North South University

Project Report

CSE311

Project Name: DG Health Portal Management System of National Health

SUBMITTED BY: AL AF MUKTADIR. ID: 2232975642

TANVIR AHMED. ID:2231047642

SUBMITTED TO: ABU SAYER TOQUE (SLF)

LAB INSTRUCTOR: SHABBIR AHMED

SECTION: 12

GROUP: 4

SUBMISSION DATE: 02/12/2024

DG Health Portal Management System Of National Health

1. Introduction

The **DG Health Portal Management System** is a robust, web-based application developed to centralize healthcare services at a national level. It simplifies patient appointments, resource management, and access to additional services like blood banks, ambulance coordination, and health tips. The project employs **XAMPP** for hosting on a local server and leverages **PHP**, **HTML**, **CSS**, and **SQL** for development.

2. Objectives

- Centralize healthcare service management.
- Streamline patient and hospital interactions.
- Maintain secure, efficient, and accurate data management.
- Enhance accessibility to additional health-related services.

3. Technologies Used

- **Back-end Development**: PHP
- **Front-end Development**: HTML, CSS,TAILWIND
- **Database**: MySQL (via XAMPP)
- **Server**: Apache (via XAMPP)
- **Development Environment**: Localhost

4. Features

1. **Appointment Scheduling**: Patients can book appointments based on department and doctor availability.
2. **Hospital Resource Management**: Tracks staff, departments, and facilities.
3. **Pharmacy Management**: Monitors medicine inventory and availability.
4. **Support Services**: Includes blood banks, ambulance services, and health tips.

5. Database Design

The project follows a relational database model with entities categorized into **strong**, **weak**, and **role entities**. Below is the detailed entity classification, attributes, and relationships.

Entities

1. Appointments

- **Entity Type**: Strong, Transactional
- **Description**: Represents patient bookings with doctors in specific hospitals and departments.
- **Attributes**:
  - `appointment_id` (Primary Key)
  - `patient_name`, `patient_age`, `patient_gender`, `patient_email`, `contact_number`
  - `hospital_id` (Foreign Key referencing `Hospitals`)
  - `dept_id` (Foreign Key referencing `Departments`)
  - `doctor_id` (Foreign Key referencing `Doctors`)
  - `appointment_date`, `patient_id` (Foreign Key referencing `Patients`)
- **Relationships**:
  - One-to-Many with `Hospitals`, `Departments`, and `Doctors`.
  - Many-to-Many with `Patients`.

2.Admin

- **Entity Type**: Strong, Master Data
- **Description**: Represents every management within hospitals.
- **Attributes**:
  - `id` (Primary Key), `name`, `name`, `username`, `password`,`email`,`contact_number`,`created_at`.
- **Relationships**:
  - Many-to-One with `all`.

3. Doctors

- **Entity Type**: Strong, Role
- **Description**: Represents doctors working in hospitals.

- **Attributes**:
  - `doctor_id` (Primary Key), `name`, `email`, `password`, `experience`, `department_id` (Foreign Key referencing `Departments`), `hospital_id` (Foreign Key referencing `Hospitals`)
- **Relationships**:
  - Many-to-One with `Departments` and `Hospitals`.
  - Linked with `Appointments` for consultations.

4. Medicine

- **Entity Type**: Weak, Inventory
- **Description**: Represents medicine stocks in hospital pharmacies.
- **Attributes**:
  - `medicine_id` (Primary Key), `pharmacy_id` (Foreign Key referencing `Pharmacy`), `name`, `type`, `manufacturer`, `expiry_date`, `quantity`
- **Relationships**:
  - Many-to-One with `Pharmacy`.

5. Hospital Staff

- **Entity Type**: Strong, Role
- **Description**: Represents administrative and support staff of hospitals.
- **Attributes**:
  - `staff_id` (Primary Key), `name`, `designation`, `contact_number`, `email`, shift,`hospital_id` (Foreign Key referencing `Hospitals`)
- **Relationships**:
  - Many-to-One with `Hospitals`.

6. Pharmacy

- **Entity Type**: Strong, Facility
- **Description**: Represents pharmacies linked to hospitals.
- **Attributes**:
  - `pharmacy_id` (Primary Key), `name`, `location`, `contact_number`, `email`, `hospital_id` (Foreign Key referencing `Hospitals`)
- **Relationships**:
  - One-to-One with `Hospitals`.
  - One-to-Many with `Medicine`.

7. Hospitals

- **Entity Type**: Strong, Core
- **Description**: Represents hospitals in the system.
- **Attributes**:
  - `hospital_id` (Primary Key), `name`, `location`, `contact`, `email`
- **Relationships**:
  - One-to-Many with `Departments`, `Doctors`, `Appointments`, `Pharmacy`, and `Hospital Staff`.

8. Patients

- **Entity Type**: Strong, User
- **Description**: Represents patients using the portal.
- **Attributes**:
  - `patient_id` (Primary Key), `name`, `email`, `password`, `gender`, `age`, `contact`
- **Relationships**:
  - Many-to-Many with `Appointments`.

9. Ambulance

- **Entity Type**: Weak, Service
- **Description**: Represents ambulance services provided by hospitals.
- **Attributes**:
  - `ambulance_id` (Primary Key), `vehicle_number`, `type`, `status`, `location.`

10. bookambulance

- **Entity Type**: Strong, Service
- **Description**: Represents ambulance booking services provided by hospitals.
- **Attributes**:
  - booking`_id` (Primary Key), `patient_name`, `contact_number`, pickup_location,dropoff_location,booking_time,ambulance`_id` (Foreign Key),patient_id(Foreign Key).
- **Relationships**:
  - Many-to-One with Patient.

11. Health Tips

- **Entity Type**: Informational Entity
- **Description**: Represents health-related guidance.
- **Attributes**:
  - `tip_id` (Primary Key), `title`, `description`,created_at

12.DEPARTMENTS

- **Entity Type**: Strong, Master Data
- **Description**:Represents departments within hospitals.
- Attributes:
  - `dept_id (Primary Key), name, hospital_id (Foreign Key referencing Hospitals), doctor_id (Foreign Key referencing Doctors)`

```
Relationships:
Many-to-One with Hospitals.
One-to-Many with Doctors.
```

6. Implementation Details

Backend (PHP & SQL)

- PHP scripts developed to handle CRUD operations for all entities.
- SQL used for efficient data retrieval and relationship integrity.

Frontend (HTML & CSS)

- HTML used for creating structure and forms for user interactions.
- CSS applied for styling and responsive design.

