



Inspiring Excellence

CSE370 : Database Systems
Project Report
Project Title : Hospital Management System

Group No : 2, CSE370 Lab Section : 2, Fall 2023		
ID	Name	Contribution
21201080	MD. ABRAR RAHMAN SHAFIN	Backend and frontend for everything related to home page, login, signup and admin profile
21301661	MEHRUN CHOWDHURY DOLA	Patient request for Appointment and html for it
23341111	FARDIN HASSAN TAMIM	Frontend and backend for everything related to patient Profile.
21301142	AL-AMIN SAGOR	
20101466	MD. GOLAM SHAHNEWAZ	Doctors viewing medical records

Table of Contents

Section No	Content	Page No
1	Introduction	3
2	Project Features	4
3	ER/EER Diagram	5
4	Schema Diagram	6
5	Frontend Development	7-55
6	Backend Development	56-84
7	Conclusion	85
8	References	85

Introduction

Welcome to our Healthcare Management System (HMS), a robust platform designed to streamline and enhance healthcare services. Our system is meticulously crafted to meet the needs of medical practitioners, staff, and patients alike.

Within our HMS, we've structured a comprehensive framework to manage user profiles efficiently. Patients and doctors have dedicated profiles, each equipped with essential information ensuring a personalized and secure experience. Appointments can be conveniently scheduled, tracked, and managed, ensuring smooth interactions between patients and doctors. Financial aspects are also streamlined through our billing system, allowing transparent management of payments and ensuring a hassle-free experience for patients. Wards and bed allocations have been meticulously organized, providing a structured environment for patient admissions and ensuring optimal resource utilization. In essence, our Healthcare Management System embodies efficiency, precision, and care, aiming to elevate healthcare experiences for both providers and recipients.

Project Features

Admin Privileges: The admin account holds the authority to add, remove, edit, and view patient, doctor, and other entity records within the system.

Role-Based Actions: Doctor accounts can access patient profiles, manage appointments, while patient accounts can request for appointments.

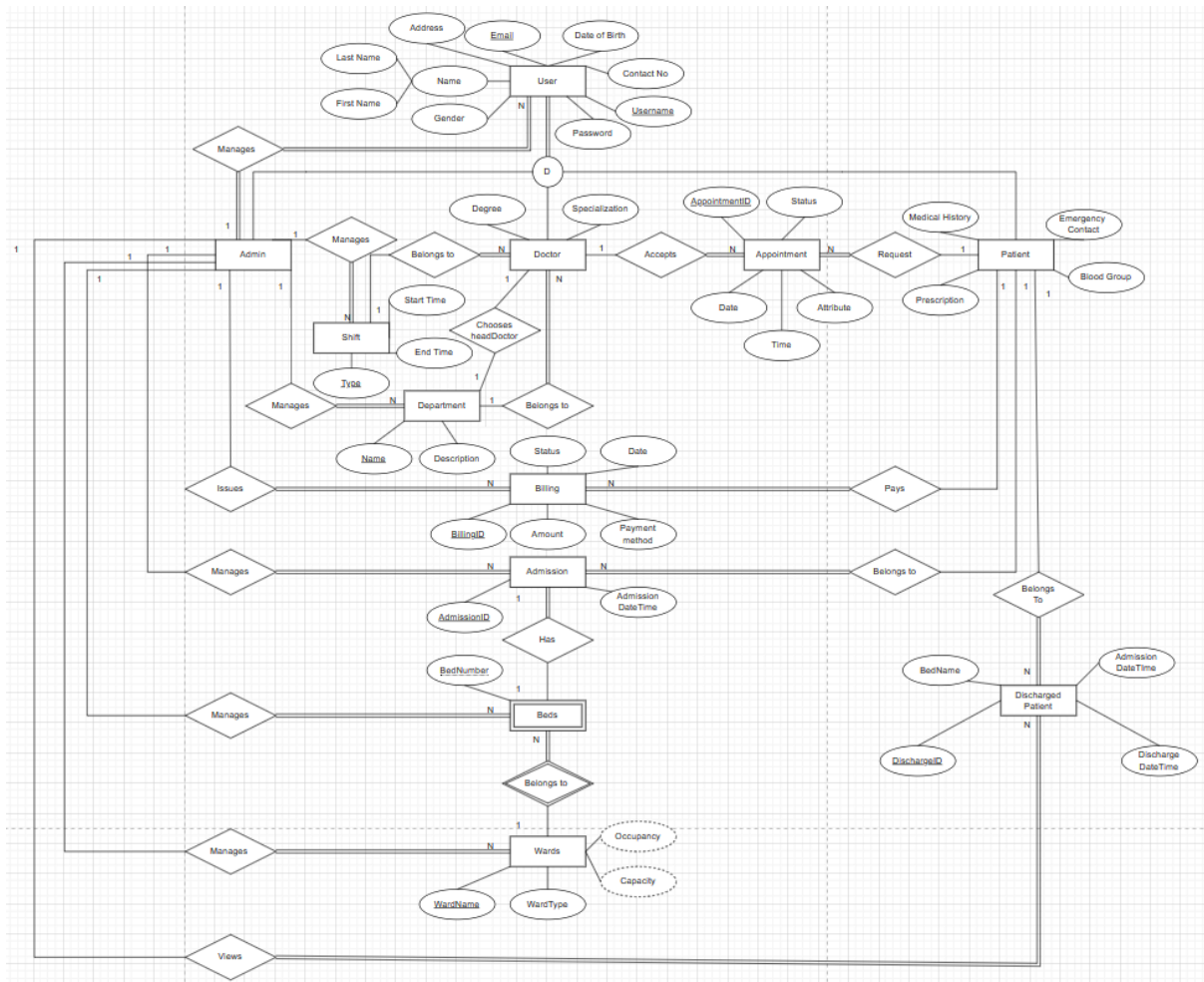
Appointment Management: Users can accept, reject, or schedule appointments, with the system handling time clashes.

Ward Occupancy Control: Real-time occupancy tracking prevents exceeding ward capacities.

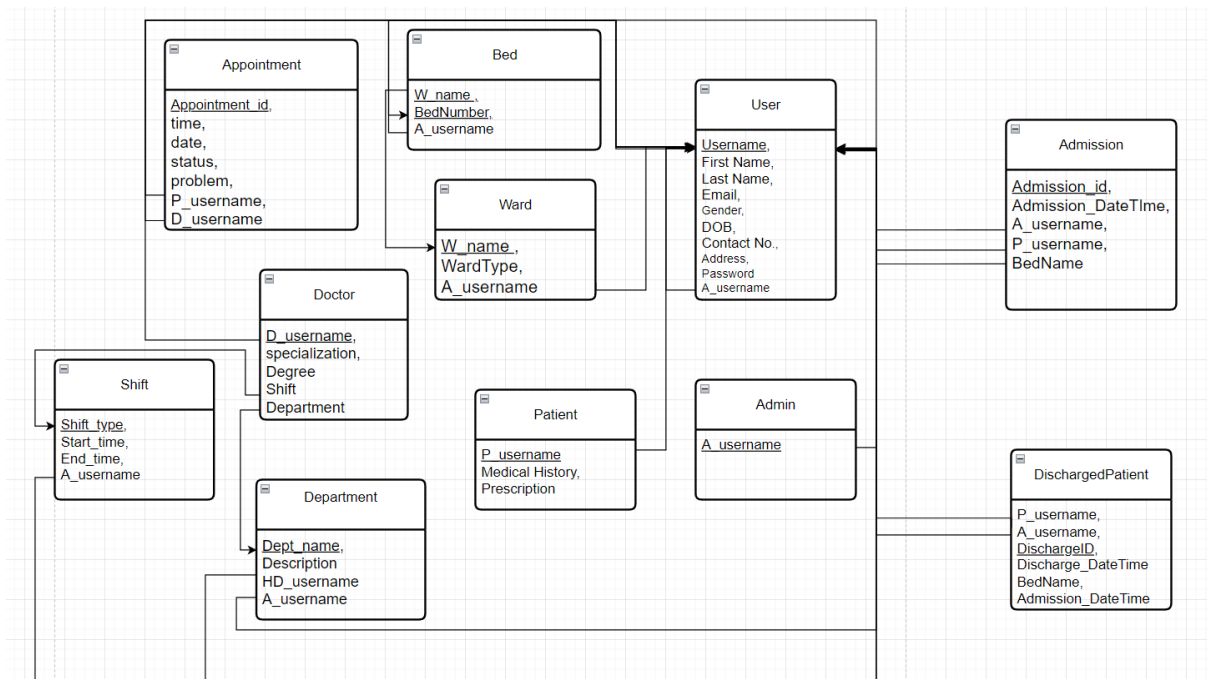
Search, Filter and Sorting Functionality: Users can search, sort and filter patients, doctors, and other entities based on criteria such as department, name, or ID for ease of access.

Discharged Patient Tracking: The system keeps track of discharged patients, maintaining records of discharge dates and details.

ER/EER Diagram



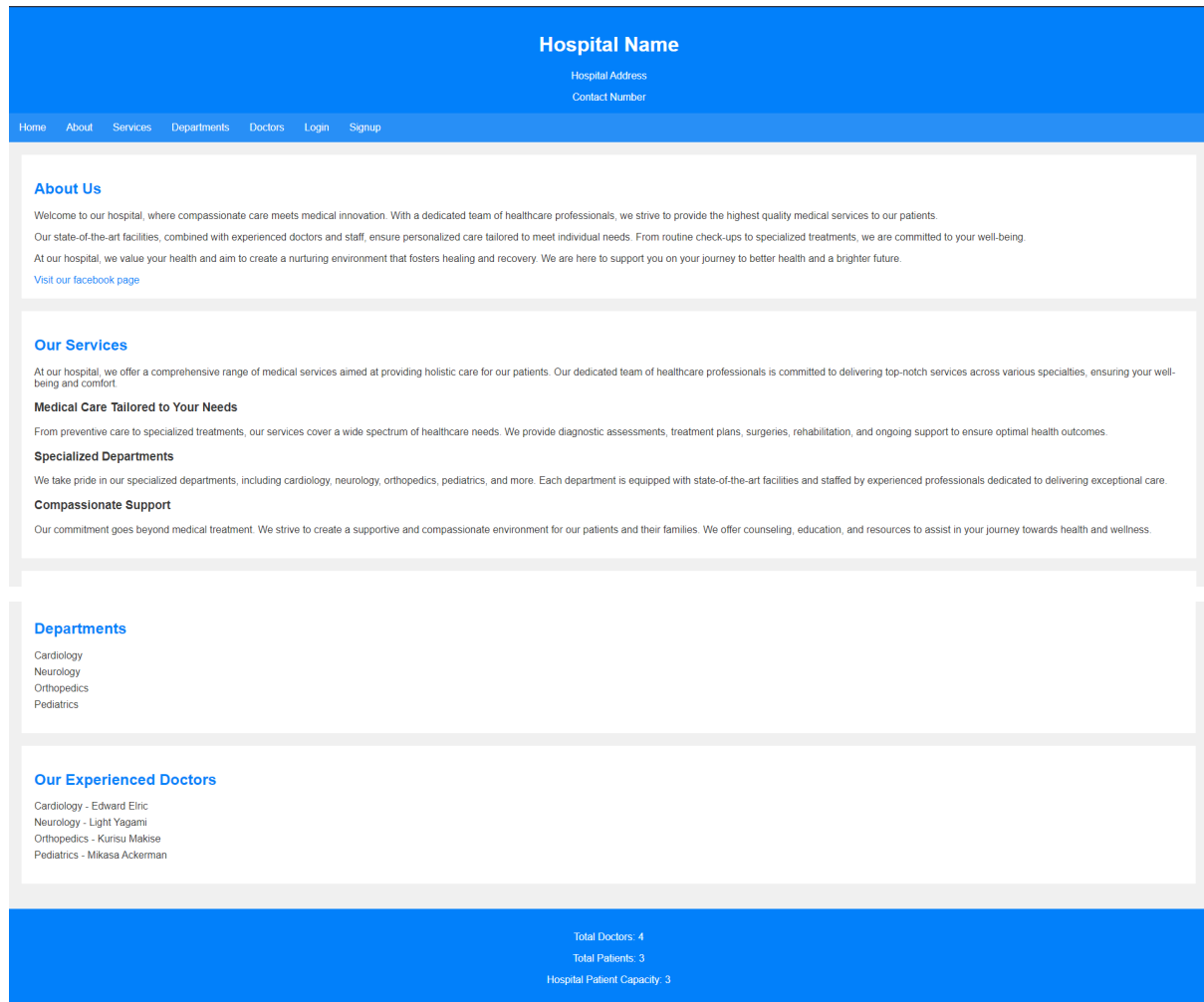
Schema Diagram



Frontend Development

Contribution of ID : 21201080, Name : Md. Abrar Rahman Shafin

Home Page: This page can be viewed by everyone



CSS for Home(Department, Doctor) :

```
body {
    font-family: 'Roboto', sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f0f0f0;
    color: #333;
}

header {
    background-color: #007bff;
    color: #fff;
    text-align: center;
    padding: 20px 0 0 0;
}

nav ul {
```

```

    list-style-type: none;
    margin: 0;
    padding: 0;
    background-color: #268cfa;
    overflow: hidden;
    text-align: right;
}
nav li {
    float: left;
}
nav li a {
    display: block;
    color: #fff;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
    transition: background-color 0.3s;
}
nav li a:hover {
    background-color: #0056b3;
}
main {
    padding: 20px;
}
.about-us {
    background-color: #fff;
    padding: 20px;
    margin-bottom: 20px;
}
.about-us h2 {
    color: #007bff;
}
.services {
    background-color: #fff;
    padding: 20px;
    margin-bottom: 20px;
}
.services h2 {
    color: #007bff;
}
.about-us a {
    color: #007bff;
    text-decoration: none;
    transition: color 0.3s;
}
.about-us a:hover {
    color: #0056b3;
}
.departments {
    background-color: #fff;
    padding: 20px;
    margin-bottom: 20px;
}

