CSE471: System Analysis and Design
Project Report
Project Title: Hospital Management System

Group No: 1, CSE471 Lab Section: 2, Fall 2024	
ID	**Name**
22101334 / 24241284	Md. Rumman Shahriar
22101736	Md. Asif Hasan Biplob
21301417	Rodoshie Reheean

Submission Date: 08/01/2025

Table of Contents

1. System Request

Business need:

To ensure that proper healthcare is provided to the people in need, appointments, prescriptions, and billing must be managed efficiently and on time. Since healthcare facilities in Bangladesh often suffer from scheduling appointments properly, a digital solution is necessary to make the system efficient and allow it to focus on enhancing patient care while taking away the administrative burden. Therefore, the proposed Hospital Management System addresses these challenges by integrating modern technology into healthcare management and ensures smooth operations and improved service quality.

Business requirements:

- **Patient Management:** Patients can manage their profiles, book, access prescriptions, handle billing, and provide feedback.
- **Doctor Management:** Doctors can manage their profiles, and schedules, maintain appointments, and write prescriptions.
- **Pharmacy and Billing:** Pharmacists can manage prescriptions, handle billing, and maintain medicine inventory.
- **Administrative Control:** Provide admins to oversee feedback left by patients and manage healthcare professionals.
- **Role-based Access:** Role-based access system to ensure data privacy and secure handling of sensitive information.
- **Scalability:** The system can be accommodated for future growth in functionality and users.
- **User Experience:** The system has intuitive interfaces to make it user-friendly.

Business value:

Patients are more likely to appreciate systems that allow easy scheduling and appointment-making. Instead of waiting in the queue for hours on end just for a regular doctor visit, the patient can use the system to make appointments and visit the hospital on time without extra hassle. It is also convenient for the patient to use the system to get desired medicines for themselves and pay the bill online instead of visiting the hospital in person for such small tasks. Therefore, the proposed system brings good business value by ensuring the following aspects:

- **Enhanced Operational Support:** The system makes scheduling, billing, and prescriptions easier for patients by letting doctors and pharmacists manage them manually.
- **Patient Satisfaction:** The appointment booking is simplified and the system provides easy access to medical records and prescriptions while also allowing patients to leave feedback if necessary.
- **Data Security and Privacy:** The system ensures that sensitive data is securely stored and accessed only by authorized personnel.

Special issues or constraints:

- **Lack of Automation:** The actions taken by the Pharmacists are not automated, they have to manually approve the billing which needs to be automated for efficiency.
- **No Transaction API Available:** The payment uses no API and therefore the payment is made immediately upon a click without any authentication.
- **No Chat System:** The system provides no live support for the users to navigate with ease.

2. Functional Requirements

	Functional Requirements
Module 0: Authentication	**1. Patient Login and Register** **2. Doctor Login** **3. Pharmacist Login** **4. Admin Login**
Module 1: Patient Management	1. **-View Dashboard:** Patients can view the dashboard of the hospital. 2. - **Profile Management:** Patients can view and edit their profiles. 3. - **Appointment Management:** Patients can book appointments with doctors. 4. - **Prescription Viewing:** Patients can access their past prescriptions. 5. - **Billing and Payment:** Patients can pay their bills and payments for appointments. 6. - **Feedback:** Patients can provide feedback on services.

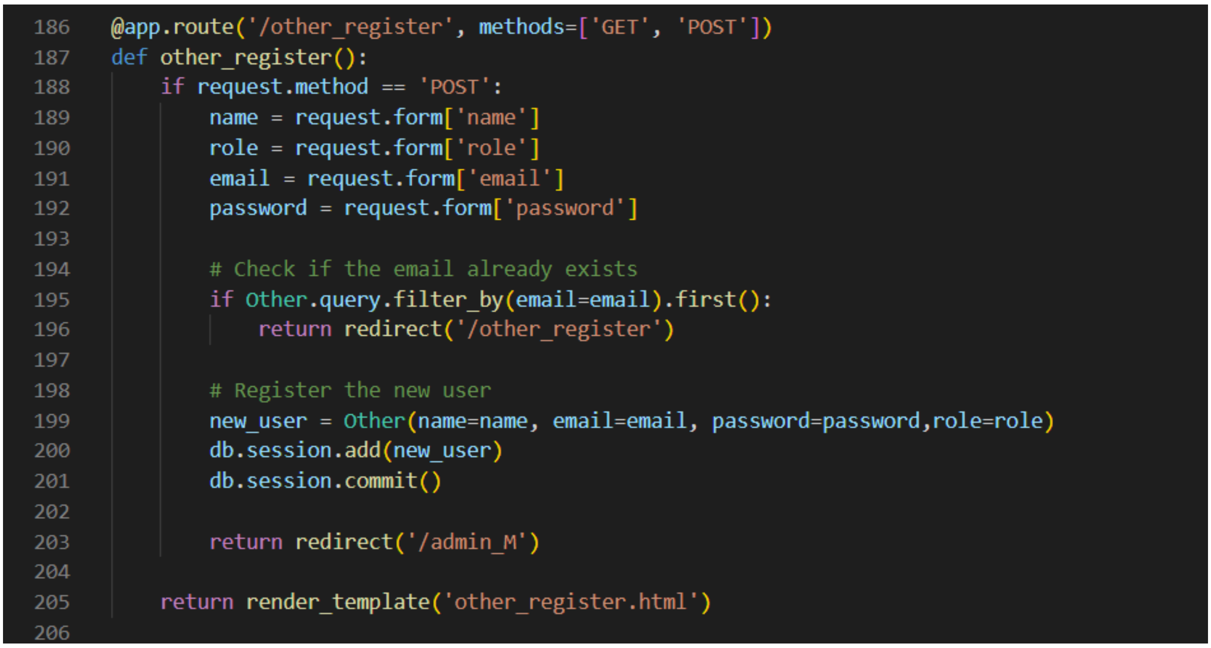
Module 2: Doctor Management	1. **-Profile Management:** Doctors can view and edit their profiles and edit their information. 2. - **Schedule Management:** Doctors can view and manage their appointments according to their schedules. 3. - **Appointment Approval:** Doctors can approve, decline, or reschedule patient appointments. 4. - **Prescription Writing:** Doctors can prescribe medications for patients.
Module 3: Pharmacy and Billing	1. - **Prescription Management:** Pharmacists can view the prescriptions of patients. 2. - **Medicine Management:** Pharmacists can add medicines to inventory and also change and edit the medicine price and description. 3. **- Storage Management:** Pharmacists can manage the medicine stock of the hospital store.
Module 4: Admin Control	1. - **Appoint Doctor and Pharmacist:** Admins can add Doctors and Pharmacists in the system accordingly. 2. - **Doctor Management:** Admins can add, remove, and edit doctors and doctor's public information. 3. - **Pharmacist Management:** Admins can add, remove, and edit pharmacists and pharmacist's public information. 4. - **Check Feedback:** Admins can view patient feedback and delete them after taking proper steps.

3. Technology (Framework, Languages)

Frontend: HTML, and CSS for building user interfaces.
Backend: Flask for server-side logic, routing, and database management.
Database: Mydatabase.db for storing and managing user and system data.

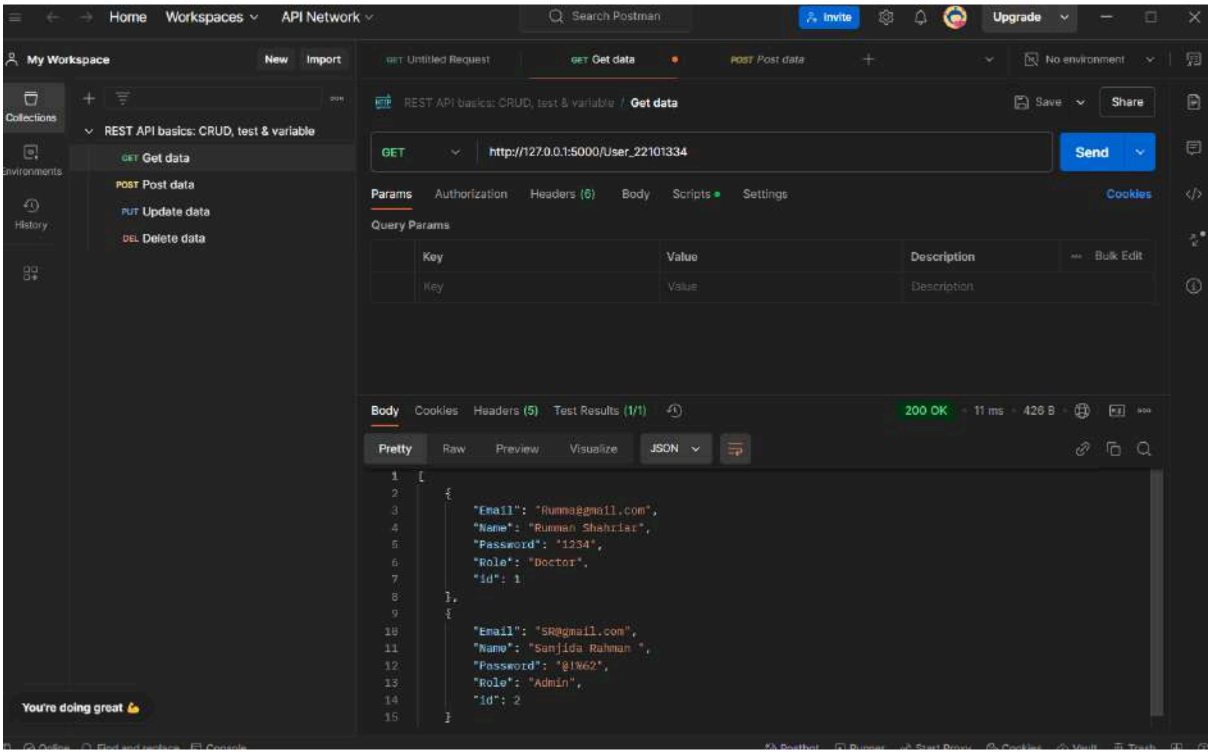
4. Backend Development

API 1:



```python
@app.route('/other_register', methods=['GET', 'POST'])
def other_register():
    if request.method == 'POST':
        name = request.form['name']
        role = request.form['role']
        email = request.form['email']
        password = request.form['password']

        # Check if the email already exists
        if Other.query.filter_by(email=email).first():
            return redirect('/other_register')

        # Register the new user
        new_user = Other(name=name, email=email, password=password,role=role)
        db.session.add(new_user)
        db.session.commit()

        return redirect('/admin_M')

    return render_template('other_register.html')

```

The code captures user inputs like name, role, email, and password, checks if the email is already registered to prevent duplicates, and then saves the new user in the database. Upon successful registration, it redirects to the admin dashboard (`/admin_M`) for further actions. This setup ensures basic validation and a smooth registration process, though it could be improved with password hashing, error messages, and additional validations for better security and usability.

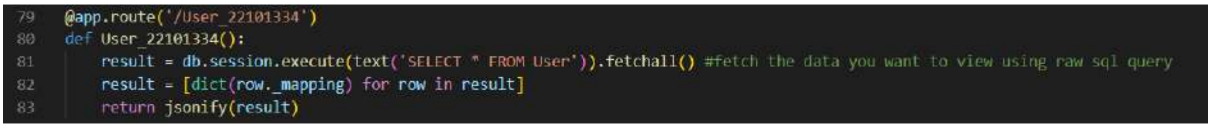


REST API basics: CRUD, test & variable / Get data

GET http://127.0.0.1:5000/User_22101334

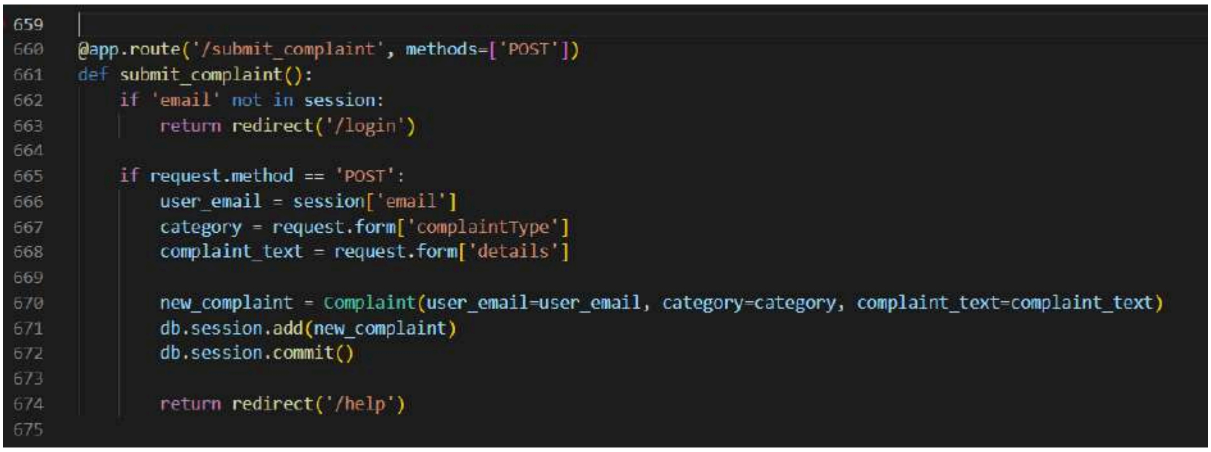
200 OK · 11 ms · 426 B

```
[
    {
        "Email": "Rumma@gmail.com",
        "Name": "Rumman Shahriar",
        "Password": "1234",
        "Role": "Doctor",
        "id": 1
    },
    {
        "Email": "SR@gmail.com",
        "Name": "Sanjida Rahman ",
        "Password": "@1%62",
        "Role": "Admin",
        "id": 2
    }
```



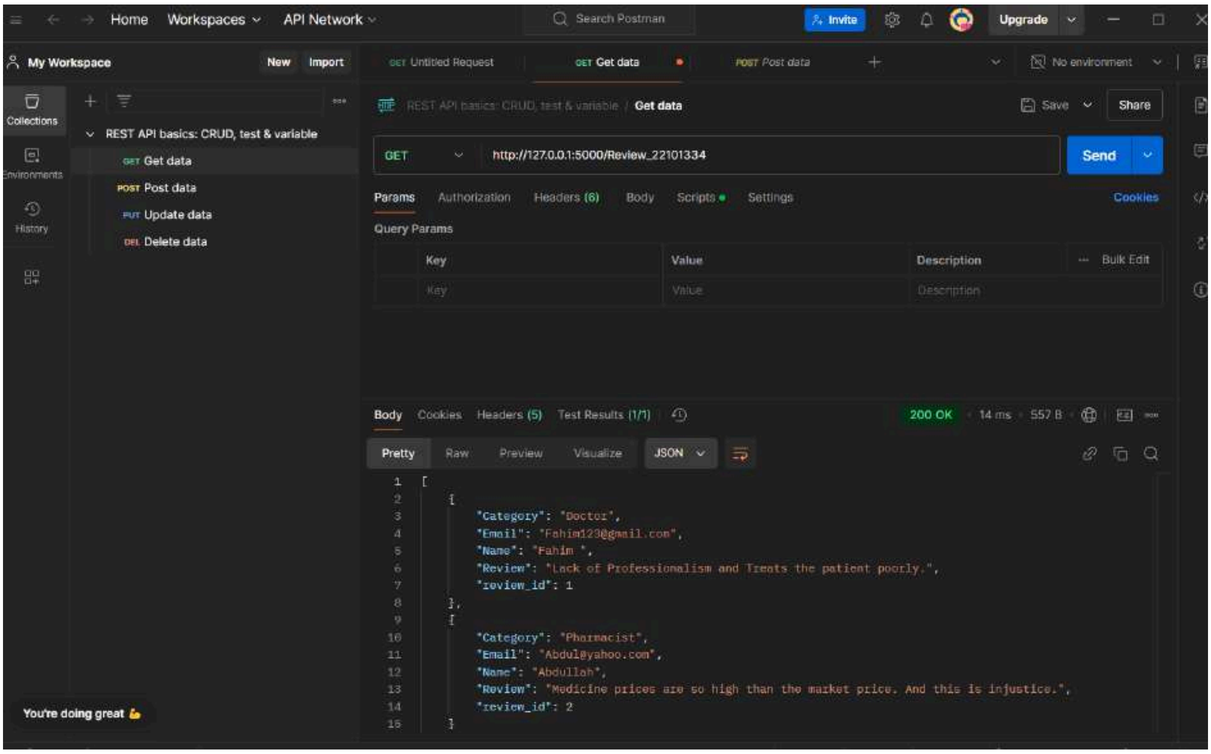
```python
@app.route('/User_22101334')
def User_22101334():
    result = db.session.execute(text('SELECT * FROM User')).fetchall() #fetch the data you want to view using raw sql query
    result = [dict(row._mapping) for row in result]
    return jsonify(result)
```

API 2:



```python

@app.route('/submit_complaint', methods=['POST'])
def submit_complaint():
    if 'email' not in session:
        return redirect('/login')

    if request.method == 'POST':
        user_email = session['email']
        category = request.form['complaintType']
        complaint_text = request.form['details']

        new_complaint = Complaint(user_email=user_email, category=category, complaint_text=complaint_text)
        db.session.add(new_complaint)
        db.session.commit()

        return redirect('/help')

```

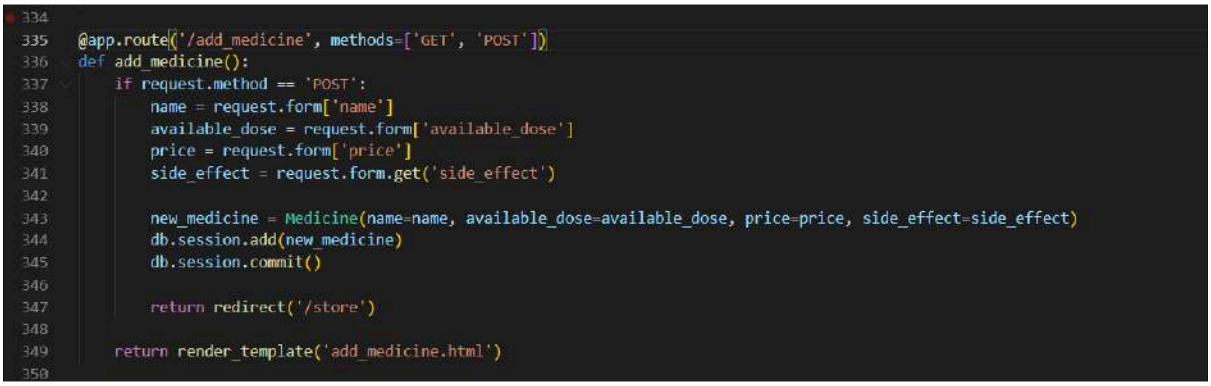
The `submit_complaint` function handles the submission of a complaint by a user in a Flask web application. It first checks if the user is logged in by verifying the presence of an email in the session; if not, it redirects the user to the login page. Upon receiving a POST request, the function retrieves the user's email, complaint category, and text from the form data. A new `Complaint` object is created with this information and added to the database, after which the changes are committed. Finally, the user is redirected to the help page.



GET http://127.0.0.1:5000/Review_22101334

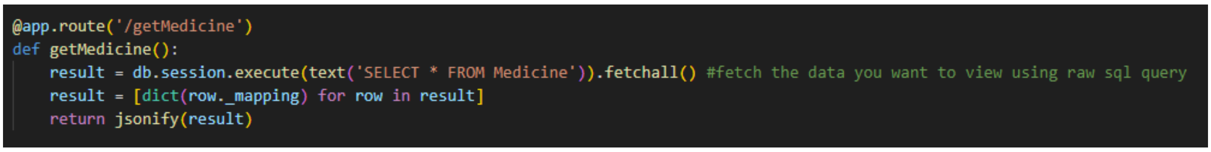
```json
[
    {
        "Category": "Doctor",
        "Email": "Fahim123@gmail.com",
        "Name": "Fahim ",
        "Review": "Lack of Professionalism and Treats the patient poorly.",
        "review_id": 1
    },
    {
        "Category": "Pharmacist",
        "Email": "Abdul@yahoo.com",
        "Name": "Abdullah",
        "Review": "Medicine prices are so high than the market price. And this is injustice.",
        "review_id": 2
    }
```

API 3:

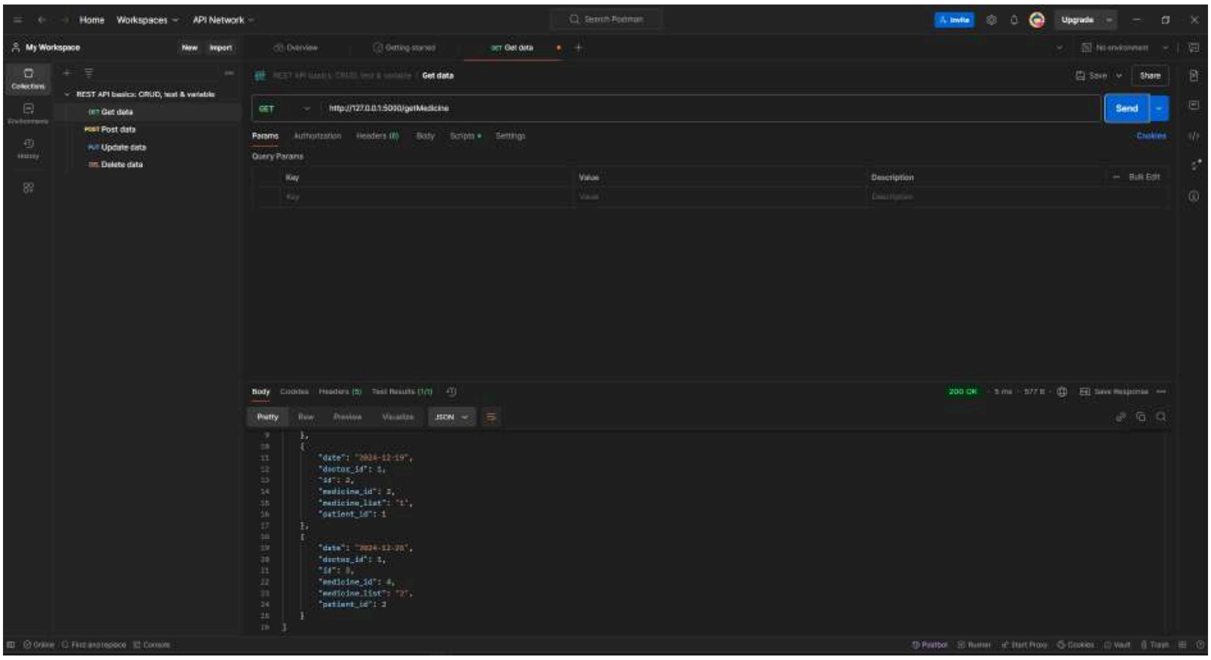


```
@app.route('/add_medicine', methods=['GET', 'POST'])
def add_medicine():
    if request.method == 'POST':
        name = request.form['name']
        available_dose = request.form['available_dose']
        price = request.form['price']
        side_effect = request.form.get('side_effect')

        new_medicine = Medicine(name=name, available_dose=available_dose, price=price, side_effect=side_effect)
        db.session.add(new_medicine)
        db.session.commit()

        return redirect('/store')

    return render_template('add_medicine.html')
```

The `add_medicine` function in a Flask web application handles the process of adding new medicine details to the database. It first checks if the request method is POST, indicating that the form has been submitted. The function then retrieves the medicine's name, available dose, price, and any side effects from the form data. A new `Medicine` object is created with this information and added to the database. After committing the changes, the user is redirected to the store page. If the request method is GET, it renders the `add_medicine.html` template to display the form.



```
@app.route('/getMedicine')
def getMedicine():
    result = db.session.execute(text('SELECT * FROM Medicine')).fetchall() #fetch the data you want to view using raw sql query
    result = [dict(row._mapping) for row in result]
    return jsonify(result)
```

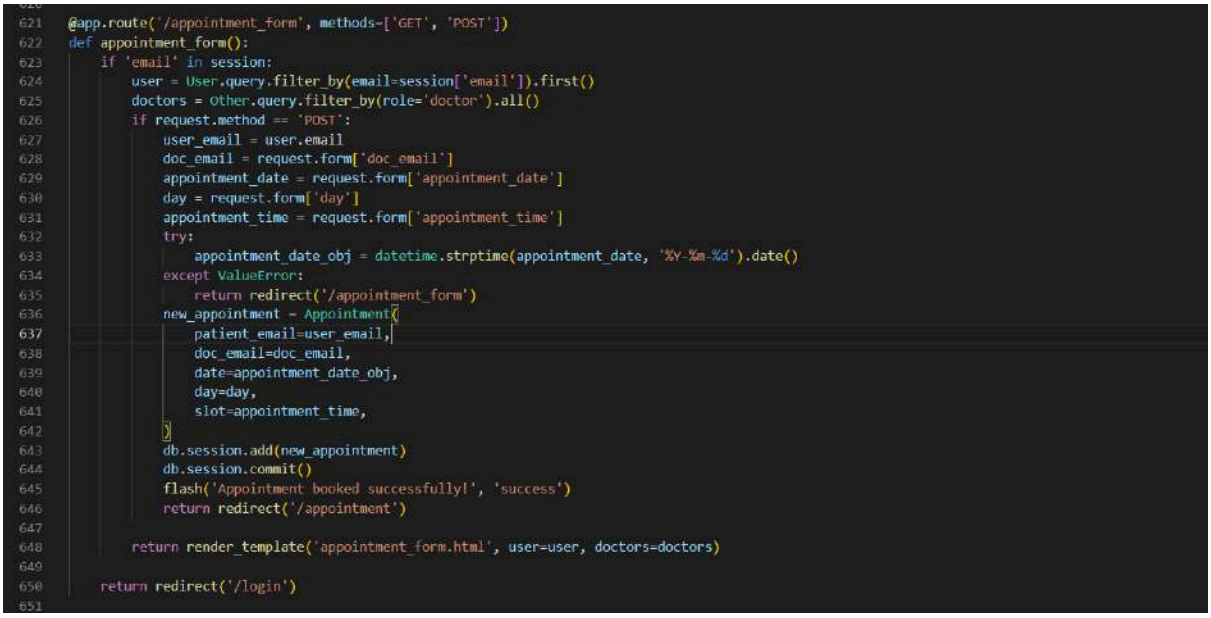


GET http://127.0.0.1:5000/getMedicine

```
    },
    {
        "date": "2024-12-19",
        "doctor_id": 1,
        "id": 2,
        "medicine_id": 2,
        "medicine_list": "1",
        "patient_id": 1
    },
    {
        "date": "2024-12-25",
        "doctor_id": 1,
        "id": 3,
        "medicine_id": 4,
        "medicine_list": "2",
        "patient_id": 2
    }
]
```

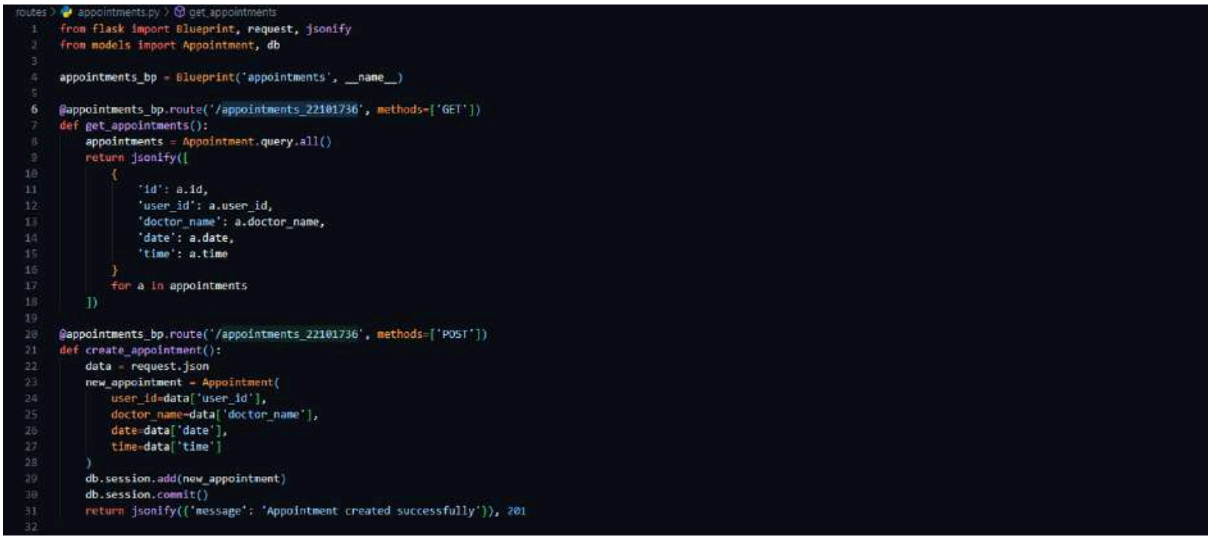
The above code captures data such as Medicine ID, Patient ID, Doctor ID, Medicine List, and Date and saves it in the database. After saving the data, the user is redirected back to the /medicine page. In the backend, the route/getMedicine provides a way to retrieve stored data from the database. It executes a raw SQL query to fetch all records from the Medicine table.

