**Patient Medicine and Appointment System**

**1. Project Overview**

The Hospital Management System (HMS) is a web-based application developed using **Spring Boot 6**, **HTML**, **CSS**, **JavaScript**, and **Thymeleaf**. It streamlines hospital operations by managing appointments, doctors, patients, medications, and profiles.

**2. Key Modules and Features**

**2.1. Patient Management**

* **Registration & Login:** Patients can register and log in.
* **Dashboard:** View upcoming appointments and medications.
* **Profile Management:** Update profile information.
* **Logout:** End session securely.

**2.2. Doctor Management**

* **Registration & Login:** Doctors can register and log in.
* **Dashboard:** View appointments and patient details.
* **Profile Management:** Update profile information.
* **Logout:** End session securely.

**2.3. Appointment Management**

* **Book Appointment:** Patients can schedule appointments.
* **View Appointments:** List of all upcoming and past appointments.
* **Update/Cancel Appointment:** Reschedule or delete appointments.

**2.4. Medication Management**

* **Add Medication:** Add prescription details for patients.
* **View & Update:** Manage medication information.
* **Delete Medication:** Remove medication records.

**3. Technology Stack**

**Backend:**

* **Spring Boot 6:** Main application framework.
* **Java 17+**: Programming language.
* **JPA/Hibernate:** Database ORM.
* **H2 / MySQL:** Database options.

**Frontend:**

* **HTML5 / CSS3 / Bootstrap:** UI design.
* **Thymeleaf:** Template engine.
* **JavaScript:** Frontend interactivity.

**4. Database Design**

**4.1. Database Tables**

* **Patient:** id, name, email, password, phone
* **Doctor:** id, name, email, password, specialization
* **Appointment:** id, doctor\_id, patient\_id, date\_time
* **Medication:** id, name, dosage, instructions, patient\_id

**5. Controller Structure**

**5.1. PatientController**

* /patients/register – Show registration form
* /patients/login – Handle login
* /patients/dashboard – Show patient dashboard
* /patients/profile – Update patient profile
* /patients/logout – Logout patient

**5.2. DoctorController**

* /doctors/doctor-registration – Show doctor registration form
* /doctors/doctor-login – Handle login
* /doctors/doctor-dashboard – Show doctor dashboard
* /doctors/profile – Update doctor profile
* /doctors/logout – Logout doctor

**5.3. AppointmentController**

* /appointments/bookform – Show booking form
* /appointments/book – Handle booking
* /appointments/{id} – Show appointment details
* /appointments/edit/{id} – Update appointment
* /appointments/delete/{id} – Delete appointment

**5.4. MedicationController**

* /medications/add – Show add medication form
* /medications/{id} – Show medication details
* /medications/edit/{id} – Update medication
* /medications/delete/{id} – Delete medication

**6. API Endpoints**

* **GET /patients/{id}** – Get patient details
* **POST /patients/register** – Register new patient
* **POST /doctors/register** – Register new doctor
* **POST /appointments/book** – Book an appointment
* **PUT /appointments/edit/{id}** – Update appointment
* **DELETE /appointments/delete/{id}** – Cancel appointment

**7. Unit Testing**

* **JUnit 5 & Mockito:** Used for writing unit tests.
* **Controller, Service, Repository:** Comprehensive test coverage.

**8. Error Handling**

* **ResourceNotFoundException:** For missing records.
* **ResourceAlreadyExistsException:** For duplicate records.