```



```

.departments h2 {
    color: #007bff;
}

.departments ul {
    list-style: none;
    padding: 0;
}

.departments li {
    margin-bottom: 8px;
}

.head-doctors {
    background-color: #fff;
    padding: 20px;
    margin-bottom: 20px;
}

.head-doctors h2 {
    color: #007bff;
}

.head-doctors ul {
    list-style: none;
    padding: 0;
}

.head-doctors li {
    margin-bottom: 8px;
}

footer {
    background-color: #007bff;
    color: #fff;
    text-align: center;
    padding: 20px 0;
}

footer ul {
    list-style: none;
    padding: 0;
}

footer ul li {
    margin-bottom: 8px;
}

table {
    width: 100%;
    border-collapse: collapse;
}

th {
    background-color: #007bff;
    color: white;
    font-weight: bold;
    text-align: left;
    padding: 12px;
}

td {
    border-bottom: 1px solid #ddd;
    padding: 12px;
}

```

```
tr:hover {
    background-color: #f5f5f5;
```

HTML:

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Hospital</title>
    <link rel="stylesheet" href="{% static 'css/home.css' %}">
</head>
<body>
    <header>
        <div class="hospital-info">
            <h1>Hospital Name</h1>
            <p>Hospital Address</p>
            <p>Contact Number</p>
        </div>

        <nav>
            <ul>
                <li><a href="#home">Home</a></li>
                <li><a href="#about">About</a></li>
                <li><a href="#services">Services</a></li>
                <li><a href="{% url 'department_user_view' %}">Departments</a></li>
                <li><a href="{% url 'doctor_user_view' %}">Doctors</a></li>
                <li><a href="{% url 'login' %}">Login</a></li>
                <li><a href="{% url 'signup' %}">Signup</a></li>
            </ul>
        </nav>
    </header>

    <main>
        <section class="about-us" id="about">
            <h2>About Us</h2>
            <p>Welcome to our hospital, where compassionate care meets medical innovation. With a dedicated team of healthcare professionals, we strive to provide the highest quality medical services to our patients.</p>
            <p>Our state-of-the-art facilities, combined with experienced doctors and staff, ensure personalized care tailored to meet individual needs. From routine check-ups to specialized treatments, we are committed to your well-being.</p>
            <p>At our hospital, we value your health and aim to create a nurturing environment that fosters healing and recovery. We are here to support you on your journey to better health and a brighter future.</p>
            <a href="https://www.facebook.com/">Visit our facebook page</a>
        </section>

        <section class="services" id="services">
            <h2>Our Services</h2>
```

```
<p>At our hospital, we offer a comprehensive range of medical services aimed at providing holistic care for our patients. Our dedicated team of healthcare professionals is committed to delivering top-notch services across various specialties, ensuring your well-being and comfort.</p>
```

```
<h3>Medical Care Tailored to Your Needs</h3>
```

```
<p>From preventive care to specialized treatments, our services cover a wide spectrum of healthcare needs. We provide diagnostic assessments, treatment plans, surgeries, rehabilitation, and ongoing support to ensure optimal health outcomes.</p>
```

```
<h3>Specialized Departments</h3>
```

```
<p>We take pride in our specialized departments, including cardiology, neurology, orthopedics, pediatrics, and more. Each department is equipped with state-of-the-art facilities and staffed by experienced professionals dedicated to delivering exceptional care.</p>
```

```
<h3>Compassionate Support</h3>
```

```
<p>Our commitment goes beyond medical treatment. We strive to create a supportive and compassionate environment for our patients and their families. We offer counseling, education, and resources to assist in your journey towards health and wellness.</p>
```

```
</section>
```

```
<section class="departments" id="departments">
```

```
<h2>Departments</h2>
```

```
<ul>
```

```
  {% for department in departments %}
```

```
    <li>{{ department.get_name_display }}</li>
```

```
  {% endfor %}
```

```
</ul>
```

```
</section>
```

```
<section class="head-doctors" id="doctors">
```

```
<h2>Our Experienced Doctors</h2>
```

```
<ul>
```

```
  {% for department in departments %}
```

```
    <li>{{ department.get_name_display }} -
```

```
      {% if department.head_doctor %}
```

```
        {{ department.head_doctor.user_profile.get_full_name }}
```

```
      {% else %}
```

```
        No head doctor assigned
```

```
      {% endif %}
```

```
    </li>
```

```
  {% endfor %}
```

```
</ul>
```

```
</section>
```

```
</main>
```

```
<footer>
```

```
<p>Total Doctors: {{ approved_doctors_count }}</p>
```

```
<p>Total Patients: {{ patients_count }}</p>
```

```

    <p>Hospital Patient Capacity: {{ total_beds }}</p>
  </footer>
</body>
</html>

```

Department page: This page can be viewed by everyone

Hospital Name

Hospital Address
Contact Number

Home
Departments
Doctors
Login
Signup

All Departments

Cardiology

Cardiology is the study of the heart. Cardiology is a branch of medicine that deals with disorders of the heart and the cardiovascular system. The field includes medical diagnosis and treatment of congenital heart defects, coronary artery disease, heart failure, valvular heart disease, and electrophysiology.

Head Doctor: Edward Elic

Total Doctors: 1

Neurology

Neurology Neurology is the branch of medicine dealing with the diagnosis and treatment of all categories of conditions and disease involving the nervous system, which comprises the brain, the spinal cord and the peripheral nerves.

Head Doctor: Light Yagami

Total Doctors: 1

Orthopedics

Orthopedic surgery or orthopedics is the branch of surgery concerned with conditions involving the musculoskeletal system. Orthopedic surgeons use both surgical and nonsurgical means to treat musculoskeletal trauma, spine diseases, sports injuries, degenerative diseases, infections, tumors, and congenital disorders.

Head Doctor: Kurisu Makise

Total Doctors: 1

Pediatrics

Pediatrics is the branch of medicine dealing with the health and medical care of infants, children, and adolescents from birth up to the age of 18. The word "paediatrics" means "healer of children"

Head Doctor: Mikasa Ackerman

Total Doctors: 1

HTML:

```

<main>
  <section class="departments">
    <h2>All Departments</h2>
    <ul>
      {% for data in department_data %}
        <li>
          <h3>{{ data.department.get_name_display }}</h3>
          <p>{{ data.department.description }}</p>
          <p>Head Doctor:
            {% if data.department.head_doctor %}
              {{ data.department.head_doctor.user_profile.get_full_name }}
            {% else %}
              No head doctor assigned
            {% endif %}
          </p>
          <p>Total Doctors: {{ data.approved_doctors_count }}</p>
        </li>
      {% endfor %}
    </ul>
  </section>/.departments
</main>

```

Doctor page: This page can be viewed by everyone

Hospital Name			
Hospital Address			
Contact Number			
Home	Departments	Doctors	Login Signup
Search by Name: <input type="text"/>			
Department: <input type="text" value="Select department"/>			
Search/Filter Cancel Filter			
Name	Specialization	Degree	Department
Edward Elinc	Cardiologist	MBBS, MS	Cardiology Department
Light Yagami	Neurologist	FCPS	Neurology Department
Kurusu Makise	Oncologist	MBBS, FCPS	Orthopedics Department
Mikasa Ackerman	Pediatrician	MBBS, DMRT, FCPS	Pediatrics Department

HTML:

```
<table>

  <thead>
    <tr>
      <th>Name</th>
      <th>Specialization</th>
      <th>Degree</th>
      <th>Department</th>
    </tr>
  </thead>
  <tbody>
    {% for doctor in all_doctors %}
    <tr>
      <td>{{ doctor.user_profile.full_name }}</td>
      <td>{{ doctor.specialization }}</td>
      <td>{{ doctor.qualifications }}</td>
      <td>{{ doctor.department }}</td>
    </tr>
    {% empty %}
    <tr>
      <td>No registered doctors available.</td>
    </tr>
    {% endfor %}
  </tbody>
</table>
```

User Sign up Page:

User Signup

First Name:

Last Name:

Gender:

Address:

Email:

Date of Birth:

Contact Number:

Username:

Password:

Confirm Password:

Account Type:

Next

[Back to Home](#)

HTML:

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>User Signup</title>
  <link rel="stylesheet" href="{% static 'css/signup.css' %}">
</head>
<body>
  <div class="signup-form">
    <h2>User Signup</h2>
    {% if error %}
      <p>Error: {{ error }}</p>
    {% endif %}
    <form method="post">
      {% csrf_token %}
      <div class="form-group">
        <label for="first_name">First Name:</label>
        <input type="text" id="first_name" name="first_name" required>
      </div>
      <div class="form-group">
```

```

        <label for="last_name">Last Name:</label>
        <input type="text" id="last_name" name="last_name" required>
    </div>
    <div class="form-group">
        <label for="gender">Gender:</label>
        <select id="gender" name="gender" required>
            <option value="M">Male</option>
            <option value="F">Female</option>
            <option value="O">Other</option>
        </select>
    </div>
    <div class="form-group">
        <label for="address">Address:</label>
        <input type="text" id="address" name="address" required>
    </div>
    <div class="form-group">
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>
    </div>
    <div class="form-group">
        <label for="dob">Date of Birth:</label>
        <input type="date" id="dob" name="dob" required>
    </div>
    <div class="form-group">
        <label for="contact_no">Contact Number:</label>
        <input type="text" id="contact_no" name="contact_no" required>
    </div>
    <div class="form-group">
        <label for="username">Username:</label>
        <input type="text" id="username" name="username" required>
    </div>
    <div class="form-group">
        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required>
    </div>
    <div class="form-group">
        <label for="confirm_password">Confirm Password:</label>
        <input type="password" id="confirm_password" name="confirm_password" required>
    </div>
    <div class="form-group">
        <label for="account_type">Account Type:</label>
        <select id="account_type" name="account_type">
            <option value="patient">Patient</option>
            <option value="doctor">Doctor</option>
        </select>
    </div>
    <input type="submit" value="Next">
</form>
</div>
<div class="additional-links">
    <p><a href="{% url 'home' %}">Back to Home</a></p>
</div>
</body>
</html>

```

Patient Signup Form(Comes after user form if account type is patient):

Patient Information

Medical History:

Prescriptions:

Submit

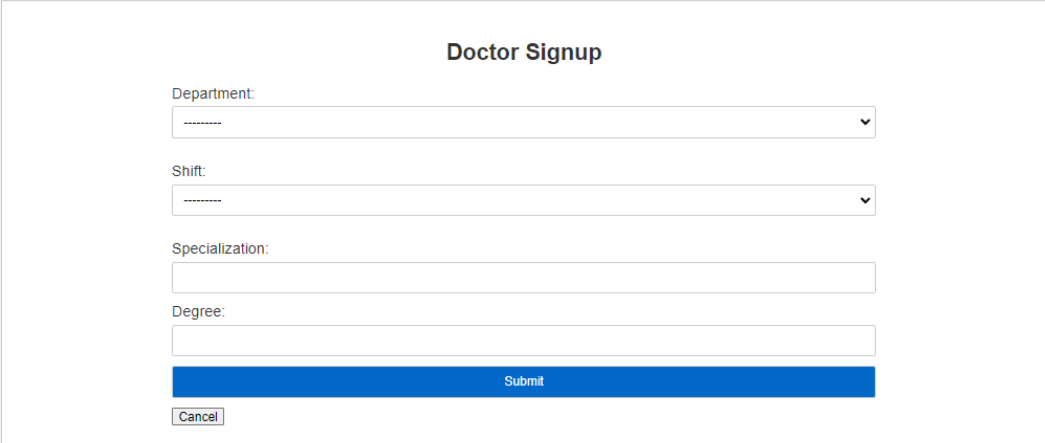
Cancel

HTML:

```
{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Patient Information</title>
  <link rel="stylesheet" href="{% static 'css/signup.css' %}">
</head>
<body>
  <h2>Patient Information</h2>
  <form method="post">
    {% csrf_token %}
    <div class="form-group">
      <label for="medical_history">Medical History:</label>
      <textarea id="medical_history" name="medical_history" rows="4" required></textarea>
    </div>/.form-group
    <div class="form-group">
      <label for="prescriptions">Prescriptions:</label>
      <textarea id="prescriptions" name="prescriptions" rows="4" required></textarea>
    </div>/.form-group
    <input type="submit" value="Submit">
    <a href="{% url 'cancel_signup' user_profile.id %}"><button type="button">Cancel</button></a>
  </form>
</body>
</html>
```


Doctor Signup Form(Comes after user form if account type is doctor):

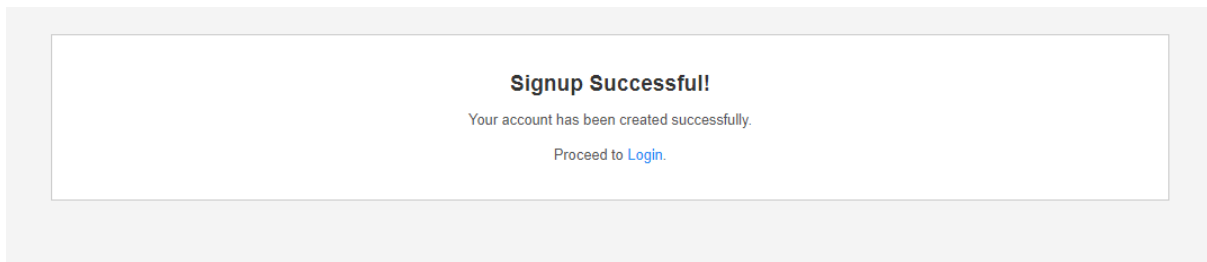


HTML:

```
{%load static%}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Doctor Signup</title>
  <link rel="stylesheet" href="{% static 'css/signup.css' %}">
</head>
<body>
  <div class="signup-form">
    <h2>Doctor Signup</h2>
    <form method="post">
      {% csrf_token %}

      <div class="form-group">
        {{ form.as_p }}
      </div>/.form-group
      <div class="form-group">
        <label for="specialization">Specialization:</label>
        <input type="text" id="specialization" name="specialization" required>
      </div>/.form-group
      <div class="form-group">
        <label for="degree">Degree:</label>
        <input type="text" id="degree" name="degree" required>
      </div>/.form-group
      <input type="submit" value="Submit">
      <a href="{% url 'cancel_signup' user_profile.id %}"><button type="button">Cancel</button></a>
    </form>
  </div>/.signup-form
</body>
</html>
```

Signup Success Page:



Html:

```
{%load static%}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Signup Success</title>
    <link rel="stylesheet" href="{% static 'css/success.css' %}">
</head>
<body>
    <div class="success-page">
        <h2>Signup Successful!</h2>
        <p>Your account has been created successfully.</p>
        <p>Proceed to <a href="{% url 'login' %}">Login</a>.</p>
    </div>/.success-page
</body>
</html>
```