Hosting (XAMPP)

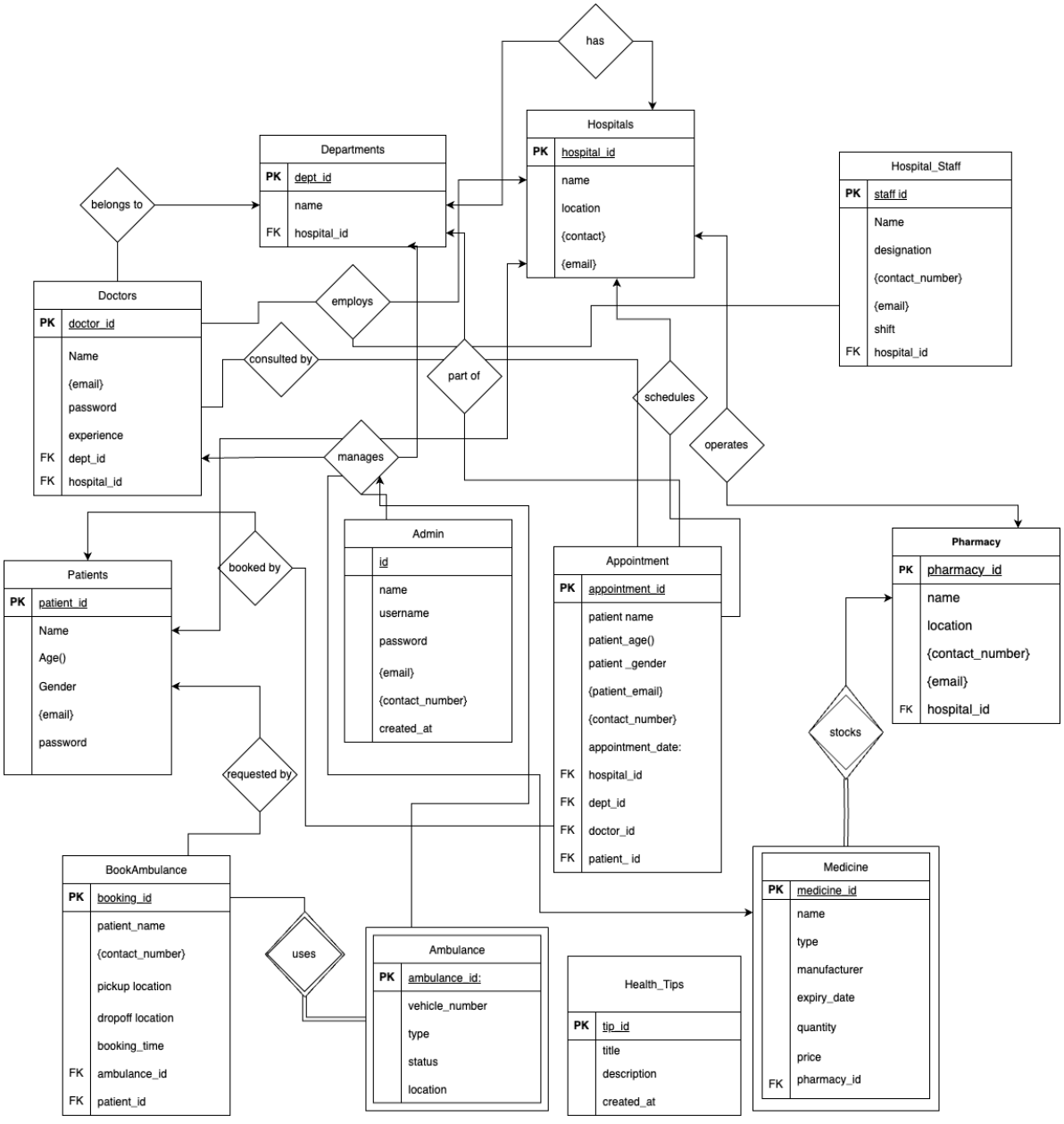
- Apache server hosts PHP scripts on a local environment.

7. Challenges and Solutions

- **Challenge**: Managing complex entity relationships.
  Solution: Applied normalization to simplify database structure.

- **Challenge**: Designing user-friendly interfaces.
  Solution: Conducted iterative testing for optimal UI/UX.

Entity-Relationship Diagram (ERD):



Relation Schema:

1. Hospitals(**hospital_id**, name, location)

2. Departments(**dept_id**, name, hospital_id)

3. Doctors(**doctor_id**, name, email, password, experience, dept_id, hospital_id)

4. Patients(**patient_id**, name, age, gender, password)

5. Appointments(**appointment_id**, patient_name, patient_gender, patient_email, appointment_date, hospital_id, dept_id, doctor_id, patient_id)

6. Hospital_Staff(**staff_id**, name, designation, shift, hospital_id)

7. Pharmacy(**pharmacy_id**, name, location, hospital_id)

8. Medicine(**medicine_id**, name, type, manufacturer, expiry_date, quantity, price, pharmacy_id)

9. Ambulance(**ambulance_id**, vehicle_number, type, status, location)

10. BookAmbulance(**booking_id**, patient_name, pickup_location, dropoff_location, booking_time, ambulance_id, patient_id)

11.a**dmin** (id, name, username, password, email, contact_number, created_at)

12. Health_Tips(**tip_id**, title, description, created_at)

13. **belongs_to** (dept_id, hospital_id)

14. **has** (dept_id, hospital_id)

15. **employs** (doctor_id, hospital_id,)

16. **consulted_by** (doctor_id, patient_id)

17. **booked_by** (patient_id, appointment_id)

18. **requested_by** (patient_id, appointment_id)

19. **scheduled_for** (hospital_id, appointment_id,staff_id)

20. **operates** (hospital_id, pharmacy_id)

21.**stocks** (pharmacy_id, medicine_id)

22. **uses** (patient_id, booking_id)

23.**booked_for** (ambulance_id, booking_id)

24.**created_for** (tip_id, hospital_id)

25.hospital_contact(hospital_id,contact)

26.hospital_email(hospital_id,email)

27.patient_email(patient_id,email)

28.appointment_contact(appointment_id,contact_number)

29.Hospital_Staff_contact(staff_id,contact_number)

30.Hospital_Staff_email(staff_id,email)

31.Pharmacy_contact(pharmacy_id,contact_number)

32.Pharmacy_email(pharmacy_id,email)

33.BookAmbulance_contact_number(ambulance_id,contact_number)

34.admin_email(id, email)

35.admin_contact_number(id,contact_number)

TENARY RELATIONSHIP:

scheduled_for (hospital_id, appointment_id,staff_id)

SQL DDL for the Relation Schema:

1. Appointments:

```
CREATE TABLE Appointments (
    appointment_id INT PRIMARY KEY,
    patient_name VARCHAR(100),
    patient_age INT,
    patient_gender VARCHAR(10),
    patient_email VARCHAR(100),
    contact_number VARCHAR(15),
    hospital_id INT,
    dept_id INT,
    doctor_id INT,
    appointment_date DATE,
    patient_id INT,
    FOREIGN KEY (hospital_id) REFERENCES Hospitals(hospital_id),
    FOREIGN KEY (dept_id) REFERENCES Departments(dept_id),
    FOREIGN KEY (doctor_id) REFERENCES Doctors(doctor_id),
    FOREIGN KEY (patient_id) REFERENCES Patients(patient_id)
);
```