API:4

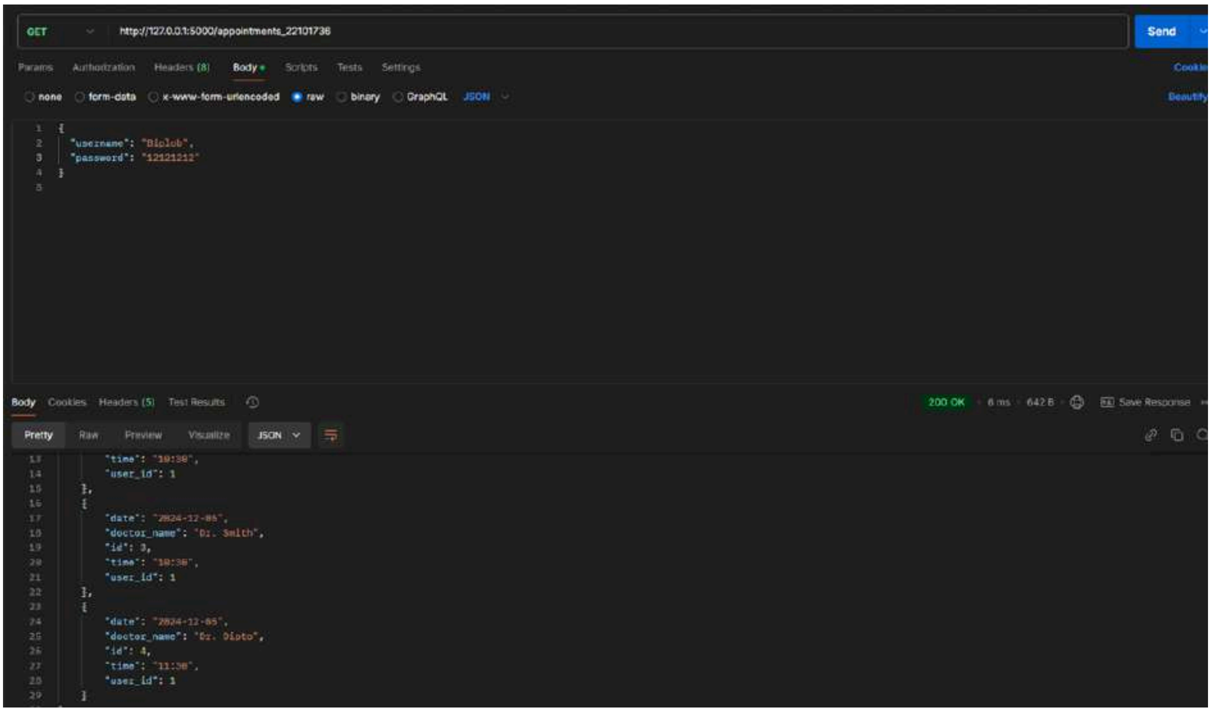


```python
@app.route('/appointment_form', methods=['GET', 'POST'])
def appointment_form():
    if 'email' in session:
        user = User.query.filter_by(email=session['email']).first()
        doctors = Other.query.filter_by(role='doctor').all()
        if request.method == 'POST':
            user_email = user.email
            doc_email = request.form['doc_email']
            appointment_date = request.form['appointment_date']
            day = request.form['day']
            appointment_time = request.form['appointment_time']
            try:
                appointment_date_obj = datetime.strptime(appointment_date, '%Y-%m-%d').date()
            except ValueError:
                return redirect('/appointment_form')
            new_appointment = Appointment(
                patient_email=user_email,
                doc_email=doc_email,
                date=appointment_date_obj,
                day=day,
                slot=appointment_time,
            )
            db.session.add(new_appointment)
            db.session.commit()
            flash('Appointment booked successfully!', 'success')
            return redirect('/appointment')

        return render_template('appointment_form.html', user=user, doctors=doctors)

    return redirect('/login')
```

The appointment_form function in a Flask web application handles the process of booking an appointment. If the user is logged in, it fetches the user's details and a list of doctors from the database. When the form is submitted via POST, the function retrieves the appointment details, such as the doctor's email, date, day, and time. It then attempts to convert the appointment date to a proper date object. If successful, a new Appointment object is created and saved to the database. A success message is displayed, and the user is redirected to the appointment page. If the user is not logged in, they are redirected to the login page.



```python
from flask import Blueprint, request, jsonify
from models import Appointment, db

appointments_bp = Blueprint('appointments', __name__)

@appointments_bp.route('/appointments_22101736', methods=['GET'])
def get_appointments():
    appointments = Appointment.query.all()
    return jsonify([
        {
            'id': a.id,
            'user_id': a.user_id,
            'doctor_name': a.doctor_name,
            'date': a.date,
            'time': a.time
        }
        for a in appointments
    ])

@appointments_bp.route('/appointments_22101736', methods=['POST'])
def create_appointment():
    data = request.json
    new_appointment = Appointment(
        user_id=data['user_id'],
        doctor_name=data['doctor_name'],
        date=data['date'],
        time=data['time']
    )
    db.session.add(new_appointment)
    db.session.commit()
    return jsonify({'message': 'Appointment created successfully'}), 201
```



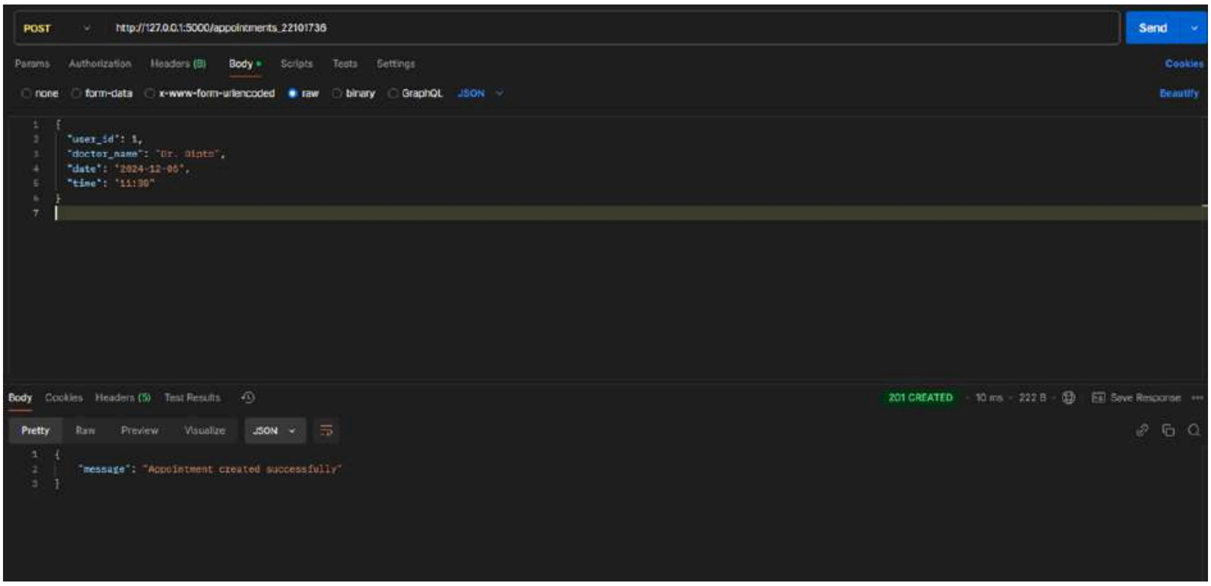
GET http://127.0.0.1:5000/appointments_22101736

Send
Params
Authorization
Headers (8)
Body
Scripts
Tests
Settings
Cookie
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON
Beautify

```
{
  "username": "Biplob",
  "password": "12121212"
}
```

200 OK · 6 ms · 642 B

```
        "time": "10:30",
        "user_id": 1
    },
    {
        "date": "2024-12-05",
        "doctor_name": "Dr. Smith",
        "id": 3,
        "time": "10:30",
        "user_id": 1
    },
    {
        "date": "2024-12-05",
        "doctor_name": "Dr. Dipto",
        "id": 4,
        "time": "11:30",
        "user_id": 1
    }
```



POST http://127.0.0.1:5000/appointments_22101736

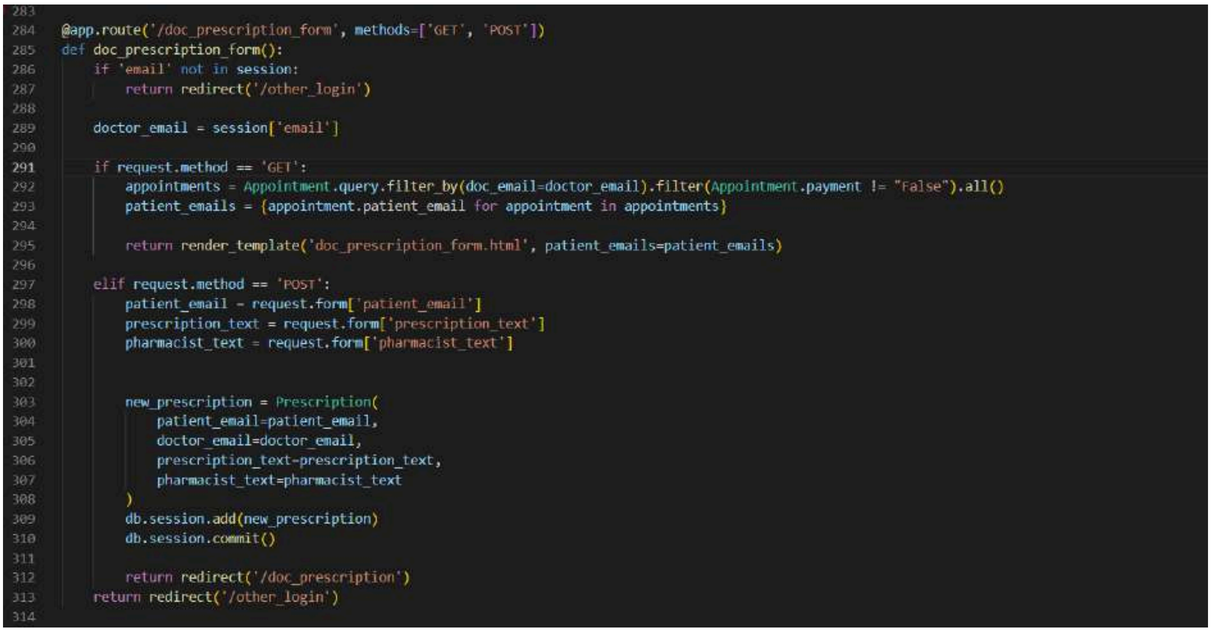
```
{
  "user_id": 1,
  "doctor_name": "Dr. Dipto",
  "date": "2024-12-05",
  "time": "11:30"
}
```

201 CREATED · 10 ms · 222 B

```
{
    "message": "Appointment created successfully"
}
```

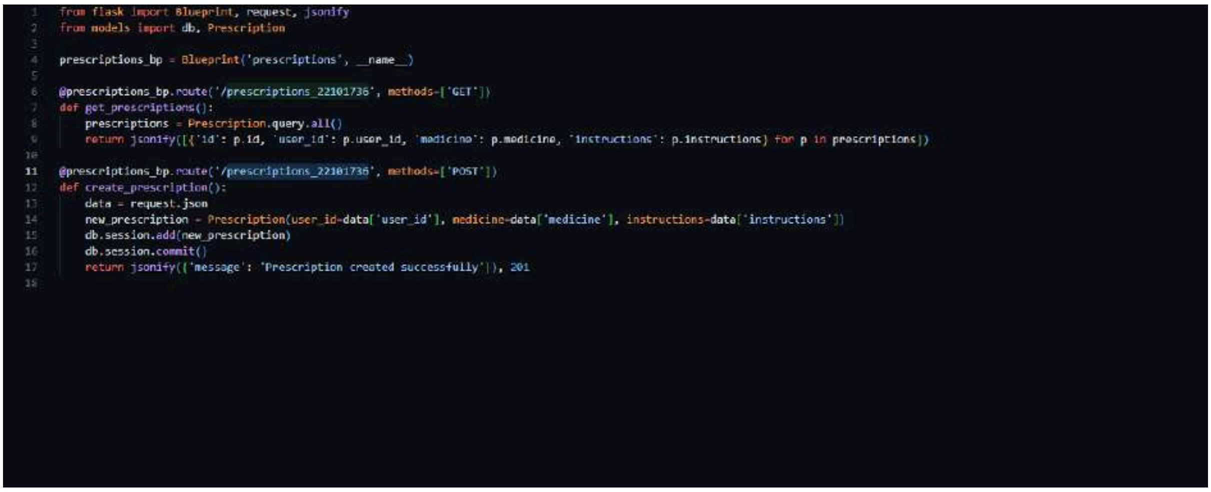
This API handles appointment scheduling. It collects details such as Appointment ID, Patient ID, Doctor ID, Date, Time, and Status through a form. When a user submits the form, the /appointment_submit route saves this data into the database. After submission, the user is redirected to the /appointments page. The /getAppointments route fetches all scheduled appointments, formats them as JSON, and returns the data. This setup allows users to schedule, view, and manage appointments easily.

API:5

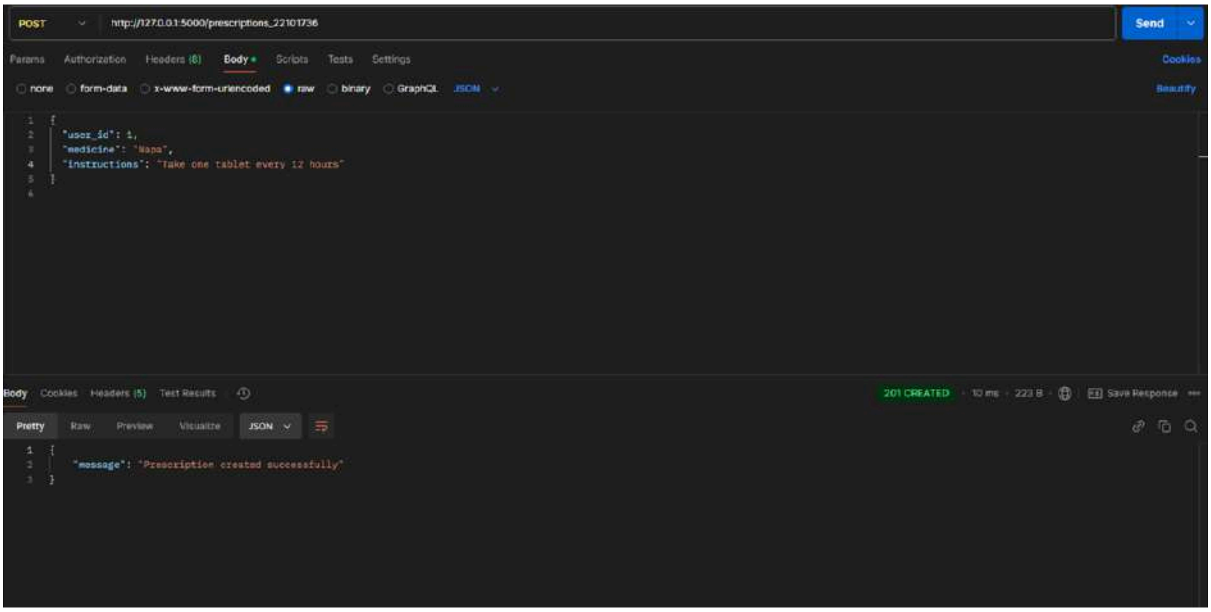


```
@app.route('/doc_prescription_form', methods=['GET', 'POST'])
def doc_prescription_form():
    if 'email' not in session:
        return redirect('/other_login')

    doctor_email = session['email']

    if request.method == 'GET':
        appointments = Appointment.query.filter_by(doc_email=doctor_email).filter(Appointment.payment != "False").all()
        patient_emails = {appointment.patient_email for appointment in appointments}

        return render_template('doc_prescription_form.html', patient_emails=patient_emails)

    elif request.method == 'POST':
        patient_email = request.form['patient_email']
        prescription_text = request.form['prescription_text']
        pharmacist_text = request.form['pharmacist_text']


        new_prescription = Prescription(
            patient_email=patient_email,
            doctor_email=doctor_email,
            prescription_text=prescription_text,
            pharmacist_text=pharmacist_text
        )
        db.session.add(new_prescription)
        db.session.commit()

        return redirect('/doc_prescription')
    return redirect('/other_login')
```

The `doc_prescription_form` function in a Flask web application allows doctors to create and manage prescriptions for patients. If the doctor is logged in, the function retrieves appointments for the doctor, excluding those that have not been paid. The form displays a list of patients associated with those appointments. When the form is submitted via POST, the function collects the patient's email, prescription text, and pharmacist instructions, then creates a new `Prescription` object, which is saved to the database. After successful submission, the doctor is redirected to the prescription page. If the doctor is not logged in, they are redirected to the login page.



```
from flask import Blueprint, request, jsonify
from models import db, Prescription

prescriptions_bp = Blueprint('prescriptions', __name__)

@prescriptions_bp.route('/prescriptions_22101736', methods=['GET'])
def get_prescriptions():
    prescriptions = Prescription.query.all()
    return jsonify([{'id': p.id, 'user_id': p.user_id, 'medicine': p.medicine, 'instructions': p.instructions} for p in prescriptions])

@prescriptions_bp.route('/prescriptions_22101736', methods=['POST'])
def create_prescription():
    data = request.json
    new_prescription = Prescription(user_id=data['user_id'], medicine=data['medicine'], instructions=data['instructions'])
    db.session.add(new_prescription)
    db.session.commit()
    return jsonify({'message': 'Prescription created successfully'}), 201
```

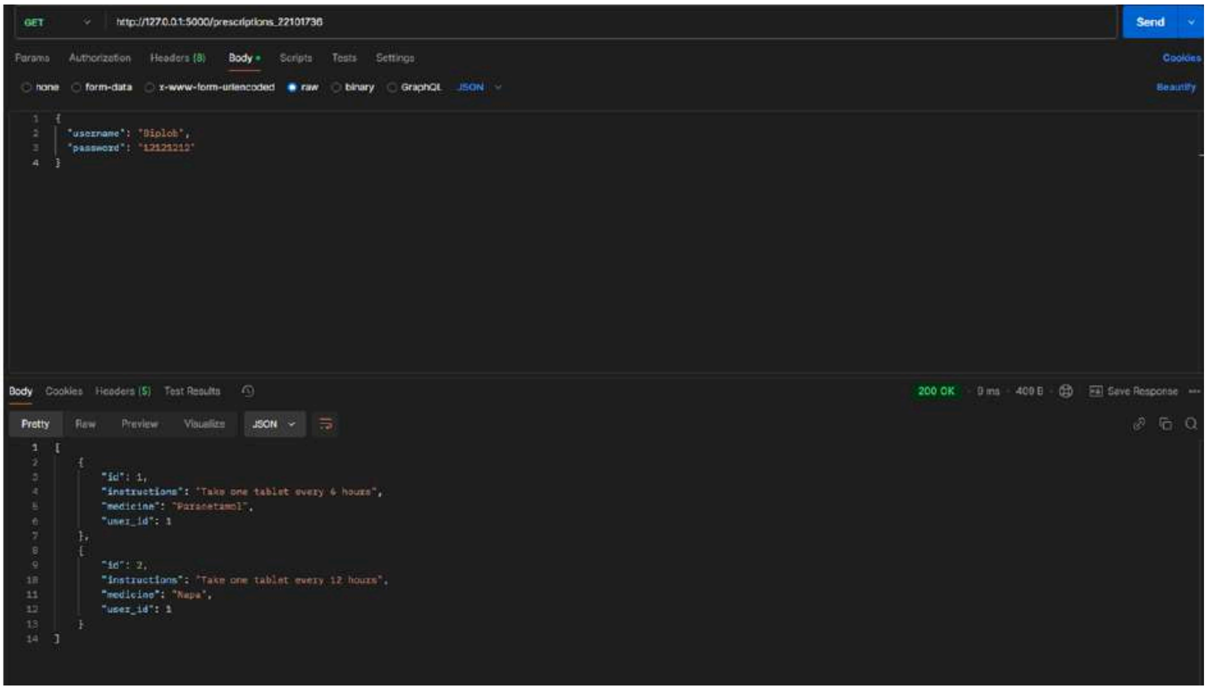


POST http://127.0.0.1:5000/prescriptions_22101736

```
{
  "user_id": 1,
  "medicine": "Napa",
  "instructions": "Take one tablet every 12 hours"
}
```

201 CREATED · 10 ms · 223 B

```
{
  "message": "Prescription created successfully"
}
```



GET http://127.0.0.1:5000/prescriptions_22101736

```
{
  "username": "Biplob",
  "password": "12121212"
}
```

200 OK · 9 ms · 409 B

```
[
    {
        "id": 1,
        "instructions": "Take one tablet every 6 hours",
        "medicine": "Paracetamol",
        "user_id": 1
    },
    {
        "id": 2,
        "instructions": "Take one tablet every 12 hours",
        "medicine": "Napa",
        "user_id": 1
    }
]
```

This API manages prescriptions issued by doctors. It gathers information like Prescription ID, Patient ID, Doctor ID, Medication Details, Dosage, and Notes. When the form is submitted, the /prescription_submit route stores the data securely in the database. Users are then redirected to the /prescriptions page. The /getPrescriptions route retrieves all prescriptions in JSON format, making it simple to display and manage prescription records across the application.

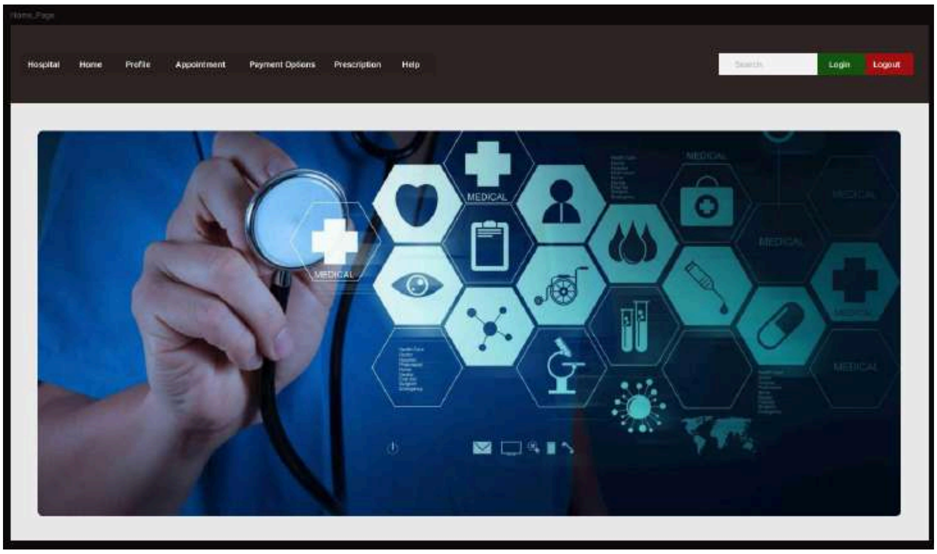
5. User Interface Design

Index Interface:

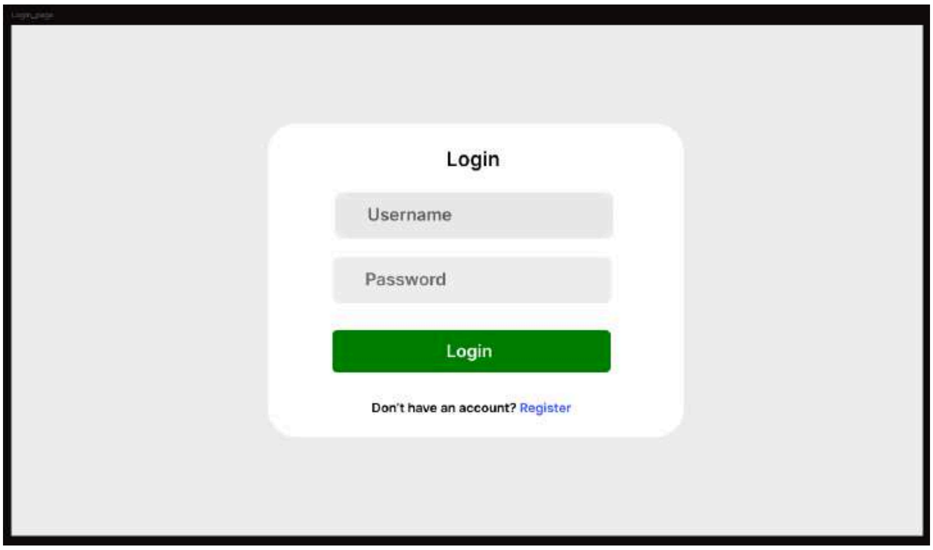
Figma Link:
https://www.figma.com/design/e8ljELRmNdz2aRRJioOxhC/Hospital_managementSystem(grp-01)(sec_02)MdRummanShahriar?node-id=0-1&m=dev&t=TrhAK1gWwjPlusXO-1

Prototype Link:
https://www.figma.com/proto/e8ljELRmNdz2aRRJioOxhC/Hospital_managementSystem(grp-01)(sec_02)MdRummanShahriar?node-id=0-1&t=TrhAK1gWwjPlusXO-1

Home Page:

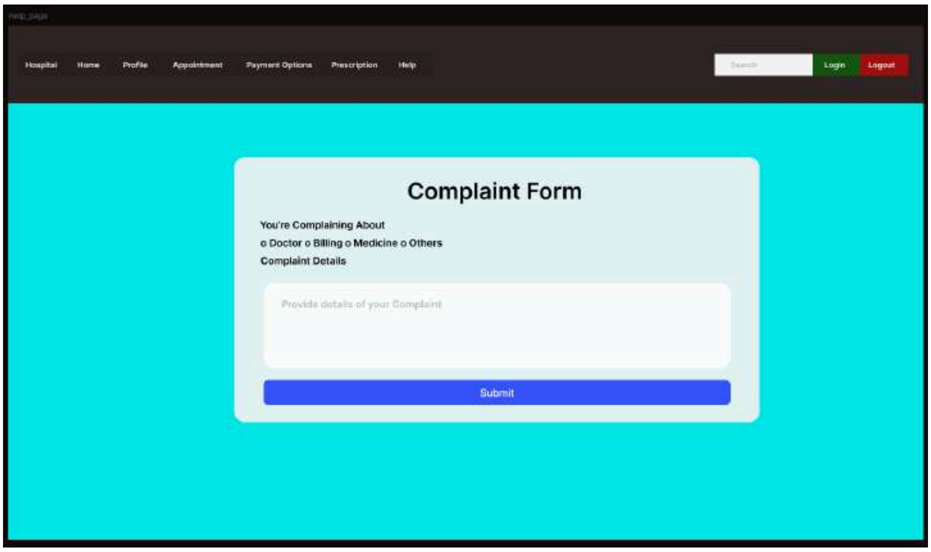


Login Page:



Don't have an account? Register

Help Page:



You're Complaining About

○ Doctor ○ Billing ○ Medicine ○ Others

Complaint Details

Provide details of your Complaint

Admin Interface:

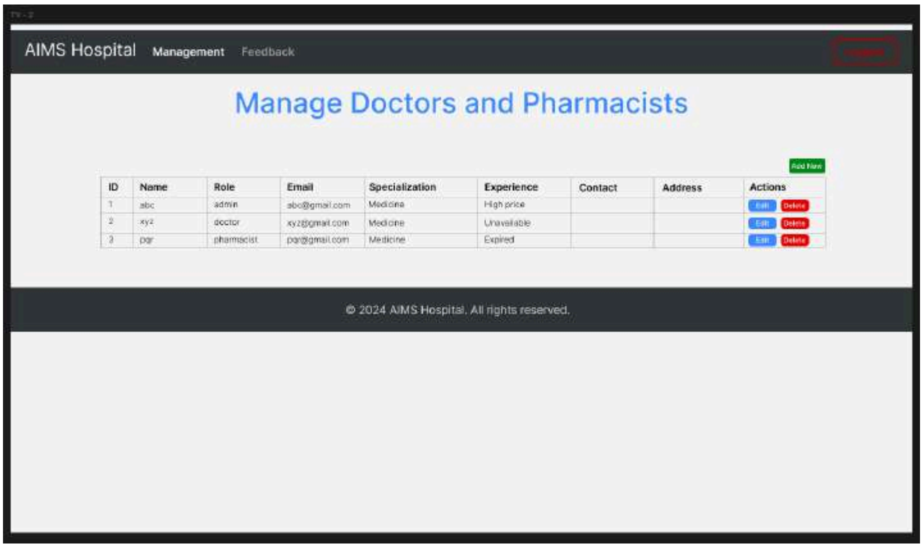
Figma Link:

https://www.figma.com/design/qor05RKw770yufyMFZT4xE/Admin-Page?node-id=0-1&m=dev&t=gNku6kv277QGAtgn-1

Prototype Link:

https://www.figma.com/proto/qor05RKw770yufyMFZT4xE/Admin-Page?node-id=0-1&t=gNku6kv277QGAtgn-1

Main Admin User Interface:

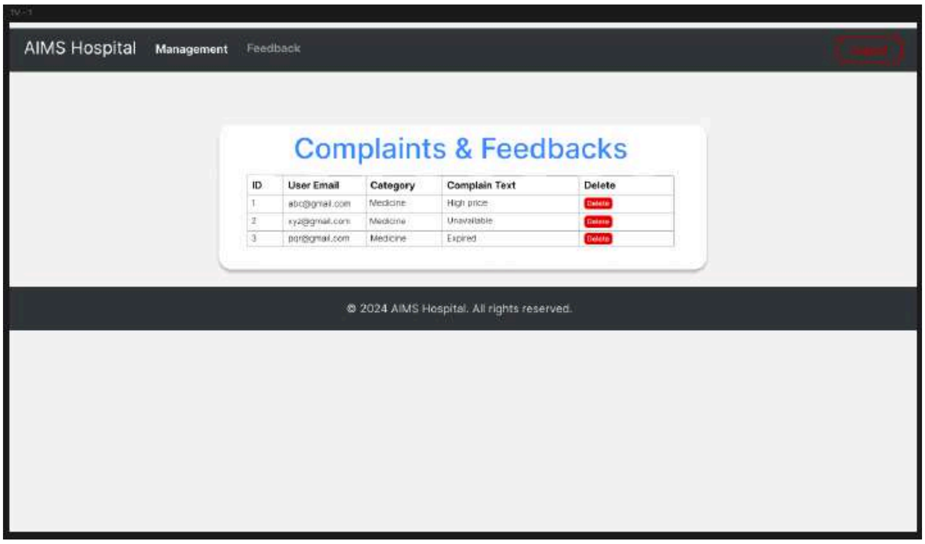


Manage Doctors and Pharmacists

ID	Name	Role	Email	Specialization	Experience	Contact	Address	Actions
1	abc	admin	abc@gmail.com	Medicine	High price			Edit Delete
2	xyz	doctor	xyz@gmail.com	Medicine	Unavailable			Edit Delete
3	pqr	pharmacist	pqr@gmail.com	Medicine	Expired			Edit Delete

© 2024 AIMS Hospital. All rights reserved.

Feedback interface:

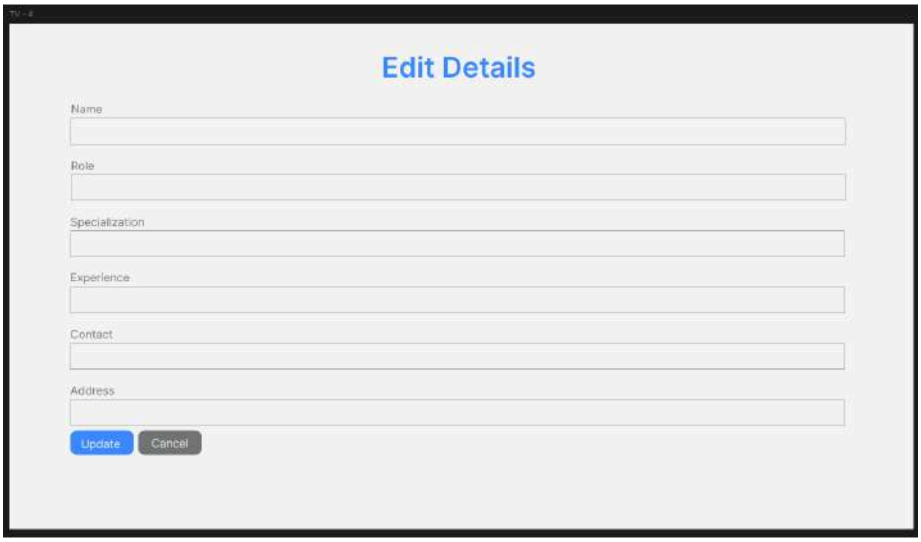


Complaints & Feedbacks

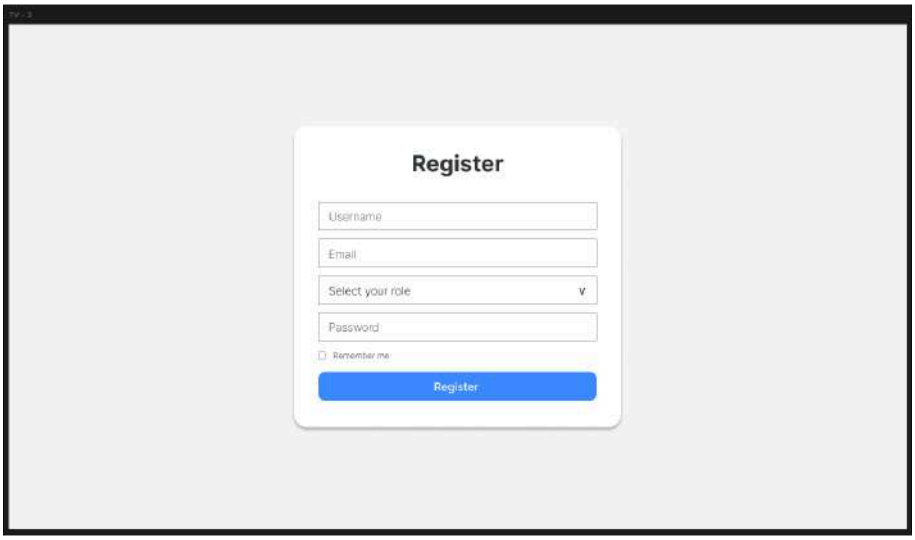
ID	User Email	Category	Complain Text	Delete
1	abc@gmail.com	Medicine	High price	Delete
2	xyz@gmail.com	Medicine	Unavailable	Delete
3	pqr@gmail.com	Medicine	Expired	Delete

© 2024 AIMS Hospital. All rights reserved.

Edit Doctor/Pharmacist Interface:



Add New Doctor/Pharmacist Interface:

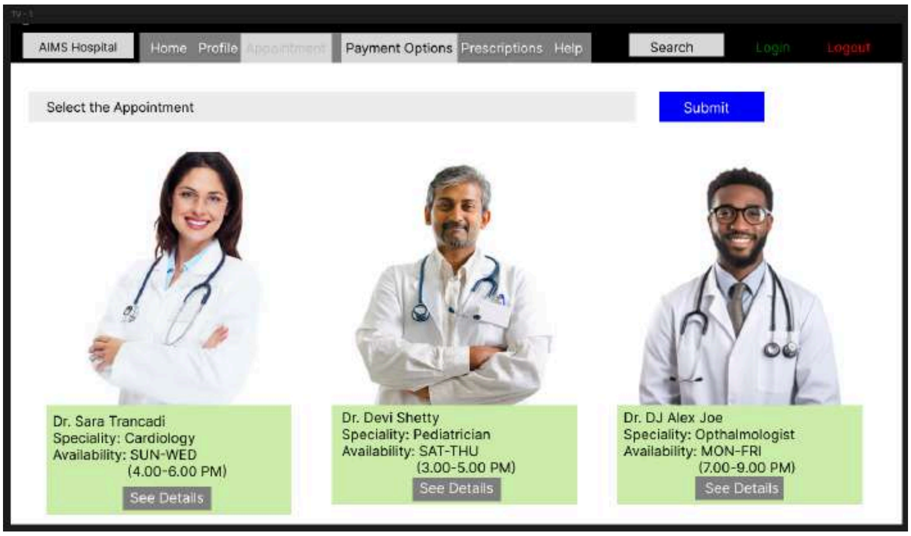


User Interface

Figma Link:
https://www.figma.com/design/6IBzYCEYThW7FRwtsZJWPG/Figma-Assignment-for-CSE471-Project?node-id=0-1&m=dev&t=WJ8EGUQxg0WbVc6r-1

Prototype Link:
https://www.figma.com/proto/6IBzYCEYThW7FRwtsZJWPG/Figma-Assignment-for-CSE471-Project?node-id=0-1&t=WJ8EGUQxg0WbVc6r-1

Appointment Page:

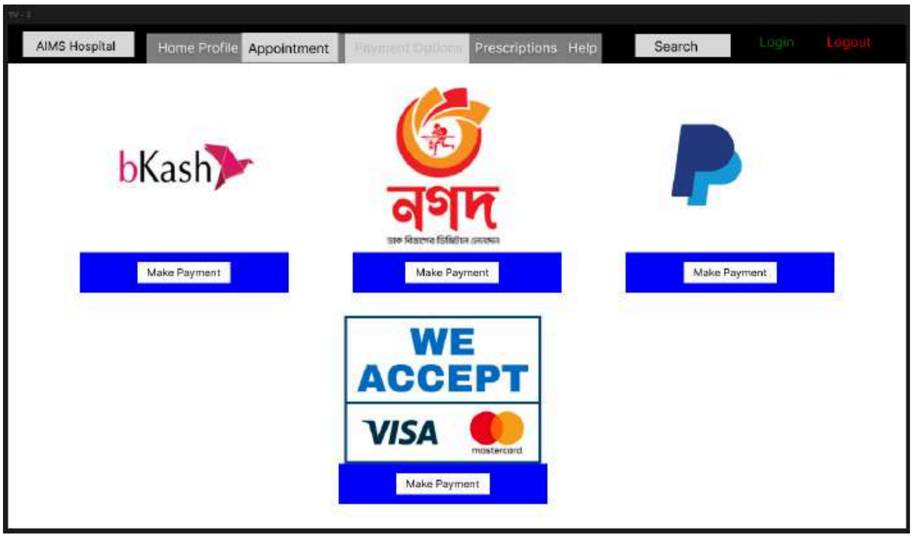


Dr. Sara Trancadi
Speciality: Cardiology
Availability: SUN-WED
(4.00-6.00 PM)

Dr. Devi Shetty
Speciality: Pediatrician
Availability: SAT-THU
(3.00-5.00 PM)

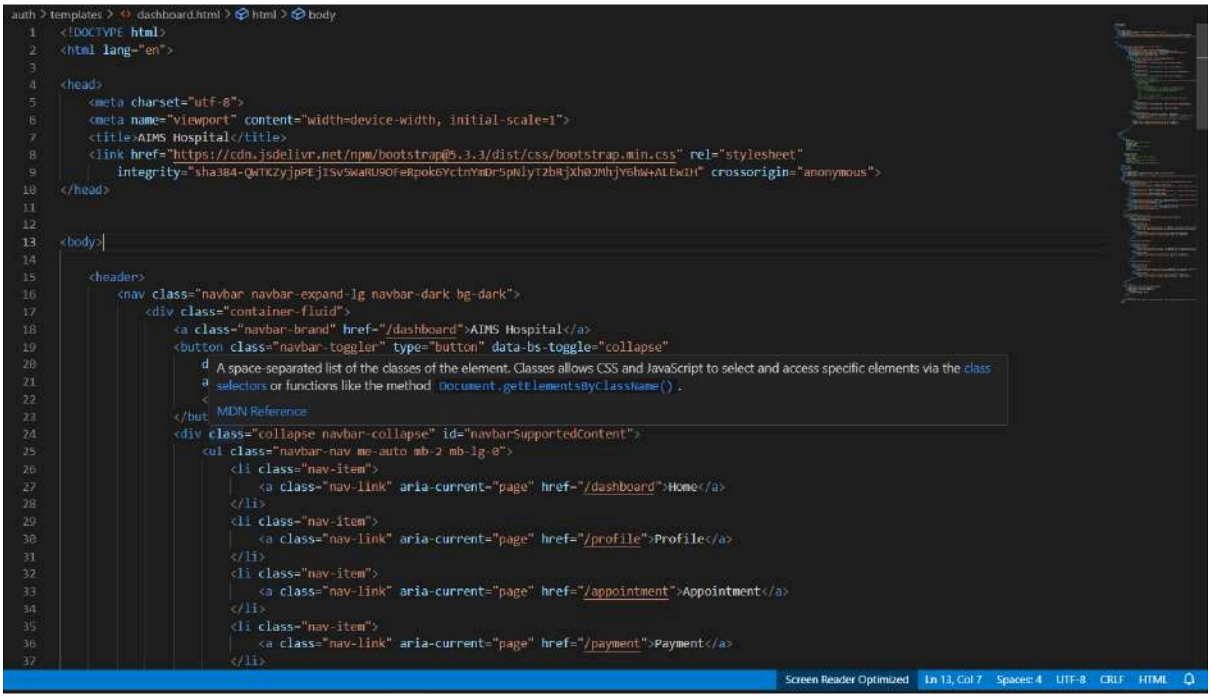
Dr. DJ Alex Joe
Speciality: Opthalmologist
Availability: MON-FRI
(7.00-9.00 PM)

Payment Page:



6. Frontend Development

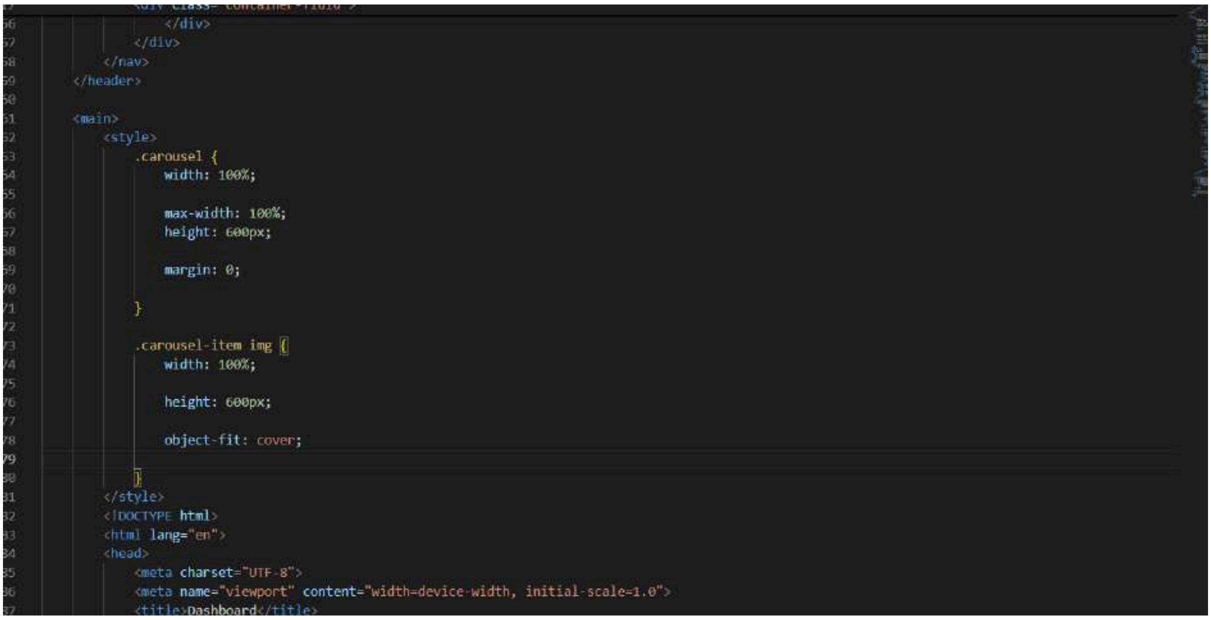
1.dashboard.html



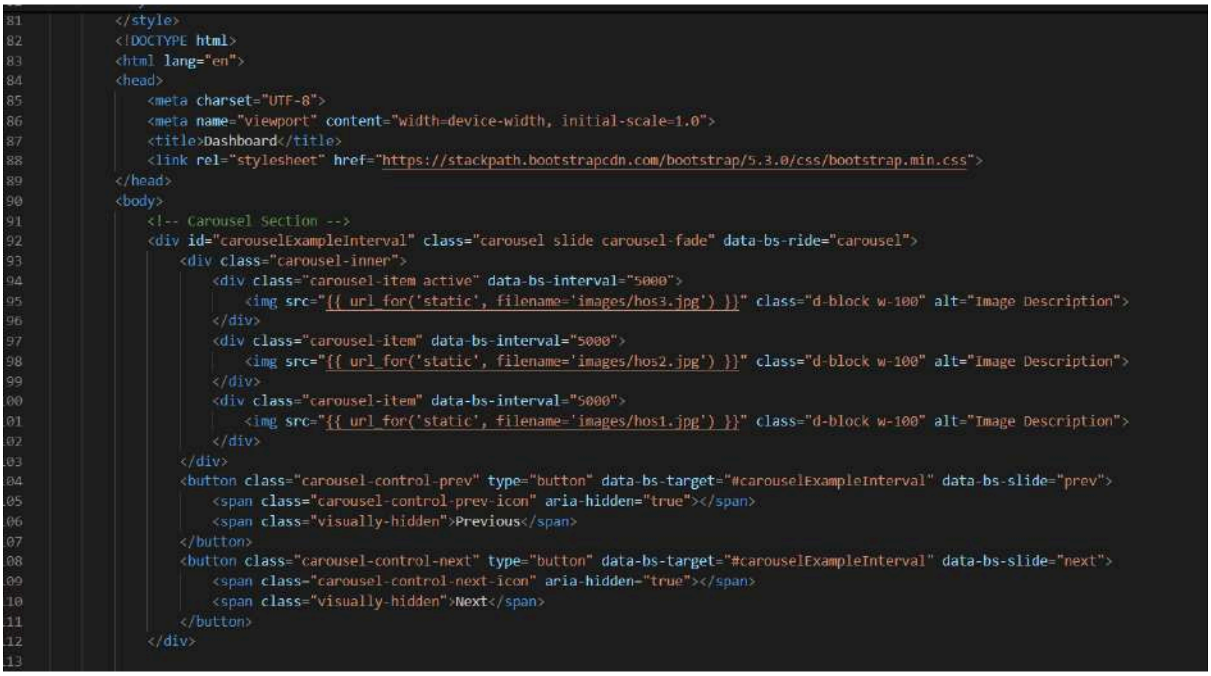
```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>AIMS Hospital</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
</head>


<body>

    <header>
        <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
            <div class="container-fluid">
                <a class="navbar-brand" href="/dashboard">AIMS Hospital</a>
                <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
                    d
                    a
                    <
                </but
                <div class="collapse navbar-collapse" id="navbarSupportedContent">
                    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                        <li class="nav-item">
                            <a class="nav-link" aria-current="page" href="/dashboard">Home</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" aria-current="page" href="/profile">Profile</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" aria-current="page" href="/appointment">Appointment</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" aria-current="page" href="/payment">Payment</a>
                        </li>
```

A space-separated list of the classes of the element. Classes allows CSS and JavaScript to select and access specific elements via the class selectors or functions like the method Document.getElementsByClassName() .

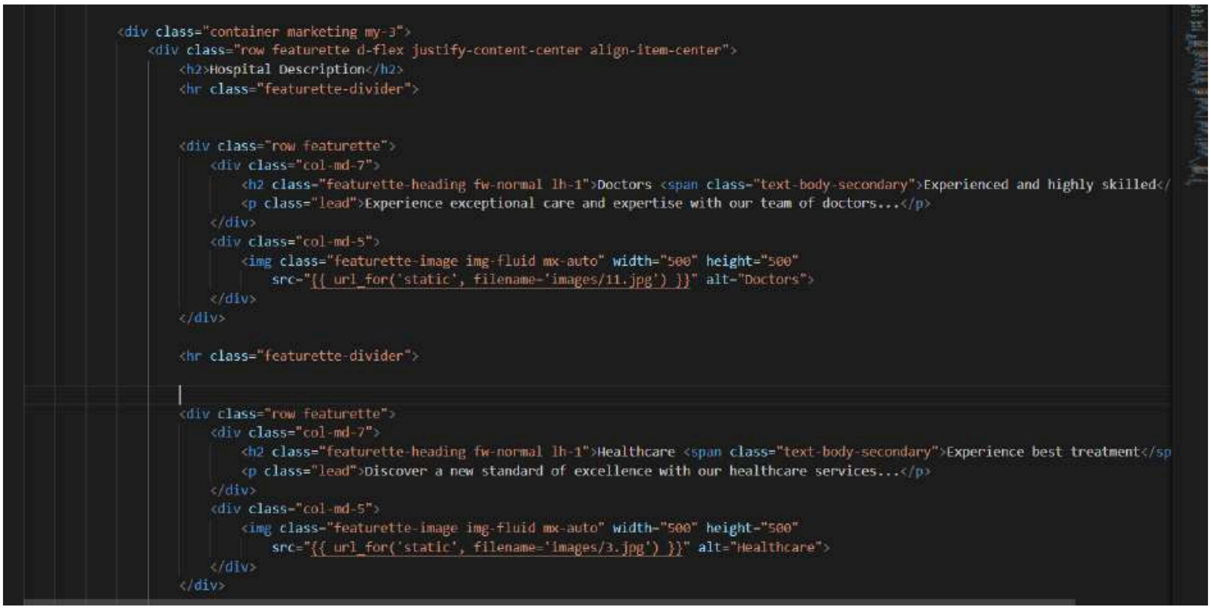
MDN Reference



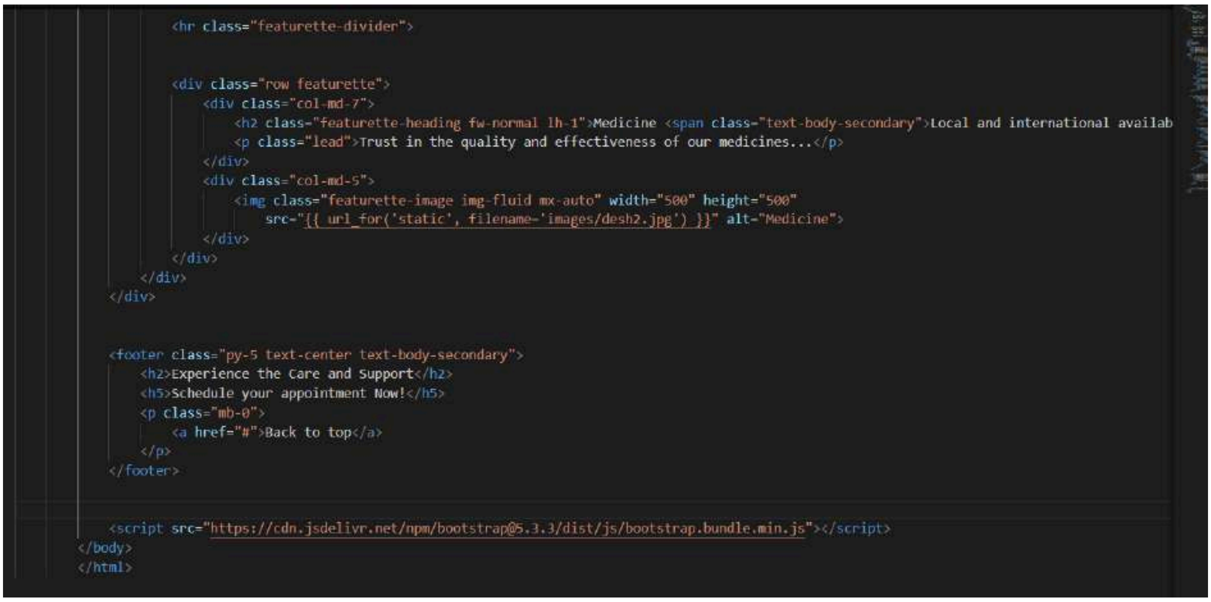
```html
            <div class="container-fluid">
                    </div>
                </div>
            </nav>
        </header>

        <main>
            <style>
                .carousel {
                    width: 100%;

                    max-width: 100%;
                    height: 600px;

                    margin: 0;

                }

                .carousel-item img {
                    width: 100%;

                    height: 600px;

                    object-fit: cover;

                }
            </style>
            <!DOCTYPE html>
            <html lang="en">
            <head>
                <meta charset="UTF-8">
                <meta name="viewport" content="width=device-width, initial-scale=1.0">
                <title>Dashboard</title>
```



```html
</style>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dashboard</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/5.3.0/css/bootstrap.min.css">
</head>
<body>
    <!-- Carousel Section -->
    <div id="carouselExampleInterval" class="carousel slide carousel-fade" data-bs-ride="carousel">
        <div class="carousel-inner">
            <div class="carousel-item active" data-bs-interval="5000">
                <img src="{{ url_for('static', filename='images/hos3.jpg') }}" class="d-block w-100" alt="Image Description">
            </div>
            <div class="carousel-item" data-bs-interval="5000">
                <img src="{{ url_for('static', filename='images/hos2.jpg') }}" class="d-block w-100" alt="Image Description">
            </div>
            <div class="carousel-item" data-bs-interval="5000">
                <img src="{{ url_for('static', filename='images/hos1.jpg') }}" class="d-block w-100" alt="Image Description">
            </div>
        </div>
        <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleInterval" data-bs-slide="prev">
            <span class="carousel-control-prev-icon" aria-hidden="true"></span>
            <span class="visually-hidden">Previous</span>
        </button>
        <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleInterval" data-bs-slide="next">
            <span class="carousel-control-next-icon" aria-hidden="true"></span>
            <span class="visually-hidden">Next</span>
        </button>
    </div>