CSS:

```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f4f4f4;
}

.success-page {
    width: 50%;
    margin: 50px auto;
    padding: 20px;
    border: 1px solid #ccc;
    background-color: #fff;
    text-align: center;
}

h2 {
    color: #333;
    margin-bottom: 10px;
}

p {
    color: #555;
    margin-bottom: 20px;
}

a {
    color: #268cfa;
    text-decoration: none;
    transition: color 0.3s;
}

a:hover {
    color: #0056b3;
}
```

Login Page:

Login

Username:

Password:

Login

Don't have an account? [Sign up here](#)
[Back to Home](#)

HTML:

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Login</title>
  <link rel="stylesheet" href="{% static 'css/login.css' %}">
</head>
<body>
  {% if messages %}
    <ul class="messages">
      {% for message in messages %}
        <li{% if message.tags %} class="{{ message.tags }}" {% endif %}>{{ message }}</li>
      {% endfor %}
    </ul>/.messages
  {% endif %}
  <h1>Login</h1>
  <form method="post">
    {% csrf_token %}
    <div>
      <label for="username">Username:</label>
      <input type="text" id="username" name="username" required>
    </div>
    <div>
      <label for="password">Password:</label>
      <input type="password" id="password" name="password" required>
    </div>
    <div>
      <input type="submit" value="Login">
    </div>
  </form>
  <div class="additional-links">
    <p>Don't have an account? <a href="{% url 'signup' %}">Sign up here</a></p>
    <p><a href="{% url 'home' %}">Back to Home</a></p>
  </div>/.additional-links
</body>
</html>
```

CSS (Signup and login):

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f4f4f4;
}

.container {
  width: 50%;
  margin: 50px auto;
  padding: 20px;
  border: 1px solid #ccc;
  background-color: #fff;
}

h1 {
  color: #333;
  text-align: center;
  margin-bottom: 20px;
}

form {
  width: 70%;
  margin: 0 auto;
}

label {
  display: block;
  margin-bottom: 5px;
  color: #333;
}

input[type="text"],
input[type="password"],
input[type="submit"] {
  width: 100%;
  padding: 8px;
  margin-bottom: 10px;
  border-radius: 3px;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

input[type="submit"] {
  background-color: #0066cc;
  color: #fff;
  cursor: pointer;
}

input[type="submit"]:hover {
  background-color: #0052a3;
}
```

```

.additional-links {
    margin-top: 20px;
    text-align: center;
}

.additional-links p {
    margin: 5px 0;
}

.additional-links a {
    color: #268cfa;
    text-decoration: none;
    transition: color 0.3s;
}

.additional-links a:hover {
    color: #0056b3;
}

.messages {
    list-style: none;
    padding: 0;
    margin: 0;
}

.messages li {
    margin-bottom: 5px;
    padding: 10px;
    border-radius: 5px;
}

.error {
    color: #721c24;
    background-color: #f8d7da;
    border-color: #f5c6cb;
}

.success {
    color: #155724;
    background-color: #d4edda;
    border-color: #c3e6cb;
}

.info {
    color: #0c5460;
    background-color: #d1ecf1;
    border-color: #bee5eb;
}

.warning {
    color: #856404;
    background-color: #fff3cd;
    border-color: #ffeeba;
}

```

Admin Profile:

Hospital Admin Dashboard	
Dashboard	Edit Admin Profile
Users	Patients
Doctors	Departments
Shifts	Wards
Beds	Admissions
Billings	Discharged Patients
Doctor Requests	Logout

Quick Links
Create User
Create Patient
Create Doctor
Create Admission
Create Billing
Create Ward
Create Bed

Counts
Total Doctors: 4
Total Patients: 3
Total Admitted Patients: 0
Total Billings: 3
Total Wards: 1
Total Beds: 3
Pending Doctor Requests: 0

CSS:

```
body {
    font-family: 'Roboto', sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f0f0f0;
}

header {
    background-color: #007bff;
    color: #fff;
    text-align: center;
    padding: 20px 0;
}

nav ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    background-color: #268cfa;
    overflow: hidden;
    text-align: right;
}

nav li {
    float: left;
}

nav li a {
    display: block;
    color: #fff;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
```

```

    transition: background-color 0.3s;
}

nav li a:hover {
    background-color: #0056b3;
}

main {
    padding: 20px;
}

.admin-panel {
    display: flex;
    flex-wrap: wrap;
    gap: 20px;
    justify-content: space-between;
}

.shortcuts,
.dashboard-counts {
    flex: 1 1 calc(50% - 20px);
    background-color: #2196f3;
    color: #fff;
    border-radius: 8px;
    padding: 20px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    height: fit-content;
    min-height: 200px;
    margin-bottom: 20px;
}

.shortcuts h3,
.dashboard-counts h3 {
    color: #fff;
    margin-bottom: 15px;
    text-align: center;
}

.shortcuts ul,
.dashboard-counts ul {
    padding: 0;
    list-style: none;
}

.shortcuts li,
.dashboard-counts li {
    padding: 10px;
    margin-bottom: 5px;
    background-color: #1976d2;
    border-radius: 5px;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
    transition: all 0.3s ease;
}

```

```

    text-align: center;
}

.shortcuts li:hover,
.dashboard-counts li:hover {
    transform: translateY(-2px);
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}

.shortcuts ul li a {
    color: #fff;
    text-decoration: none;
    display: block;
    padding: 10px;
}

@media screen and (max-width: 768px) {
    nav li {
        float: none;
        display: block;
        text-align: center;
    }

    .admin-panel {
        flex-direction: column;
    }

    .dashboard-counts,
    .shortcuts {
        flex: 1 1 100%;
    }
}

```

HTML: (Nav Bar)

```

{%load static%}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Admin Dashboard</title>
    <link rel="stylesheet" href="{% static 'css/custom_admin_dashboard.css' %}">
</head>
<body>
    <header>
        <h1>Hospital Admin Dashboard</h1>
    </header>

    <nav>
        <ul>
            <li><a href="{% url 'custom_admin_dashboard' %}">Dashboard</a></li>
            <li><a href="{% url 'edit_admin_profile' %}">Edit Admin Profile</a></li>
            <li><a href="{% url 'view_users' %}">Users</a></li>
            <li><a href="{% url 'view_patients' %}">Patients</a></li>
            <li><a href="{% url 'view_doctors' %}">Doctors</a></li>
            <li><a href="{% url 'view_departments' %}">Departments</a></li>
            <li><a href="{% url 'view_shifts' %}">Shifts</a></li>
            <li><a href="{% url 'view_wards' %}">Wards</a></li>
            <li><a href="{% url 'view_beds' %}">Beds</a></li>
            <li><a href="{% url 'view_admissions' %}">Admissions</a></li>
            <li><a href="{% url 'view_billings' %}">Billings</a></li>
            <li><a href="{% url 'view_discharged_patients' %}">Discharged Patients</a></li>
            <li><a href="{% url 'doctor_nurse_requests' %}">Doctor Requests</a></li>
            <li><a href="{% url 'logout' %}">Logout</a></li>
        </ul>
    </nav>
    <main>

```


(Main)

```
<main>
  <section class="admin-panel">
    <div class="shortcuts">
      <h3>Quick Links</h3>
      <ul class="quick-links">
        <li><a href="{% url 'create_user' %}">Create User</a></li>
        <li><a href="{% url 'create_patient' %}">Create Patient</a></li>
        <li><a href="{% url 'create_doctor' %}">Create Doctor</a></li>
        <li><a href="{% url 'create_admission' %}">Create Admission</a></li>
        <li><a href="{% url 'create_billing' %}">Create Billing</a></li>
        <li><a href="{% url 'create_ward' %}">Create Ward</a></li>
        <li><a href="{% url 'create_bed' %}">Create Bed</a></li>
      </ul>/.quick-links
    </div>/.shortcuts
    <div class="dashboard-counts">
      <h3>Counts</h3>
      <ul class="counts">
        <li>Total Doctors: {{ total_doctors }}</li>
        <li>Total Patients: {{ total_patients }}</li>
        <li>Total Admitted Patients: {{ total_admitted_patients }}</li>
        <li>Total Billings: {{ total_billings }}</li>
        <li>Total Wards: {{ total_wards }}</li>
        <li>Total Beds: {{ total_beds }}</li>
        <li>Pending Doctor Requests: {{ pending_doctor_requests }}</li>
      </ul>/.counts
    </div>/.dashboard-counts
  </section>/.admin-panel
</main>
```

CSS for all list view pages:

```
body {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f0f0f0;
}

table {
  width: 100%;
  border-collapse: collapse;
}

th {
  background-color: #007bff;
  color: white;
  font-weight: bold;
  text-align: left;
  padding: 12px;
}

td {
  border-bottom: 1px solid #ddd;
  padding: 12px;
}

tr:hover {
  background-color: #f5f5f5;
}
```

```

nav ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    background-color: #268cfa;
    overflow: hidden;
    text-align: right;
}

nav li {
    float: left;
}

nav li a {
    display: block;
    color: #fff;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
    transition: background-color 0.3s;
}

nav li a:hover {
    background-color: #0056b3;
    color: #fff;
}

a {
    color: #268cfa;
    text-decoration: none;
    transition: color 0.3s;
}

a:hover {
    color: #0056b3;
}

header {
    background-color: #007bff;
    color: #fff;
    text-align: center;
    padding: 20px 0;
}

```

User List Page:

Users						
Dashboard Edit Admin Profile Users Patients Doctors Admissions Departments Wards Shifts Beds Billings Discharged Patients Doctor Requests Logout						
Create User						
Search by Username: <input type="text"/>						
Gender: <input type="text"/>						
Account type: <input type="text"/>						
Search/Filter Cancel Filter						
Username	Name	Email	Gender	Contact Number	Account Type	Action
Abnar	Abnar Kahman	abnarahman80@gmail.com	Male	01998825928	Patient	Edit Delete
Edward	Edward Elic	edward@gmail.com	Male	01999999955	Doctor	Edit Delete
Gupu	Gupu Lupu	gupu@gmail.com	Female	01888888888	Patient	Edit Delete
Kashem	Kashem Molla	kashem420@gmail.com	Male	01999999999	Patient	Edit Delete
Kurisu	Kurisu Makise	kurisu@gmail.com	Female	01999999954	Doctor	Edit Delete
Light	Light Yagami	light@gmail.com	Male	01584638231	Doctor	Edit Delete
Mikasa	Mikasa Ackerman	mikasa@gmail.com	Female	01998825927	Doctor	Edit Delete

HTML:

```

<a href="{% url 'create_user' %}">Create User</a>
<form method="GET">
    {{ form.as_p }}
    <button type="submit">Search/Filter</button>
    <a href="{% url 'view_users' %}?cancel_filter=True">Cancel Filter</a>
</form>
<table>
    <thead>
        <tr>
            <th>Username</th>
            <th>Name</th>
            <th>Email</th>
            <th>Gender</th>
            <th>Contact Number</th>
            <th>Account Type</th>
            <th>Action</th>
        </tr>
    </thead>
    <tbody>

        {% for user in all_users %}
            <tr>
                <td>{{ user.username }}</td>
                <td>{{ user.full_name }}</td>
                <td>{{ user.email }}</td>
                <td>{{ user.get_gender_display }}</td>
                <td>0{{ user.contact_no }}</td>
                <td>{{ user.get_account_type_display }}</td>
                <td>
                    <a href="{% url 'edit_user' user.id %}">Edit</a>
                    <a href="{% url 'delete_user' user.id %}">Delete</a>
                </td>
            </tr>
        {% endfor %}

    </tbody>
</table>

```

Patient List Page:

Patients						
Dashboard Edit Admin Profile Users Patients Doctors Admissions Departments Wards Shifts Beds Billings Discharged Patients Doctor Requests Logout						
Create Patient						
Search by Username: <input type="text"/>						
Search/Filter Cancel Filter						
Username	Name	Medical History	Prescribed Medications	Admission ID	Billing ID	Action
Abrar	Abrar Rahman	dsadsadad	dsadsadada			Edit Delete
Gupu	Gupu Lupu	dsdsadad	dadsada			Edit Delete
Kashem	Kashem Molla	dfdfdf	fdfdfdf			Edit Delete

HTML:

```
<a href="{% url 'create_patient' %}">Create Patient</a>
<form method="GET">
    {{ form.as_p }}
    <button type="submit">Search/Filter</button>
    <a href="{% url 'view_patients' %}?cancel_filter=True">Cancel Filter</a>
</form>
<table>
    <thead>
        <tr>
            <th>Username</th>
            <th>Name</th>
            <th>Medical History</th>
            <th>Prescribed Medications</th>
            <th>Admission ID</th>
            <th>Billing ID</th>
            <th>Action</th>
        </tr>
    </thead>
    <tbody>
        {% for patient in all_patients %}
        <tr>
            <td>{{ patient.user_profile.username }}</td>
            <td>{{ patient.user_profile.full_name }}</td>
            <td>{{ patient.medical_history }}</td>
            <td>{{ patient.prescribed_medications }}</td>
            <td>{{ patient.admission.id }}</td>
            <td>{{ patient.billing.id }}</td>
            <td>
                <a href="{% url 'edit_patient' patient.id %}">Edit</a>
                <a href="{% url 'delete_patient' patient.id %}">Delete</a>
            </td>
        </tr>
        {% empty %}
        <tr>
            <td>No patients available.</td>
        </tr>
        {% endfor %}
    </tbody>
</table>
```

Doctor List Page:

Doctors

[Dashboard](#)
[Edit Admin Profile](#)
[Users](#)
[Patients](#)
[Doctors](#)
[Admissions](#)
[Departments](#)
[Wards](#)
[Shifts](#)
[Beds](#)
[Billings](#)
[Discharged Patients](#)
[Doctor Requests](#)
[Logout](#)

Create Doctor

Search by Username:

Department

Select department

Search/Filter

Cancel Filter

Username	Name	Department	Shift	Specialization	Degree	Action
Edward	Edward Eric	Cardiology Department	Day Shift	Cardiologist	MBBS, MS	Edit Delete
Kursu	Kursu Makise	Orthopedics Department	Evening Shift	Oncologist	MBBS, FCPS	Edit Delete
Light	Light Yagami	Neurology Department	Day Shift	Neurologist	FCPS	Edit Delete
Mikasa	Mikasa Ackerman	Pediatrics Department	Night Shift	Pediatrician	MBBS, DMRT, FCPS	Edit Delete

