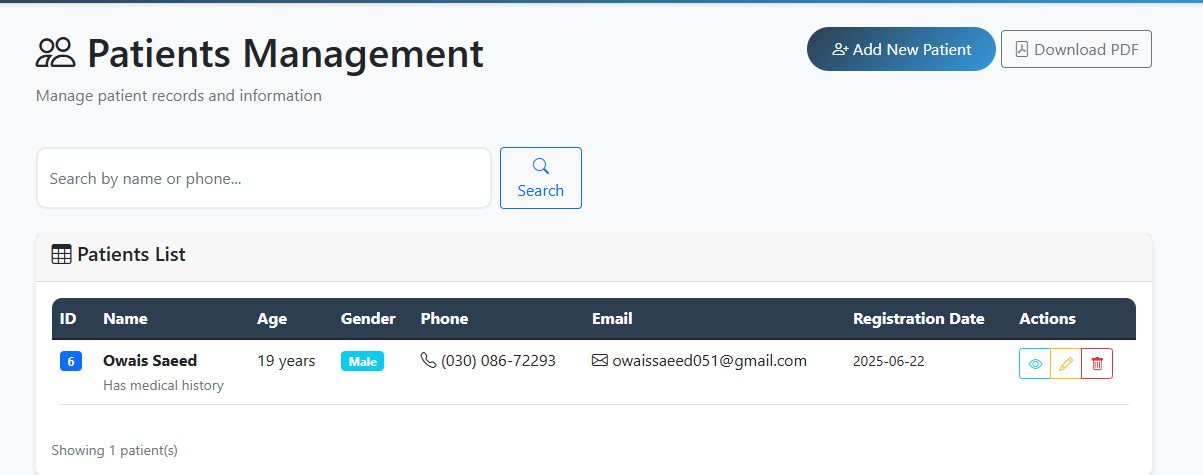
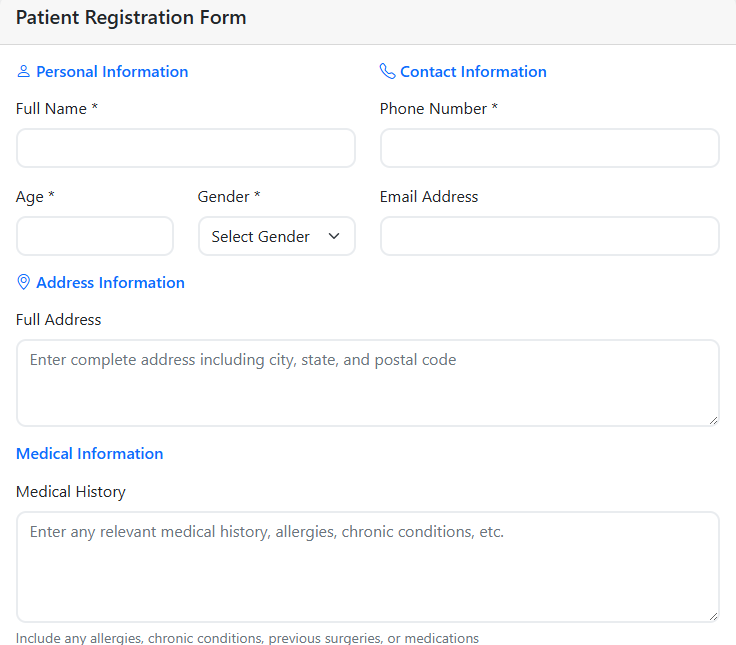
**Hospital Management System - Project Report**