2. Departments

```
CREATE TABLE Departments (
    dept_id INT PRIMARY KEY,
    name VARCHAR(100),
    hospital_id INT,
    FOREIGN KEY (hospital_id) REFERENCES Hospitals(hospital_id)
    FOREIGN KEY (doctor_id) REFERENCES Doctors(doctor_id)
);
```

3. Doctors

```
CREATE TABLE Doctors (
    doctor_id INT PRIMARY KEY,
    name VARCHAR(100),
```

```
    email VARCHAR(100),
    password VARCHAR(100),
    experience INT,
    department_id INT,
    hospital_id INT,
    FOREIGN KEY (department_id) REFERENCES Departments(dept_id),
    FOREIGN KEY (hospital_id) REFERENCES Hospitals(hospital_id)
);
```

4. Medicine

```
CREATE TABLE Medicine (
    medicine_id INT PRIMARY KEY,
    pharmacy_id INT,
    name VARCHAR(100),
    type VARCHAR(50),
    manufacturer VARCHAR(100),
    purpose VARCHAR(100),
    expiry_date DATE,
    quantity INT,
    price INT,
    FOREIGN KEY (pharmacy_id) REFERENCES Pharmacy(pharmacy_id)
);
```

5. Hospital Staff

```
CREATE TABLE Hospital_Staff (
    staff_id INT PRIMARY KEY,
    name VARCHAR(100),
    designation VARCHAR(50),
    contact_number VARCHAR(15),
    email VARCHAR(100),
    hospital_id INT,
    Shift INT;
    FOREIGN KEY (hospital_id) REFERENCES Hospitals(hospital_id)
);
```

6. Pharmacy

```
CREATE TABLE Pharmacy (
    pharmacy_id INT PRIMARY KEY,
    name VARCHAR(100),
```

```
    location VARCHAR(200),
    contact_number VARCHAR(15),
    email VARCHAR(100),
    hospital_id INT UNIQUE,
    FOREIGN KEY (hospital_id) REFERENCES Hospitals(hospital_id)
);
```

7. Hospitals

```
CREATE TABLE Hospitals (
    hospital_id INT PRIMARY KEY,
    name VARCHAR(100),
    location VARCHAR(200),
    contact VARCHAR(15),
    email VARCHAR(100)
);
```

8. Patients

```
CREATE TABLE Patients (
    patient_id INT PRIMARY KEY,
    name VARCHAR(100),
    email VARCHAR(100),
    password VARCHAR(100),
    gender VARCHAR(10),
    age INT,
    contact VARCHAR(15)
);
```

9. Ambulance

```
CREATE TABLE Ambulance (
    ambulance_id INT PRIMARY KEY,
    vehicle_number VARCHAR(20),
    type VARCHAR(50),
    status VARCHAR(20),
    location VARCHAR(200)
);
```

10. BookAmbulance

```
CREATE TABLE BookAmbulance (
    booking_id INT PRIMARY KEY,
    patient_name VARCHAR(100),
```

```
    contact_number VARCHAR(15),
    pickup_location VARCHAR(200),
    dropoff_location VARCHAR(200),
    booking_time TIMESTAMP,
    ambulance_id INT,
    patient_id INT,
    FOREIGN KEY (ambulance_id) REFERENCES Ambulance(ambulance_id),
    FOREIGN KEY (patient_id) REFERENCES Patients(patient_id)
);
```

11. Health Tips

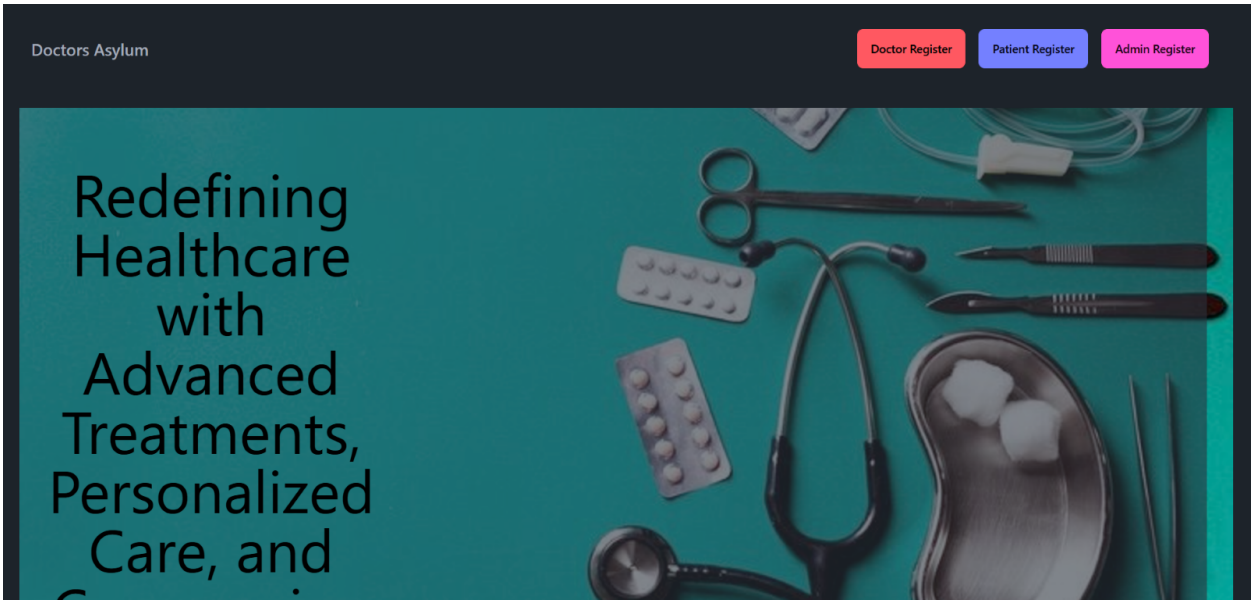
```
CREATE TABLE Health_Tips (
    tip_id INT PRIMARY KEY,
    title VARCHAR(200),
    description TEXT,
    created_at TIMESTAMP
);
```