HTML:

```

<a href="{% url 'create_doctor' %}">Create Doctor</a>
<form method="GET">
    {{ form.as_p }}
    <button type="submit">Search/Filter</button>
    <a href="{% url 'view_doctors' %}?cancel_filter=True">Cancel Filter</a>
</form>
<table>
    <thead>
        <tr>
            <th>Username</th>
            <th>Name</th>
            <th>Department</th>
            <th>Shift</th>
            <th>Specialization</th>
            <th>Degree</th>
            <th>Action</th>
        </tr>
    </thead>
    <tbody>
        {% for doctor in all_doctors %}
        <tr>
            <td>{{ doctor.user_profile.username }}</td>
            <td>{{ doctor.user_profile.full_name }}</td>
            <td>{{ doctor.department }}</td>
            <td>{{ doctor.shift.get_type_display }}</td>
            <td>{{ doctor.specialization }}</td>
            <td>{{ doctor.qualifications }}</td>
            <td>
                <a href="{% url 'edit_doctor' doctor.user_profile.id %}">Edit</a>
                <a href="{% url 'delete_doctor' doctor.user_profile.id %}">Delete</a>
            </td>
        </tr>
        {% empty %}
        <tr>
            <td>No registered doctors available.</td>
        </tr>
        {% endfor %}
    </tbody>
</table>

```

Admissions List Page:

Admissions						
Dashboard Edit Admin Profile Users Patients Doctors Admissions Departments Wards Shifts Beds Billings Discharged Patients Doctor Requests Logout						
Admit Patient						
Search by Username or Name: <input type="text"/>						
Ward Type: <input type="text"/>						
Sort by: <input type="text"/>						
Search/Filter Cancel Filter						
Admission ID	Patient Username	Name	Ward Type	Bed	Admission Datetime	Action
19	Abbar	Abbar Rahman	General Ward	A1	Dec. 8, 2023, 10:52 p.m.	Edit Delete
20	Kashem	Kashem Molla	General Ward	A2	Dec. 8, 2023, 10:52 p.m.	Edit Delete
21	Gupu	Gupu Lupu	General Ward	A3	Dec. 8, 2023, 10:53 p.m.	Edit Delete

HTML:

```
<a href="{% url 'create_admission' %}">Admit Patient</a>
<form method="GET">
    {{ form.as_p }}
    <button type="submit">Search/Filter</button>
    <a href="{% url 'view_admissions' %}?cancel_filter=True">Cancel Filter</a>
</form>
<table>
    <thead>
        <tr>
            <th>Admission ID</th>
            <th>Patient Username</th>
            <th>Name</th>
            <th>Ward Type</th>
            <th>Bed</th>
            <th>Admission Datetime</th>
            <th>Action</th>
        </tr>
    </thead>
    <tbody>
        {% for admission in admissions %}
        <tr>
            <td>{{ admission.id }}</td>
            <td>{{ admission.patient.user_profile.username }}</td>
            <td>{{ admission.patient.user_profile.full_name }}</td>
            <td>{{ admission.bed.ward.get_ward_type_display }}</td>
            <td>{{ admission.bed.bed_name }}</td>
            <td>{{ admission.admission_datetime }}</td>
            <td>
                <a href="{% url 'edit_admission' admission.id %}">Edit</a>
                <a href="{% url 'delete_admission' admission.id %}">Delete</a>
            </td>
        </tr>
        {% endfor %}
    </tbody>
</table>
```

Department List Page:

Departments		
<div>DashboardEdit Admin ProfileUsersPatientsDoctorsAdmissionsDepartmentsWardsShiftsBedsBillingsDischarged PatientsDoctor RequestsLogout</div>		
Add Department		
Name	Head Doctor	Action
Cardiology	Edward Ethic	Edit Delete
Neurology	Light Yagami	Edit Delete
Orthopedics	Kurisu Makise	Edit Delete
Pediatrics	Mikasa Ackerman	Edit Delete

HTML:

```
<a href="{% url 'create_department' %}">Add Department</a>
<table>
  <thead>
    <tr>
      <th>Name</th>
      <th>Head Doctor</th>
      <th>Action</th>
    </tr>
  </thead>
  <tbody>
    {% for department in departments %}
      <tr>
        <td>{{ department.get_name_display }}</td>
        <td>{{ department.head_doctor }}</td>
        <td>
          <a href="{% url 'edit_department' department.id %}">Edit</a>
          <a href="{% url 'delete_department' department.id %}">Delete</a>
        </td>
      </tr>
    {% endfor %}
  </tbody>
</table>
```

Ward List Page:

Wards				
<div>DashboardEdit Admin ProfileUsersPatientsDoctorsAdmissionsDepartmentsWardsShiftsBedsBillingsDischarged PatientsDoctor RequestsLogout</div>				
Create Ward				
Ward Type: <div></div>				
<div>SearchFilterCancel Filter</div>				
Ward Name	Ward Type	Capacity	Occupancy	Action
A	General Ward	3	3	Edit Delete
B	Specialized Ward	0	0	Edit Delete
C	Isolation Ward	0	0	Edit Delete
D	Isolation Ward	0	0	Edit Delete

HTML:

```
<a href="{% url 'create_ward' %}">Create Ward</a>
<form method="GET">
  {{ form.as_p }}
  <button type="submit">Search/Filter</button>
  <a href="{% url 'view_wards' %}?cancel_filter=True">Cancel Filter</a>
</form>
<table>
  <thead>
    <tr>
      <th>Ward Name</th>
      <th>Ward Type</th>
      <th>Capacity</th>
      <th>Occupancy</th>
      <th>Action</th>
    </tr>
  </thead>
  <tbody>
    {% for ward in wards %}
    <tr>
      <td>{{ ward.name }}</td>
      <td>{{ ward.get_ward_type_display }}</td>
      <td>{{ ward.capacity }}</td>
      <td>{{ ward.occupancy }}</td>
      <td>
        <a href="{% url 'edit_ward' ward.id %}">Edit</a>
        <a href="{% url 'delete_ward' ward.id %}">Delete</a>
      </td>
    </tr>
    {% endfor %}
  </tbody>
</table>
```

Shift List Page:

Shifts			
Dashboard Edit Admin Profile Users Patients Doctors Admissions Departments Wards Shifts Beds Billings Discharged Patients Doctor Requests Logout			
Create Shift			
Shift	Start time	End time	Action
Day Shift	7 a.m.	3 p.m.	Edit Delete
Evening Shift	3 p.m.	11 p.m.	Edit Delete
Night Shift	11 p.m.	7 a.m.	Edit Delete

HTML:

```
<a href="{% url 'create_shift' %}">Create Shift</a>
<table>
  <thead>
    <tr>
      <th>Shift </th>
      <th>Start time</th>
      <th>End time</th>
      <th>Action</th>
    </tr>
  </thead>
  <tbody>
    {% for shift in shifts %}
    <tr>
      <td>{{ shift.get_type_display }}</td>
      <td>{{ shift.start_time }}</td>
      <td>{{ shift.end_time }}</td>
      <td>
        <a href="{% url 'edit_shift' shift.id %}">Edit</a>
        <a href="{% url 'delete_shift' shift.id %}">Delete</a>
      </td>
    </tr>
    {% endfor %}
  </tbody>
</table>
```

Bed List Page:

Beds			
Dashboard Edit Admin Profile Users Patients Doctors Admissions Departments Wards Shifts Beds Billings Discharged Patients Doctor Requests Logout			
Create Bed			
Bed Number: <input type="text"/>			
Ward: <input type="text"/>			
<input type="button" value="Search/Filter"/> <input type="button" value="Cancel Filter"/>			
Bed	Ward Type	Admission ID	Action
A1	General Ward	19	Edit Delete
A2	General Ward	20	Edit Delete
A3	General Ward	21	Edit Delete

HTML:

```
<a href="{% url 'create_bed' %}">Create Bed</a>
<form method="GET">
  {{ form.as_p }}
  <button type="submit">Search/Filter</button>
  <a href="{% url 'view_beds' %}?cancel_filter=True">Cancel Filter</a>
</form>
<table>
  <thead>
    <tr>
      <th>Bed</th>
      <th>Ward Type</th>
      <th>Admission ID</th>
      <th>Action</th>
    </tr>
  </thead>
  <tbody>
    {% for bed in beds %}
    <tr>
      <td>{{ bed.bed_name }}</td>
      <td>{{ bed.ward.get_ward_type_display }}</td>
      <td>{{ bed.admission.id }}</td>
      <td>
        <a href="{% url 'edit_bed' bed.id %}">Edit</a>
        <a href="{% url 'delete_bed' bed.id %}">Delete</a>
      </td>
    </tr>
    {% endfor %}
  </tbody>
</table>
```

Billing List Page:

Billings						
Dashboard Edit Admin Profile Users Patients Doctors Admissions Departments Wards Shifts Beds Billings Discharged Patients Doctor Requests Logout						
Create Billing						
Search by Billing ID: <input type="text"/>						
Date: <input type="text"/>						
Status: <input type="text"/>						
Payment Method: <input type="text"/>						
Search/Filter Cancel Filter						
Billing ID	Patient	Total Amount	Date	Status	Payment Method	Action
13	Gupu	10000.00	Dec. 8, 2023, 6:57 p.m.	Paid	Cash	Edit Delete
12	Kashem	10000.00	Dec. 8, 2023, 6:55 p.m.	Paid	Cash	Edit Delete
11	Abrar	10000.00	Dec. 8, 2023, 6:54 p.m.	Paid	Cash	Edit Delete

HTML:

```
<a href="{% url 'create_billing' %}">Create Billing</a>
<form method="GET">
  {{ form.as_p }}
  <button type="submit">Search/Filter</button>
  <a href="{% url 'view_billings' %}?cancel_filter=True">Cancel Filter</a>
</form>
<table>
  <thead>
    <tr>
      <th>Billing ID</th>
      <th>Patient</th>
      <th>Total Amount</th>
      <th>Date</th>
      <th>Status</th>
      <th>Payment Method</th>
      <th>Action</th>
    </tr>
  </thead>
  <tbody>
    {% for billing in billings %}
    <tr>
      <td>{{ billing.id }}</td>
      <td>{{ billing.patient.user_profile.username }}</td>
      <td>{{ billing.total_amount }}</td>
      <td>{{ billing.date }}</td>
      <td>{{ billing.get_status_display }}</td>
      <td>{{ billing.get_payment_method_display }}</td>
      <td>
        <a href="{% url 'edit_billing' billing.id %}">Edit</a>
        <a href="{% url 'delete_billing' billing.id %}">Delete</a>
      </td>
    </tr>
    {% endfor %}
  </tbody>
</table>
```

Discharged Patient List:

Discharged Patients													
Dashboard	Edit Admin Profile	Users	Patients	Doctors	Admissions	Departments	Wards	Shifts	Beds	Billings	Discharged Patients	Doctor Requests	Logout
ID	Patient Username		Patient Name		Bed		Admission Date		Discharge Date				
9	Abrar		Abrar Rahman		A1		Dec. 8, 2023, 6:54 p.m.		Dec. 8, 2023, 1 p.m.				
8	Kashem		Kashem Molla		A2		Dec. 8, 2023, 6:54 p.m.		Dec. 8, 2023, 12:59 p.m.				
7	Gupu		Gupu Lupu		A3		Dec. 8, 2023, 6:57 p.m.		Dec. 8, 2023, 12:58 p.m.				
6	Abrar		Abrar Rahman		A1		Dec. 8, 2023, 6:42 p.m.		Dec. 8, 2023, 12:49 p.m.				
5	Abrar		Abrar Rahman		A1		Dec. 8, 2023, 6:11 p.m.		Dec. 8, 2023, 12:41 p.m.				
4	Abrar		Abrar Rahman		A1		Dec. 8, 2023, 6:08 p.m.		Dec. 8, 2023, 12:09 p.m.				
3	Abrar		Abrar Rahman		A1		Dec. 8, 2023, 5:28 p.m.		Dec. 8, 2023, 12:08 p.m.				
2	Abrar		Abrar Rahman		A1		Dec. 8, 2023, 5:19 p.m.		Dec. 8, 2023, 11:22 a.m.				

HTML:

```
<table>
  <thead>
    <tr>
      <th>ID</th>
      <th>Patient Username</th>
      <th>Patient Name</th>
      <th>Bed</th>
      <th>Admission Date</th>
      <th>Discharge Date</th>
    </tr>
  </thead>
  <tbody>
    {% for discharged_patient in discharged_patients %}
      <tr>
        <td>{{ discharged_patient.id }}</td>
        <td>{{ discharged_patient.patient.user_profile.username }}</td>
        <td>{{ discharged_patient.patient.user_profile.full_name }}</td>
        <td>{{ discharged_patient.bed_name }}</td>
        <td>{{ discharged_patient.admission_datetime }}</td>
        <td>{{ discharged_patient.discharge_datetime }}</td>
      </tr>
    {% endfor %}
  </tbody>
</table>
```

Doctor Request List:

Doctor Requests					
Dashboard Edit Admin Profile Users Patients Doctors Admissions Departments Wards Shifts Beds Billings Discharged Patients Doctor Requests Logout					
Search by Username: <input type="text"/>					
Search/Filter Cancel Filter					
Username	Name	Specialization	Degree	Date	Action
Sakura		Cardiologist	MBBS, DA, FCPS	Dec. 8, 2023, 5:01 p.m.	Approve Reject

HTML:

```
<form method="GET">
  {{ form.as_p }}
  <button type="submit">Search/Filter</button>
  <a href="{% url 'doctor_nurse_requests' %}?cancel_filter=True">Cancel Filter</a>
</form>
<table>
  <thead>
    <tr>
      <th>Username</th>
      <th>Name</th>
      <th>Specialization</th>
      <th>Degree</th>
      <th>Date</th>
      <th>Action</th>
    </tr>
  </thead>
  <tbody>
    {% for doctor in doctor_details %}
    <tr>
      <td>{{ doctor.username }}</td>
      <td>{{ doctor.full_name }}</td>
      <td>{{ doctor.specialization }}</td>
      <td>{{ doctor.degree }}</td>
      <td>{{ doctor.date_joined }}</td>
      <td>
        <a href="{% url 'approve_account' doctor.id %}">Approve</a>
        <a href="{% url 'reject_account' doctor.id %}">Reject</a>
      </td>
    </tr>
    {% empty %}
    <tr>
      <td>No pending doctor requests.</td>
    </tr>
    {% endfor %}
  </tbody>
</table>
```

CSS For ALL update/create form:

```
form {
  width: 50%;
  margin: 50px auto;
  padding: 20px;
  border: 1px solid #ccc;
  background-color: #fff;
}

h1 {
  color: #333;
  text-align: center;
  margin-bottom: 20px;
}

form label {
  display: block;
  margin-bottom: 5px;
  color: #333;
}

form input[type="text"],
form input[type="password"],
form input[type="submit"] {
  width: 100%;
  padding: 8px;
  margin-bottom: 10px;
  border-radius: 3px;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

form input[type="submit"] {
  background-color: #0066cc;
  color: #fff;
  cursor: pointer;
}

form input[type="submit"]:hover {
  background-color: #0052a3;
}
```

Admin Profile Edit Page:

Edit Admin Profile

Username:

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

First name:

Last name:

Email:

Gender:

Address:

Date of birth:

Contact no:

Password:
No password set.
Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using [this form](#).