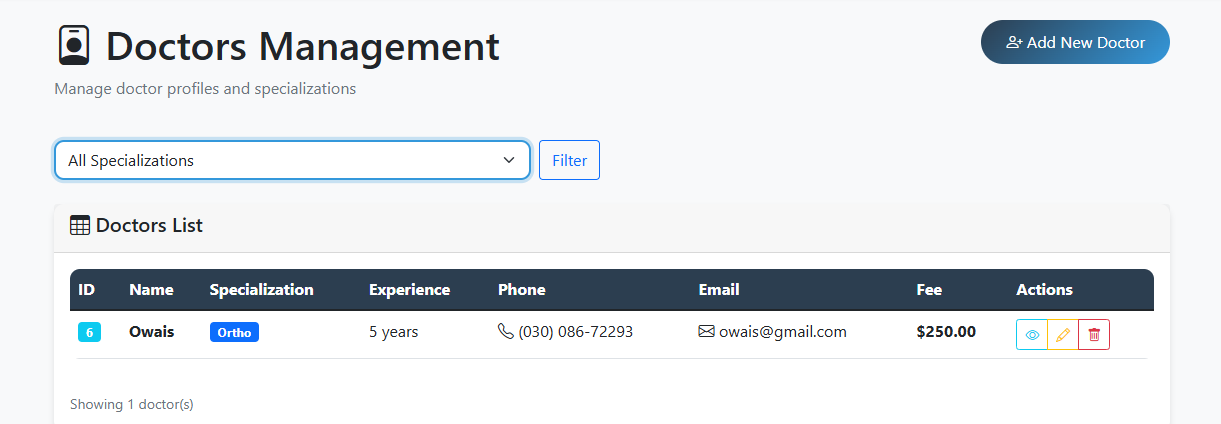
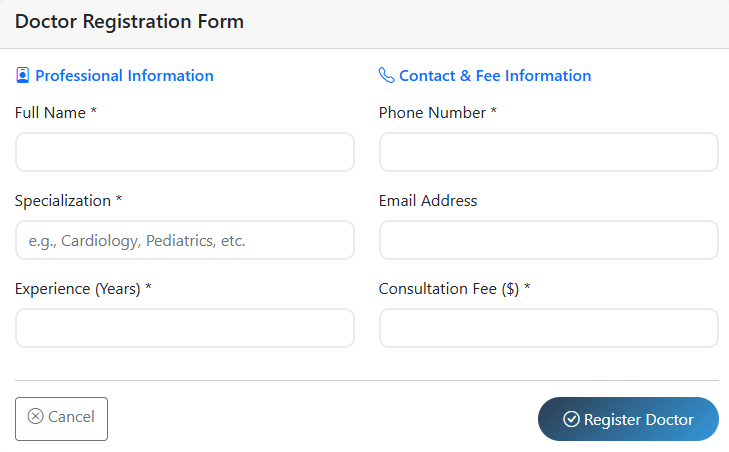
**Overview**  
The Hospital Management System (HMS) is a web-based software solution built using **Flask (Python)** for the backend and **Bootstrap** for the frontend. It is designed to efficiently manage hospital operations such as patient registrations, doctor records, appointment scheduling, and staff administration. The system integrates a structured MySQL database and provides a user-friendly interface for hospital personnel to manage daily healthcare operations.