12. Admin

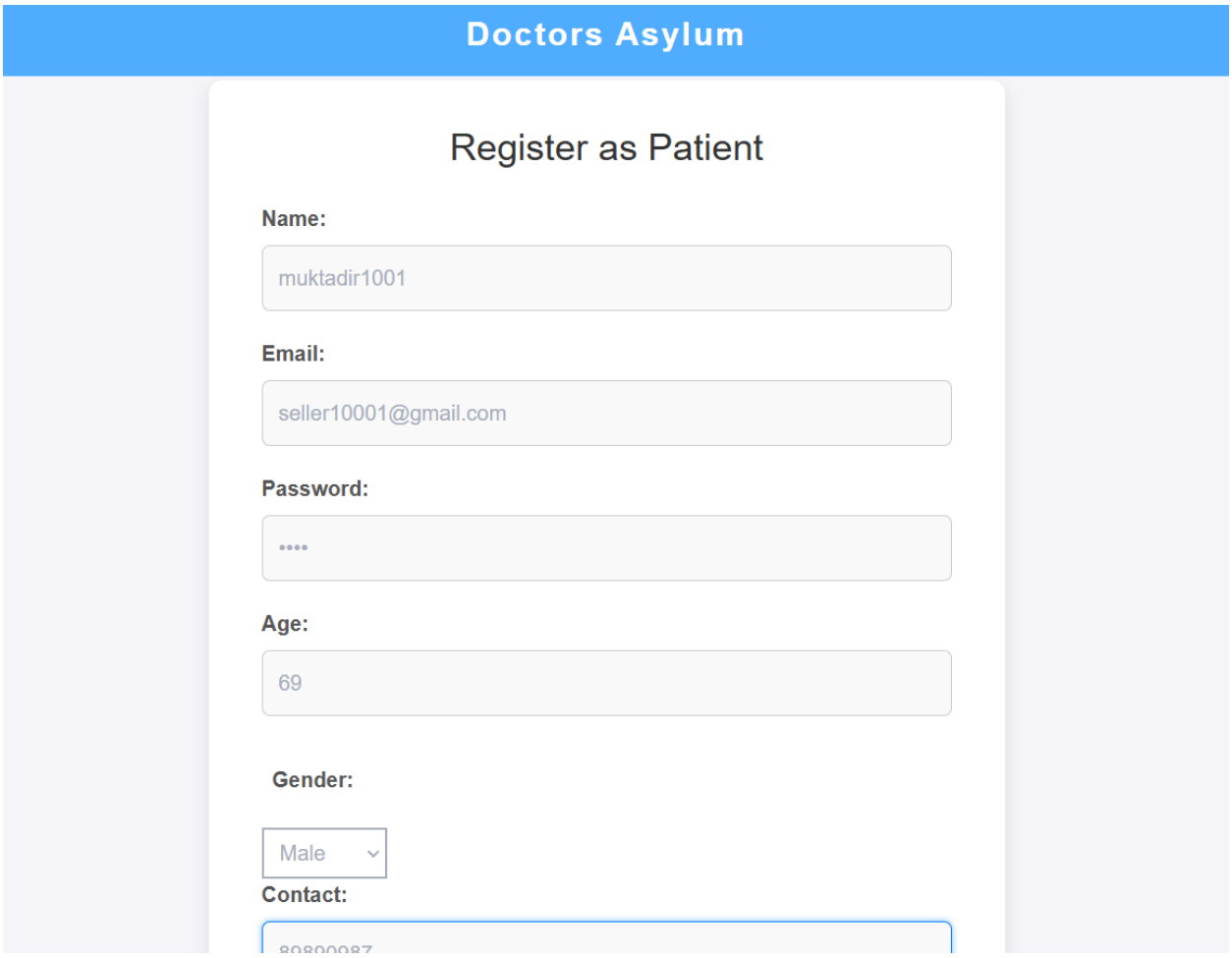
```
CREATE TABLE Admin (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    username VARCHAR(255) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
    contact_number VARCHAR(15),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  );
```

screenshots :