HTML:

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Edit Admin Profile</title>
  <link rel="stylesheet" href="{% static 'css/update_create.css' %}">
</head>
<body>
  <h1>Edit Admin Profile</h1>
  <form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Save Changes</button>
    <a href="{% url 'custom_admin_dashboard' %}"><button type="button">Cancel</button></a>
  </form>
</body>
</html>
```

All Create and Edit Forms:

Create User

Username:

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

First name:

Last name:

Email:

Gender:

▼

Address:

Date of birth:

Contact no:

Account type:

▼

Is approved:

☒

Create

Cancel

HTML:

```
{%load static%}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Create User</title>
    <link rel="stylesheet" href="{% static 'css/update_create.css' %}">
</head>
<body>
    <h1>Create User</h1>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit" value="Create">
        <a href="{% url 'view_users' %}"><button type="button">Cancel</button></a>
    </form>
</body>
</html>
```

Edit User

Username:	<input type="text" value="Abrar"/>
Required. 150 characters or fewer. Letters, digits and @./+/_ only.	
First name:	<input type="text" value="Abrar"/>
Last name:	<input type="text" value="Rahman"/>
Email:	<input type="text" value="abrarrahman80@gmail.com"/>
Gender:	<input type="text" value="Male"/>
Address:	<input type="text" value="wapdaroad, west rampura, dhaka"/>
Date of birth:	<input type="text" value="2023-12-08"/>
Contact no:	<input type="text" value="1998825928"/>
Account type:	<input type="text" value="Patient"/>
Is approved:	<input checked="" type="checkbox"/>
<input type="button" value="Save Changes"/>	
<input type="button" value="Cancel"/>	

HTML:

```
{%load static%}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Edit User</title>
  <link rel="stylesheet" href="{% static 'css/update_create.css' %}">
</head>
<body>
  <h1>Edit User</h1>
  <form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <input type="submit" value="Save Changes">
    <a href="{% url 'view_users' %}"><button type="button">Cancel</button></a>
  </form>
</body>
</html>
```


Create Patient Profile

User profile:

Medical history:

Prescribed medications:

HTML:

```
{%load static%}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Create Patient</title>
  <link rel="stylesheet" href="{% static 'css/update_create.css' %}">
</head>
<body>
  <h1>Create Patient Profile</h1>
  <form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <input type="submit" value="Create">
    <a href="{% url 'view_patients' %}"><button type="button">Cancel</button></a>
  </form>
</body>
</html>
```

Edit Patient Profile

User profile:

Medical history:

Prescribed medications:

HTML:

```
{%load static%}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Edit Patient Profile</title>
  <link rel="stylesheet" href="{% static 'css/update_create.css' %}">
</head>
<body>
  <h1>Edit Patient Profile</h1>
  <form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <input type="submit" value="Update">
    <a href="{% url 'view_patients' %}"><button type="button">Cancel</button></a>
  </form>
</body>
</html>
```

Create Doctor

User profile:

Department:

Shift:

Specialization:

Qualifications:

HTML:

```
{%load static%}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Create Doctor</title>
  <link rel="stylesheet" href="{% static 'css/update_create.css' %}">
</head>
<body>
  <h1>Create Doctor</h1>
  <form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Create Doctor</button>
    <a href="{% url 'view_doctors' %}"><button type="button">Cancel</button></a>
  </form>
</body>
</html>
```

Edit Doctor

User profile:

Department:

Shift:

Specialization:

Qualifications:

HTML:

```
{%load static%}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Edit Doctor</title>
  <link rel="stylesheet" href="{% static 'css/update_create.css' %}">
</head>
<body>
  <h1>Edit Doctor</h1>

  <form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <input type="submit" value="Save Changes">
    <a href="{% url 'view_doctors' %}"><button type="button">Cancel</button></a>
  </form>
</body>
</html>
```

Admit Patient

Patient:

Bed:

Admission datetime:

HTML:

```
{%load static%}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Admit Patient</title>
  <link rel="stylesheet" href="{% static 'css/update_create.css' %}">
</head>
<body>
  <h1>Admit Patient</h1>
  <form method="post" >
    {% csrf_token %}
    {{ form.as_p }}
    <input type="submit" value="Admit">
    <a href="{% url 'view_admissions' %}"><button type="button">Cancel</button></a>
  </form>
</body>
</html>
```

Edit Admission

Patient:

Bed:

Admission datetime:

HTML:

```
{%load static%}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Edit Admission</title>
    <link rel="stylesheet" href="{% static 'css/update_create.css' %}">
</head>
<body>
    <h1>Edit Admission</h1>

    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit" value="Save Changes">
        <a href="{% url 'view_admissions' %}"><button type="button">Cancel</button></a>
    </form>
</body>
</html>
```

Create Department

Name:

Description:

Head doctor:

HTML:

```
{%load static%}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Create Department</title>
    <link rel="stylesheet" href="{% static 'css/update_create.css' %}">
</head>
<body>
    <h1>Create Department</h1>
    <form method="post" >
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit" value="Create Department">
        <a href="{% url 'view_departments' %}"><button type="button">Cancel</button></a>
    </form>
</body>
</html>
```

Edit Department

Name:

Description:

Cardiology is the study of the heart. Cardiology is a branch of medicine that deals with disorders of the heart and the cardiovascular system. The field includes medical diagnosis and treatment of congenital heart defects, coronary artery disease, heart failure, valvular heart disease, and electrophysiology.

Head doctor:

HTML:

```
{%load static%}  
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <title>Edit Department</title>  
  <link rel="stylesheet" href="{% static 'css/update_create.css' %}">  
</head>  
<body>  
  <h1>Edit Department</h1>  
  <form method="post">  
    {% csrf_token %}  
    {{ form.as_p }}  
    <input type="submit" value="Save Changes">  
    <a href="{% url 'view_departments' %}"><button type="button">Cancel</button></a>  
  </form>  
</body>  
</html>
```

Create Ward

Name:

Ward type:

HTML:

```
{%load static%}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Create Ward</title>
    <link rel="stylesheet" href="{% static 'css/update_create.css' %}">
</head>
<body>
    <h1>Create Ward</h1>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit" value="Create Ward">
        <a href="{% url 'view_wards' %}"><button type="button">Cancel</button></a>
    </form>
</body>
</html>
```

Edit Ward

Name:

Ward type:

HTML:

```
{%load static%}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Edit Ward</title>
    <link rel="stylesheet" href="{% static 'css/update_create.css' %}">
</head>
<body>
    <h1>Edit Ward</h1>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit" value="Save Changes">
        <a href="{% url 'view_wards' %}"><button type="button">Cancel</button></a>
    </form>
</body>
</html>
```


Create Shift

Type:

Start time:

End time:

HTML:

```
{%load static%}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Create Shift</title>
  <link rel="stylesheet" href="{% static 'css/update_create.css' %}">
</head>
<body>
  <h1>Create Shift</h1>
  <form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <input type="submit" value="Create Shift">
    <a href="{% url 'view_shifts' %}"><button type="button">Cancel</button></a>
  </form>
</body>
</html>
```

Edit Shift

Type:

Start time:

End time:

HTML:

```
{%load static%}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Edit Shift</title>
    <link rel="stylesheet" href="{% static 'css/update_create.css' %}">
</head>
<body>
    <h1>Edit Shift</h1>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit" value="Save Changes">
        <a href="{% url 'view_shifts' %}"><button type="button">Cancel</button></a>
    </form>
</body>
</html>
```

Create a New Bed

Ward:

Bed number:

```
{%load static%}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Create Bed</title>
    <link rel="stylesheet" href="{% static 'css/update_create.css' %}">
</head>
<body>
    <h1>Create a New Bed</h1>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit" value="Create Bed">
        <a href="{% url 'view_beds' %}"><button type="button">Cancel</button></a>
    </form>
</body>
</html>
```

Edit Bed

Ward:

Bed number:

```
{%load static%}  
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <title>Edit Bed</title>  
  <link rel="stylesheet" href="{% static 'css/update_create.css' %}">  
</head>  
<body>  
  <h1>Edit Bed</h1>  
  
  <form method="post">  
    {% csrf_token %}  
    {{ form.as_p }}  
    <input type="submit" value="Save Changes">  
    <a href="{% url 'view_beds' %}"><button type="button">Cancel</button></a>  
  </form>  
</body>  
</html>
```

Create Billing

Patient:

Total amount:

Date:

Status:

Payment method:

```
{%load static%}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Create Billing</title>
    <link rel="stylesheet" href="{% static 'css/update_create.css' %}">
</head>
<body>
    <h1>Create Billing</h1>
    {% if messages %}
    <ul class="messages">
        {% for message in messages %}
            <li{% if message.tags %} class="{% message.tags %}"{% endif %}>{{ message }}</li>
        {% endfor %}
    </ul>/.messages
    {% endif %}
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit" value="Create Billing">
        <a href="{% url 'view_billings' %}"><button type="button">Cancel</button></a>
    </form>
</body>
</html>
```

Edit Billing

Patient:

Total amount:

Date:

Status:

Payment method:

HTML:

```
{%load static%}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Edit Billing</title>
    <link rel="stylesheet" href="{% static 'css/update_create.css' %}">
</head>
<body>
    <h1>Edit Billing</h1>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit" value="Save Changes">
        <a href="{% url 'view_billings' %}"><button type="button">Cancel</button></a>
    </form>
</body>
</html>
```

Contribution of ID : 23341111, Name : Fardin Hassan Tamim

Patient Profile Dashboard:

Profile Update and Doctor Search

Welcome to Your Dashboard, tamiem

Your Information

First Name

tamiem

Last Name

ddd

Address

adsdd

Mail

tamiem@gmail.com

Contacts

1303866626

Update Profile

Search Doctors:

Enter doctor's name

Patient Additional information like Medical History:

Payment Info: Tk

Medical History:

fever

Prescribed Medications:

NA

Admission Details

Name:

Ward:

Bed:

Admission date-time:

Discharge date-time:

Logout

HTML for Profile Update and Doctor Search

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Patient Dashboard</title>
</head>
<body>
  <div class="container">

    <h1>Welcome to Your Dashboard, {{ user.first_name }}</h1>
    <h2>Your Information</h2>
    <div>
      <form action="{% url 'update_profile' %}" method="post" >
        {% csrf_token %}
        <div>
          <label for="">First Name</label>
          <input class="firstname" type="text" name="first_name" value="{{user.first_name}}">
        </div>
        <div>
          <label for="">Last Name</label>
          <input class="lastname" type="text" name="last_name" value="{{user.last_name}}">
        </div>
        <div>
          <label for="">Address</label>
          <input class="address" type="text" name="address" value="{{user.address}}">
        </div>
        <div>
          <label for="">Mail</label>
          <input class="mail" type="text" name="mail" value="{{user.email}}">
        </div>
        <div>
          <label for="">Contacts</label>
          <input class="mail" type="text" name="con" value="{{user.contact_no}}">
        </div>
        <button type="submit">Update Profile</button>
      </form>
    </div>
  </div>
</body>
</html>
```

HTML for Patient Additional information like Medical History

```
{% if paymentStatus%}
  <h2>Payment Info:</h2>
  <p>Total Amount: {{ paymentStatus.total_amount }}</p>
  <p>Payment Status: {{ paymentStatus.status }}</p>
{% endif %}
<h2>Medical History:</h2>
<p>{{ medical_history }}</p>

<h2>Prescribed Medications:</h2>
<p>{{ prescribed_medications }}</p>

<h2>Admission Details</h2>

<li>Name: {{admission.patient}}</li>
<li>Ward: {{admission.ward}}</li>
<li>Bed: {{admission.bed}}</li>
<li>Admission date-time: {{admission.admission_datetime}}</li>
<li>Discharge date-time: {{admission.discharge_datetime}}</li>

<a href="{% url 'logout' %}"> <button>Logout </button></a>

</div>
</div>
</html>
```

Contribution of ID : 21301661, Name : Meherun Chowdhury Dola
Booking Appointment

The screenshot shows a web interface for booking appointments. At the top, there's a header "All Doctors:". Below it, there are two buttons: "Sort Ascending" and "Sort Descending". A list of doctors is displayed, each with a "Book Appointment" button. The doctors listed are "monir dasdas - sdasda" and "mahmud sdasdas - sdasdas". Below the doctors list, there's a section titled "Appointments:" which shows a message: "No appointments scheduled." At the bottom, there's a red text "Payment Info: Tk".

HTML:

```
<h2>Appointments:</h2>

{% for appointment in appointments %}
  <li>{{ appointment.date }} at {{ appointment.time }} - {{ appointment.problem }}</li>
{% empty %}
  <li>No appointments scheduled.</li>
{% endfor %}
```

Contribution of ID : 21301142, Name : AL-AMIN SAGOR

Contribution of ID : 20101466, Name : MD. GOLAM SHAHNEWAZ

Backend Development

Contribution of ID : 21201080, Name : Md. Abrar Rahman Shafin

The Django ORM (Object-Relational Mapping) has been utilized to manage the backend of the project, handling database operations through Pythonic abstractions.

All existing tables shown in Built-in Django Admin Panel :

HOSPITAL
Admissions
Appointments
Beds
Billings
Departments
Discharged patients
Doctor profiles
Patient profiles
Shifts
Users
Wards

Table Creation using Django ORM:

Importing AbstractUser for user model, database models, project settings, validators, and date class for date handling in Django.

```
from django.contrib.auth.models import AbstractUser
from django.db import models
from django.conf import settings
from django.core.validators import MinValueValidator, MaxValueValidator, RegexValidator
from datetime import date
```


UserProfile Table:

Django's AbstractUser provides username, password, first_name, last_name etc.