```



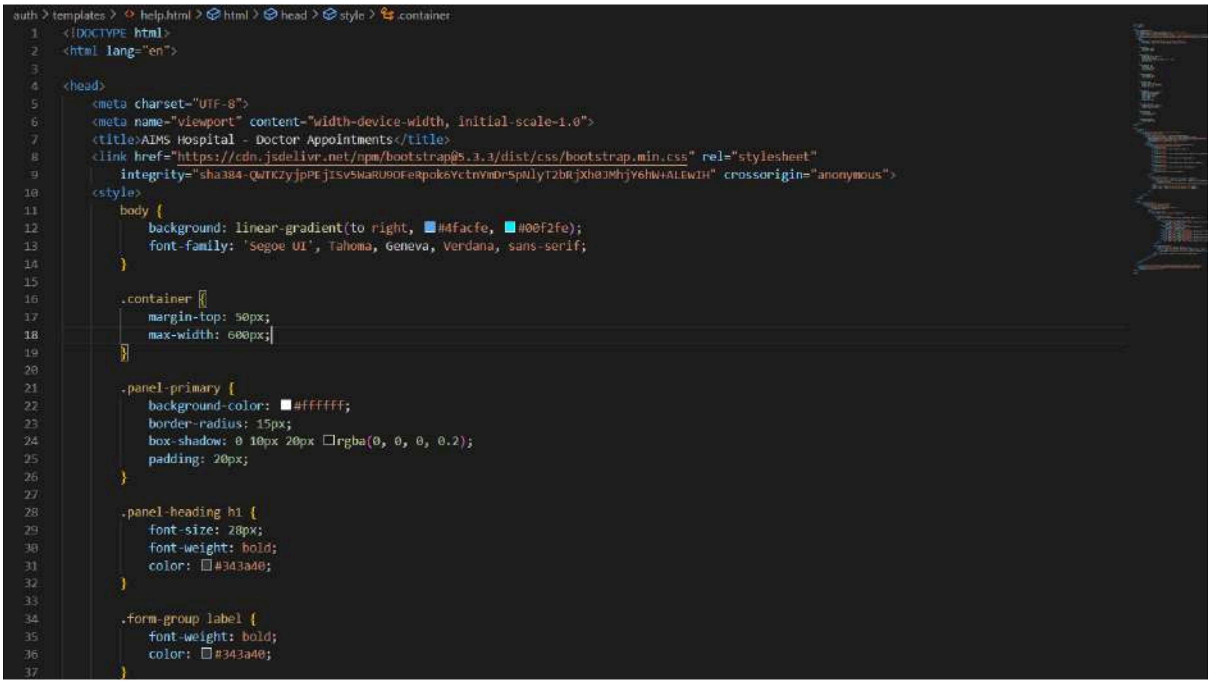
```html
<div class="container marketing my-3">
    <div class="row featurette d-flex justify-content-center align-item-center">
        <h2>Hospital Description</h2>
        <hr class="featurette-divider">


        <div class="row featurette">
            <div class="col-md-7">
                <h2 class="featurette-heading fw-normal lh-1">Doctors <span class="text-body-secondary">Experienced and highly skilled</
                <p class="lead">Experience exceptional care and expertise with our team of doctors...</p>
            </div>
            <div class="col-md-5">
                <img class="featurette-image img-fluid mx-auto" width="500" height="500"
                    src="{{ url_for('static', filename='images/11.jpg') }}" alt="Doctors">
            </div>
        </div>

        <hr class="featurette-divider">

        
        <div class="row featurette">
            <div class="col-md-7">
                <h2 class="featurette-heading fw-normal lh-1">Healthcare <span class="text-body-secondary">Experience best treatment</sp
                <p class="lead">Discover a new standard of excellence with our healthcare services...</p>
            </div>
            <div class="col-md-5">
                <img class="featurette-image img-fluid mx-auto" width="500" height="500"
                    src="{{ url_for('static', filename='images/3.jpg') }}" alt="Healthcare">
            </div>
        </div>
```



```
            <hr class="featurette-divider">


            <div class="row featurette">
                <div class="col-md-7">
                    <h2 class="featurette-heading fw-normal lh-1">Medicine <span class="text-body-secondary">Local and international availab
                    <p class="lead">Trust in the quality and effectiveness of our medicines...</p>
                </div>
                <div class="col-md-5">
                    <img class="featurette-image img-fluid mx-auto" width="500" height="500"
                        src="{{ url_for('static', filename='images/desh2.jpg') }}" alt="Medicine">
                </div>
            </div>
        </div>
    </div>


    <footer class="py-5 text-center text-body-secondary">
        <h2>Experience the Care and Support</h2>
        <h5>Schedule your appointment Now!</h5>
        <p class="mb-0">
            <a href="#">Back to top</a>
        </p>
    </footer>


    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

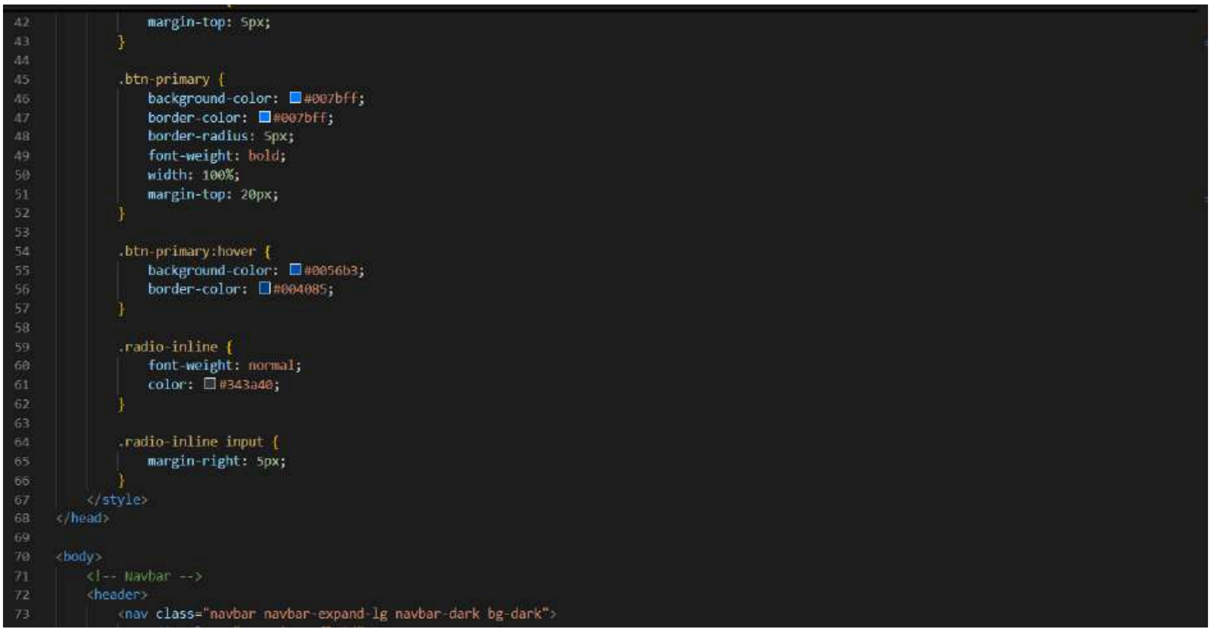
This `dashboard.html` file serves as the main frontend page for the **AIMS Hospital Management System**, offering a sleek, responsive, and user-friendly interface designed with **Bootstrap 5**. The page begins with a well-structured navigation bar that allows quick access to key sections such as **appointments**, **prescriptions**, **payments**, and **help**, ensuring smooth navigation throughout the platform. A dynamic **image carousel** follows, effectively showcasing various hospital visuals to engage visitors. The page also includes several informative marketing sections, emphasizing the hospital's core services such as **experienced doctors**, **advanced healthcare technology**, and **high-quality medicines**, all organized using Bootstrap's flexible grid system for consistency across different screen sizes. Additionally, the footer encourages users to **book appointments** and provides essential contact information. With a professional, modern design and a focus on accessibility, this page ensures an intuitive and seamless experience for users interacting with the hospital management system.

2.help.html

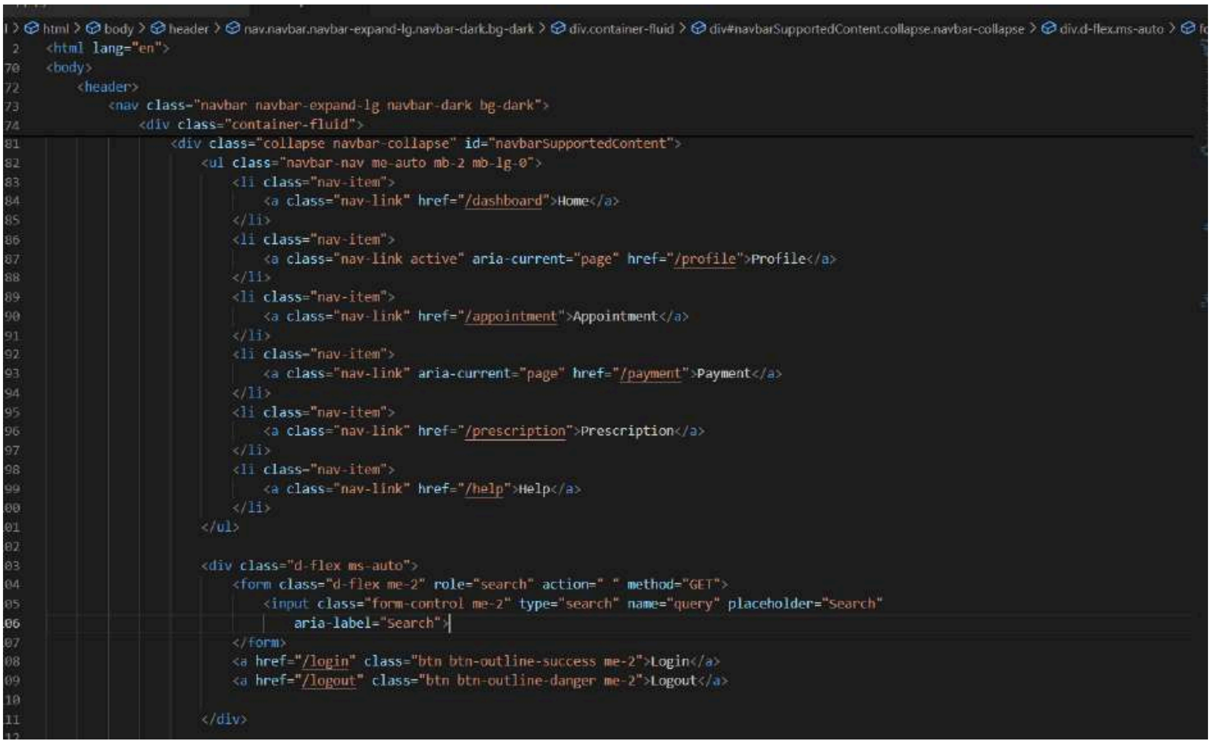


auth > templates > help.html > html > head > style > .container

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>AIMS Hospital - Doctor Appointments</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
    <style>
        body {
            background: linear-gradient(to right, #4facfe, #00f2fe);
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
        }

        .container {
            margin-top: 50px;
            max-width: 600px;
        }

        .panel-primary {
            background-color: #ffffff;
            border-radius: 15px;
            box-shadow: 0 10px 20px rgba(0, 0, 0, 0.2);
            padding: 20px;
        }

        .panel-heading h1 {
            font-size: 28px;
            font-weight: bold;
            color: #343a40;
        }

        .form-group label {
            font-weight: bold;
            color: #343a40;
        }
```

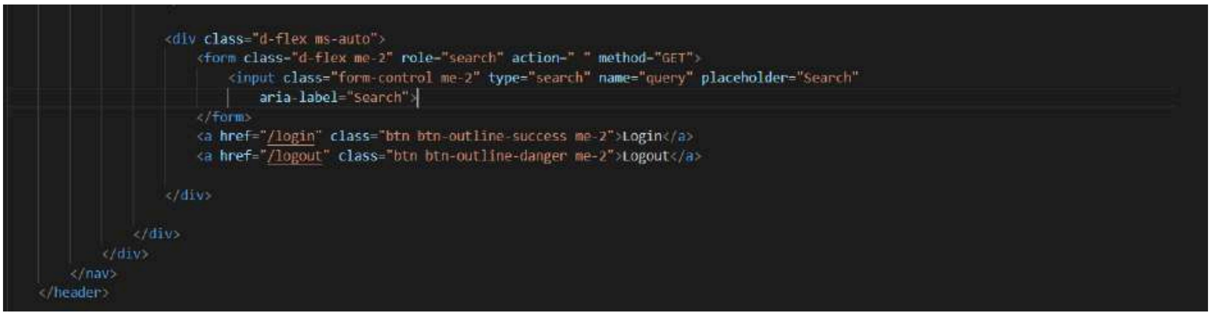


```html
            margin-top: 5px;
        }

        .btn-primary {
            background-color: #007bff;
            border-color: #007bff;
            border-radius: 5px;
            font-weight: bold;
            width: 100%;
            margin-top: 20px;
        }

        .btn-primary:hover {
            background-color: #0056b3;
            border-color: #004085;
        }

        .radio-inline {
            font-weight: normal;
            color: #343a40;
        }

        .radio-inline input {
            margin-right: 5px;
        }
    </style>
</head>

<body>
    <!-- Navbar -->
    <header>
        <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
```

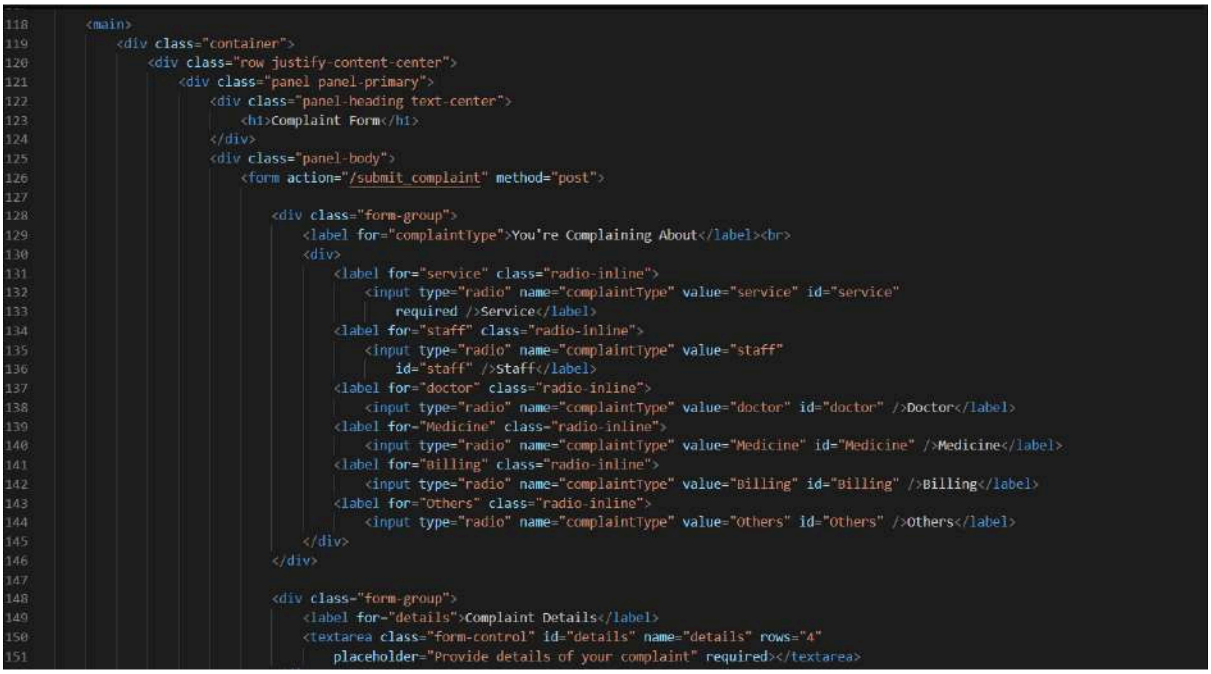


html > body > header > nav.navbar.navbar-expand-lg.navbar-dark.bg-dark > div.container-fluid > div#navbarSupportedContent.collapse.navbar-collapse > div.d-flex.ms-auto > f

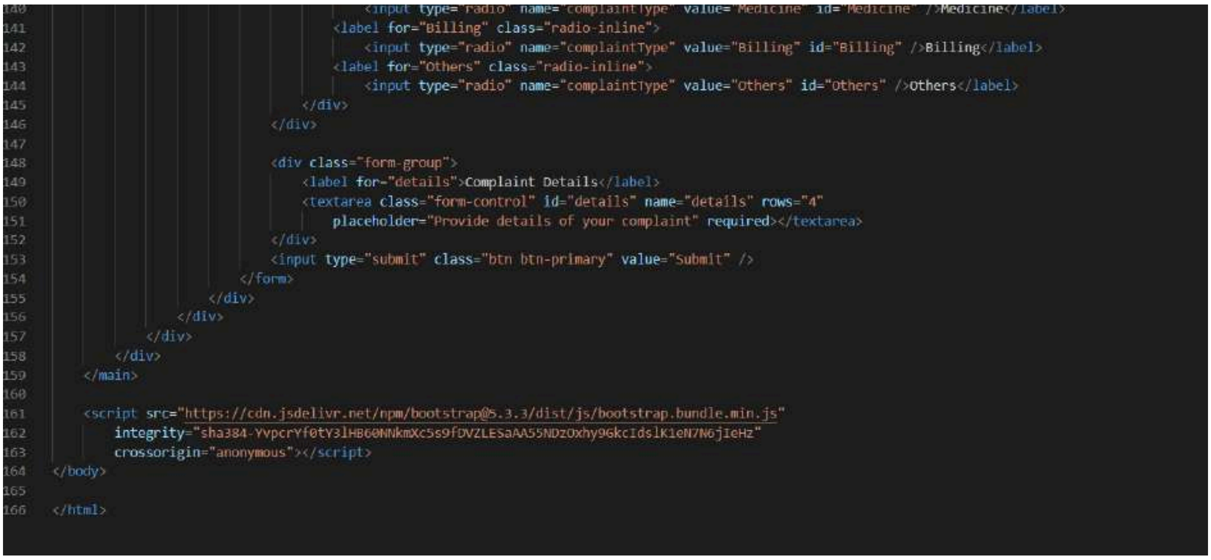
```html
<html lang="en">
<body>
    <header>
        <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
            <div class="container-fluid">
                <div class="collapse navbar-collapse" id="navbarSupportedContent">
                    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                        <li class="nav-item">
                            <a class="nav-link" href="/dashboard">Home</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link active" aria-current="page" href="/profile">Profile</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="/appointment">Appointment</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" aria-current="page" href="/payment">Payment</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="/prescription">Prescription</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="/help">Help</a>
                        </li>
                    </ul>

                    <div class="d-flex ms-auto">
                        <form class="d-flex me-2" role="search" action=" " method="GET">
                            <input class="form-control me-2" type="search" name="query" placeholder="Search"
                                aria-label="Search">
                        </form>
                        <a href="/login" class="btn btn-outline-success me-2">Login</a>
                        <a href="/logout" class="btn btn-outline-danger me-2">Logout</a>

                    </div>
```



```html
                    <div class="d-flex ms-auto">
                        <form class="d-flex me-2" role="search" action=" " method="GET">
                            <input class="form-control me-2" type="search" name="query" placeholder="Search"
                                aria-label="Search">
                        </form>
                        <a href="/login" class="btn btn-outline-success me-2">Login</a>
                        <a href="/logout" class="btn btn-outline-danger me-2">Logout</a>

                    </div>

                </div>
            </div>
        </nav>
    </header>
```



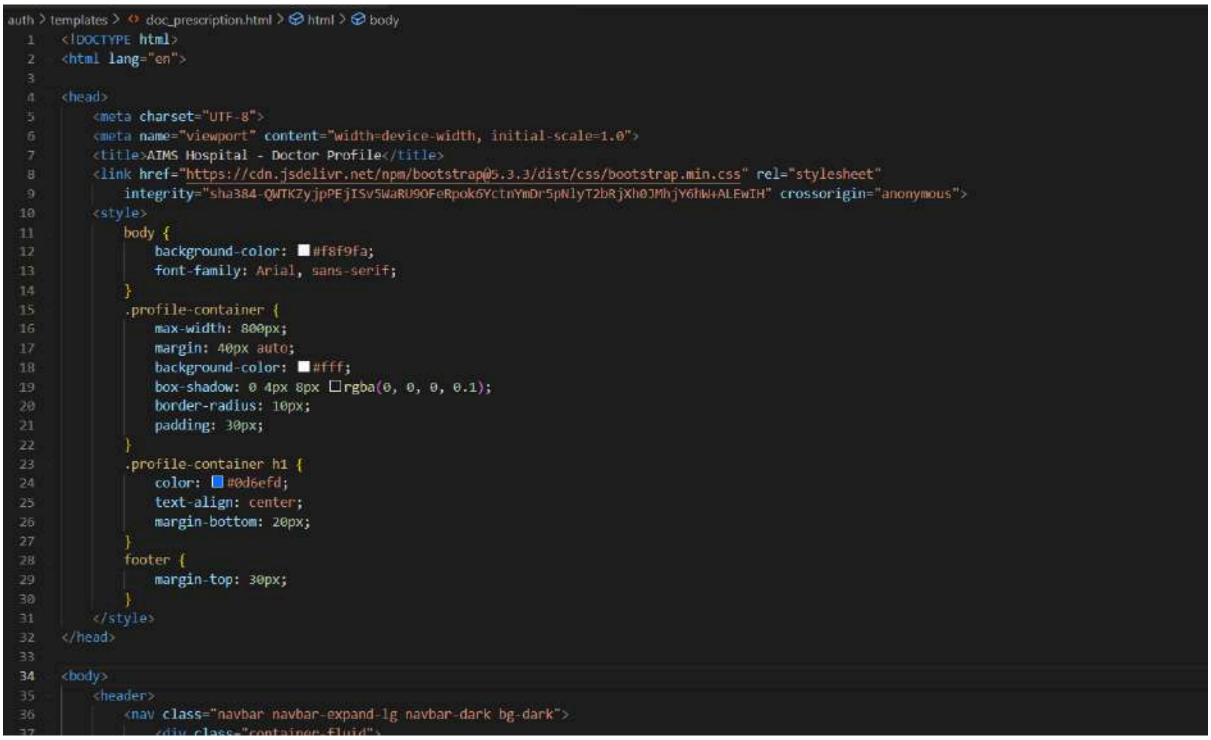
```html
    <main>
        <div class="container">
            <div class="row justify-content-center">
                <div class="panel panel-primary">
                    <div class="panel-heading text-center">
                        <h1>Complaint Form</h1>
                    </div>
                    <div class="panel-body">
                        <form action="/submit_complaint" method="post">

                            <div class="form-group">
                                <label for="complaintType">You're Complaining About</label><br>
                                <div>
                                    <label for="service" class="radio-inline">
                                        <input type="radio" name="complaintType" value="service" id="service"
                                            required />Service</label>
                                    <label for="staff" class="radio-inline">
                                        <input type="radio" name="complaintType" value="staff"
                                            id="staff" />Staff</label>
                                    <label for="doctor" class="radio-inline">
                                        <input type="radio" name="complaintType" value="doctor" id="doctor" />Doctor</label>
                                    <label for="Medicine" class="radio-inline">
                                        <input type="radio" name="complaintType" value="Medicine" id="Medicine" />Medicine</label>
                                    <label for="Billing" class="radio-inline">
                                        <input type="radio" name="complaintType" value="Billing" id="Billing" />Billing</label>
                                    <label for="Others" class="radio-inline">
                                        <input type="radio" name="complaintType" value="Others" id="Others" />Others</label>
                                </div>
                            </div>

                            <div class="form-group">
                                <label for="details">Complaint Details</label>
                                <textarea class="form-control" id="details" name="details" rows="4"
                                    placeholder="Provide details of your complaint" required></textarea>
```



```html
                                        <input type="radio" name="complaintType" value="Medicine" id="Medicine" />Medicine</label>
                                    <label for="Billing" class="radio-inline">
                                        <input type="radio" name="complaintType" value="Billing" id="Billing" />Billing</label>
                                    <label for="Others" class="radio-inline">
                                        <input type="radio" name="complaintType" value="Others" id="Others" />Others</label>
                                </div>
                            </div>

                            <div class="form-group">
                                <label for="details">Complaint Details</label>
                                <textarea class="form-control" id="details" name="details" rows="4"
                                    placeholder="Provide details of your complaint" required></textarea>
                            </div>
                            <input type="submit" class="btn btn-primary" value="Submit" />
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </main>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
        crossorigin="anonymous"></script>
</body>

</html>
```

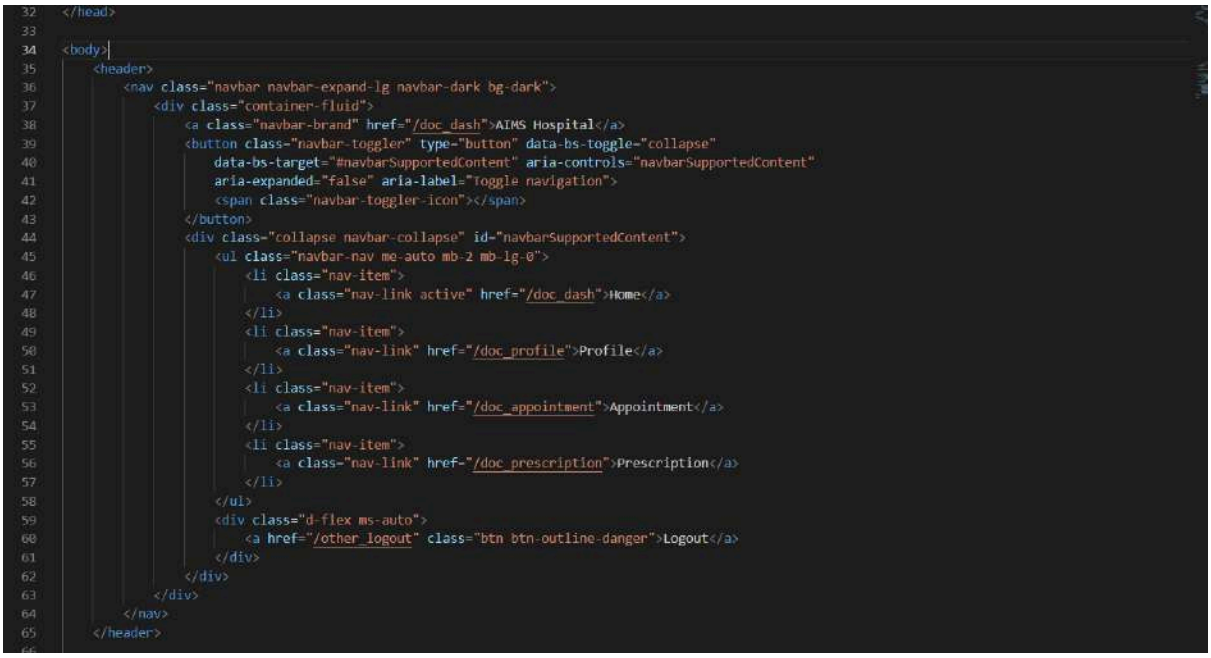
The `doc_prescription.html` page is an integral part of the **AIMS Hospital Management System**, designed to streamline prescription management for doctors. This page features a clean, user-friendly interface with a **responsive layout** built using **Bootstrap 5**, ensuring optimal viewing on various devices. At the top, a well-organized **navigation bar** provides quick and easy access to key sections such as the **homepage**, **doctor profiles**, **appointments**, and **prescriptions**, allowing doctors to efficiently navigate between tasks. The main content area highlights a **button** that enables doctors to **add new prescriptions**, simplifying the process of updating patient care. The page also prioritizes security and convenience with a **logout option**, ensuring that doctors can securely exit the system. With its modern, intuitive design, the page enhances the workflow of healthcare professionals, offering an efficient way to manage prescriptions while maintaining a professional, secure environment.