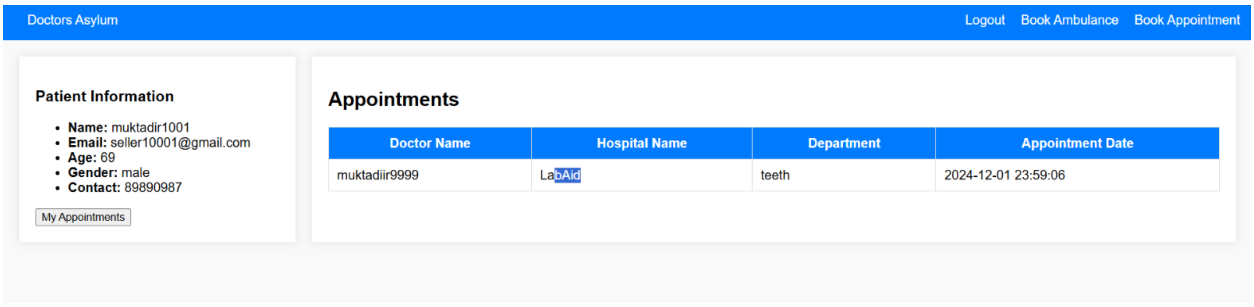
Front Page-



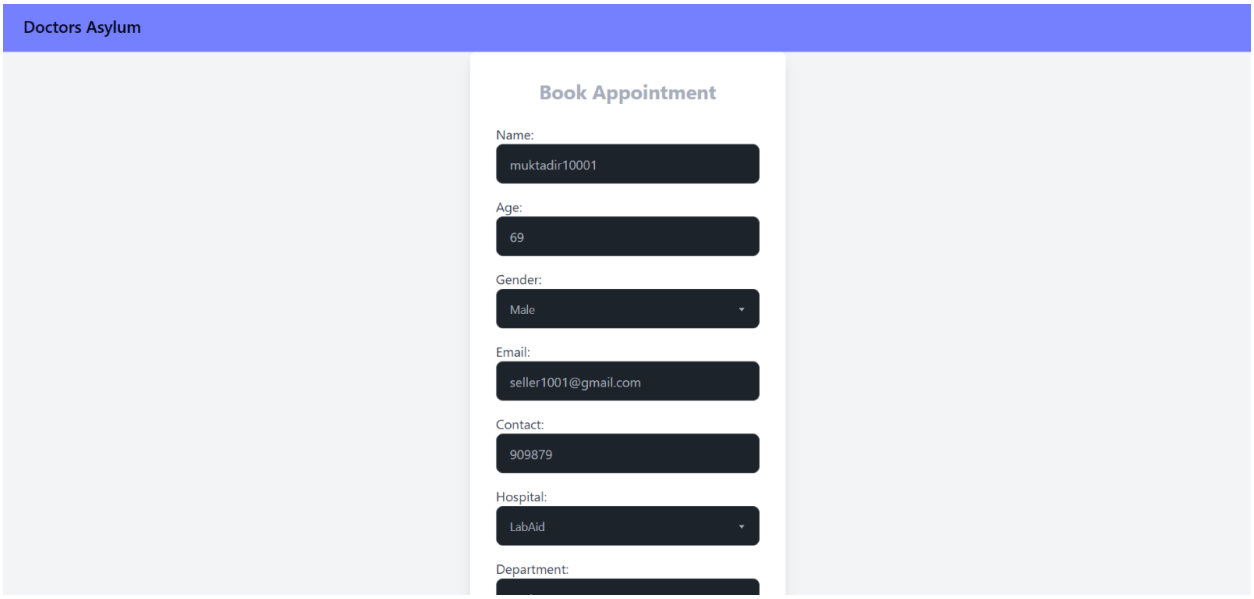
Patient Register Page-



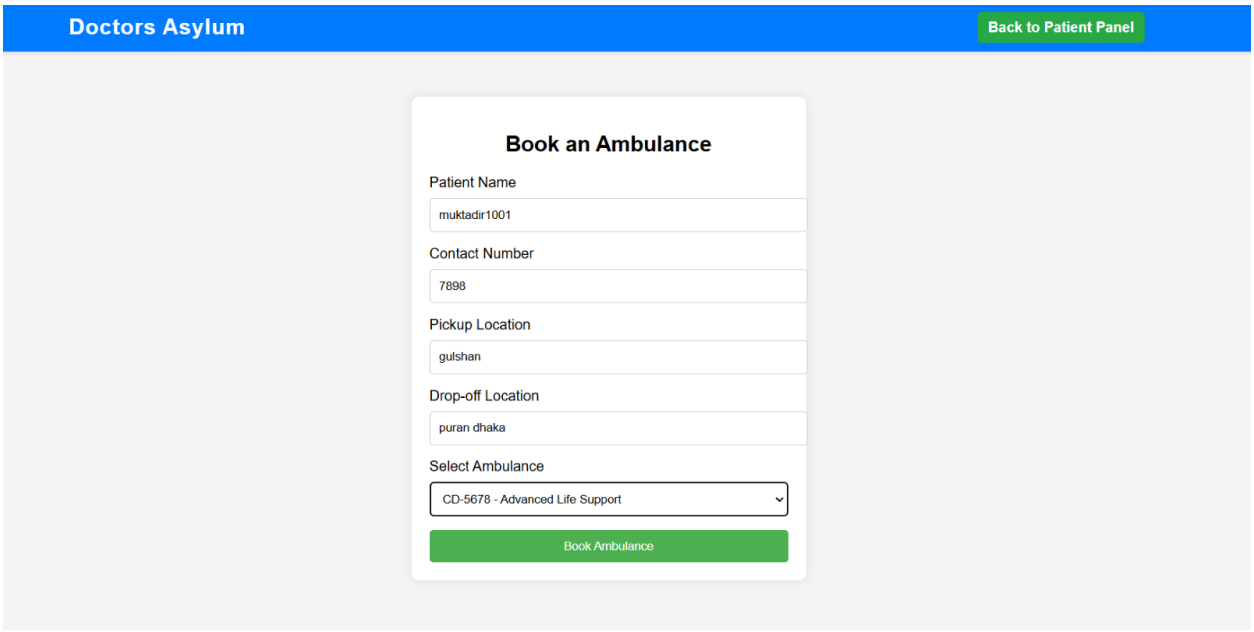
After filing up the patient register form the patient information will shown-



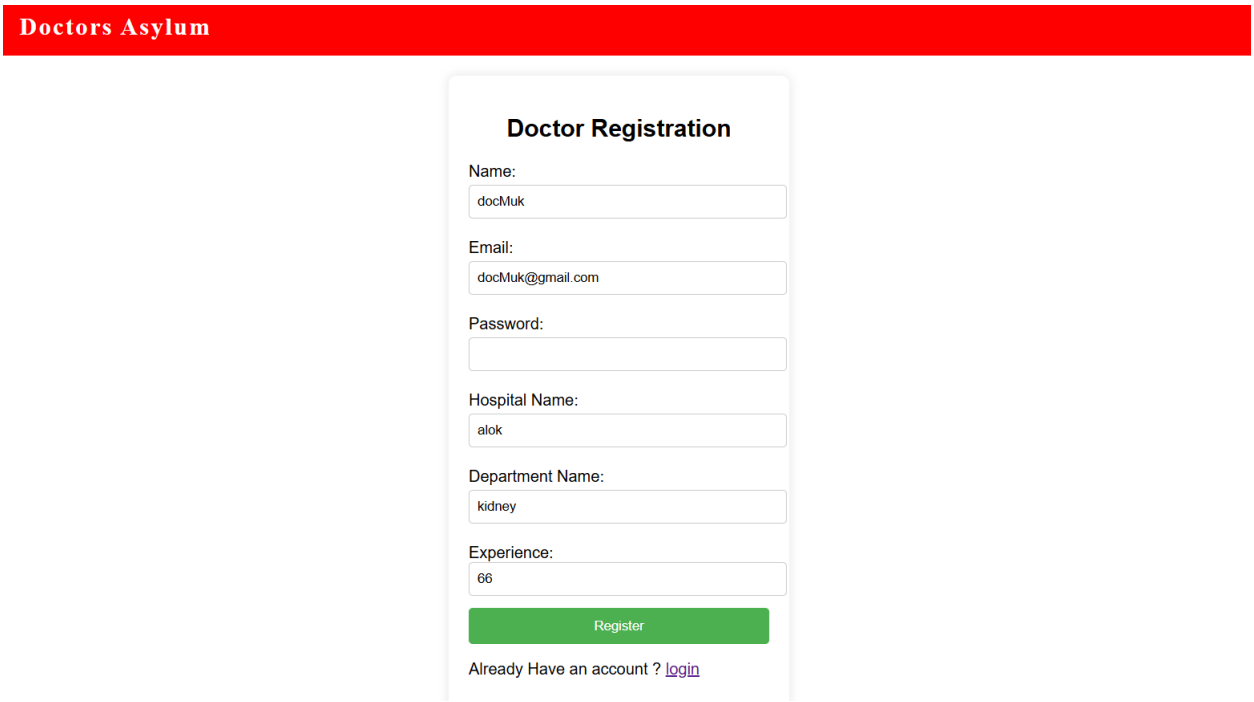
From patient information page we can go to appointment page using book appointmentment button-