UserProfile inherited these attributes from abstractuser, for which they are not mentioned explicitly while creating the UserProfile table. Validator has been used to accept more valid contact numbers(Bangladesh Only). Full_name is composed of first_name and last_name.

```
class UserProfile(AbstractUser):
    email = models.EmailField(unique=True)
    GENDER_CHOICES = [
        ('M', 'Male'),
        ('F', 'Female'),
        ('O', 'Other'),
    ]
    gender = models.CharField(max_length=1, choices=GENDER_CHOICES)
    address = models.CharField(max_length=200)
    date_of_birth = models.DateField(default=date.today)
    contact_no = models.IntegerField(
        validators=[
            MinValueValidator(1500000000),
            MaxValueValidator(1999999999)
        ], unique=True, default=0000000000
    )
    ACCOUNT_TYPE_CHOICES = [
        ('patient', 'Patient'),
        ('doctor', 'Doctor'),
    ]
    account_type = models.CharField(max_length=20, choices=ACCOUNT_TYPE_CHOICES, blank=False, null=False)
    is_approved = models.BooleanField(default=True)

    @property
    def full_name(self):
        return f"{self.first_name} {self.last_name}"

    def __str__(self):
        return f"{self.username}"
```

Department, Shift Table

```
class Department(models.Model):
    DEPARTMENT_CHOICES = [
        ('cardiology', 'Cardiology'),
        ('neurology', 'Neurology'),
        ('orthopedics', 'Orthopedics'),
        ('pediatrics', 'Pediatrics')
    ]

    name = models.CharField(max_length=100, unique=True, choices=DEPARTMENT_CHOICES)
    description = models.TextField()
    head_doctor = models.OneToOneField('DoctorProfile', related_name='head_of_department', unique=True, null=True, blank=True, on_delete=models.SET_NULL)

    def __str__(self):
        return f"{self.get_name_display()} Department"

class Shift(models.Model):
    SHIFT_CHOICES = [
        ('day', 'Day Shift'),
        ('evening', 'Evening Shift'),
        ('night', 'Night Shift')
    ]

    type = models.CharField(max_length=20, unique=True, choices=SHIFT_CHOICES)
    start_time = models.TimeField()
    end_time = models.TimeField()

    def __str__(self):
        return f"{self.get_type_display()} - {self.start_time} to {self.end_time}"
```

Doctor and Patient Profile Table:

```
class DoctorProfile(models.Model):
    user_profile = models.OneToOneField(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    specialization = models.CharField(max_length=100)
    qualifications = models.TextField()

    department = models.ForeignKey(Department, on_delete=models.CASCADE)
    shift = models.ForeignKey(Shift, on_delete=models.SET_NULL, null=True)

    def __str__(self):
        return f"{self.user_profile.first_name} {self.user_profile.last_name}"

class PatientProfile(models.Model):
    user_profile = models.OneToOneField(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    medical_history = models.TextField()
    prescribed_medications = models.TextField()

    def __str__(self):
        return self.user_profile.username
```

Appointment Table:

```
class Appointment(models.Model):
    STATUS_CHOICES = [
        ('scheduled', 'Scheduled'),
        ('confirmed', 'Confirmed'),
        ('in_progress', 'In Progress'),
        ('completed', 'Completed'),
        ('cancelled', 'Cancelled'),
        ('no_show', 'No Show'),
        ('rescheduled', 'Rescheduled'),
        ('pending', 'Pending'),
    ]
    date = models.DateField()
    time = models.TimeField()
    patient = models.ForeignKey(PatientProfile, on_delete=models.CASCADE)
    doctor = models.ForeignKey(DoctorProfile, on_delete=models.CASCADE)
    status = models.CharField(max_length=20, choices=STATUS_CHOICES, default='scheduled')
    problem = models.TextField()

    def __str__(self):
        return f"Appointment on {self.date} at {self.time} with Dr. {self.doctor} for {self.patient}: {self.get_status_display()}"
```

Billing Table:

```
class Billing(models.Model):

    STATUS_CHOICES = [
        ('pending', 'Pending'),
        ('paid', 'Paid'),
    ]

    PAYMENT_METHOD_CHOICES = [
        ('cash', 'Cash'),
        ('online', 'Online')
    ]

    patient = models.ForeignKey(PatientProfile, on_delete=models.CASCADE)
    total_amount = models.DecimalField(max_digits=10, decimal_places=2)
    date = models.DateTimeField()
    status = models.CharField(max_length=20, choices=STATUS_CHOICES, default='pending')
    payment_method = models.CharField(max_length=20, choices=PAYMENT_METHOD_CHOICES)

    def __str__(self):
        return f"Billing for {self.patient} - Total Amount: {self.total_amount}, Status: {self.status}"
```

Admission Table and DischargedPatient Table:

```
class Admission(models.Model):
    patient = models.OneToOneField(PatientProfile, on_delete=models.CASCADE)
    bed = models.OneToOneField(Bed, on_delete=models.CASCADE)
    admission_datetime = models.DateTimeField()

    def __str__(self):
        return f"Admission for {self.patient} in {self.ward}"

class DischargedPatient(models.Model):
    patient = models.ForeignKey(PatientProfile, on_delete=models.CASCADE)
    bed_name = models.CharField(max_length=100)
    admission_datetime = models.DateTimeField()
    discharge_datetime = models.DateTimeField()

    def __str__(self):
        return f"Discharged Patient: {self.patient.user_profile.full_name}"
```

Ward Table:

Used validators for ward name so that ward name can be of exact 1 letter (A to Z)

Capacity derived from total number of beds in a ward.

Occupancy derived from total number of admission in a ward.

```
class Ward(models.Model):
    WARD_TYPES = [
        ('general', 'General Ward'),
        ('specialized', 'Specialized Ward'),
        ('isolation', 'Isolation Ward'),
        ('hdu', 'High Dependency Unit (HDU)'),
        ('private', 'Private Ward')
    ]

    name = models.CharField(
        max_length=1,
        unique=True,
        validators=[
            RegexValidator(
                regex='^[A-Z]$',
                message='Ward name should be a single letter from A to Z.',
                code='invalid_ward_name'
            )
        ]
    )
    ward_type = models.CharField(max_length=50, choices=WARD_TYPES)

    @property
    def capacity(self):
        return self.bed_set.count()

    @property
    def occupancy(self):
        return Admission.objects.filter(bed__ward=self).count()

    def __str__(self):
        return f"{self.name} ({self.get_ward_type_display()})"
```

Bed Table:

Bed is a weak entity, it can be uniquely identified by composing ward name and bed number together.

```
class Bed(models.Model):

    ward = models.ForeignKey(Ward, on_delete=models.CASCADE)
    bed_number = models.IntegerField()
    class Meta:
        unique_together = ('ward', 'bed_number')

    @property
    def bed_name(self):
        return f'{self.ward.name}{self.bed_number}'
    def __str__(self):
        return f'{self.ward.name}{self.bed_number}({self.ward.ward_type})"
```

Installing Hospital App:

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'hospital'
]
```

Connecting to Database:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'myHospital',
        'HOST': 'localhost',
        'USER': 'root',
        'PASSWORD': ''
    }
}
```

Using Hospital app's UserProfile as user model:

```
AUTH_USER_MODEL = 'hospital.UserProfile'
```

Ensuring strong password while user signup:

```
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
]
```

Importing builtin functions that are needed for getting data from database and using it in frontend:

```
from django.shortcuts import render, redirect, get_object_or_404
from django.contrib.auth import authenticate, login, logout
from django.contrib.auth.decorators import login_required
from django.db import IntegrityError
from django.http import HttpResponse
from django.contrib import messages
from django.contrib.auth.password_validation import validate_password
from django.db.models import Q
from django.utils import timezone
from django.core.exceptions import ValidationError
from .models import UserProfile, PatientProfile, DoctorProfile,
Department, Appointment, Admission, Billing, Bed, Ward, Shift,
DischargedPatient
from .forms import DeptShifts, AdminProfileForm, AdmissionForm,
DepartmentForm, WardForm, ShiftForm, BedForm, BillingForm, DoctorForm,
UserProfileForm, PatientForm, UserFilterForm, PatientFilterForm,
DoctorFilterForm, DoctorUserFilterForm, WardFilterForm, BedFilterForm,
BillingFilterForm, DoctorRequestFilterForm, AdmissionFilterForm
```

For viewing existing departments, head doctor of each departments, total patient, doctors and hospital capacity in Home Page:

```
#Home
def home_view(request):
    departments = Department.objects.all()

    head_doctors = {
        department.name: department.head_doctor.user_profile.get_full_name() if department.head_doctor else "No head doctor assigned"
        for department in departments
    }

    patients_count = PatientProfile.objects.count()

    approved_doctors_count = DoctorProfile.objects.filter(user_profile__is_approved=True).count()

    total_beds = Bed.objects.count()

    return render(request, 'home.html', {
        'departments': departments,
        'head_doctors': head_doctors,
        'patients_count': patients_count,
        'approved_doctors_count': approved_doctors_count,
        'total_beds': total_beds,
    })
```

For viewing department details without logging in:

```
def department_user_view(request):
    departments = Department.objects.all()
    department_data = []
    for department in departments:
        approved_doctors_count = DoctorProfile.objects.filter(
            department=department, user_profile__is_approved=True
        ).count()
        department_info = {
            'department': department,
            'approved_doctors_count': approved_doctors_count,
        }
        department_data.append(department_info)
    context = {'department_data': department_data}
    return render(request, 'department_user.html', context)
```

For viewing, searching, filtering doctors without logging in:

```
def doctor_user_view(request):
    form = DoctorUserFilterForm(request.GET)
    all_doctors = DoctorProfile.objects.select_related('user_profile').filter(user_profile__is_approved=True)

    if form.is_valid():
        search_query = form.cleaned_data.get('search_query')
        department = form.cleaned_data.get('department')

        if search_query:
            all_doctors = all_doctors.filter(user_profile__first_name__icontains=search_query) | all_doctors.filter(user_profile__last_name__icontains=search_query)

        if department:
            all_doctors = all_doctors.filter(department = department)

    return render(request, 'doctor_user.html', {'all_doctors': all_doctors, 'form': form})
```

For collecting data from SignUp form and storing it into the database:

```
def signup_view(request):
    error = None
    if request.method == 'POST':
        first_name = request.POST.get('first_name')
        last_name = request.POST.get('last_name')
        gender = request.POST.get('gender')
        address = request.POST.get('address')
        email = request.POST.get('email')
        date_of_birth = request.POST.get('dob')
        contact_no = request.POST.get('contact_no')
        account_type = request.POST.get('account_type')
        username = request.POST.get('username')
        password = request.POST.get('password')
        confirm_password = request.POST.get('confirm_password')

        if password != confirm_password:
            return render(request, 'signup.html', {'error': 'Passwords do not match'})

        try:
            validate_password(password)
        except ValidationError as e:
            error = ', '.join(e.messages)
            return render(request, 'signup.html', {'error': error})

        try:
            user_profile = UserProfile.objects.create_user(
                username=username,
                email=email,
                password=password,
                first_name=first_name,
                last_name=last_name,
                gender=gender,
                address=address,
                date_of_birth=date_of_birth,
                contact_no=contact_no,
                account_type=account_type
            )

            if user_profile.account_type == 'patient':
                return redirect('patient_signup', user_profile_id=user_profile.id)
            elif user_profile.account_type == 'doctor':
                user_profile.is_approved = False
                user_profile.save()
                return redirect('doctor_signup', user_profile_id=user_profile.id)

            return redirect('success')
```



```

except IntegrityError as e:
    error = None
    if 'username' in str(e):
        error = 'Username already exists.'
    elif 'email' in str(e):
        error = 'Email already exists.'
    elif 'contact_no' in str(e):
        error = 'Contact number already exists.'

return render(request, 'signup.html', {'error': error})

```

For Logging into all time of user using a single login page:

```

def login_view(request):
    if request.method == 'POST':
        username = request.POST.get('username')
        password = request.POST.get('password')

        user = authenticate(request, username=username, password=password)

        if user is not None:
            if user.account_type == 'patient':
                login(request, user)
                return redirect('patient_dashboard')

            elif user.account_type == 'doctor':
                login(request, user)
                return redirect('doctor_dashboard')

            |
            login(request, user)
            return redirect('custom_admin_dashboard')

        messages.error(request, 'Invalid username or password')
        return render(request, 'login.html')

return render(request, 'login.html')

```

Logout: If a user logs out, they will not be able to go back and do things that requires login.

SignUp cancellation: If someone creates a userprofile and ends up creating neither doctor nor patient profile, then the user profile will be automatically deleted.