3.doc_prescription.html

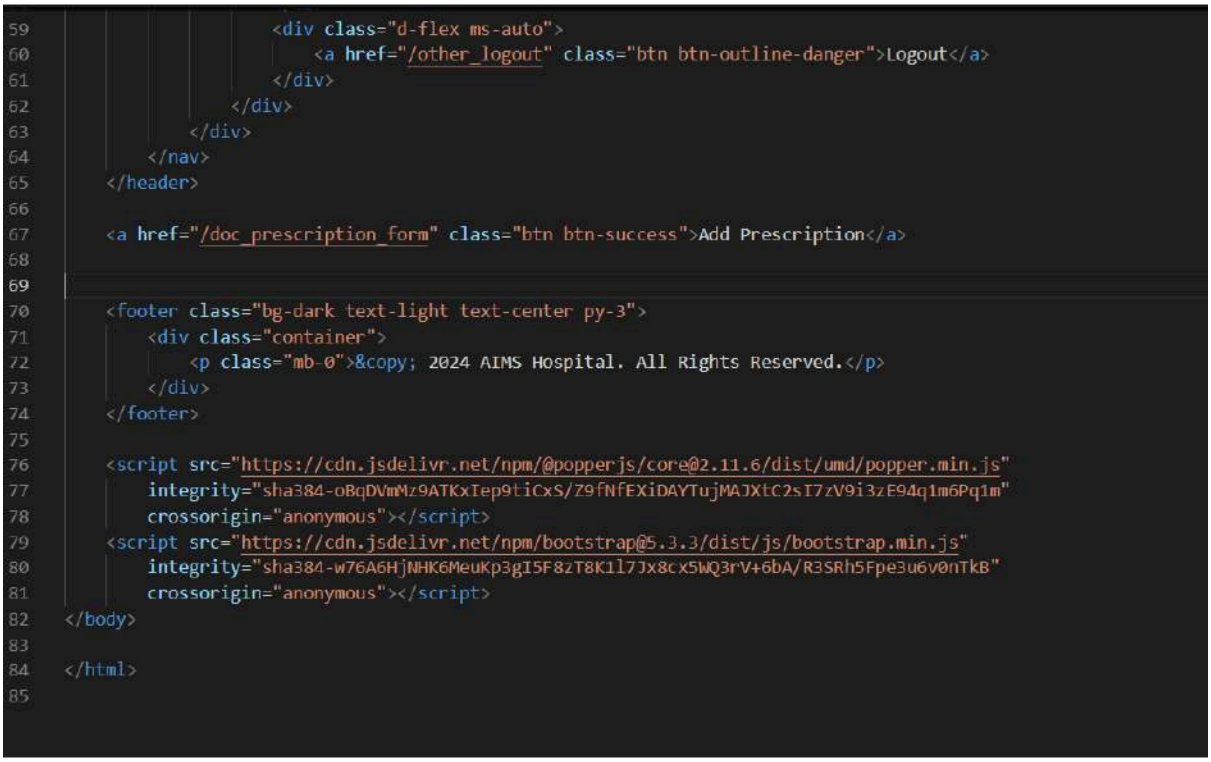


auth > templates > doc_prescription.html > html > body

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>AIMS Hospital - Doctor Profile</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
    <style>
        body {
            background-color: #f8f9fa;
            font-family: Arial, sans-serif;
        }
        .profile-container {
            max-width: 800px;
            margin: 40px auto;
            background-color: #fff;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
            border-radius: 10px;
            padding: 30px;
        }
        .profile-container h1 {
            color: #0d6efd;
            text-align: center;
            margin-bottom: 20px;
        }
        footer {
            margin-top: 30px;
        }
    </style>
</head>

<body>
    <header>
        <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
            <div class="container-fluid">
```



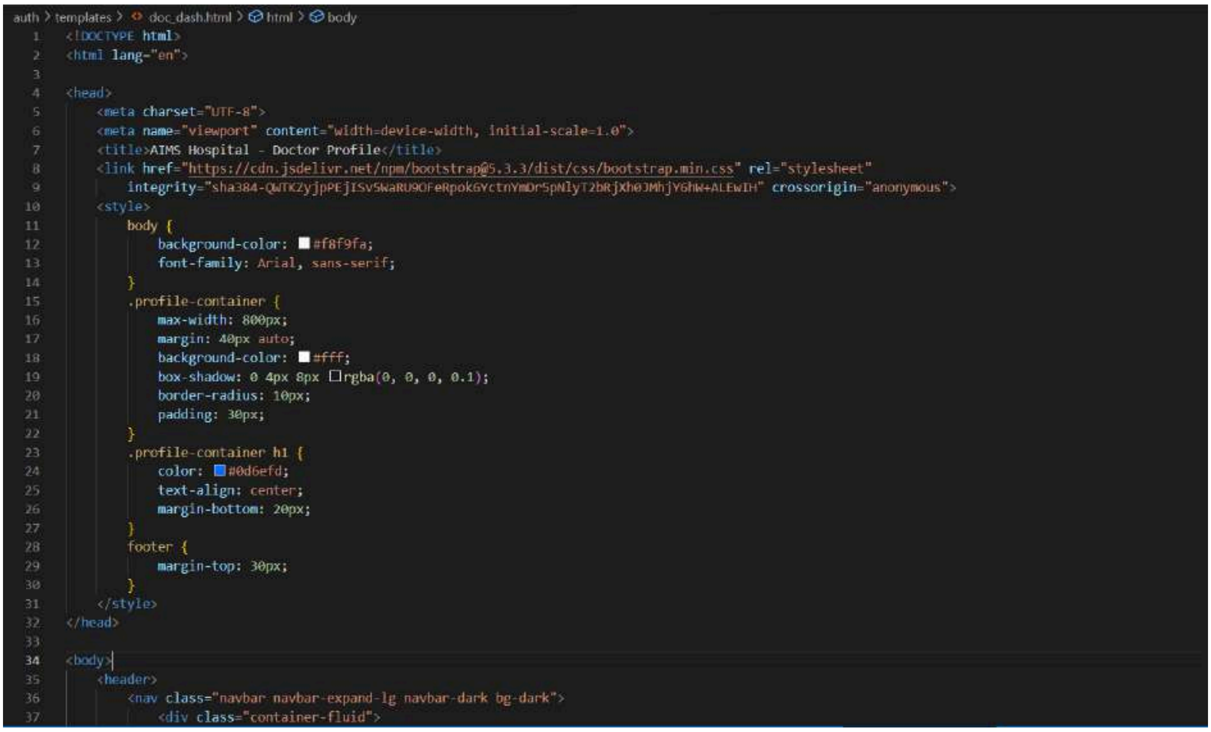
```html
</head>

<body>
    <header>
        <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
            <div class="container-fluid">
                <a class="navbar-brand" href="/doc_dash">AIMS Hospital</a>
                <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
                    data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
                    aria-expanded="false" aria-label="Toggle navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="collapse navbar-collapse" id="navbarSupportedContent">
                    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                        <li class="nav-item">
                            <a class="nav-link active" href="/doc_dash">Home</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="/doc_profile">Profile</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="/doc_appointment">Appointment</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="/doc_prescription">Prescription</a>
                        </li>
                    </ul>
                    <div class="d-flex ms-auto">
                        <a href="/other_logout" class="btn btn-outline-danger">Logout</a>
                    </div>
                </div>
            </div>
        </nav>
    </header>
```



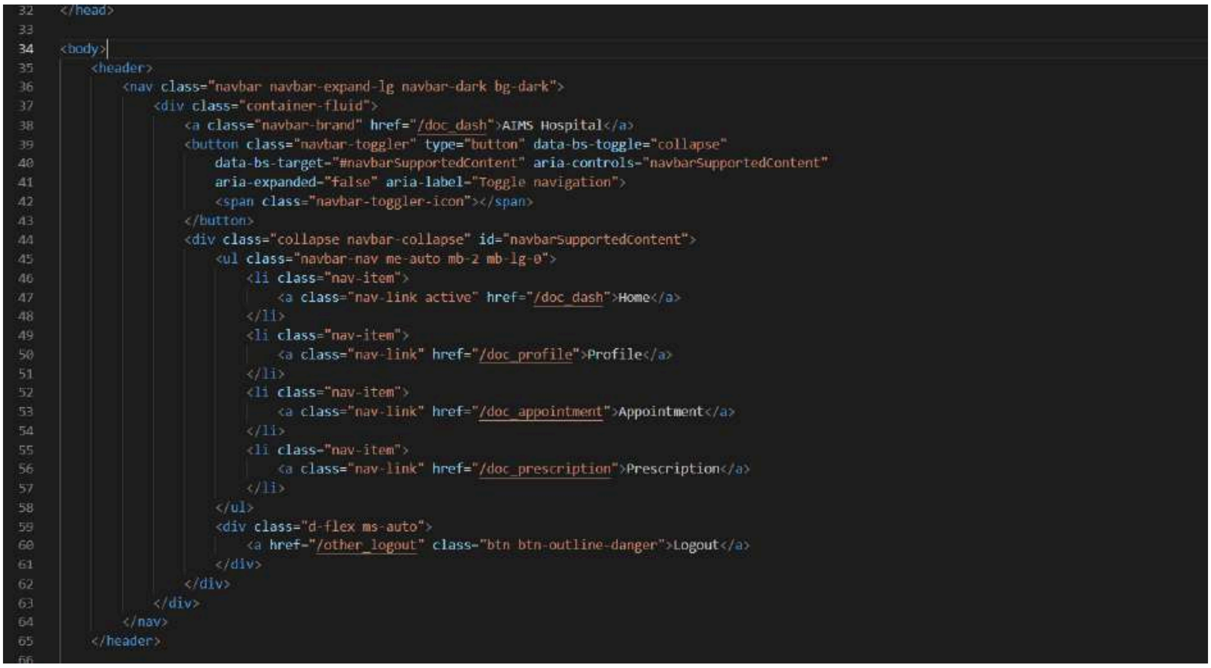
```
                    <div class="d-flex ms-auto">
                        <a href="/other_logout" class="btn btn-outline-danger">Logout</a>
                    </div>
                </div>
            </div>
        </nav>
    </header>

    <a href="/doc_prescription_form" class="btn btn-success">Add Prescription</a>


    <footer class="bg-dark text-light text-center py-3">
        <div class="container">
            <p class="mb-0">&copy; 2024 AIMS Hospital. All Rights Reserved.</p>
        </div>
    </footer>

    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"
        integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAJXtC2sI7zV9i3zE94q1m6Pq1m"
        crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.min.js"
        integrity="sha384-w76A6HjNHK6MeuKp3gI5F8zT8K1l7Jx8cx5WQ3rV+6bA/R3SRh5Fpe3u6v0nTkB"
        crossorigin="anonymous"></script>
</body>

</html>

```

The **doc_profile.html** page is a crucial component of the **AIMS Hospital Management System**, designed to manage and display doctor-related information in an organized, user-friendly manner. The page utilizes a **responsive Bootstrap 5 layout**, ensuring that the design adapts seamlessly across various screen sizes and devices. At the top, a well-structured **navigation bar** allows doctors to quickly access key sections such as **appointments**, **prescriptions**, and their **profile**, making it easy to navigate between tasks. The main content area is dedicated to displaying and editing the doctor's personal and professional details, ensuring that information is easily accessible and manageable. Additionally, a prominent **button** is provided for doctors to **add new prescriptions**, streamlining patient care management. The page also includes a secure **logout functionality**, ensuring that doctors can safely exit the system after use. With its clean design and intuitive navigation, the page enhances the user experience by providing doctors with an efficient, secure platform for managing their professional information and tasks.

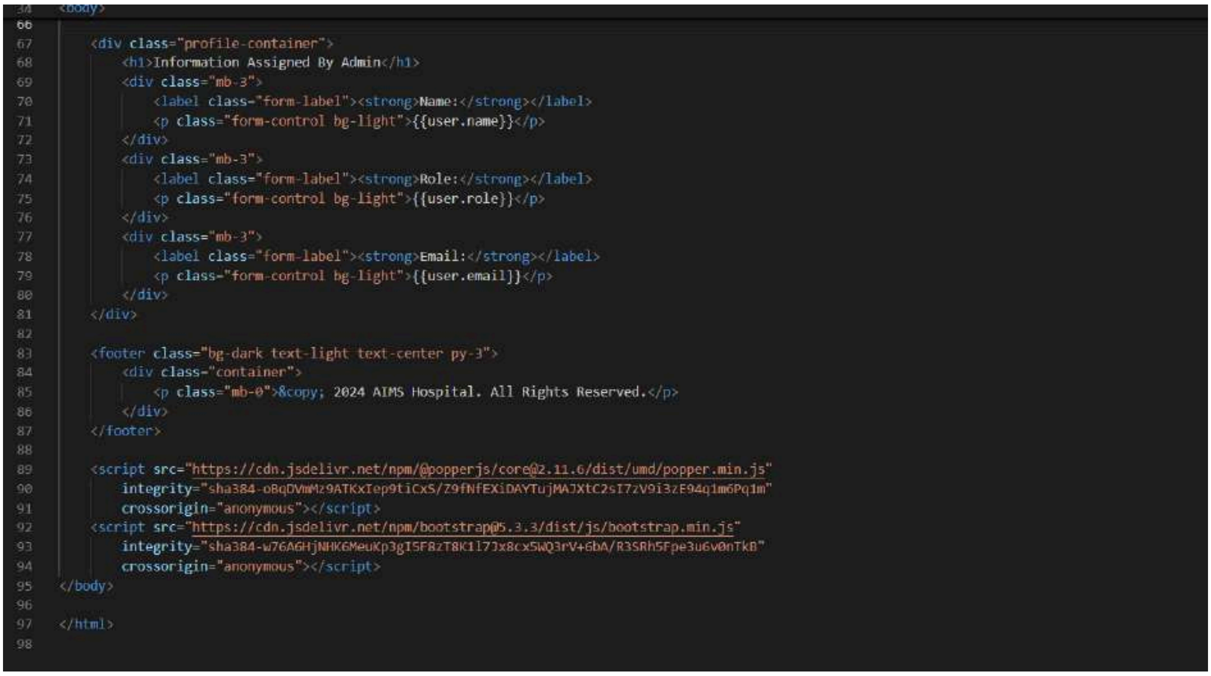
4.doc_dash.html



```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>AIMS Hospital - Doctor Profile</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpokGYctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
    <style>
        body {
            background-color: ■#f8f9fa;
            font-family: Arial, sans-serif;
        }
        .profile-container {
            max-width: 800px;
            margin: 40px auto;
            background-color: ■#fff;
            box-shadow: 0 4px 8px □rgba(0, 0, 0, 0.1);
            border-radius: 10px;
            padding: 30px;
        }
        .profile-container h1 {
            color: ■#0d6efd;
            text-align: center;
            margin-bottom: 20px;
        }
        footer {
            margin-top: 30px;
        }
    </style>
</head>

<body>
    <header>
        <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
            <div class="container-fluid">
```



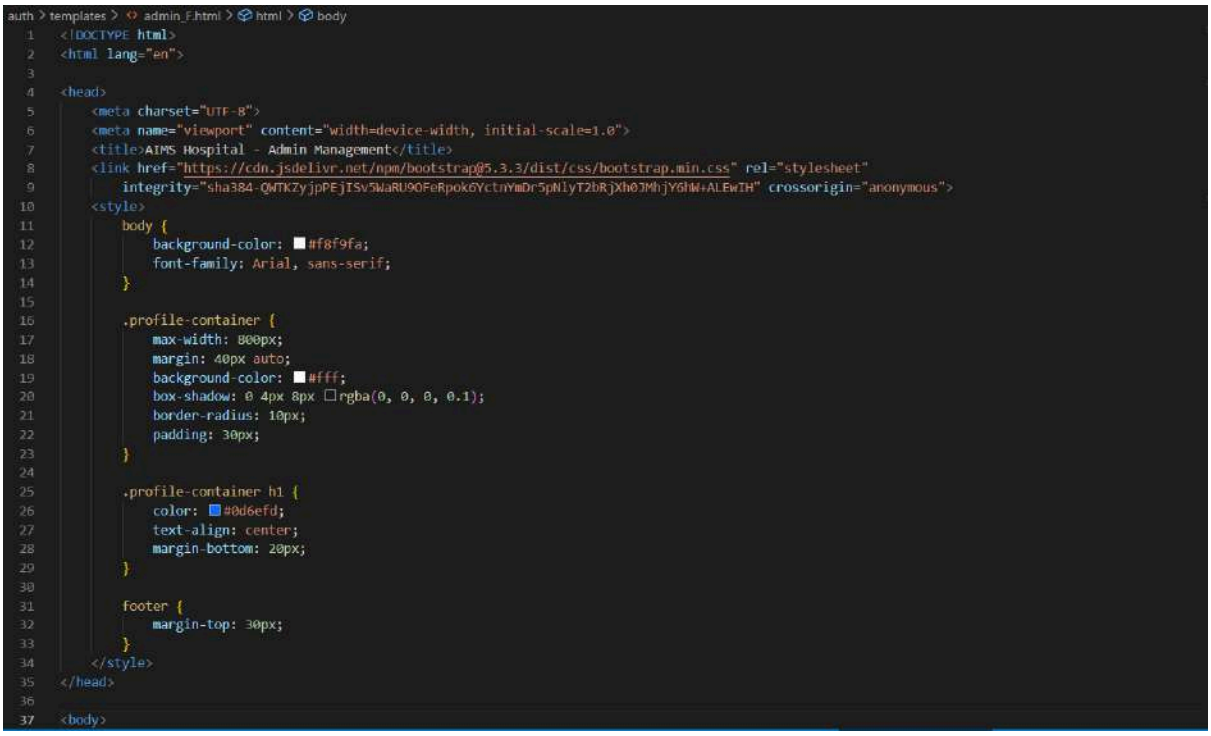
```html
</head>

<body>
    <header>
        <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
            <div class="container-fluid">
                <a class="navbar-brand" href="/doc_dash">AIMS Hospital</a>
                <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
                    data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
                    aria-expanded="false" aria-label="Toggle navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="collapse navbar-collapse" id="navbarSupportedContent">
                    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                        <li class="nav-item">
                            <a class="nav-link active" href="/doc_dash">Home</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="/doc_profile">Profile</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="/doc_appointment">Appointment</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="/doc_prescription">Prescription</a>
                        </li>
                    </ul>
                    <div class="d-flex ms-auto">
                        <a href="/other_logout" class="btn btn-outline-danger">Logout</a>
                    </div>
                </div>
            </div>
        </nav>
    </header>
```



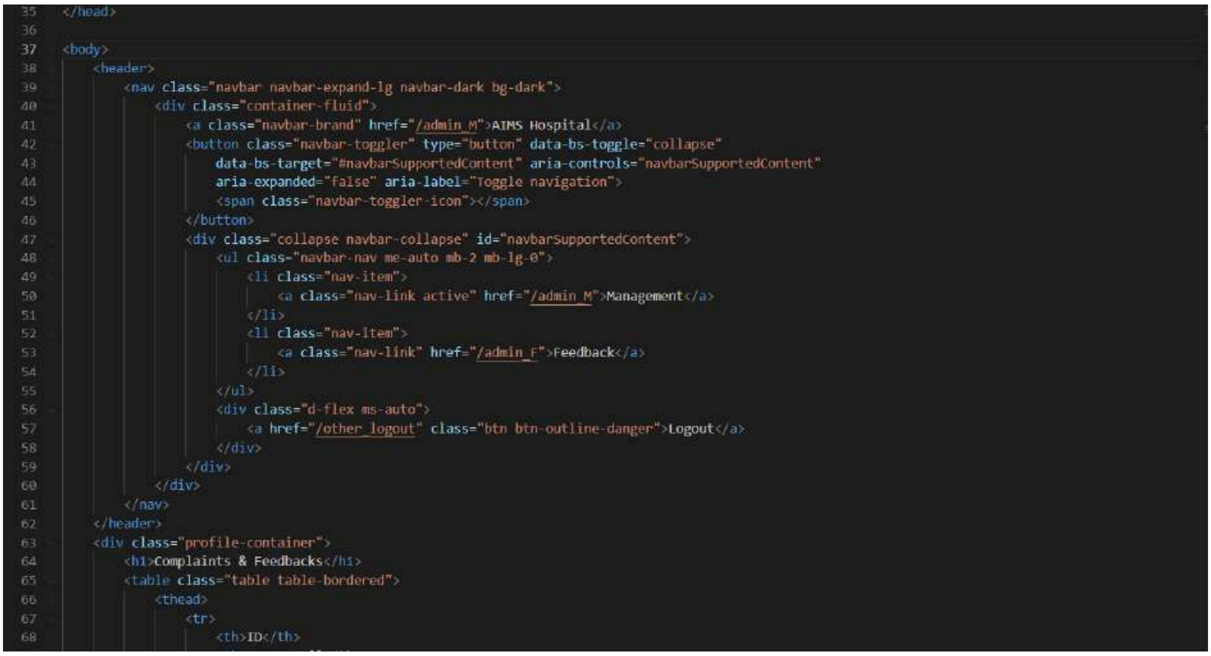
```html
<body>

    <div class="profile-container">
        <h1>Information Assigned By Admin</h1>
        <div class="mb-3">
            <label class="form-label"><strong>Name:</strong></label>
            <p class="form-control bg-light">{{user.name}}</p>
        </div>
        <div class="mb-3">
            <label class="form-label"><strong>Role:</strong></label>
            <p class="form-control bg-light">{{user.role}}</p>
        </div>
        <div class="mb-3">
            <label class="form-label"><strong>Email:</strong></label>
            <p class="form-control bg-light">{{user.email]}</p>
        </div>
    </div>

    <footer class="bg-dark text-light text-center py-3">
        <div class="container">
            <p class="mb-0">&copy; 2024 AIMS Hospital. All Rights Reserved.</p>
        </div>
    </footer>

    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"
        integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAJXtC2sI7zV9i3zE94q1m6Pq1m"
        crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.min.js"
        integrity="sha384-w76AGHjNHK6MeuKp3gI5F8zT8K1l7Jx8cx5WQ3rV+6bA/R3SRh5Fpe3u6v0nTkB"
        crossorigin="anonymous"></script>
</body>

</html>
```

The `doc_dash.html` page serves as the central dashboard for doctors within the **AIMS Hospital Management System**, providing a comprehensive overview of essential information assigned by the admin, such as the **doctor's name**, **role**, and **email**. The page features a **responsive navigation bar** that allows doctors to quickly access key sections like the **homepage**, **profile**, **appointments**, and **prescriptions**, ensuring smooth and efficient navigation throughout the system. Designed with **Bootstrap 5**, the layout is clean, modern, and fully responsive, ensuring compatibility across different devices and screen sizes. The page is tailored for ease of use, allowing doctors to manage their profile and hospital-related tasks from a single location. Additionally, the **secure logout option** ensures that doctors can safely exit the system when finished. This dashboard page is a critical hub for doctors to oversee and manage their tasks, making it an essential part of the hospital management system.

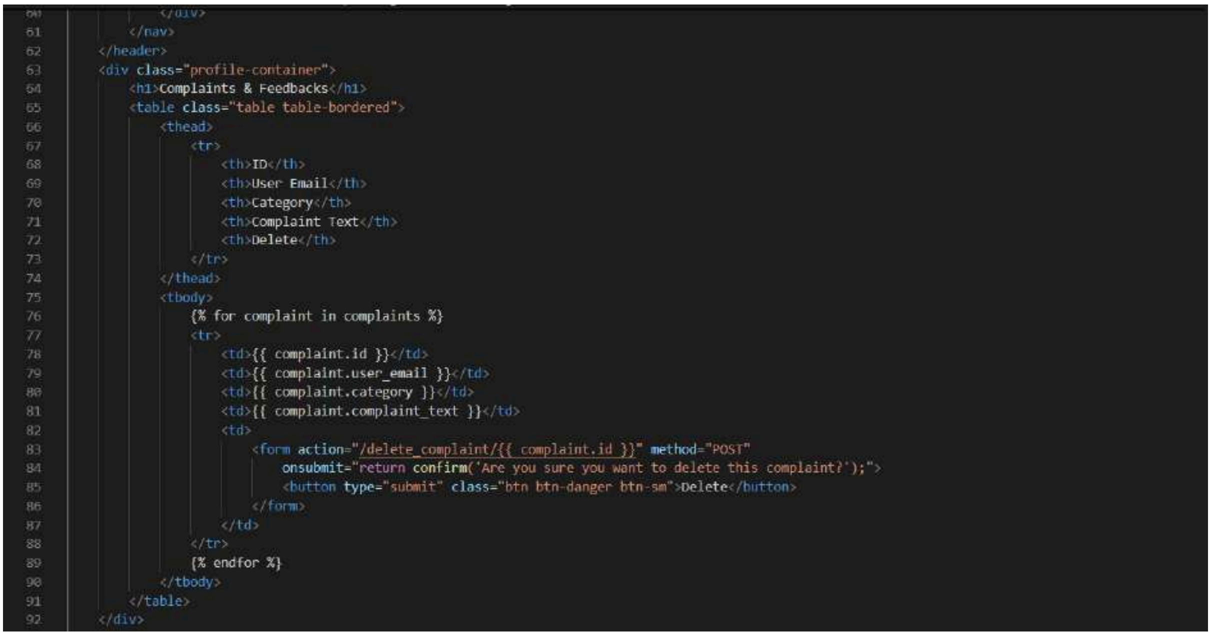
5.admin_F.html



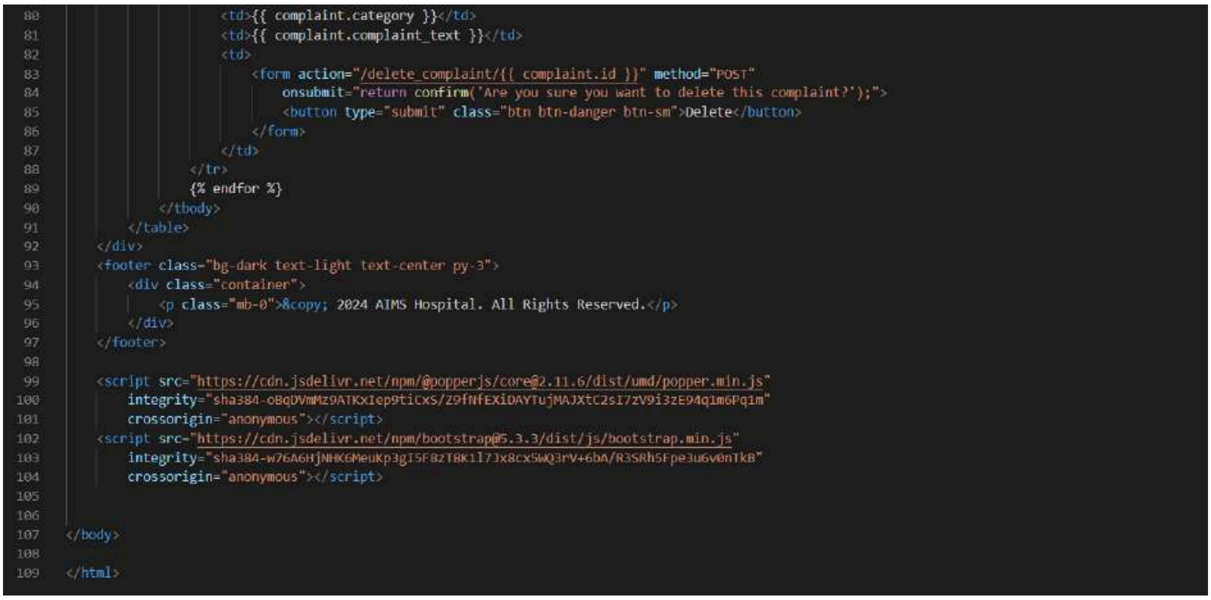
```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>AIMS Hospital - Admin Management</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
    <style>
        body {
            background-color: #f8f9fa;
            font-family: Arial, sans-serif;
        }