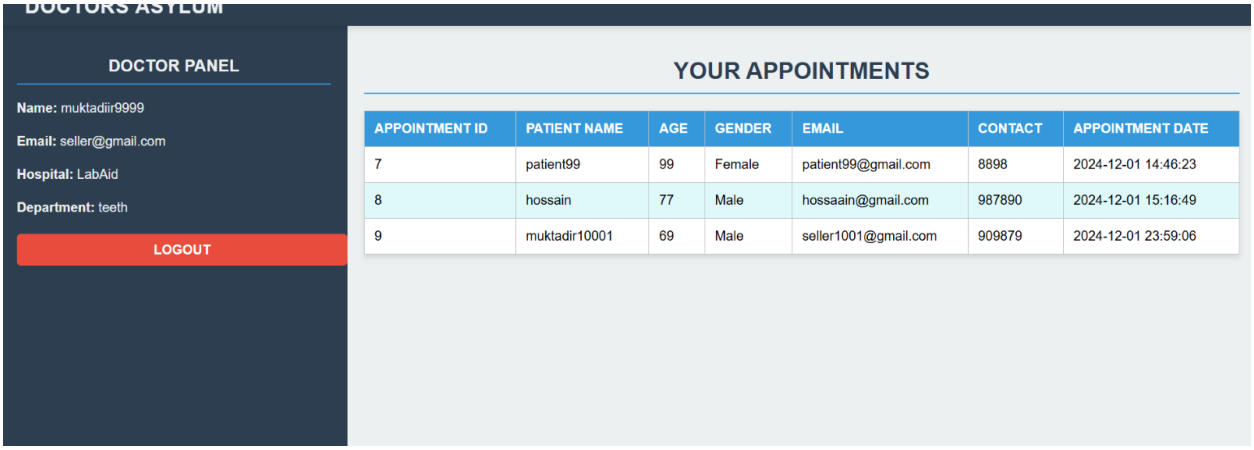
We can also go for ambulance booking-



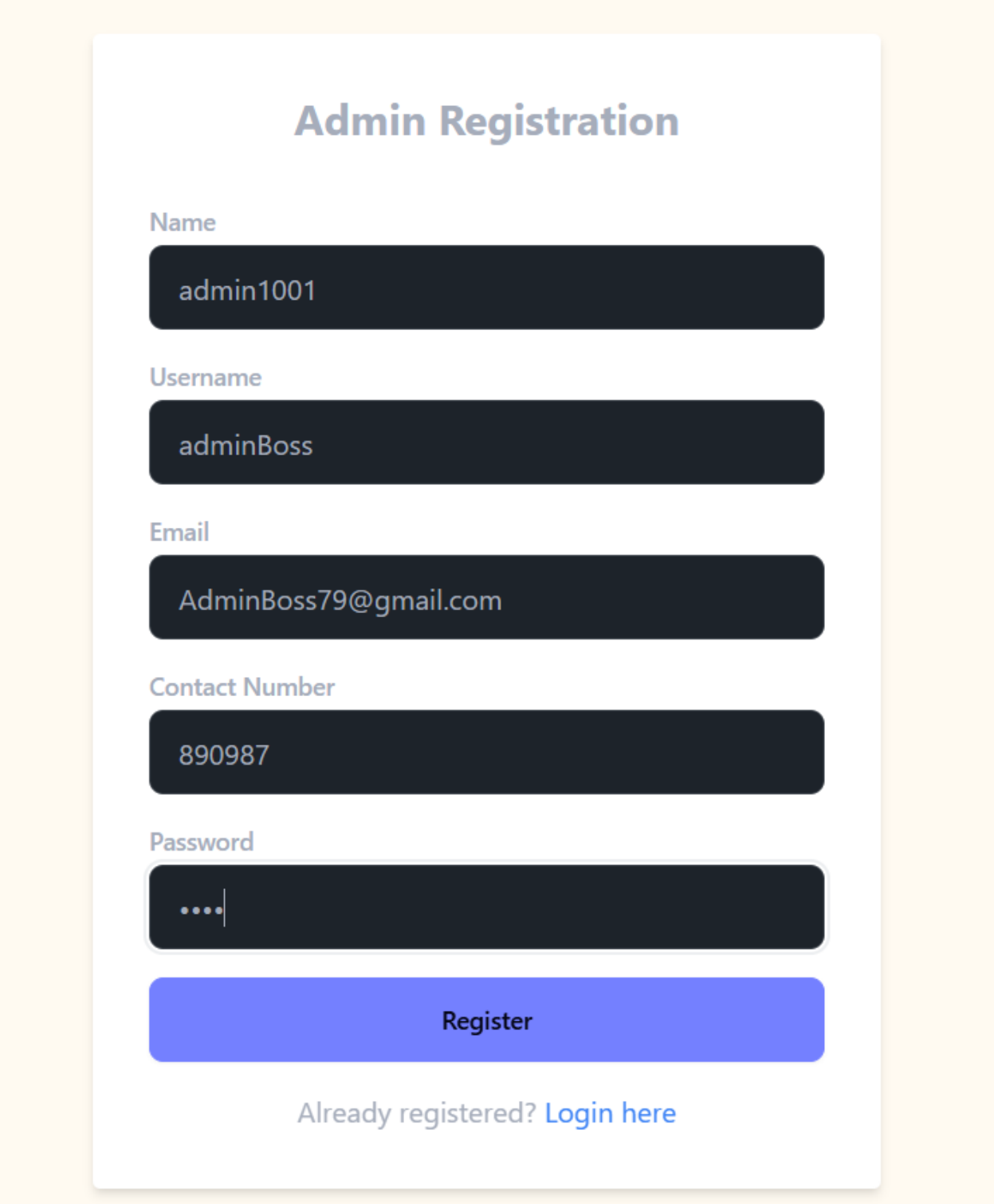
We have doctor registration page -



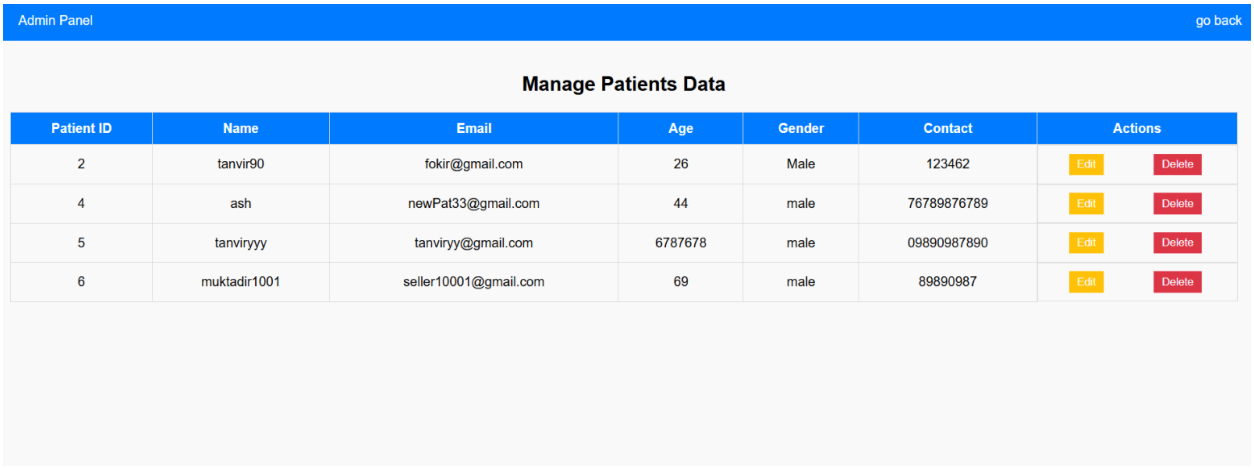
We can see doctor panel with appointments-

