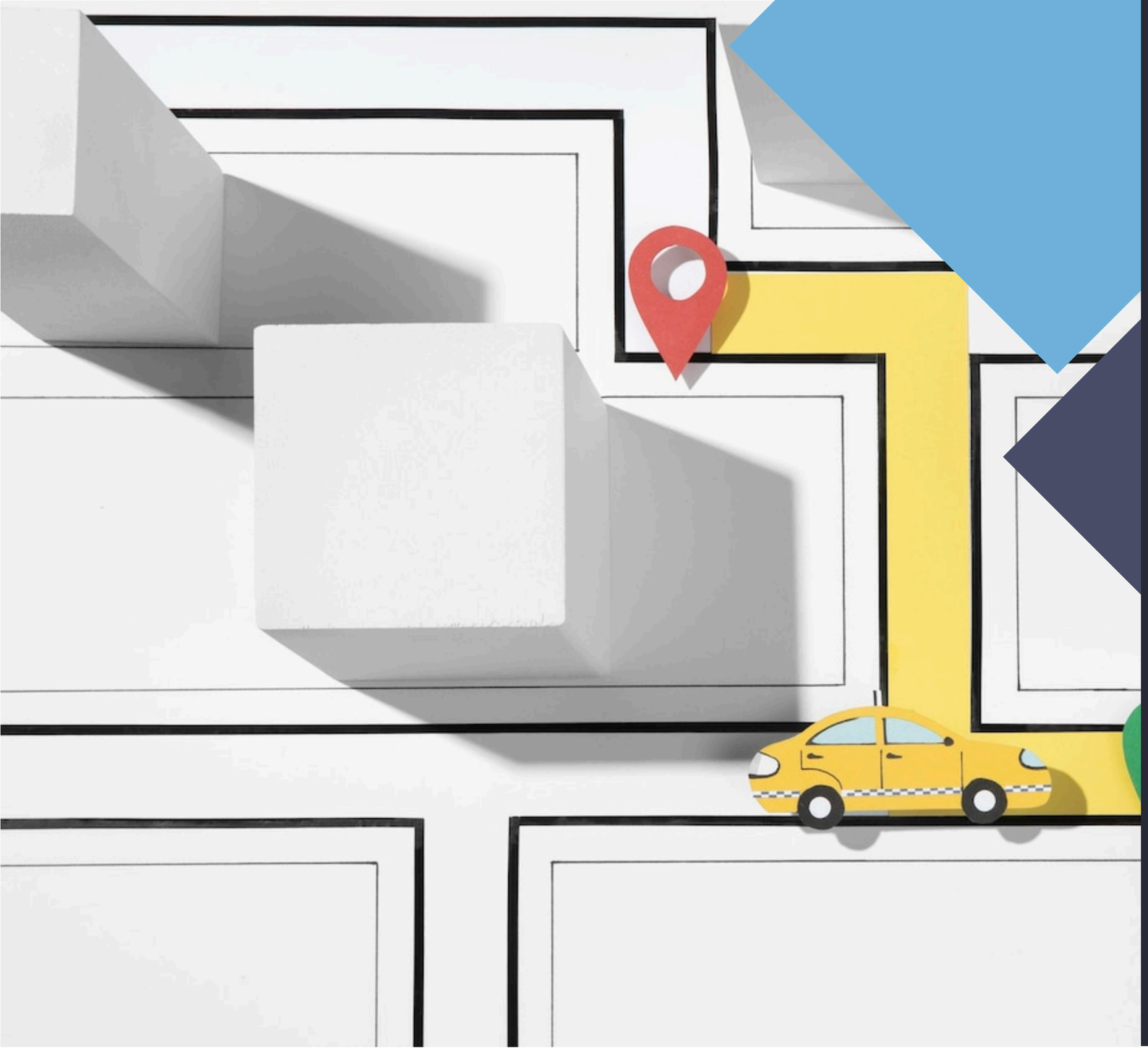
Name: Zunaira Hassan
Title: Your requested successfully accepted
Message: Your doctor request has been approved

Name: Ch Faizan
Title: New Appointment Request
Message: A new appointment request has been made by Ch Faizan

Name: Ch Faizan
Title: New Appointment Request
Message: A new appointment request has been made by Ch Faizan

Name: Test User
Title: New Appointment Request
Message: A new appointment request has been made by Test User

Name: Salman Muazam
Title: New Appointment Request
Message: A new appointment request has been made by Salman Muazam



Future Enhancements

After deployment, we can explore **future enhancements** such as integrating telehealth features, improving the user interface, and implementing advanced analytics. Continuous improvement ensures that the system remains relevant and meets the evolving needs of users.

Conclusion

In conclusion, building a **Doctor Appointment System** with the **MERN Stack** offers a practical solution to streamline healthcare processes. By leveraging modern technologies, we can enhance patient experiences and improve operational efficiency in healthcare settings.
Thank you for your attention!

THANK YOU