        .profile-container {
            max-width: 800px;
            margin: 40px auto;
            background-color: #fff;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
            border-radius: 10px;
            padding: 30px;
        }

        .profile-container h1 {
            color: #0d6efd;
            text-align: center;
            margin-bottom: 20px;
        }

        footer {
            margin-top: 30px;
        }
    </style>
</head>

<body>
```



```html
</head>

<body>
    <header>
        <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
            <div class="container-fluid">
                <a class="navbar-brand" href="/admin_M">AIMS Hospital</a>
                <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
                    data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
                    aria-expanded="false" aria-label="Toggle navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="collapse navbar-collapse" id="navbarSupportedContent">
                    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                        <li class="nav-item">
                            <a class="nav-link active" href="/admin_M">Management</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="/admin_F">Feedback</a>
                        </li>
                    </ul>
                    <div class="d-flex ms-auto">
                        <a href="/other_logout" class="btn btn-outline-danger">Logout</a>
                    </div>
                </div>
            </div>
        </nav>
    </header>
    <div class="profile-container">
        <h1>Complaints & Feedbacks</h1>
        <table class="table table-bordered">
            <thead>
                <tr>
                    <th>ID</th>
```



```
            </div>
        </nav>
    </header>
    <div class="profile-container">
        <h1>Complaints & Feedbacks</h1>
        <table class="table table-bordered">
            <thead>
                <tr>
                    <th>ID</th>
                    <th>User Email</th>
                    <th>Category</th>
                    <th>Complaint Text</th>
                    <th>Delete</th>
                </tr>
            </thead>
            <tbody>
                {% for complaint in complaints %}
                <tr>
                    <td>{{ complaint.id }}</td>
                    <td>{{ complaint.user_email }}</td>
                    <td>{{ complaint.category }}</td>
                    <td>{{ complaint.complaint_text }}</td>
                    <td>
                        <form action="/delete_complaint/{{ complaint.id }}" method="POST"
                            onsubmit="return confirm('Are you sure you want to delete this complaint?');">
                            <button type="submit" class="btn btn-danger btn-sm">Delete</button>
                        </form>
                    </td>
                </tr>
                {% endfor %}
            </tbody>
        </table>
    </div>
```

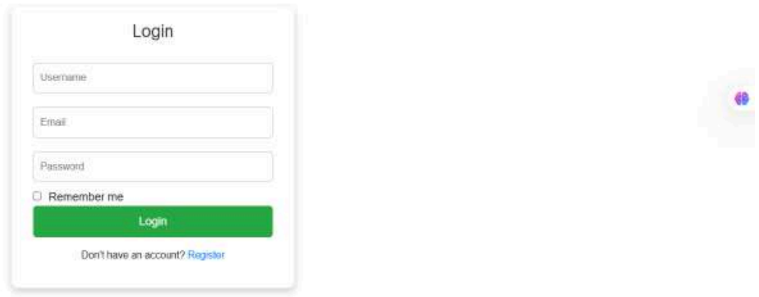


```
                    <td>{{ complaint.category }}</td>
                    <td>{{ complaint.complaint_text }}</td>
                    <td>
                        <form action="/delete_complaint/{{ complaint.id }}" method="POST"
                            onsubmit="return confirm('Are you sure you want to delete this complaint?');">
                            <button type="submit" class="btn btn-danger btn-sm">Delete</button>
                        </form>
                    </td>
                </tr>
                {% endfor %}
            </tbody>
        </table>
    </div>
    <footer class="bg-dark text-light text-center py-3">
        <div class="container">
            <p class="mb-0">&copy; 2024 AIMS Hospital. All Rights Reserved.</p>
        </div>
    </footer>

    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"
        integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAJXtC2sI7zV9i3zE94q1m6Pq1m"
        crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.min.js"
        integrity="sha384-w76A6HjNHK6MeuKp3gI5F8zT8K1l7Jx8cx5WQ3rV+6bA/R3SRh5Fpe3u6v0nTkB"
        crossorigin="anonymous"></script>