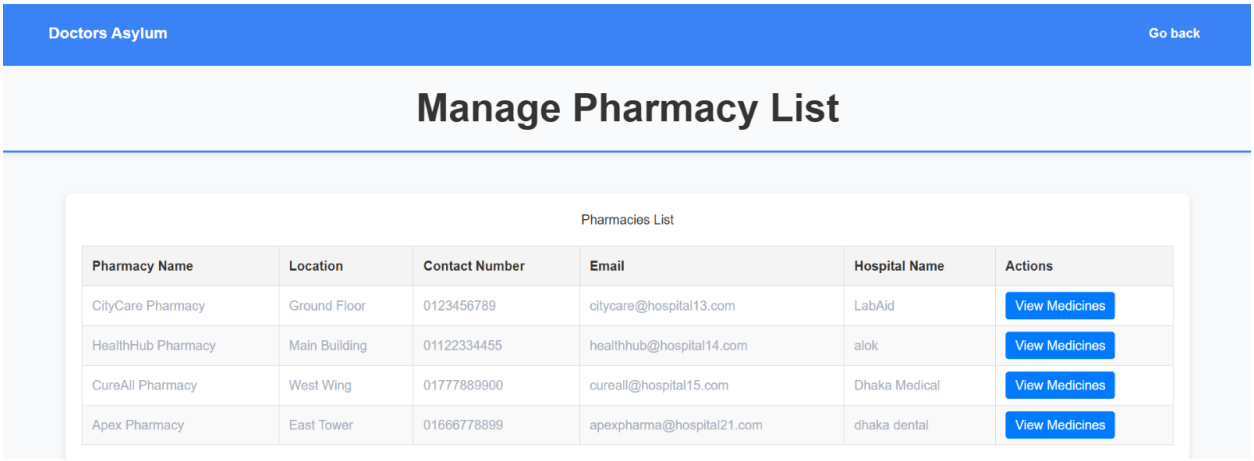


After that we can go to admin registration page-

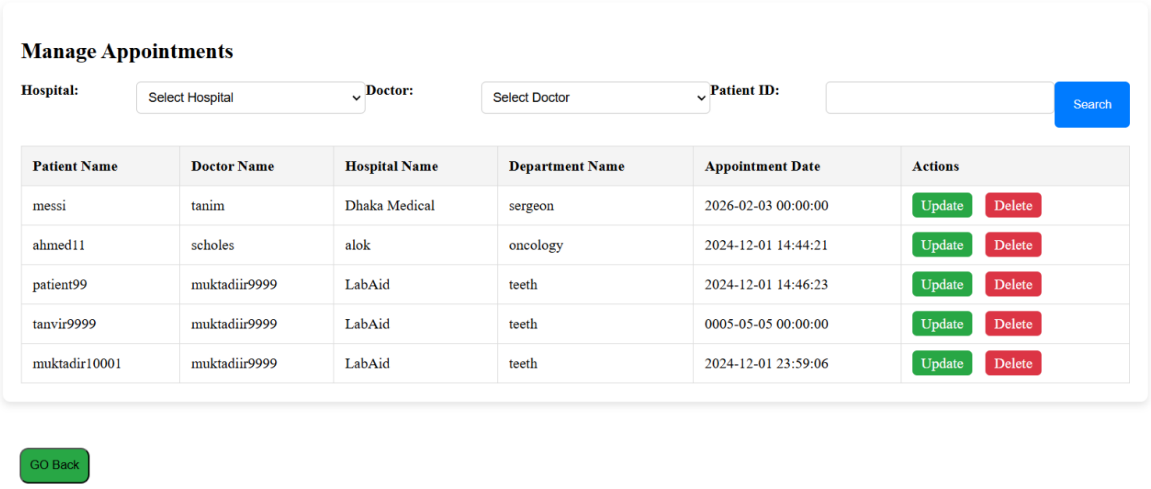


Admin panel manage all of the management system-

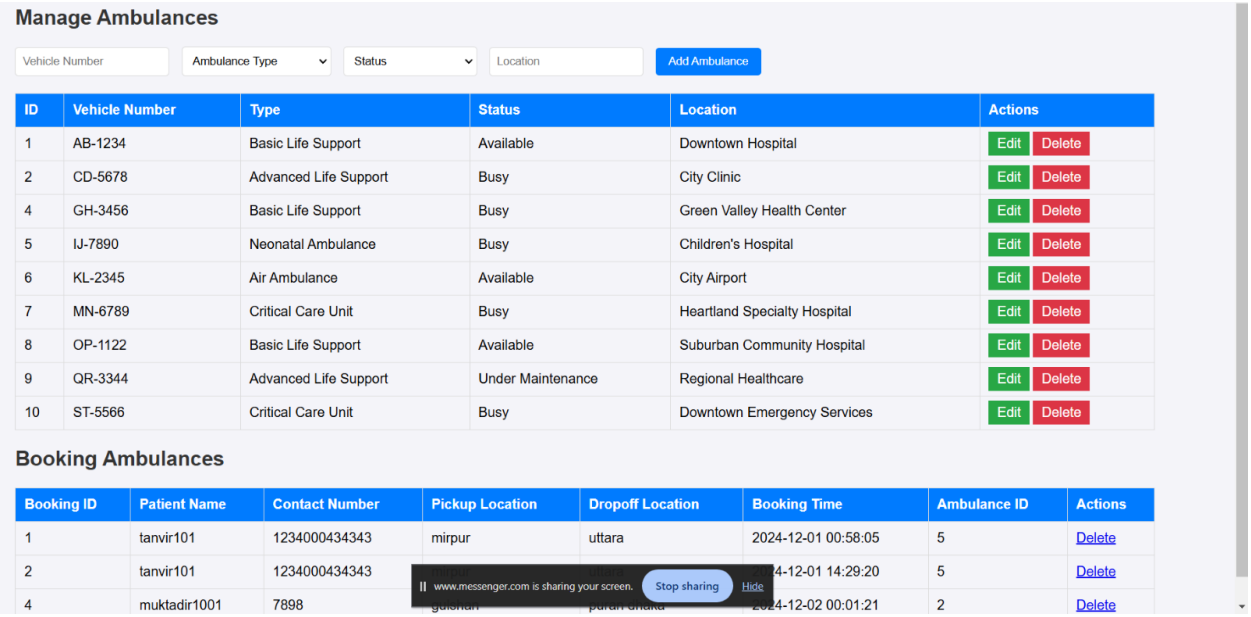




Admin can also mange appointments-



They can mange ambulances-



Manage Ambulances

Vehicle Number | Ambulance Type | Status | Location | Add Ambulance

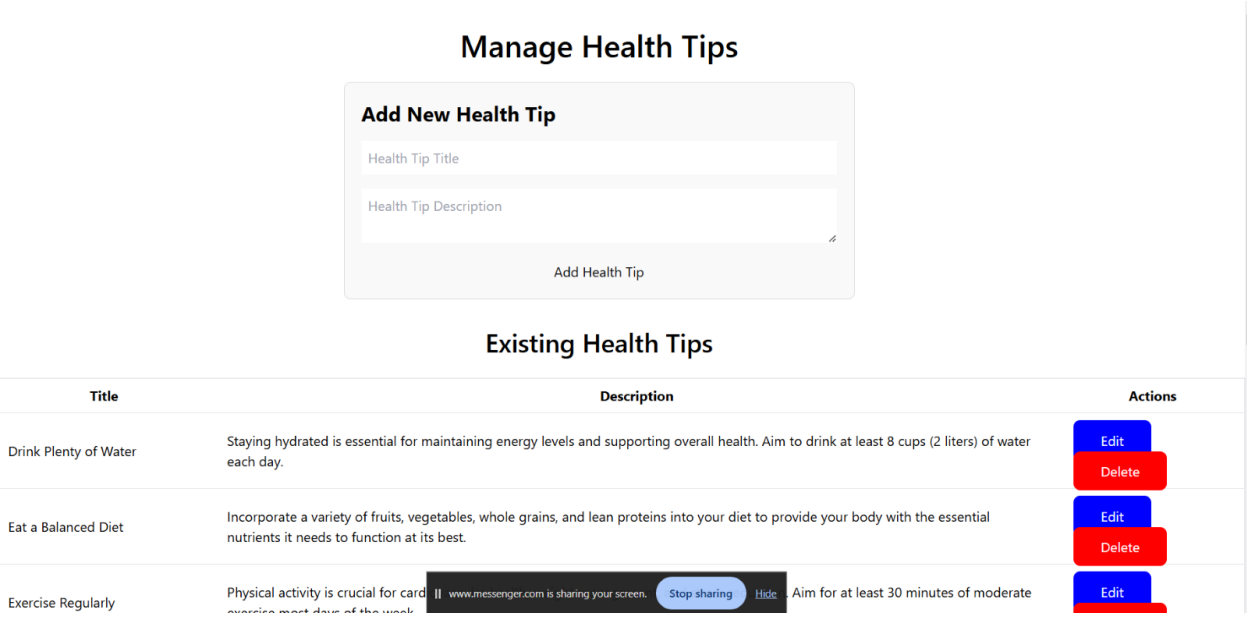
ID	Vehicle Number	Type	Status	Location	Actions
1	AB-1234	Basic Life Support	Available	Downtown Hospital	Edit Delete
2	CD-5678	Advanced Life Support	Busy	City Clinic	Edit Delete
4	GH-3456	Basic Life Support	Busy	Green Valley Health Center	Edit Delete
5	IJ-7890	Neonatal Ambulance	Busy	Children's Hospital	Edit Delete
6	KL-2345	Air Ambulance	Available	City Airport	Edit Delete
7	MN-6789	Critical Care Unit	Busy	Heartland Specialty Hospital	Edit Delete
8	OP-1122	Basic Life Support	Available	Suburban Community Hospital	Edit Delete
9	QR-3344	Advanced Life Support	Under Maintenance	Regional Healthcare	Edit Delete
10	ST-5566	Critical Care Unit	Busy	Downtown Emergency Services	Edit Delete

Booking Ambulances

Booking ID	Patient Name	Contact Number	Pickup Location	Dropoff Location	Booking Time	Ambulance ID	Actions
1	tanvir101	1234000434343	mirpur	uttara	2024-12-01 00:58:05	5	Delete