Success: A success page will be shown if the user successfully signs up.

```
def user_logout(request):
    logout(request)
    request.session.flush()
    return redirect('home')

def cancel_signup(request, user_profile_id):
    user_profile = UserProfile.objects.get(pk=user_profile_id)
    user_profile.delete()
    return redirect('home')

def signup_success_view(request):
    return render(request, 'success.html')
```

Admin Dashboard:

- Has shortcut links which have been implemented through frontend.
- Shows total count of data of some entities.

```
#Admin

@login_required
def custom_admin_dashboard(request):
    total_doctors = UserProfile.objects.filter(account_type='doctor', is_approved=True).count()
    total_patients = UserProfile.objects.filter(account_type='patient').count()
    total_admitted_patients = Admission.objects.count()
    pending_doctor_requests = UserProfile.objects.filter(account_type='doctor', is_approved=False).count()
    total_billings = Billing.objects.count()
    total_wards = Ward.objects.count()
    total_beds = Bed.objects.count()

    context = {
        'total_doctors': total_doctors,
        'total_patients': total_patients,
        'total_admitted_patients': total_admitted_patients,
        'pending_doctor_requests': pending_doctor_requests,
        'total_billings': total_billings,
        'total_wards': total_wards,
        'total_beds': total_beds,
    }

    return render(request, 'custom_admin_dashboard.html', context)
```

Importing models in Forms.py for using django Built-in form to create, edit, filter entities:

```
from django import forms
from django.contrib.auth.forms import UserChangeForm
from django.core.validators import MinValueValidator, MaxValueValidator
from .models import UserProfile, Department, Shift, Admission, Ward, Bed, PatientProfile, Billing, DoctorProfile
```

Forms for creating and editing:

```
class UserProfileForm(forms.ModelForm):
    class Meta:
        model = UserProfile
        fields = ['username', 'first_name', 'last_name', 'email', 'gender', 'address', 'date_of_birth', 'contact_no', 'account_type', 'is_approved']

class AdminProfileForm(UserChangeForm):
    gender = forms.ChoiceField(choices=UserProfile.GENDER_CHOICES)
    address = forms.CharField(max_length=200)
    date_of_birth = forms.DateField()
    contact_no = forms.IntegerField(
        validators=[
            MinValueValidator(1500000000),
            MaxValueValidator(1999999999)
        ]
    )

    class Meta:
        model = UserProfile
        fields = ['username', 'first_name', 'last_name', 'email', 'gender', 'address', 'date_of_birth', 'contact_no']

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.initial['account_type'] = 'admin'

    def save(self, commit=True):
        instance = super().save(commit=False)
        instance.account_type = 'admin'
        if commit:
            instance.save()
        return instance
```

```
class AdmissionForm(forms.ModelForm):
    class Meta:
        model = Admission
        fields = ['patient', 'bed', 'admission_datetime']
        widgets = {
            'admission_datetime': forms.DateTimeInput(attrs={'type': 'datetime-local'}),
        }

class DepartmentForm(forms.ModelForm):
    class Meta:
        model = Department
        fields = ['name', 'description', 'head_doctor']

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['head_doctor'].queryset = DoctorProfile.objects.filter(
            user_profile__is_approved=True,
            user_profile__account_type='doctor'
        )

class WardForm(forms.ModelForm):
    class Meta:
        model = Ward
        fields = ['name', 'ward_type']

class ShiftForm(forms.ModelForm):
    class Meta:
        model = Shift
        fields = ['type', 'start_time', 'end_time']
```

```

class BedForm(forms.ModelForm):
    class Meta:
        model = Bed
        fields = ['ward', 'bed_number']

class BillingForm(forms.ModelForm):
    class Meta:
        model = Billing
        fields = ['patient', 'total_amount', 'date', 'status', 'payment_method']
        widgets = {
            'date': forms.DateTimeInput(attrs={'type': 'datetime-local'}),
        }

class DoctorForm(forms.ModelForm):
    class Meta:
        model = DoctorProfile
        fields = ['user_profile', 'department', 'shift', 'specialization', 'qualifications']
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['user_profile'].queryset = UserProfile.objects.filter(
            is_approved=True,
            account_type='doctor'
        )

class PatientForm(forms.ModelForm):
    class Meta:
        model = PatientProfile
        fields = ['user_profile', 'medical_history', 'prescribed_medications']
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['user_profile'].queryset = UserProfile.objects.filter(
            is_approved=True,
            account_type='patient'
        )

```

Code implementation of creating and editing feature:

UserProfile:

```

def create_user(request):
    if request.method == 'POST':
        form = UserProfileForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('view_users')
    else:
        form = UserProfileForm()

    return render(request, 'create_user.html', {'form': form})

def update_user(request, user_id):
    user = get_object_or_404(UserProfile, pk=user_id)

    if request.method == 'POST':
        form = UserProfileForm(request.POST, instance=user)
        if form.is_valid():
            form.save()
            return redirect('view_users')
    else:
        form = UserProfileForm(instance=user)

    return render(request, 'update_user.html', {'form': form})

```

PatientProfile:

```
@login_required
def create_patient(request):
    if request.method == 'POST':
        form = PatientForm(request.POST)
        if form.is_valid():
            patient_profile = form.save(commit=False)
            patient_profile.account_type = 'patient'
            patient_profile.save()
            return redirect('view_patients')
    else:
        form = PatientForm()

    return render(request, 'create_patient.html', {'form': form})

@login_required
def update_patient(request, patient_id):
    patient_profile = get_object_or_404(PatientProfile, pk=patient_id)

    if request.method == 'POST':
        form = PatientForm(request.POST, instance=patient_profile)
        if form.is_valid():
            form.save()
            return redirect('view_patients')
    else:
        form = PatientForm(instance=patient_profile)

    return render(request, 'update_patient.html', {'form': form})
```

DoctorProfile:

```
def create_doctor(request):
    if request.method == 'POST':
        form = DoctorForm(request.POST)
        if form.is_valid():
            doctor_profile = form.save(commit=False)
            doctor_profile.account_type = 'doctor'
            doctor_profile.save()
            return redirect('view_doctors')
    else:
        form = DoctorForm()
    return render(request, 'create_doctor.html', {'form': form})

def update_doctor(request, doctor_id):
    doctor_profile = get_object_or_404(DoctorProfile, user_profile_id=doctor_id)

    if request.method == 'POST':
        form = DoctorForm(request.POST, instance=doctor_profile)
        if form.is_valid():
            form.save()
            return redirect('view_doctors')
    else:
        form = DoctorForm(instance=doctor_profile)

    return render(request, 'update_doctor.html', {'form': form})
```

Admission:

```
def create_admission(request):
    if request.method == 'POST':
        form = AdmissionForm(request.POST)
        if form.is_valid():
            patient = form.cleaned_data['patient']
            bed = form.cleaned_data['bed']
            admission_datetime = form.cleaned_data['admission_datetime']

            admission = Admission.objects.create(
                patient=patient,
                bed=bed,
                admission_datetime=admission_datetime
            )
            return redirect('view_admissions')
        else:
            form = AdmissionForm()

    return render(request, 'create_admission.html', {'form': form})
```

```
def update_admission(request, admission_id):
    admission = get_object_or_404(Admission, pk=admission_id)

    if request.method == 'POST':
        form = AdmissionForm(request.POST, instance=admission)
        if form.is_valid():
            form.save()
            return redirect('view_admissions')
        else:
            form = AdmissionForm(instance=admission)

    return render(request, 'update_admission.html', {'form': form})
```

Department:

```
def create_department(request):
    if request.method == 'POST':
        form = DepartmentForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('view_departments')
    else:
        form = DepartmentForm()
    return render(request, 'create_department.html', {'form': form})

def update_department(request, department_id):
    department = get_object_or_404(Department, pk=department_id)

    if request.method == 'POST':
        form = DepartmentForm(request.POST, instance=department)
        if form.is_valid():
            form.save()
            return redirect('view_departments')
    else:
        form = DepartmentForm(instance=department)

    return render(request, 'update_department.html', {'form': form})
```

Ward:

```
def create_ward(request):
    if request.method == 'POST':
        form = WardForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('view_wards')
    else:
        form = WardForm()

    return render(request, 'create_ward.html', {'form': form})

def update_ward(request, ward_id):
    ward = get_object_or_404(Ward, pk=ward_id)

    if request.method == 'POST':
        form = WardForm(request.POST, instance=ward)
        if form.is_valid():
            form.save()
            return redirect('view_wards')
    else:
        form = WardForm(instance=ward)

    return render(request, 'update_ward.html', {'form': form})
```

Shift:

```
def create_shift(request):
    if request.method == 'POST':
        form = ShiftForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('view_shifts')
    else:
        form = ShiftForm()

    return render(request, 'create_shift.html', {'form': form})

def update_shift(request, shift_id):
    shift = get_object_or_404(Shift, pk=shift_id)

    if request.method == 'POST':
        form = ShiftForm(request.POST, instance=shift)
        if form.is_valid():
            form.save()
            return redirect('view_shifts')
    else:
        form = ShiftForm(instance=shift)

    return render(request, 'update_shift.html', {'form': form})
```

Bed:

```
def create_bed(request):
    if request.method == 'POST':
        form = BedForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('view_beds')
    else:
        form = BedForm()

    return render(request, 'create_bed.html', {'form': form})

def update_bed(request, bed_id):
    bed = get_object_or_404(Bed, pk=bed_id)

    if request.method == 'POST':
        form = BedForm(request.POST, instance=bed)
        if form.is_valid():
            form.save()
            return redirect('view_beds')
    else:
        form = BedForm(instance=bed)

    return render(request, 'update_bed.html', {'form': form})
```


Billing:

- Billing can only be created if a patient has admission.
- If billing status is paid, then the patient will be discharged, and admission information will be stored in the dischargedPatient table.
- Billing cannot be created until the previous bill(if there is any) is paid.

```
def create_billing(request):
    if request.method == 'POST':
        form = BillingForm(request.POST)
        if form.is_valid():
            patient_id = form.cleaned_data['patient'].id
            previous_bill = Billing.objects.filter(patient_id=patient_id, status='pending').first()

            if previous_bill:
                messages.error(request, "Patient has a pending billing. Complete the previous billing first.")
                return redirect('create_billing')

            admission = Admission.objects.filter(patient_id=patient_id).first()

            if admission:
                billing_instance = form.save(commit=False)
                billing_instance.save()

                if billing_instance.status == 'paid':
                    discharged_patient = DischargedPatient.objects.create(
                        patient=admission.patient,
                        bed_name=admission.bed.bed_name,
                        admission_datetime=admission.admission_datetime,
                        discharge_datetime=billing_instance.date
                    )
                    admission.delete()

                return redirect('view_billings')
            else:
                messages.error(request, "Patient needs an active admission for billing.")
                return redirect('create_billing')
        else:
            form = BillingForm()

    return render(request, 'create_billing.html', {'form': form})
```

```
def update_billing(request, billing_id):
    billing = Billing.objects.get(pk=billing_id)
    if request.method == 'POST':
        form = BillingForm(request.POST, instance=billing)
        if form.is_valid():
            if form.cleaned_data['status'] == 'paid':
                patient_id = billing.patient.id
                admission = Admission.objects.filter(patient_id=patient_id).first()
                if admission:
                    discharged_patient = DischargedPatient.objects.create(
                        patient=admission.patient,
                        bed_name=admission.bed.bed_name,
                        admission_datetime=admission.admission_datetime,
                        discharge_datetime=timezone.now()
                    )
                    admission.delete()

                form.save()
                return redirect('view_billings')
            else:
                form = BillingForm(instance=billing)

    return render(request, 'update_billing.html', {'form': form})
```

AdminProfile(editing only):

```
@login_required
def update_admin_profile(request):
    if request.method == 'POST':
        form = AdminProfileForm(request.POST, instance=request.user)
        if form.is_valid():
            form.save()
            return redirect('custom_admin_dashboard')
    else:
        form = AdminProfileForm(instance=request.user)

    return render(request, 'update_admin_profile.html', {'form': form})
```

Forms for search and filtering data:

```
class UserFilterForm(forms.Form):
    GENDER_CHOICES = [(' ', '-----')] + UserProfile.GENDER_CHOICES
    ACCOUNT_TYPE_CHOICES = [(' ', '-----')] + UserProfile.ACCOUNT_TYPE_CHOICES

    search_query = forms.CharField(label='Search by Username', required=False)
    gender = forms.ChoiceField(choices=GENDER_CHOICES, required=False, initial='')
    account_type = forms.ChoiceField(choices=ACCOUNT_TYPE_CHOICES, required=False, initial='')

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['gender'].initial = ''
        self.fields['account_type'].initial = ''

class PatientFilterForm(forms.Form):
    search_query = forms.CharField(label='Search by Username', required=False)

class DoctorFilterForm(forms.Form):
    search_query = forms.CharField(label='Search by Username', required=False)
    department = forms.ModelChoiceField(queryset=Department.objects.all(), required=False, empty_label='Select department')

class DoctorUserFilterForm(forms.Form):
    search_query = forms.CharField(label='Search by Name', required=False)
    department = forms.ModelChoiceField(queryset=Department.objects.all(), required=False, empty_label='Select department')

class WardFilterForm(forms.Form):
    WARD_CHOICES = [(' ', '-----')] + list(Ward.WARD_TYPES)

    ward_type = forms.ChoiceField(choices=WARD_CHOICES, required=False, label='Ward Type', initial='')

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['ward_type'].initial = ''

class BedFilterForm(forms.Form):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        ward_choices = [(' ', '-----')] + [(ward.id, ward) for ward in Ward.objects.all()]
        self.fields['ward'] = forms.ChoiceField(choices=ward_choices, required=False, label='Ward')

    bed_number = forms.IntegerField(required=False, label='Bed Number')
```

```

class BillingFilterForm(forms.Form):
    search_query = forms.IntegerField(label='Search by Billing ID',required=False )
    date = forms.DateField(required=False, label='Date')
    STATUS_CHOICES = [('', '-----')] + Billing.STATUS_CHOICES
    status = forms.ChoiceField(choices=STATUS_CHOICES, required=False, label='Status')
    PAYMENT_METHOD_CHOICES = [('', '-----')] + Billing.PAYMENT_METHOD_CHOICES
    payment_method = forms.ChoiceField(choices=PAYMENT_METHOD_CHOICES, required=False, label='Payment Method')

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs) |
        self.fields['status'].initial = ''

class AdmissionFilterForm(forms.Form):
    search_query = forms.CharField(label='Search by Username or Name', required=False)
    WARD_TYPES = [('', '-----')] + Ward.WARD_TYPES
    ward_type = forms.ChoiceField(choices=WARD_TYPES, required=False, label='Ward Type')
    sort_by = forms.ChoiceField(
        label='Sort by',
        choices=(
            ('', 'Select'),
            ('admission_datetime', 'Admission Date'),
        ),
        required=False
    )

class DoctorRequestFilterForm(forms.Form):
    search_query = forms.CharField(label='Search by Username', required=False)

```

Viewing tables with search and filter functionality:

Users:

```

@login_required
def view_users(request):
    form = UserFilterForm(request.GET)
    all_users = UserProfile.objects.filter(account_type__in=['patient', 'doctor'], is_approved = True)

    if form.is_valid():
        search_query = form.cleaned_data.get('search_query')
        gender = form.cleaned_data.get('gender')
        account_type = form.cleaned_data.get('account_type')

        if search_query:
            all_users = all_users.filter(username__icontains=search_query)

        if gender:
            all_users = all_users.filter(gender=gender)

        if account_type:
            all_users = all_users.filter(account_type=account_type)

    all_users = all_users.order_by('username')

    return render(request, 'view_users.html', {'all_users': all_users, 'form': form})

```

Patients:

```

@login_required
def view_patients(request):
    form = PatientFilterForm(request.GET)
    all_patients = PatientProfile.objects.all()
    if form.is_valid():
        search_query = form.cleaned_data.get('search_query')

        if search_query:
            all_patients = all_patients.filter(user_profile__username__icontains=search_query)

    all_patients = all_patients.order_by('user_profile__username')

    return render(request, 'view_patients.html', {'all_patients': all_patients, 'form': form})