</body>

</html>
```

The **admin_F.html** page is a crucial component of the **AIMS Hospital Management System**, designed to allow administrators to efficiently manage and review patient feedback and complaints. Built with a responsive Bootstrap 5 layout, the page features a user-friendly navigation bar for quick access to the management and feedback sections. It displays a comprehensive table containing essential information about each complaint, including the complaint ID, the user’s email, the complaint category, and the text of the complaint. Administrators can easily delete complaints through a confirmation prompt to ensure that only relevant feedback remains. The page's clean, modern design makes it simple to navigate and manage patient complaints, ensuring the smooth operation of the hospital's feedback system while maintaining a high standard of customer service.

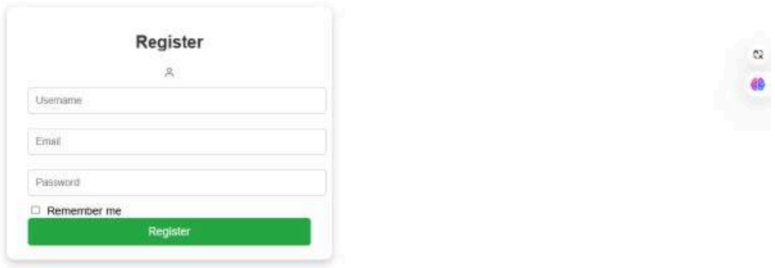
7. User Manual

The following pages are meant to be for the **stakeholder: Patient/User:**

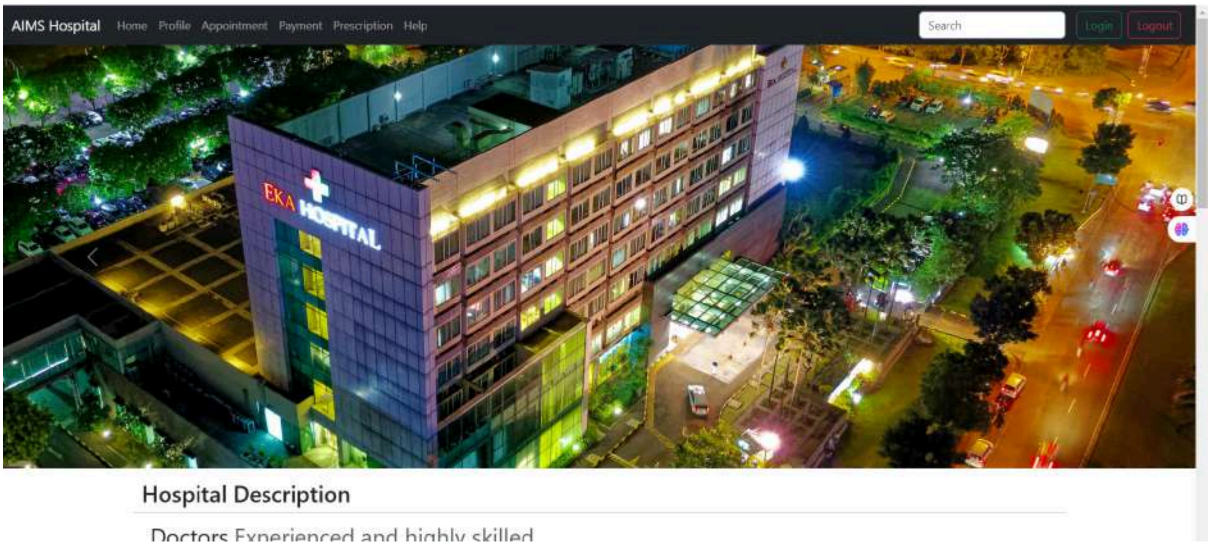


Don't have an account? Register

The User/Patient will first encounter the Login page. If the user is already registered then the user will input Username, email and Password and click the Login button to Login to the website. If the user is not registered then he/she will click the Blue color Register link.



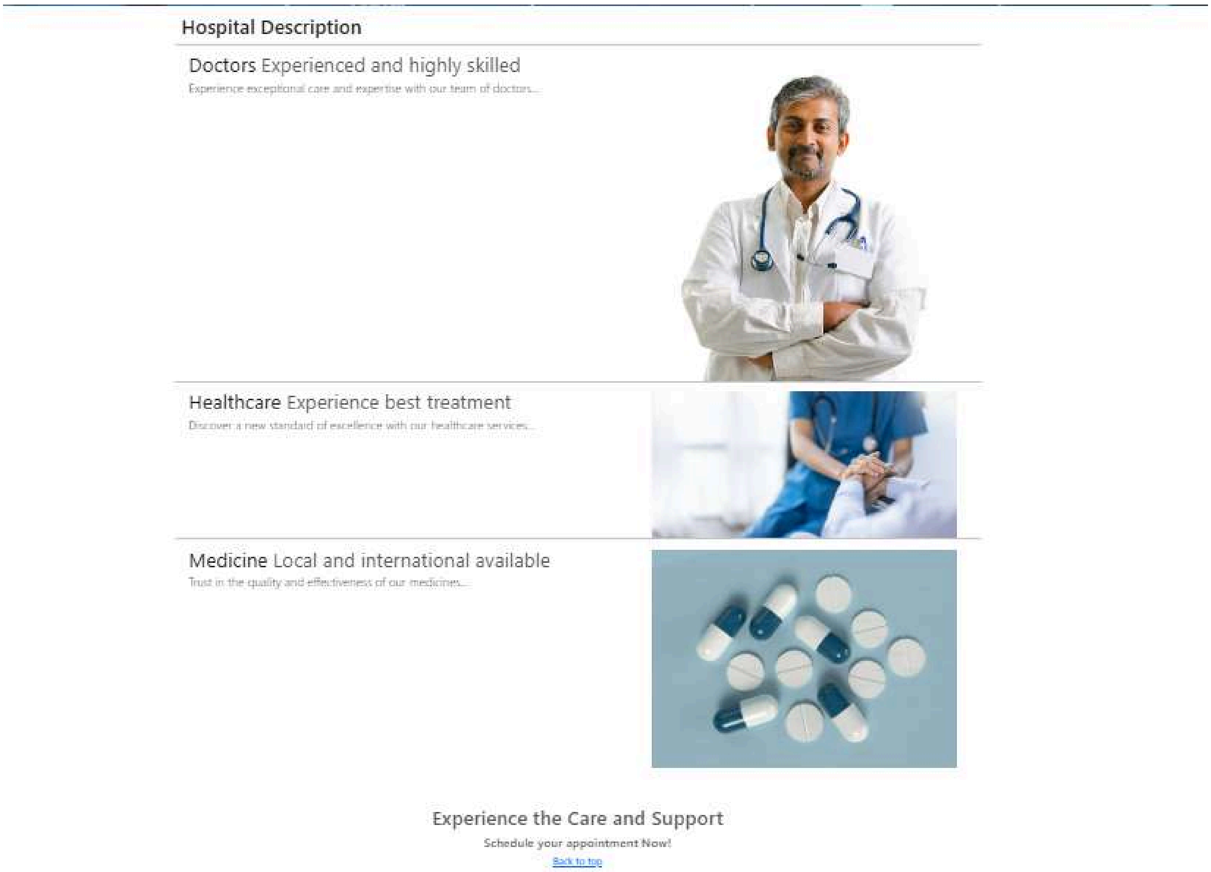
After, Clicking the register link user will see this page and after entering his/her information the user will click the Register button and that will register the User information and take the user to the login page to enter the website.



Hospital Description

Doctors Experienced and highly skilled

After successful Login the user will first see the dashboard of the hospital. It has designing elements like an auto carousel that will shift from time to time. And users will get a navbar at the top to navigate through the website. The navbar has a Login and Logout button at the top right.



Hospital Description

Doctors Experienced and highly skilled

Experience exceptional care and expertise with our team of doctors...

Healthcare Experience best treatment

Discover a new standard of excellence with our healthcare services...

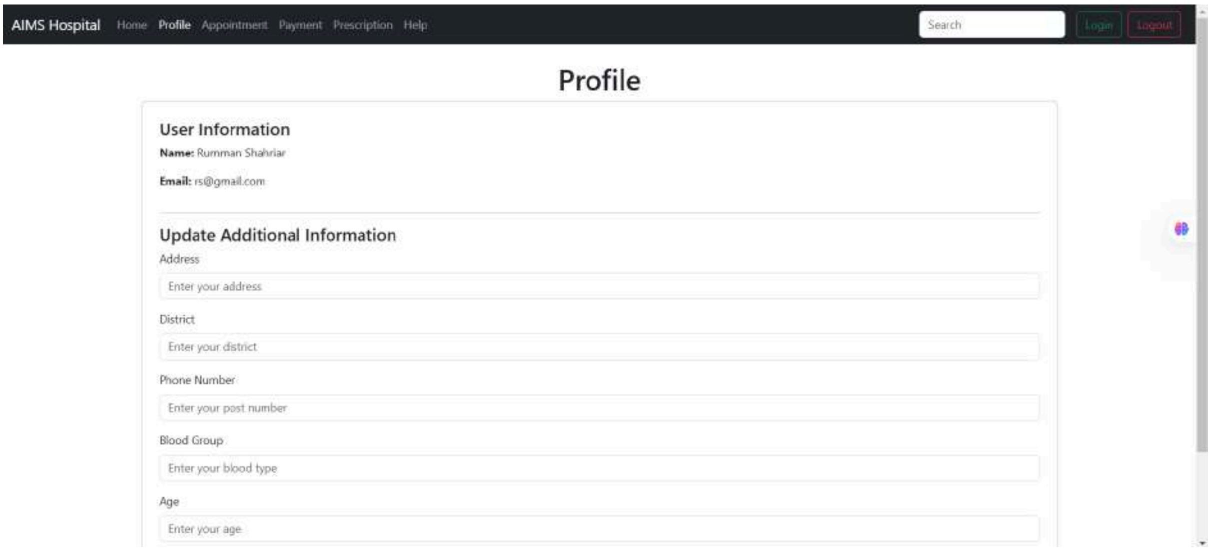
Medicine Local and international available

Trust in the quality and effectiveness of our medicines...

Experience the Care and Support

Schedule your appointment Now!

At the bottom of the dashboard user will get a short description of the hospital, doctor and services.



Profile

User Information

Name: Rumman Shahriar

Email: rs@gmail.com

Update Additional Information

Address

Enter your address

District

Enter your district

Phone Number

Enter your post number

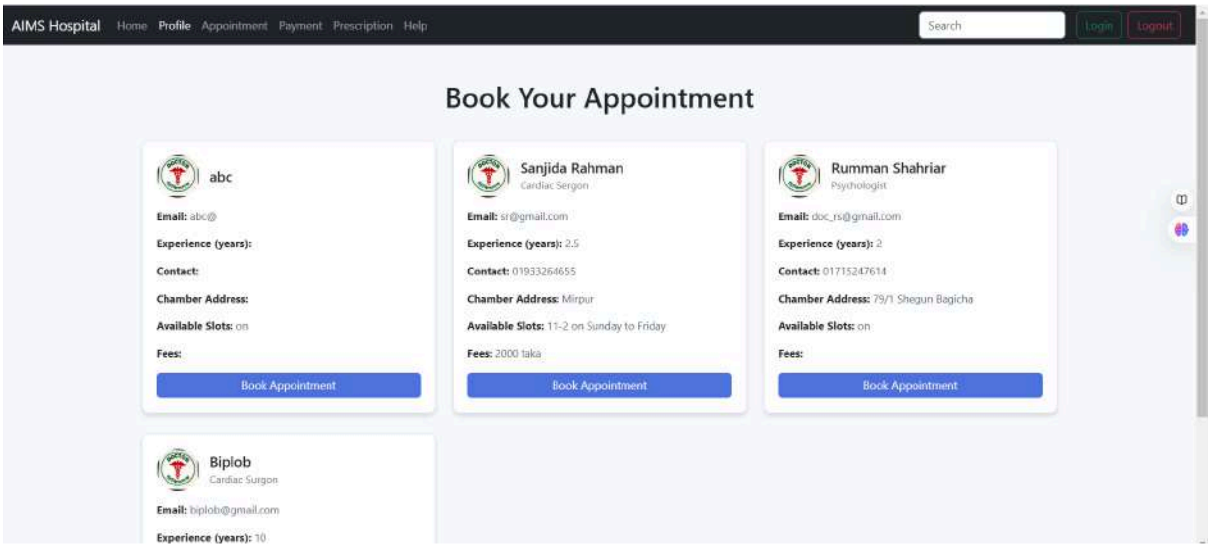
Blood Group

Enter your blood type

Age

Enter your age

Then, comes the next page Profile. Here a Patient/User can add their personal information and update that form time to time.



Book Your Appointment

abc

Email: abc@

Experience (years):

Contact:

Chamber Address:

Available Slots: on

Fees:

Sanjida Rahman
Cardiac Sergon

Email: sr@gmail.com

Experience (years): 2.5

Contact: 01933264655

Chamber Address: Mirpur

Available Slots: 11-2 on Sunday to Friday

Fees: 2000 taka

Rumman Shahriar
Psychologist

Email: doc_rs@gmail.com

Experience (years): 2

Contact: 01715247614

Chamber Address: 79/1 Shegun Bagicha

Available Slots: on

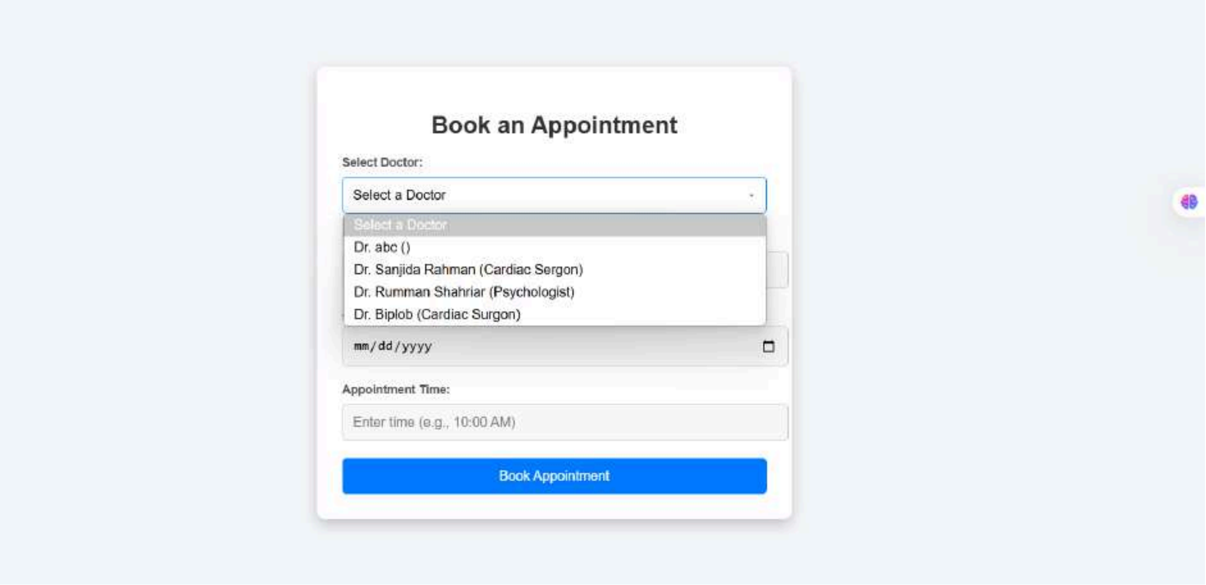
Fees:

Biplob
Cardiac Surgon

Email: biplob@gmail.com

Experience (years): 10

Then, the Patient can move to the Appointment page, this page will display all the available doctors list and their descriptions. Patients can book appointments to their desired doctor.



Book an Appointment

Select Doctor:

Select a Doctor

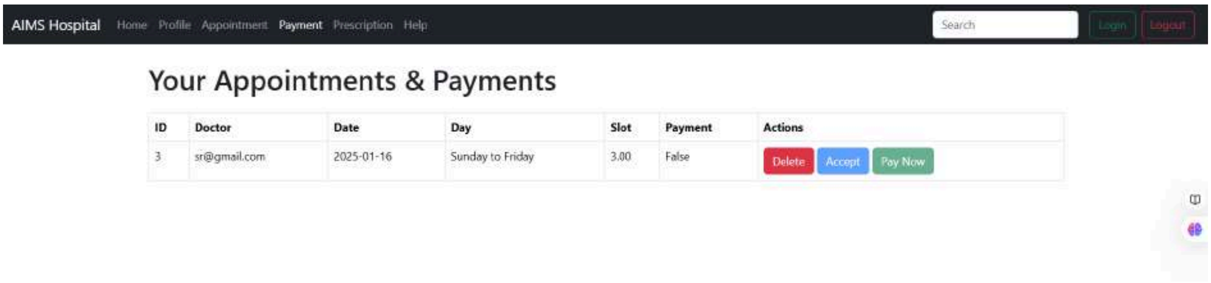
- Select a Doctor
- Dr. abc ()
- Dr. Sanjida Rahman (Cardiac Sergon)
- Dr. Rumman Shahriar (Psychologist)
- Dr. Biplob (Cardiac Surgon)

mm/dd/yyyy

Appointment Time:

Enter time (e.g., 10:00 AM)

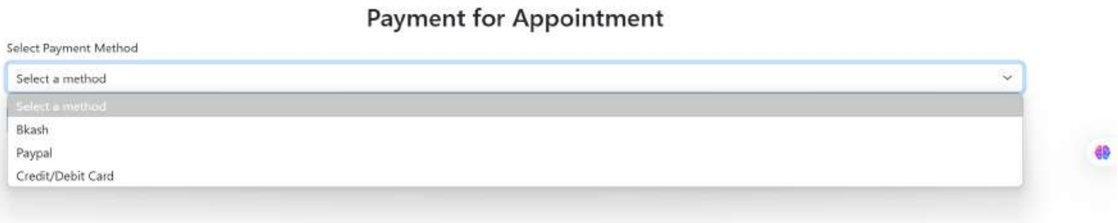
After clicking the Book Appointment button the patient will get to an Appointment form. And by filling out the form a patient can book an appointment.



Your Appointments & Payments

ID	Doctor	Date	Day	Slot	Payment	Actions
3	sr@gmail.com	2025-01-16	Sunday to Friday	3.00	False	Delete Accept Pay Now

After booking an appointment the patient can view the appointment in the Payment page. Here, the patient can delete the appointment if he/she wants. After getting approval from the doctor the patient will be able to accept the appointment from his/her end to confirm the appointment and then can click the button Pay now.

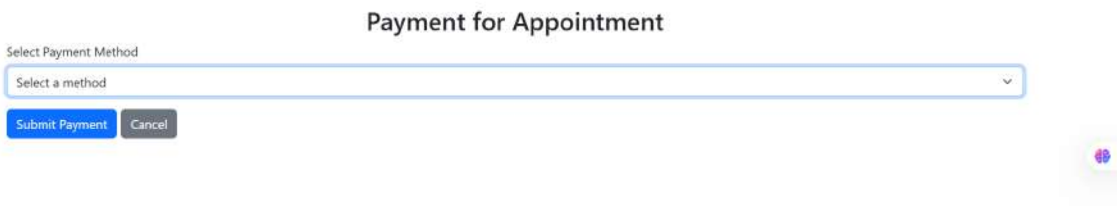


Payment for Appointment

Select Payment Method

Select a method

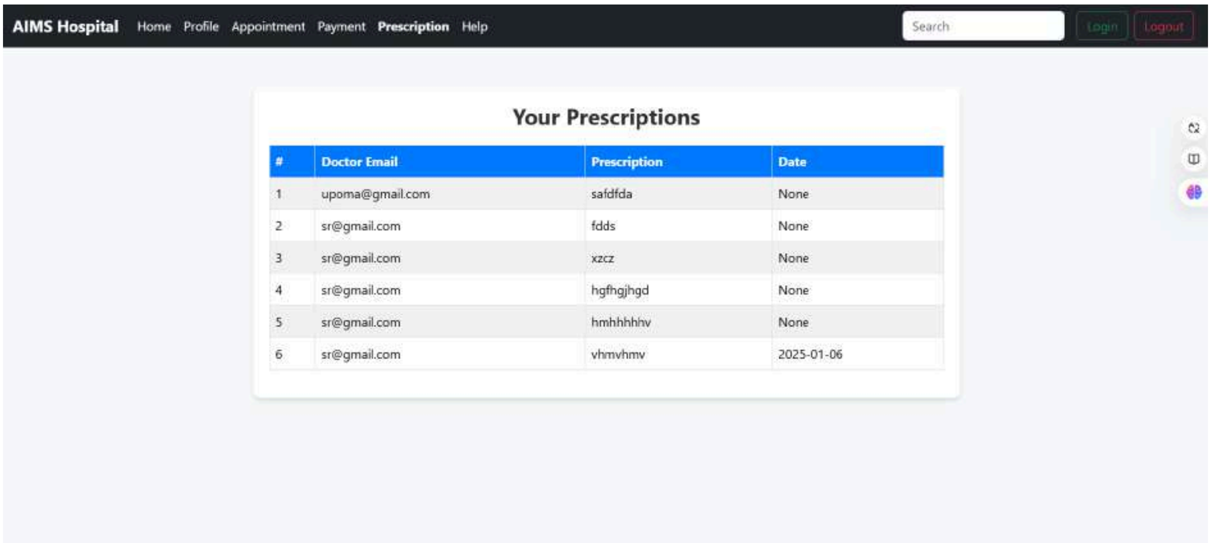
- Select a method
- Bkash
- Paypal
- Credit/Debit Card



Payment for Appointment

Select Payment Method

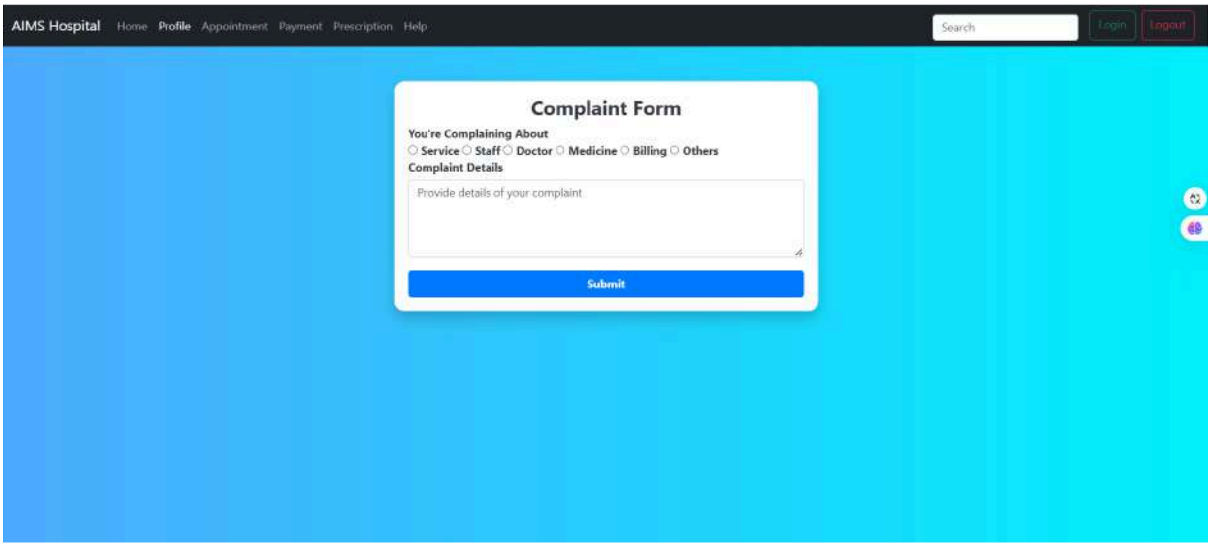
Pay now button would take the Patient to this form to choose the payment method and do the payment.



Your Prescriptions

#	Doctor Email	Prescription	Date
1	upoma@gmail.com	safdfda	None
2	sr@gmail.com	fdds	None
3	sr@gmail.com	xzcz	None
4	sr@gmail.com	hgfhgjhgd	None
5	sr@gmail.com	hmhhhhhv	None
6	sr@gmail.com	vhmvhmv	2025-01-06

Then, in the Prescription page, the patient would be able to look at all of his/her prescriptions.



Complaint Form

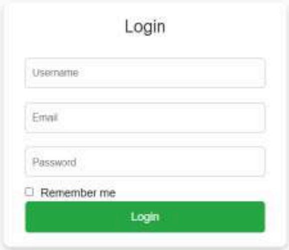
You're Complaining About

○ Service ○ Staff ○ Doctor ○ Medicine ○ Billing ○ Others

Complaint Details

Lastly, in the Help page the Patient would be able to file Complaint. First the patient would choose the designated stakeholder that he/she is complaining about then the patient would give the Complaint or feedback in the text box and click on submit. And a patient can submit multiple complaints in this page.

The following pages are meant to be for the **stakeholder: Doctor**



Login

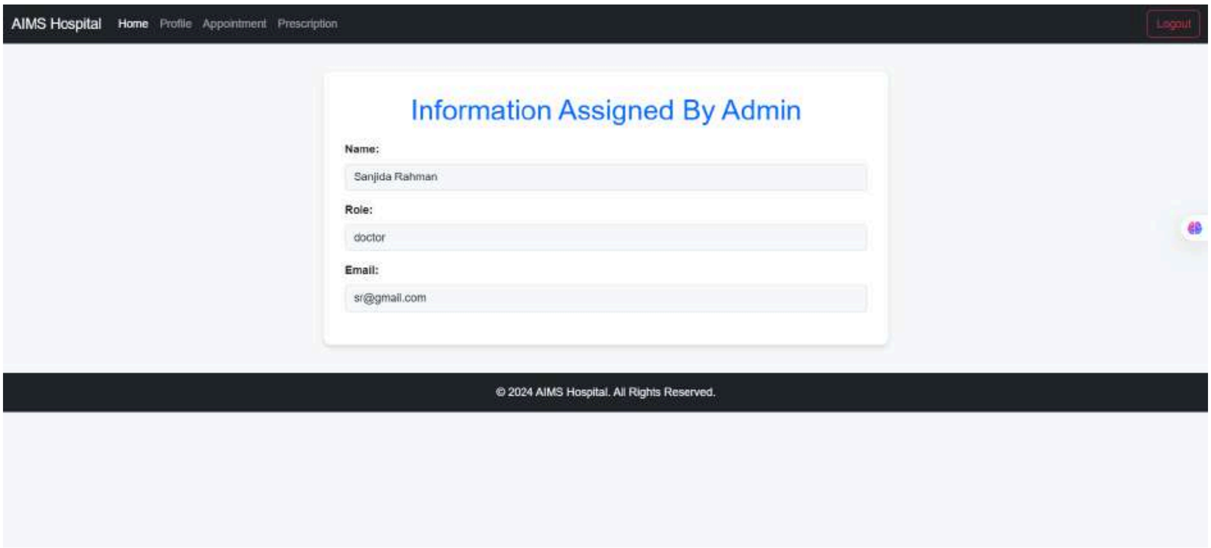
Username

Email

Password

□ Remember me

The following page is a Login page. Here, the Doctor will login to the system. For that, he/she has to register himself/herself first which will be approved by the Admin.



Information Assigned By Admin

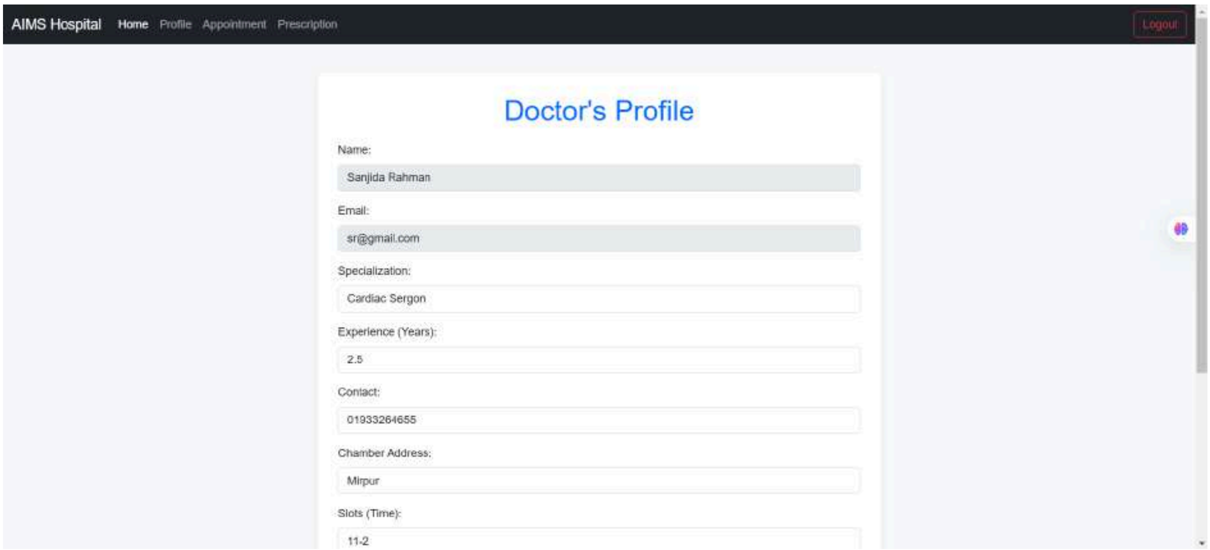
Name: Sanjida Rahman

Role: doctor

Email: sr@gmail.com

© 2024 AIMS Hospital. All Rights Reserved.

On this page, the Admin will assign the information about the Doctor containing name, role, and email.



Doctor's Profile

Name:
Sanjida Rahman

Email:
sr@gmail.com

Specialization:
Cardiac Sergon

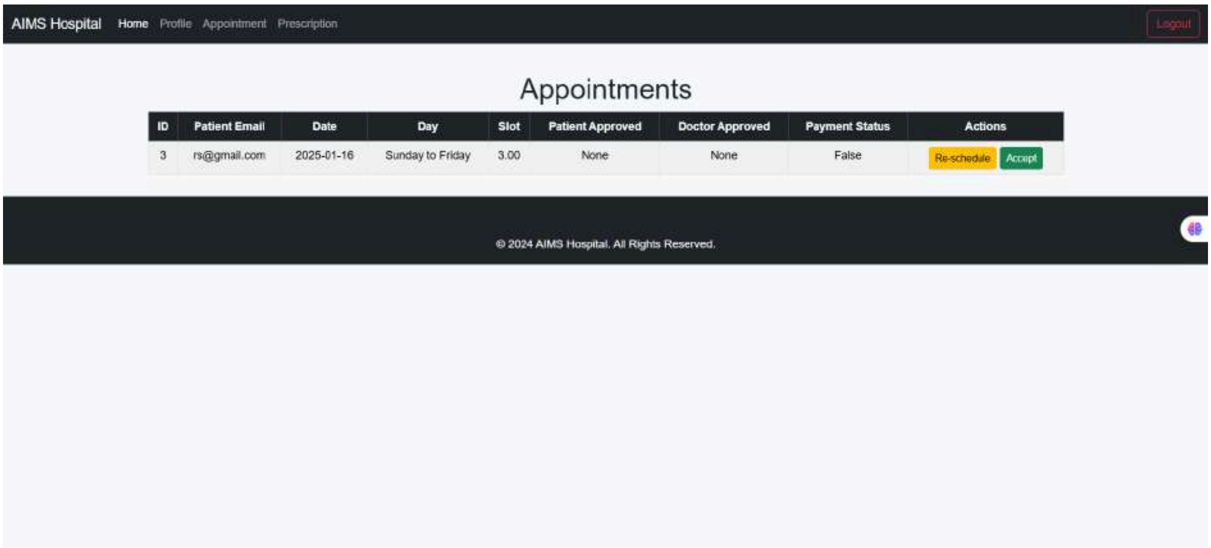
Experience (Years):
2.5

Contact:
01933264655

Chamber Address:
Mirpur

Slots (Time):
11-2

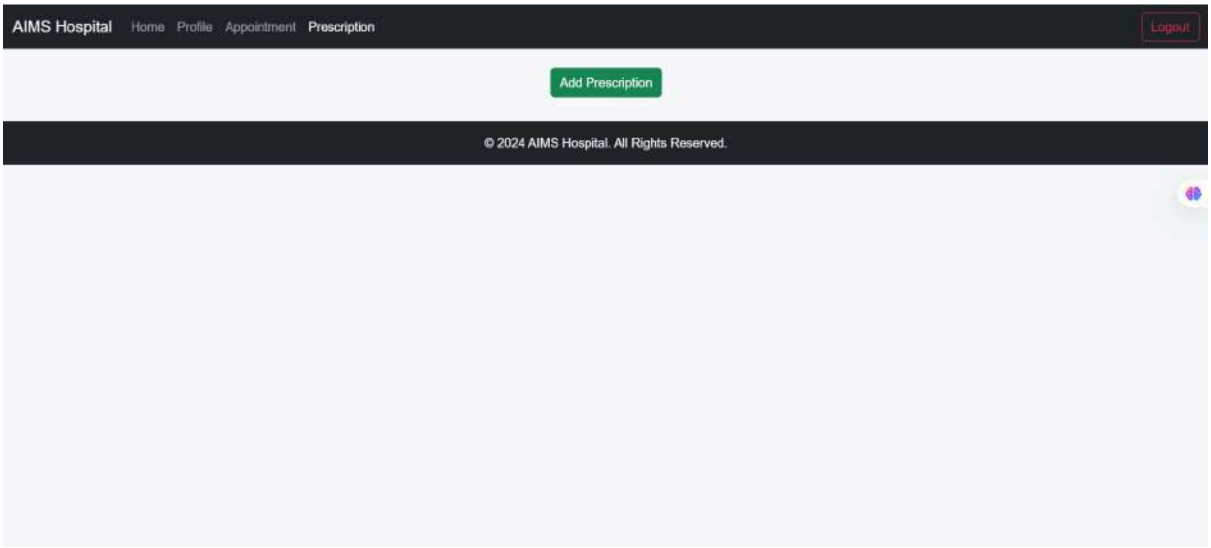
Later on, the Doctor can also edit the information regarding him/her. Therefore, the name and email can’t be changed without the approval from Admin.



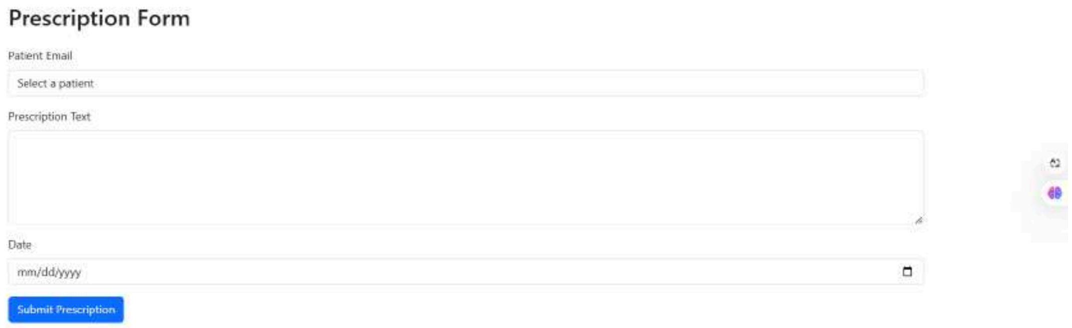
Appointments

ID	Patient Email	Date	Day	Slot	Patient Approved	Doctor Approved	Payment Status	Actions
3	rs@gmail.com	2025-01-16	Sunday to Friday	3.00	None	None	False	Re-schedule Accept

© 2024 AIMS Hospital. All Rights Reserved.



© 2024 AIMS Hospital. All Rights Reserved.


Prescription Form

Patient Email

Select a patient

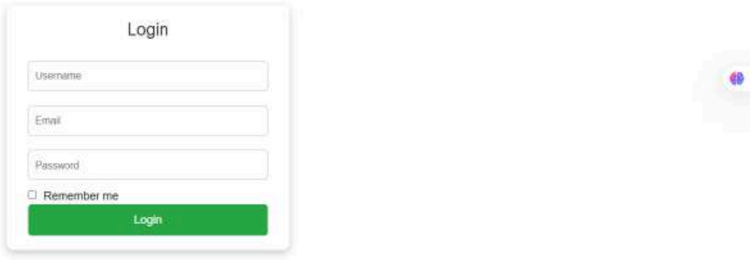
Prescription Text

Date

mm/dd/yyyy

These are the following pages that are meant to be for the Stakeholders: Patient and Doctor. Here, Patient will book an appointment to the doctor according to his/her choiceable time. Doctor also can approve or deny that appointment. After approving the appointment by the doctor, the Patient will receive the payment details and when the payment is done, the appointment will be confirmed by the Admin. Then the patient can take a consultation from a doctor and the doctor can give the necessary prescription that the patient needs.

The following pages are meant to be for the **stakeholder: Pharmacists**, where they are able to login to the system, check patients' prescriptions by searching through the database with the patient's email address, and finally manage the medicine inventory by adding and deleting medicine from the database directly from the user interface. The pharmacists will navigate through the system in the following manner:


Login

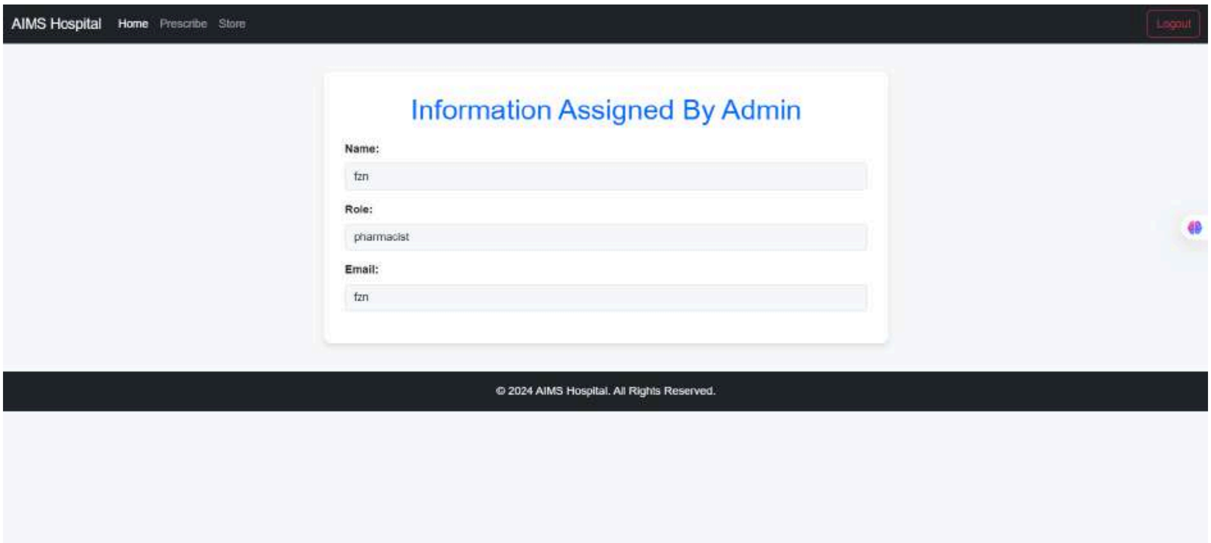
Username

Email

Password

□ Remember me

First, the pharmacists will log in to the system through the "other_login" portal, and upon successful login, they will be taken to the main pharmacist page (dashboard) which is shown in the following image:



Information Assigned By Admin

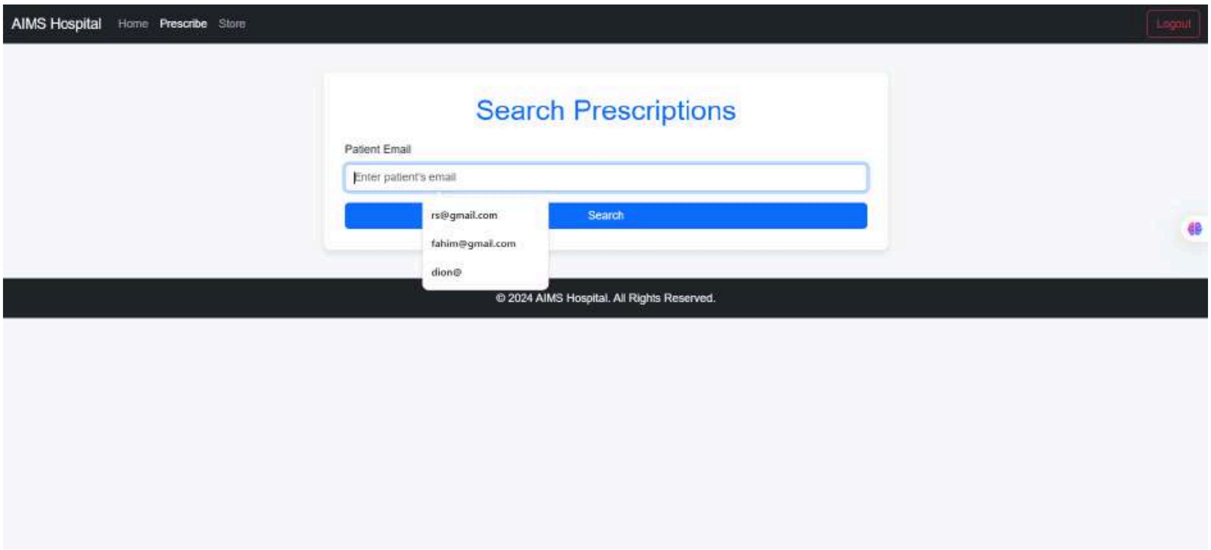
Name:
fzn

Role:
pharmacist

Email:
fzn

© 2024 AIMS Hospital. All Rights Reserved.

Now, this information displayed in the dashboard cannot be changed by the pharmacists themselves, only the admin can change them. Therefore, in order to change the pharmacist login info, they must consult the administrators. Next, the pharmacists can check patient's prescriptions by clicking on the "Prescribe" button in the navigation bar, and the system will bring up all the prescriptions prescribed to the patient, which is demonstrated in the following two images:



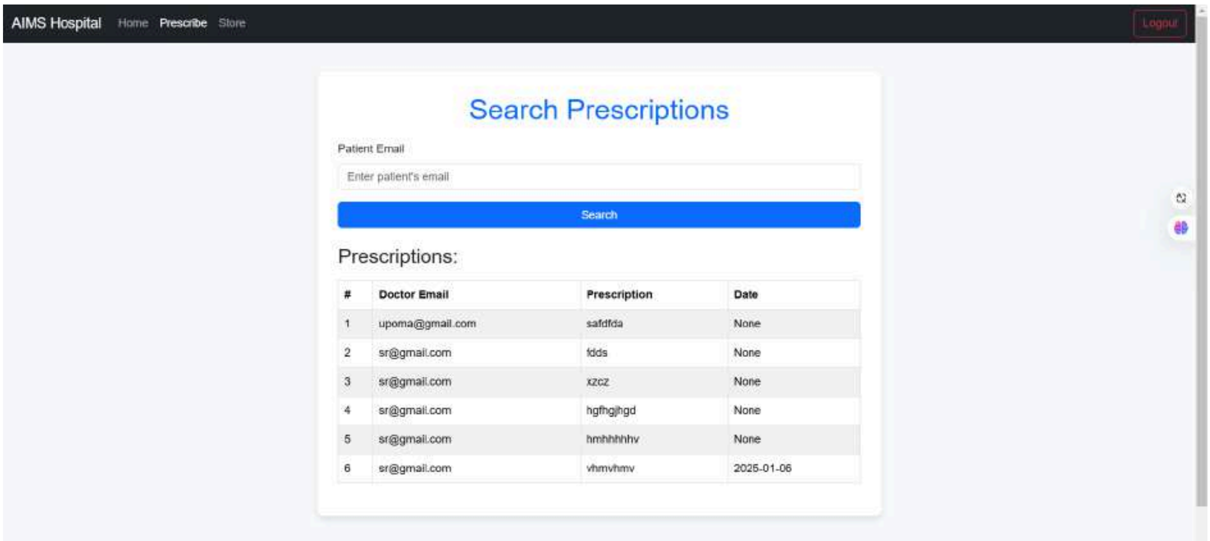
Search Prescriptions

Patient Email

Enter patient's email

rs@gmail.com
fahim@gmail.com
dion@

© 2024 AIMS Hospital. All Rights Reserved.



Search Prescriptions

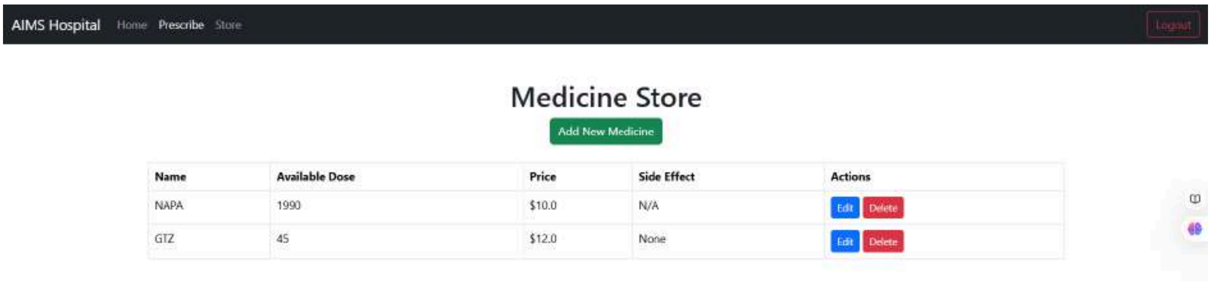
Patient Email

Enter patient's email

Prescriptions:

#	Doctor Email	Prescription	Date
1	upoma@gmail.com	safdfda	None
2	sr@gmail.com	fdds	None
3	sr@gmail.com	xzcz	None
4	sr@gmail.com	hgfhgjhgd	None
5	sr@gmail.com	hmhhhhhv	None
6	sr@gmail.com	vhmvhmv	2025-01-06

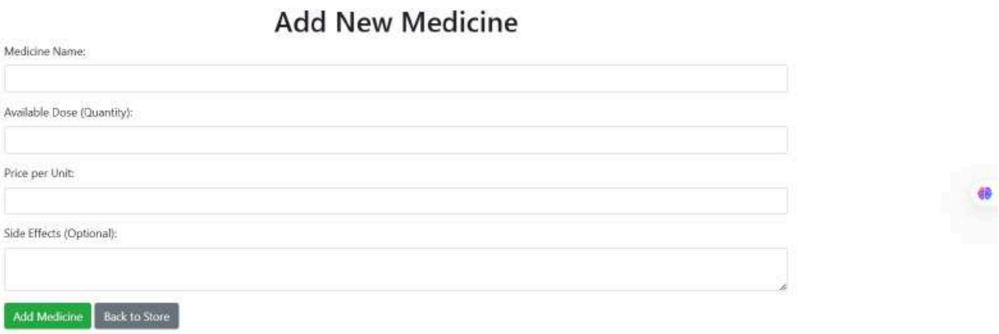
Next, the pharmacists can manage the medicine store by editing the details of the available meds in the database by clicking the blue edit button or they can delete all the medicine info by clicking the red delete button, which is shown in the following image:



Medicine Store

Name	Available Dose	Price	Side Effect	Actions
NAPA	1990	$10.0	N/A	Edit Delete
GTZ	45	$12.0	None	Edit Delete

The Pharmacists can also add new medicine to the database by clicking on the green add new medicine button and the following user interface will be shown where the pharmacists will fill out the medicine information:



Add New Medicine

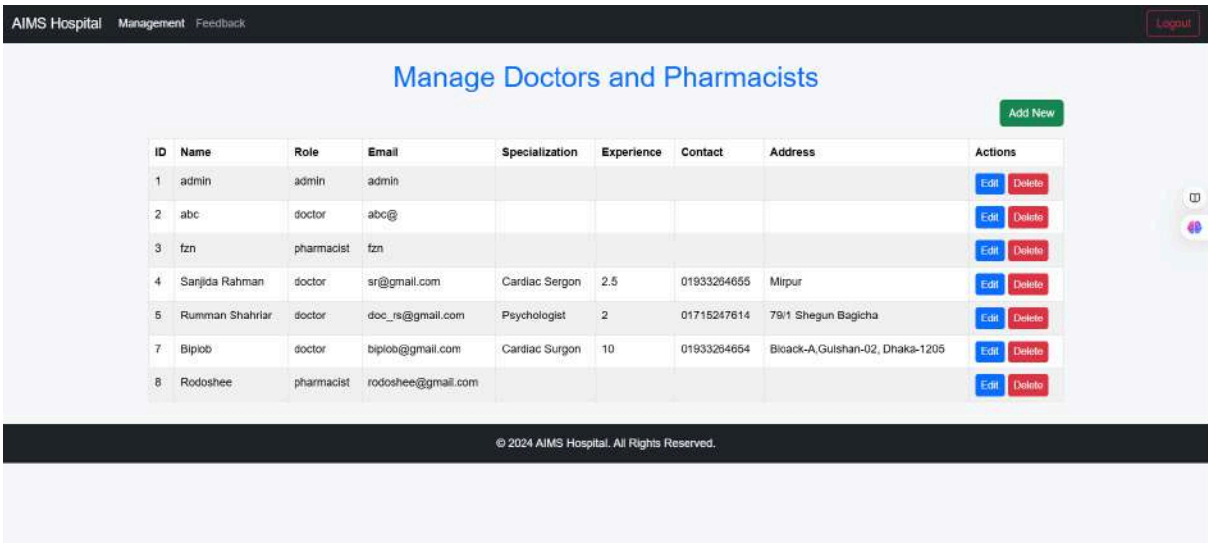
Medicine Name:

Available Dose (Quantity):

Price per Unit:

Side Effects (Optional):

The following pages are for the stakeholder: Admin where they can manage the existing stakeholders of the system such as admins, pharmacists, and doctors by editing their details or deleting their info. The following image shows the admin dashboard from where they can perform their desired tasks:

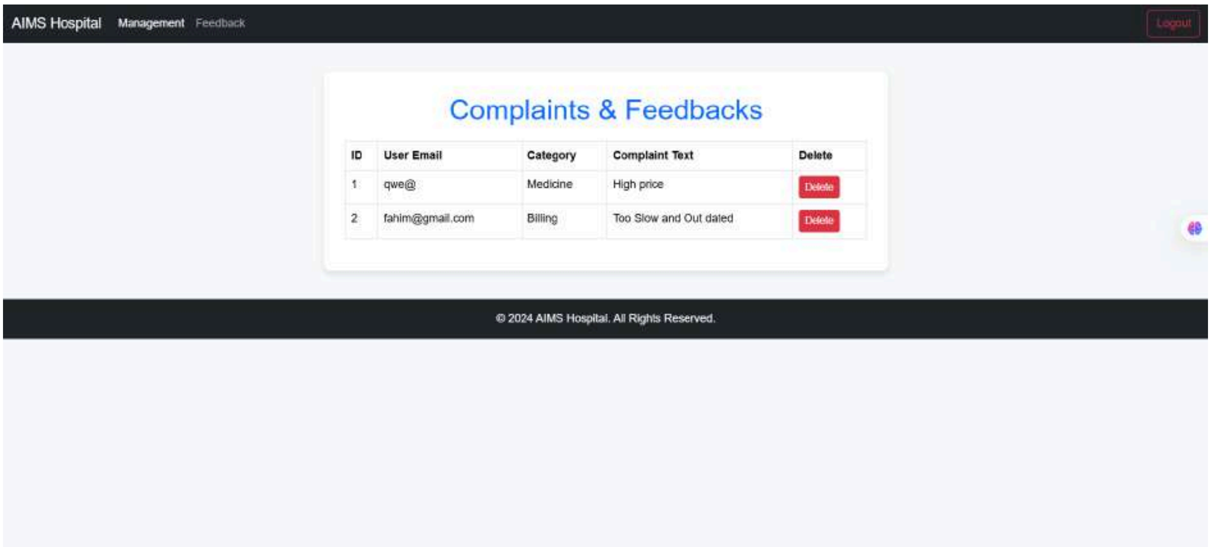


Manage Doctors and Pharmacists

ID	Name	Role	Email	Specialization	Experience	Contact	Address	Actions
1	admin	admin	admin					Edit Delete
2	abc	doctor	abc@					Edit Delete
3	fzn	pharmacist	fzn					Edit Delete
4	Sanjida Rahman	doctor	sr@gmail.com	Cardiac Sergon	2.5	01933264655	Mirpur	Edit Delete
5	Rumman Shahriar	doctor	doc_rs@gmail.com	Psychologist	2	01715247614	79/1 Shegun Bagicha	Edit Delete
7	Biplob	doctor	biplob@gmail.com	Cardiac Surgon	10	01933264654	Bloack-A,Gulshan-02, Dhaka-1205	Edit Delete
8	Rodoshee	pharmacist	rodoshee@gmail.com					Edit Delete

© 2024 AIMS Hospital. All Rights Reserved.

Lastly, the admins can check the feedback that was left by the patients for review by clicking on the feedback button. Upon clicking it, the admin will be taken to the next page (shown in the image below) and they can delete the reviews once it is resolved.



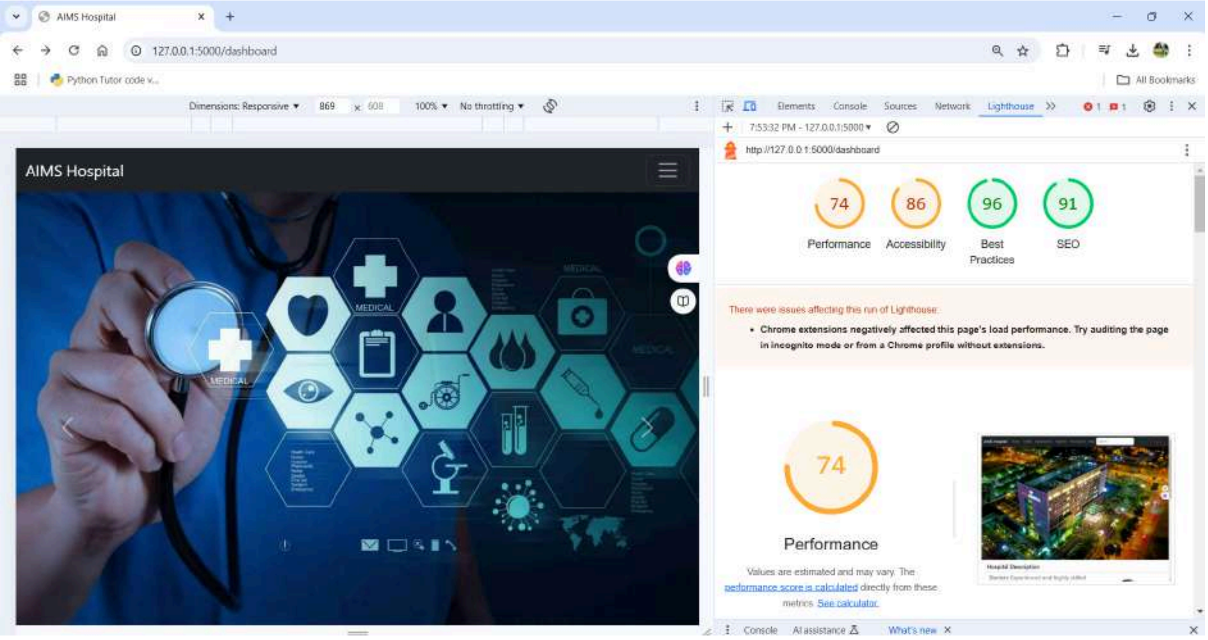
Complaints & Feedbacks

ID	User Email	Category	Complaint Text	Delete
1	qwe@	Medicine	High price	Delete
2	fahim@gmail.com	Billing	Too Slow and Out dated	Delete

© 2024 AIMS Hospital. All Rights Reserved.

8. Performance and Network Analysis

Performance and Network analysis reports using **lighthouse DevTool.**



http://127.0.0.1:5000/dashboard

74	86	96	91
Performance	Accessibility	Best Practices	SEO

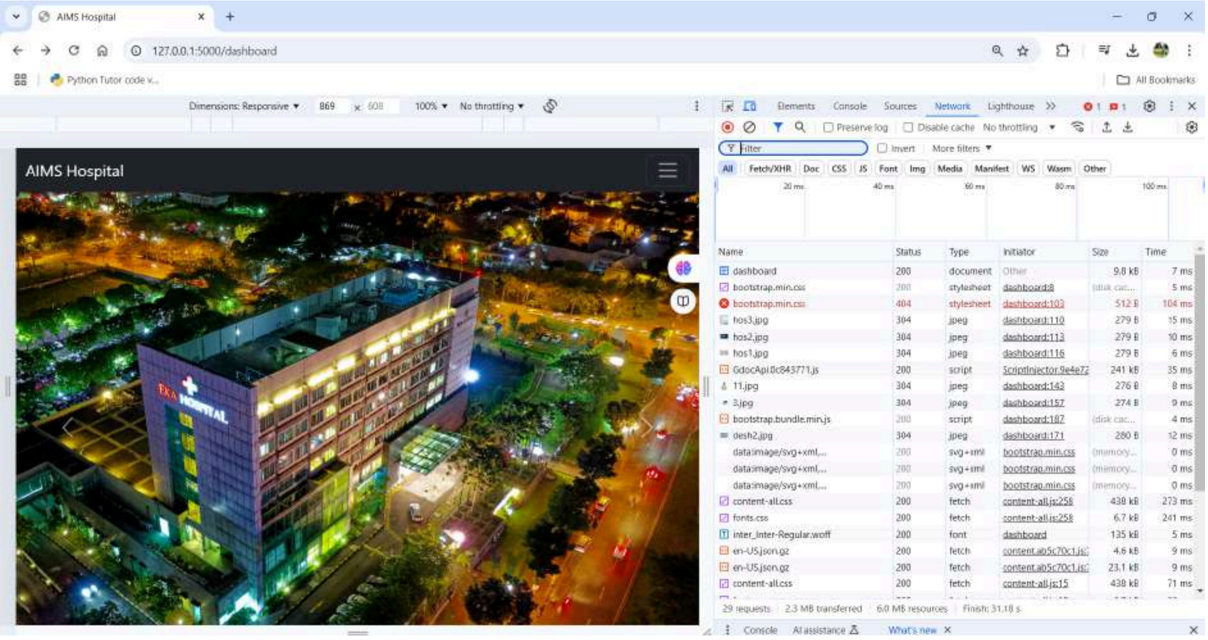
There were issues affecting this run of Lighthouse:

- **Chrome extensions negatively affected this page's load performance. Try auditing the page in incognito mode or from a Chrome profile without extensions.**

74

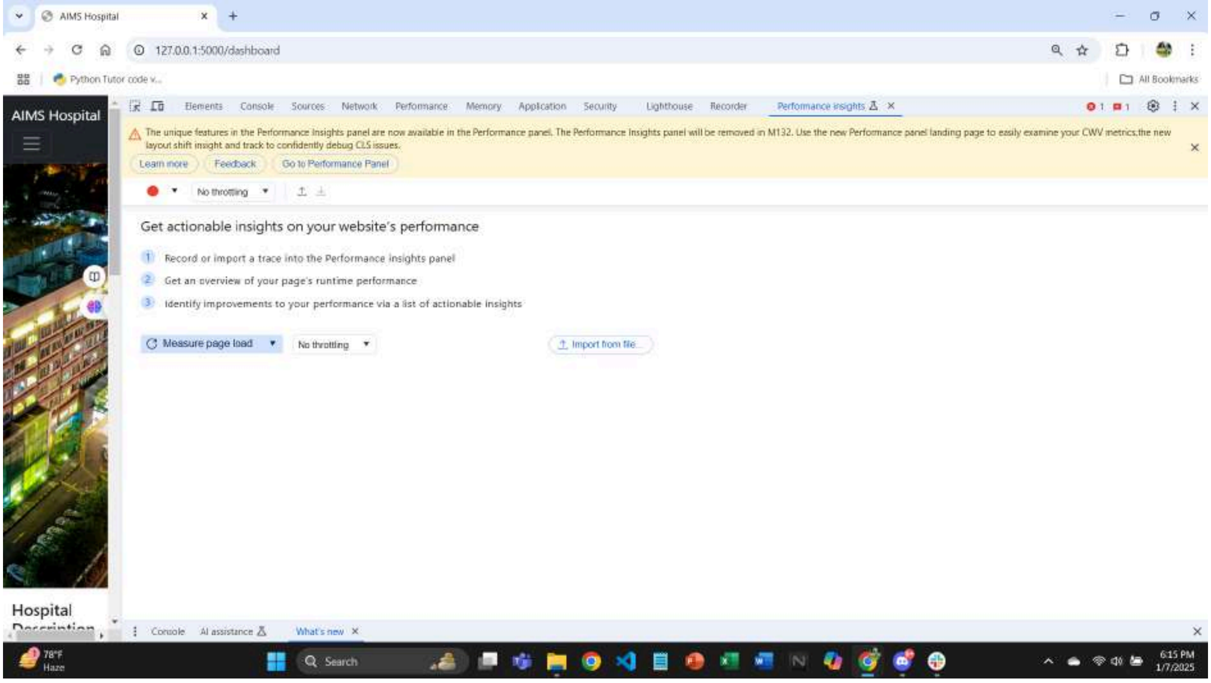
Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.



Name	Status	Type	Initiator	Size	Time
dashboard	200	document	Other	9.8 kB	7 ms
bootstrap.min.css	200	stylesheet	dashboard:8	(disk cac...	5 ms
bootstrap.min.css	404	stylesheet	dashboard:103	512 B	104 ms
hos3.jpg	304	jpeg	dashboard:110	279 B	15 ms
hos2.jpg	304	jpeg	dashboard:113	279 B	10 ms
hos1.jpg	304	jpeg	dashboard:116	279 B	6 ms
GdocApi.0c843771.js	200	script	ScriptInjector.9e4e72	241 kB	35 ms
11.jpg	304	jpeg	dashboard:143	276 B	8 ms
3.jpg	304	jpeg	dashboard:157	274 B	9 ms
bootstrap.bundle.min.js	200	script	dashboard:187	(disk cac...	4 ms
desh2.jpg	304	jpeg	dashboard:171	280 B	12 ms
data:image/svg+xml,...	200	svg+xml	bootstrap.min.css	(memory...	0 ms
data:image/svg+xml,...	200	svg+xml	bootstrap.min.css	(memory...	0 ms
data:image/svg+xml,...	200	svg+xml	bootstrap.min.css	(memory...	0 ms
content-all.css	200	fetch	content-all.js:258	438 kB	273 ms
fonts.css	200	fetch	content-all.js:258	6.7 kB	241 ms
inter_Inter-Regular.woff	200	font	dashboard	135 kB	5 ms
en-US.json.gz	200	fetch	content.ab5c70c1.js:	4.6 kB	9 ms
en-US.json.gz	200	fetch	content.ab5c70c1.js:	23.1 kB	9 ms
content-all.css	200	fetch	content-all.js:15	438 kB	71 ms

29 requests | 2.3 MB transferred | 6.0 MB resources | Finish: 31.18 s



The unique features in the Performance Insights panel are now available in the Performance panel. The Performance Insights panel will be removed in M132. Use the new Performance panel landing page to easily examine your CWV metrics,the new layout shift insight and track to confidently debug CLS issues.

Learn more | Feedback | Go to Performance Panel

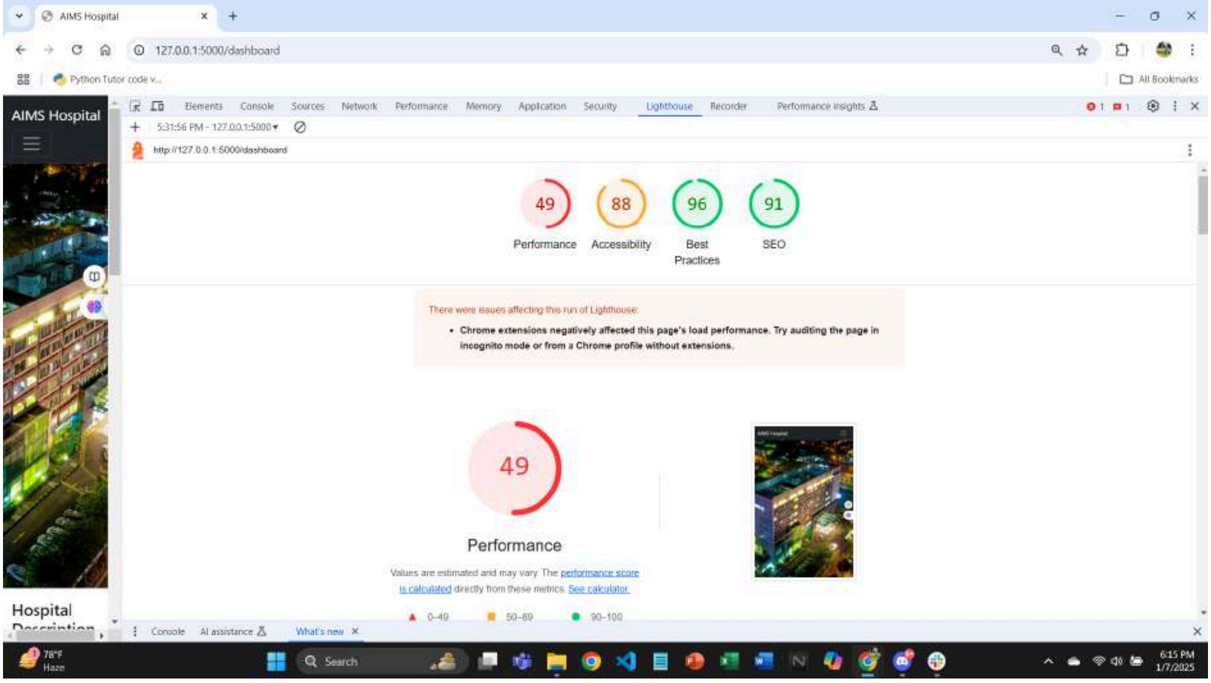
No throttling

Get actionable insights on your website's performance

1. Record or import a trace into the Performance insights panel
2. Get an overview of your page's runtime performance
3. Identify improvements to your performance via a list of actionable insights

Measure page load | No throttling | Import from file...

System in the mobile viewport and the reports.



Performance	Accessibility	Best Practices	SEO
49	88	96	91

There were issues affecting this run of Lighthouse:

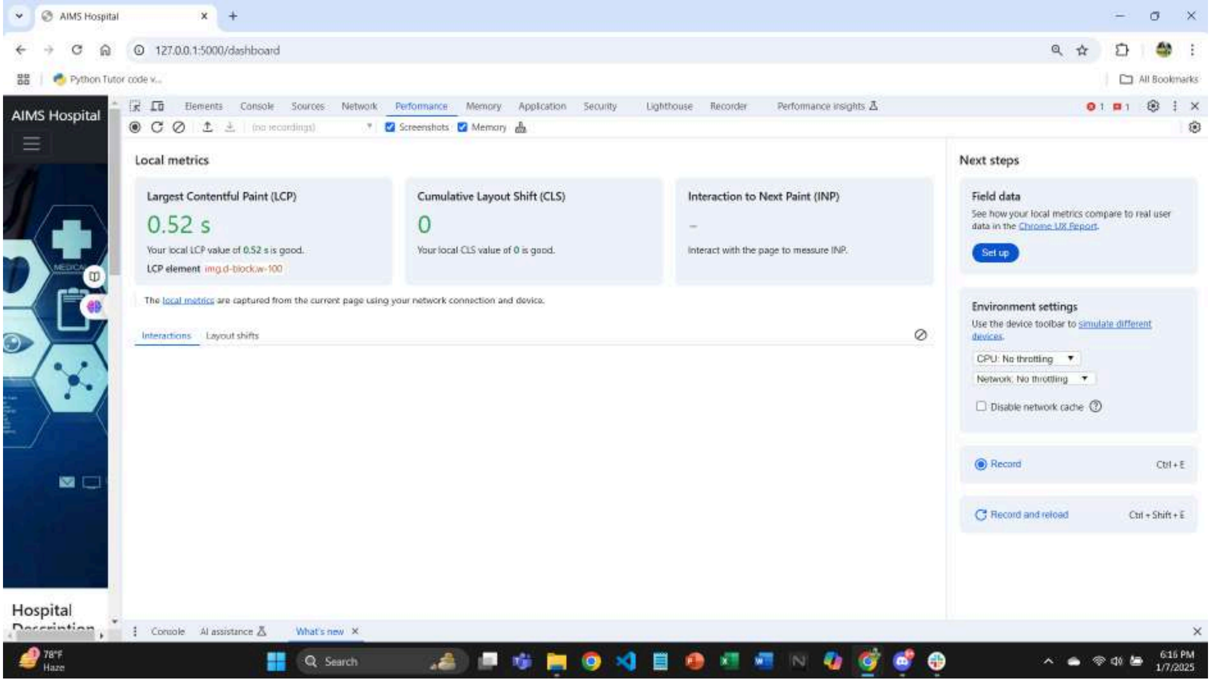
- Chrome extensions negatively affected this page's load performance. Try auditing the page in incognito mode or from a Chrome profile without extensions.

49

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100



Local metrics

Largest Contentful Paint (LCP)
0.52 s
Your local LCP value of 0.52 s is good.
LCP element img.d-block.w-100

Cumulative Layout Shift (CLS)
0
Your local CLS value of 0 is good.

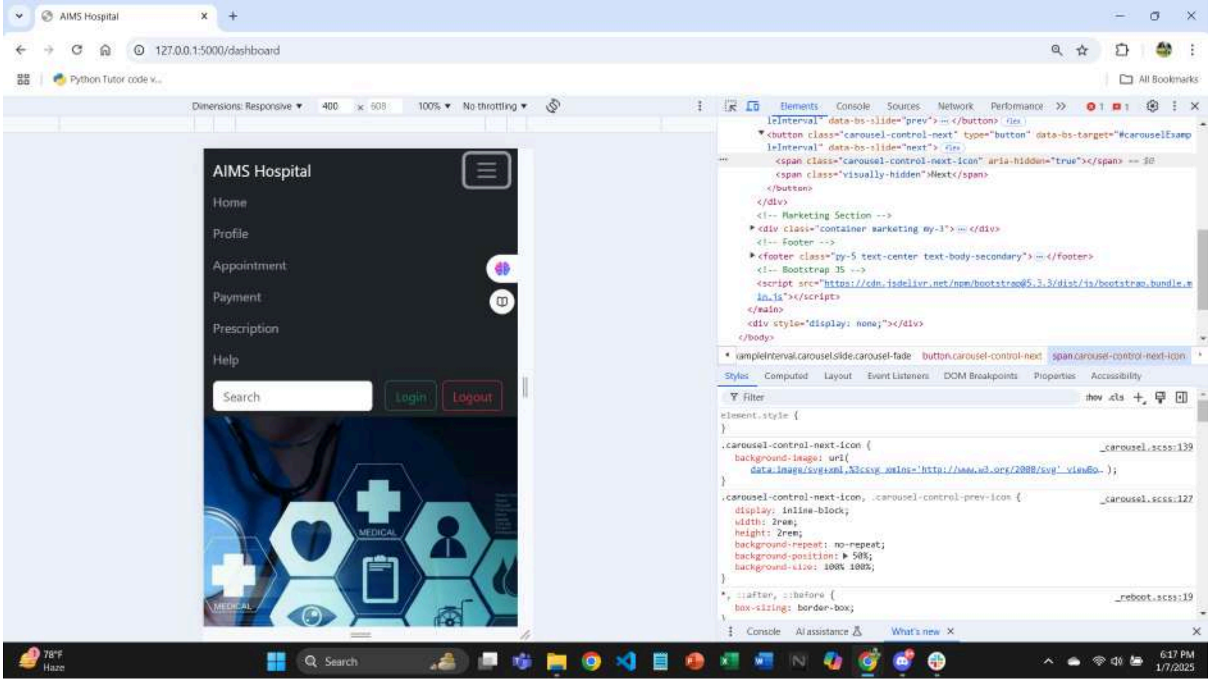
Interaction to Next Paint (INP)
-
Interact with the page to measure INP.

The local metrics are captured from the current page using your network connection and device.

Next steps

Field data
See how your local metrics compare to real user data in the Chrome UX Report.

Environment settings
Use the device toolbar to simulate different devices.
CPU: No throttling
Network: No throttling
□ Disable network cache