2	tanvir101	1234000434343			4-12-01 14:29:20	5	Delete
4	muktadir1001	7898			2024-12-02 00:01:21	2	Delete

www.messenger.com is sharing your screen. Stop sharing Hide

They can mange health tips-



Manage Health Tips

Add New Health Tip

Health Tip Title

Health Tip Description

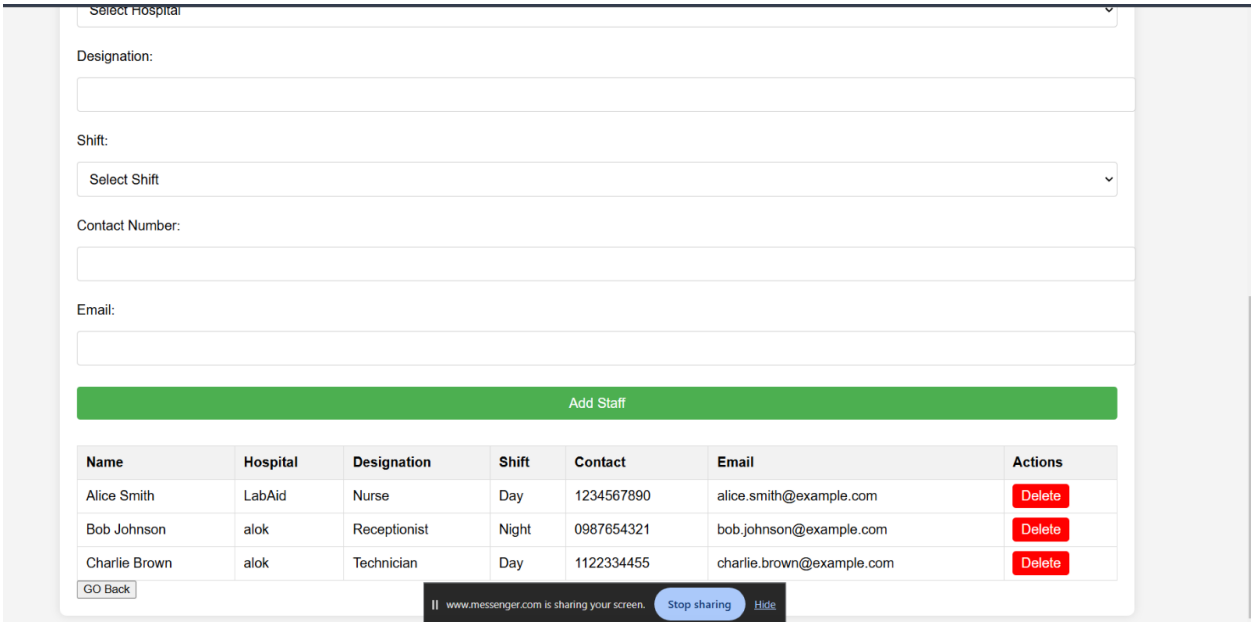
Add Health Tip

Existing Health Tips

Title	Description	Actions
Drink Plenty of Water	Staying hydrated is essential for maintaining energy levels and supporting overall health. Aim to drink at least 8 cups (2 liters) of water each day.	Edit Delete
Eat a Balanced Diet	Incorporate a variety of fruits, vegetables, whole grains, and lean proteins into your diet to provide your body with the essential nutrients it needs to function at its best.	Edit Delete
Exercise Regularly	Physical activity is crucial for card	Aim for at least 30 minutes of moderate	Edit

www.messenger.com is sharing your screen. Stop sharing Hide

They can add or remove staffs-



Conclusion:

The DG Health Portal Management System is a powerful tool for healthcare management, providing seamless coordination between patients, hospitals, and healthcare services. It ensures accurate and secure data management while maintaining scalability for future enhancements.