```

Doctors:

```
@login_required
def view_doctors(request):
    form = DoctorFilterForm(request.GET)
    all_doctors = DoctorProfile.objects.select_related('user_profile').filter(user_profile__is_approved=True)

    if form.is_valid():
        search_query = form.cleaned_data.get('search_query')
        department = form.cleaned_data.get('department')

        if search_query:
            all_doctors = all_doctors.filter(user_profile__username__icontains=search_query)

        if department:
            all_doctors = all_doctors.filter(department = department)

    all_doctors = all_doctors.order_by("user_profile__username")

    return render(request, 'view_doctors.html', {'all_doctors': all_doctors, 'form': form})
```

Doctor Account Pending Requests:

- Doctor type accounts that are not approved will be shown only

```
@login_required
def doctor_nurse_requests(request):
    form = DoctorRequestFilterForm(request.GET)
    doctor_requests = UserProfile.objects.filter(account_type='doctor', is_approved=False).order_by('-date_joined')

    search_query = request.GET.get('search_query')
    if search_query:
        doctor_requests = doctor_requests.filter(username__icontains=search_query)

    doctor_requests = doctor_requests.order_by('-date_joined')

    doctor_details = []
    for doctor_request in doctor_requests:
        doctor_info = {
            'username': doctor_request.username,
            'id': doctor_request.id,
            'first_name': doctor_request.first_name,
            'last_name': doctor_request.last_name,
            'specialization': doctor_request.doctorprofile.specialization,
            'degree': doctor_request.doctorprofile.qualifications,
            'date_joined': doctor_request.date_joined
        }
        doctor_details.append(doctor_info)

    return render(request, 'doctor_nurse_requests.html', {'doctor_details': doctor_details, 'form': form})
```

Accept Doctor:

- Approves doctor by setting is_approved attribute to true

```
@login_required
def approve_account(request, user_id):
    user_profile = UserProfile.objects.get(pk=user_id)
    user_profile.is_approved = True
    user_profile.save()
    return redirect('doctor_nurse_requests')
```

Reject Doctor:

- Rejecting deletes the request and the profile itself. (To send request again, new userprofile and doctor profile needs to be created)

```
@login_required
def reject_account(request, user_id):
    user_profile = UserProfile.objects.get(pk=user_id)
    user_profile.delete()
    return redirect('doctor_nurse_requests')
```

Admission:

```
@login_required
def view_admissions(request):
    form = AdmissionFilterForm(request.GET)
    admissions = Admission.objects.select_related('patient__user_profile')

    if form.is_valid():
        search_query = form.cleaned_data.get('search_query')
        ward_type = form.cleaned_data.get('ward_type')
        sort_by = form.cleaned_data.get('sort_by')

        if search_query:
            admissions = admissions.filter(
                Q(patient__user_profile__username__icontains=search_query) |
                Q(patient__user_profile__first_name__icontains=search_query) |
                Q(patient__user_profile__last_name__icontains=search_query)
            )

        if ward_type:
            admissions = admissions.filter(bed__ward__ward_type=ward_type)

        if sort_by:
            admissions = admissions.order_by(f'~{sort_by}')

    return render(request, 'view_admissions.html', {'admissions': admissions, 'form': form})
```

DischargedPatients:

- No search and filter option

```
def view_discharged_patients(request):
    discharged_patients = DischargedPatient.objects.all().order_by('-discharge_datetime')
    return render(request, 'view_discharged_patients.html', {'discharged_patients': discharged_patients})
```

Department:

- No search and filter, by default sorted by alphabetical order.

```
@login_required
def view_departments(request):
    departments = Department.objects.all().order_by("name")

    context = {
        'departments': departments,
    }

    return render(request, 'view_departments.html', context)
```

Wards:

```
@login_required
def view_wards(request):
    form = WardFilterForm(request.GET)
    wards = Ward.objects.all().order_by('name')

    if form.is_valid():
        name = form.cleaned_data.get('name')
        ward_type = form.cleaned_data.get('ward_type')

        if name:
            wards = wards.filter(name__icontains=name)
        if ward_type:
            wards = wards.filter(ward_type=ward_type)

    context = {
        'wards': wards,
        'form': form,
    }

    return render(request, 'view_wards.html', context)
```

Shifts:

- No search and filter

```
@login_required
def view_shifts(request):
    shifts = Shift.objects.all()
    return render(request, 'view_shifts.html', {'shifts': shifts})
```

Beds:

```
@login_required
def view_beds(request):
    form = BedFilterForm(request.GET)
    beds = Bed.objects.all().order_by('ward__name')

    ward_id = request.GET.get('ward')
    bed_number = request.GET.get('bed_number')

    if ward_id:
        beds = beds.filter(ward_id=ward_id)

    if bed_number:
        beds = beds.filter(bed_number=bed_number)

    return render(request, 'view_beds.html', {'form': form, 'beds': beds})
```

Billings:

- Sorted by latest date

```
@login_required
def view_billings(request):
    form = BillingFilterForm(request.GET)
    billings = Billing.objects.all().order_by('-date')

    if form.is_valid():
        search_query = form.cleaned_data.get('search_query')
        date = form.cleaned_data.get('date')
        status = form.cleaned_data.get('status')
        payment_method = form.cleaned_data.get('payment_method')

        if search_query:
            billings = billings.filter(id=search_query)
        if date:
            billings = billings.filter(date=date)
        if status:
            billings = billings.filter(status=status)
        if payment_method:
            billings = billings.filter(payment_method=payment_method)

    return render(request, 'view_billings.html', {'billings': billings, 'form': form})
```

Deletion of tables:

User:

```
def delete_user(request, user_id):
    user = get_object_or_404(UserProfile, pk=user_id)

    if request.method == 'POST':
        user.delete()
        return redirect('view_users')

    return render(request, 'delete_confirmation.html', {'user': user})
```

Patient:

```
def delete_patient(request, patient_id):
    patient_profile = get_object_or_404(PatientProfile, pk=patient_id)

    if request.method == 'POST':
        patient_profile.user_profile.delete()
        return redirect('view_patients')

    return render(request, 'delete_confirmation.html', {'patient_profile': patient_profile})
```

Doctor:

```
def delete_doctor(request, doctor_id):
    doctor = get_object_or_404(UserProfile, pk=doctor_id, account_type='doctor')

    if request.method == 'POST':
        doctor.delete()
        return redirect('view_doctors')

    return render(request, 'delete_confirmation.html', {'doctor': doctor})
```

Admission:

- Deleting admission deletes the billing for that admission (helps in handling billing)

```
def delete_admission(request, admission_id):
    admission = get_object_or_404(Admission, pk=admission_id)
    if request.method == 'POST':
        patient = admission.patient
        billing = Billing.objects.filter(patient=patient).first()
        if billing:
            billing.delete()
        admission.delete()
        return redirect('view_admissions')
    return render(request, 'delete_confirmation.html', {'admission': admission})
```


Department:

```
def delete_department(request, department_id):
    department = get_object_or_404(Department, pk=department_id)
    if request.method == 'POST':
        department.delete()
        return redirect('view_departments')
    return render(request, 'delete_confirmation.html', {'department': department})
```

Ward:

```
def delete_ward(request, ward_id):
    ward = get_object_or_404(Ward, pk=ward_id)
    if request.method == 'POST':
        ward.delete()
        return redirect('view_wards')
    return render(request, 'delete_confirmation.html', {'ward': ward})
```

Shift:

```
def delete_shift(request, shift_id):
    shift = get_object_or_404(Shift, pk=shift_id)
    if request.method == 'POST':
        shift.delete()
        return redirect('view_shifts')
    return render(request, 'delete_confirmation.html', {'shift': shift})
```

Bed:

```
def delete_bed(request, bed_id):
    bed = get_object_or_404(Bed, pk=bed_id)
    if request.method == 'POST':
        bed.delete()
        return redirect('view_beds')
    return render(request, 'delete_confirmation.html', {'bed': bed})
```

Billing:

```
def delete_billing(request, billing_id):
    billing = get_object_or_404(Billing, pk=billing_id)
    if request.method == 'POST':
        billing.delete()
        return redirect('view_billings')
    return render(request, 'delete_confirmation.html', {'billing': billing})
```

URLS for connecting backend with frontend:

```
#Admin
path('custom_admin_dashboard/', views.custom_admin_dashboard, name='custom_admin_dashboard'),
path('edit-admin-profile/', views.update_admin_profile, name='edit_admin_profile'),

path('custom_admin_dashboard/create_admission/', views.create_admission, name='create_admission'),
path('view_admissions/', views.view_admissions, name='view_admissions'),
path('admissions/<int:admission_id>/edit/', views.update_admission, name='edit_admission'),
path('admissions/<int:admission_id>/delete/', views.delete_admission, name='delete_admission'),
path('discharged_patients/', views.view_discharged_patients, name='view_discharged_patients'),

path('create_department/', views.create_department, name='create_department'),
path('view_departments/', views.view_departments, name='view_departments'),
path('departments/<int:department_id>/edit/', views.update_department, name='edit_department'),
path('departments/<int:department_id>/delete/', views.delete_department, name='delete_department'),

path('ward/create/', views.create_ward, name='create_ward'),
path('ward/list/', views.view_wards, name='view_wards'),
path('wards/<int:ward_id>/edit/', views.update_ward, name='edit_ward'),
path('wards/<int:ward_id>/delete/', views.delete_ward, name='delete_ward'),

path('shifts/create/', views.create_shift, name='create_shift'),
path('shift/list/', views.view_shifts, name='view_shifts'),
path('shifts/<int:shift_id>/edit/', views.update_shift, name='edit_shift'),
path('shifts/<int:shift_id>/delete/', views.delete_shift, name='delete_shift'),

path('create_billing/', views.create_billing, name='create_billing'),
path('view_billings/', views.view_billings, name='view_billings'),
path('billings/<int:billing_id>/delete/', views.delete_billing, name='delete_billing'),
path('billings/<int:billing_id>/edit/', views.update_billing, name='edit_billing'),

path('create_bed/', views.create_bed, name='create_bed'),
path('beds/', views.view_beds, name='view_beds'),
path('beds/<int:bed_id>/edit/', views.update_bed, name='edit_bed'),
path('beds/<int:bed_id>/delete/', views.delete_bed, name='delete_bed'),

path('view-patients/', views.view_patients, name='view_patients'),
path('create-patient/', views.create_patient, name='create_patient'),
path('edit-patient/<int:patient_id>/', views.update_patient, name='edit_patient'),
path('delete-patient/<int:patient_id>/', views.delete_patient, name='delete_patient'),

path('create_doctor/', views.create_doctor, name='create_doctor'),
path('view_doctors/', views.view_doctors, name='view_doctors'),
path('edit-doctor/<int:doctor_id>/', views.update_doctor, name='edit_doctor'),
path('delete-doctor/<int:doctor_id>/', views.delete_doctor, name='delete_doctor'),

path('create_user/', views.create_user, name='create_user'),
path('view_users/', views.view_users, name='view_users'),
path('edit_user/<int:user_id>/', views.update_user, name='edit_user'),
path('delete_user/<int:user_id>/', views.delete_user, name='delete_user'),

path('doctor-nurse-requests/', views.doctor_nurse_requests, name='doctor_nurse_requests'),
path('approve-account/<int:user_id>/', views.approve_account, name='approve_account'),
path('reject-account/<int:user_id>/', views.reject_account, name='reject_account'),
```

Contribution of ID : 23341111, Name : Fardin Hassan Tamim

Urls:

```
@login_required
def patient_dashboard(request):

    user_profile = request.user
    patient_profile = PatientProfile.objects.get(user_profile=user_profile)
    appointments = Appointment.objects.filter(patient=patient_profile)
    admission = Admission.objects.get(patient=patient_profile)
    paymentStatus = Billing.objects.get(patient=patient_profile)
    medical_history = patient_profile.medical_history
    prescribed_medications = patient_profile.prescribed_medications

    search_query = request.GET.get('search_query', '')
    doctors = UserProfile.objects.filter(
        Q(first_name__icontains=search_query) |
        Q(last_name__icontains=search_query),
        account_type='doctor'
    )

    sort_order = request.GET.get('sort', '')
    if sort_order == 'asc':
        doctors = doctors.order_by('first_name')
    elif sort_order == 'desc':
        doctors = doctors.order_by('-first_name')

    context = {
        'appointments': appointments,
        'medical_history': medical_history,
        'prescribed_medications': prescribed_medications,
        'doctors': doctors,
        'admission': admission,
        'paymentStatus': paymentStatus,
    }

    return render(request, 'patient_dashboard.html', context)
```

```
#PatientProfile
path('patient/dashboard/', views.patient_dashboard, name='patient_dashboard'),
path('patient/signup/<int:user_profile_id>/', views.patient_signup_view, name='patient_signup'),
path('update_profile/', views.update_profile, name='update_profile'),
```

Contribution of ID : 21301661, Name : Meherun Chowdhury Dola

```
def appointment_view(request, doctorID):
    if request.method == 'POST':

        patient_id = request.POST.get('patient')
        date = request.POST.get('date')
        time = request.POST.get('time')
        problem = request.POST.get('problem')
        status = request.POST.get('status')

        if not patient_id:
            return HttpResponse("Invalid patient ID")
        try:
            patient_id = int(patient_id)
            doctor_id = int(doctorID)
        except ValueError:
            return HttpResponse("Invalid patient or doctor ID")
        patient = get_object_or_404(PatientProfile, id=patient_id)
        doctor = get_object_or_404(DoctorProfile, id=doctor_id)
        appointment = Appointment.objects.create(
            patient=patient,
            doctor=doctor,
            date=date,
            time=time,
            problem=problem,
            status="Scheduled"
        )
        appointment.save()
        return redirect('success')
    return render(request, 'appointment.html', {'doctorId': doctorID})
```

```
path('appointment_view/<int:doctorID>/', views.appointment_view, name='appointment_view'),
```

Contribution of ID : 21301142, Name : AL-AMIN SAGOR

Contribution of ID : 20101466, Name : MD. GOLAM SHAHNEWAZ

Conclusion

Our Hospital Management System stands as a comprehensive solution aimed at enhancing healthcare operations and patient care. By leveraging Django and a robust database system, we've streamlined patient management, admissions, and billing processes. Overcoming challenges along the way, we've prioritized data security measures to safeguard sensitive patient information. Looking ahead, there's room for future expansion, including advanced analytics and reporting features. This project has been a valuable learning experience, emphasizing the importance of efficiency and security in healthcare technology. Ultimately, our HMS endeavors to make a meaningful impact on hospital administration and patient well-being.

References

 **How to Create Responsive Navigation Bar using HTML and CSS**