```
  leInterval" data-bs-slide="prev"> … </button>
 ▼<button class="carousel-control-next" type="button" data-bs-target="#carouselExamp
  leInterval" data-bs-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span> == $0
    <span class="visually-hidden">Next</span>
  </button>
 </div>
 <!-- Marketing Section -->
▶<div class="container marketing my-3"> … </div>
 <!-- Footer -->
▶<footer class="py-5 text-center text-body-secondary"> … </footer>
 <!-- Bootstrap JS -->
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.m
 in.js"></script>
</main>
<div style="display: none;"></div>
</body>
```

```
element.style {
}
.carousel-control-next-icon {                                        _carousel.scss:139
  background-image: url(
    data:image/svg+xml,%3csvg xmlns='http://www.w3.org/2000/svg' viewBo… );
}
.carousel-control-next-icon, .carousel-control-prev-icon {           _carousel.scss:127
  display: inline-block;
  width: 2rem;
  height: 2rem;
  background-repeat: no-repeat;
  background-position: ▶ 50%;
  background-size: 100% 100%;
}
*, ::after, ::before {                                                  _reboot.scss:19
  box-sizing: border-box;
```

So, the system is responsive.

9. Github Repo [Public] Link

https://github.com/RummanShahriar/Hospital.git

10. Link of Deployed Project

https://hospital-ej973dmn1-rumman-shahriars-projects.vercel.app/

11. Individual Contribution

Group member - 01	
Name: Md. Rumman Shahriar	Student ID:22101334/24241284
Functional Requirements which are developed by this member:	
1. View Dashboard: Patients can view the dashboard of the hospital.	
2. Profile Management: Patients can view and edit their profiles.	
3. Appointment Management: Patients can book appointments with doctors.	
4. Prescription Viewing: Patients can access their past prescriptions.	
5. Billing and Payment: Patients can pay their bills and payments for appointments.	
6. Feedback: Patients can provide feedback on services.	
7. Prescription Management: Pharmacists can view the prescriptions of patients.	

Group member - 02	
Name: Md. Asif Hasan Biplob	Student ID:22101736
Functional Requirements which are developed by this member:	
1. Doctor Profile Management: Doctors can view and edit their profiles and edit their information.	
2. Schedule Management: Doctors can view and manage their appointments according to their schedules	
3. Appointment Approval: Doctors can approve, decline, or reschedule patient appointments.	
4. Prescription Writing: Doctors can prescribe medications for patients.	
5. Storage Management: Pharmacists can manage the medicine stock of the hospital store.	

Group member - 03	
Name: Rodoshie Reheean	Student ID:21301417
Functional Requirements which are developed by this member:	

1. Appoint Doctor and Pharmacist: Admins can add Doctors and Pharmacists in the system accordingly.
2. Doctor Management: Admins can add, remove, and edit doctors and doctor's public information.
3. Pharmacist Management: Admins can add, remove, and edit pharmacists and pharmacist's public information.
4. Check Feedback: Admins can view patient feedback and delete them after taking proper steps.
5. Medicine Management: Pharmacists can add medicines to inventory and also change and edit the medicine price and description.

12. References

1. Flask. (n.d.). Flask documentation. Pallets Projects. Retrieved January 7, 2025, from https://flask.palletsprojects.com/en/stable/

2. **Bootstrap**. (n.d.). Bootstrap documentation. Retrieved January 7, 2025, from https://getbootstrap.com/docs/5.3/

3. **HTML**. (n.d.). HTML documentation. Mozilla Developer Network (MDN). Retrieved January 7, 2025, from https://developer.mozilla.org/en-US/docs/Web/HTML

4. **CSS**. (n.d.). CSS documentation. Mozilla Developer Network (MDN). Retrieved January 7, 2025, from https://developer.mozilla.org/en-US/docs/Web/CSS

5. **GitHub**. (n.d.). GitHub documentation. GitHub, Inc. Retrieved January 7, 2025, from https://docs.github.com/en

6. **Vercel**. (n.d.). Vercel documentation. Vercel, Inc. Retrieved January 7, 2025, from https://vercel.com/docs