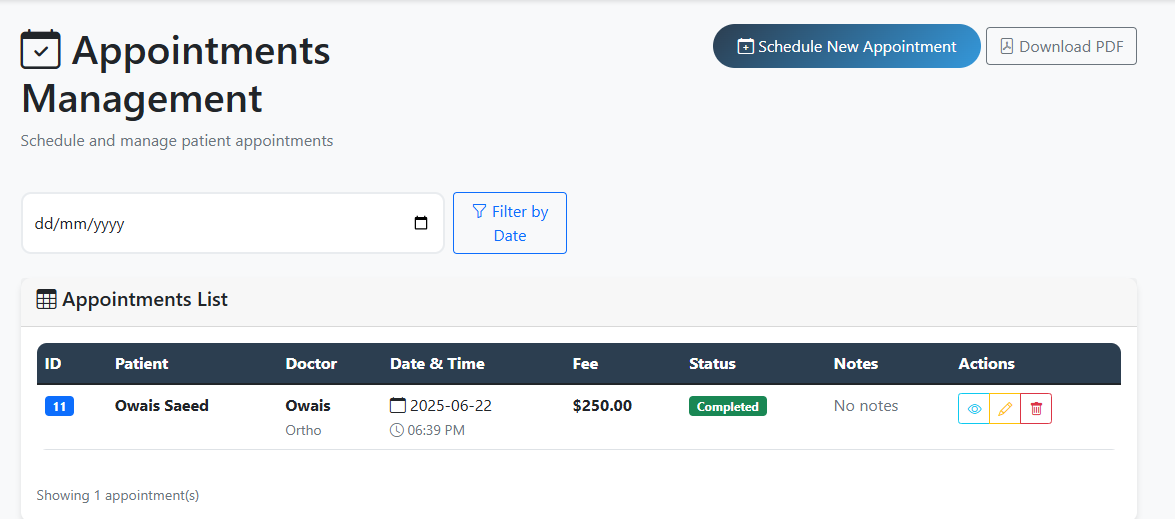
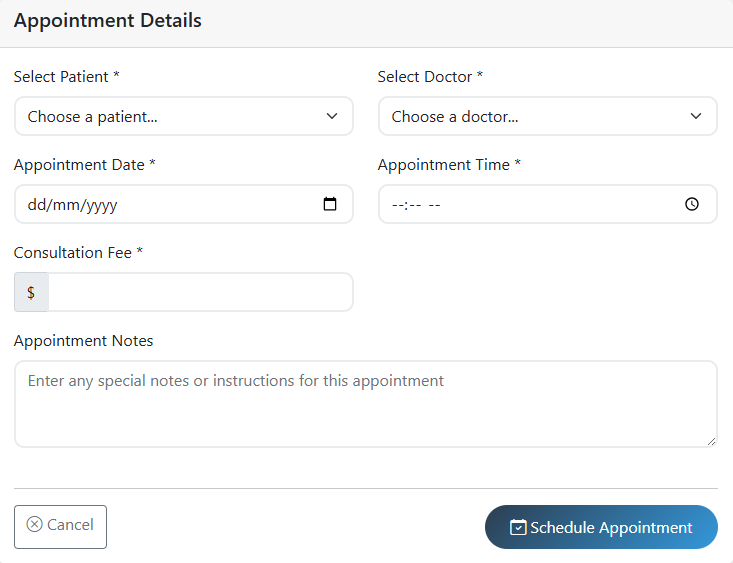
**Key Features**  
The system includes the following core functionalities:

1. **Patient Management**
   * Patient Registration: Add and manage patient details including age, gender, contact info, and medical history.
   * Search and View: Find patients by name, phone, or creation date. 



1. **Doctor Management**
   * Specialization-Based Listing: Manage doctors by their fields such as Cardiology, Pediatrics, etc.
   * Experience & Fee Details: Track doctor experience and consultation charges.

1. **Appointment Scheduling**
   * Book Appointments: Assign patients to doctors at specific times.
   * Appointment Status: Track whether appointments are Scheduled, Completed, or Cancelled.
   * View Appointments: Detailed view showing patient-doctor mapping and appointment history.  
2. **Staff Management**
   * User Roles: Define roles such as Admin, Doctor, Nurse, and Receptionist.
   * Secure Login: Uses SHA-256 password hashing for login credentials.
3. **Views and Reports**
   * Appointment Details View: Displays joined data for easier reporting.
   * Patient Summary View: Shows total appointments and recent activity.
4. **Frontend & User Interface**
   * Responsive Design: Developed with **Bootstrap**, providing a mobile-friendly and clean interface.
   * Interactive UI: Includes forms, tables, buttons, and modals for a smooth user experience.
5. **Backend Integration**
   * Developed with **Flask** in Python.
   * Database connection through Flask's MySQL integration (either raw MySQL or SQLAlchemy).
   * Routes handle CRUD operations for patients, doctors, appointments, and staff.

**Database Schema**  
The database is structured in MySQL with the following key tables:

1. **patients** – Stores patient personal and medical info
2. **doctors** – Stores doctor profile, specialization, and consultation fee
3. **appointments** – Handles booking and status tracking
4. **staff** – Admin, doctor, nurse, and receptionist credentials

**Sample SQL Commands Used:**

CREATE DATABASE IF NOT EXISTS HMS;

USE HMS;

[SQL table creation and inserts are detailed in the full script above.]

**Sample Flask Route (Python):**

@app.route('/add\_patient', methods=['POST'])

def add\_patient():

name = request.form['name']

age = request.form['age']

gender = request.form['gender']

phone = request.form['phone']

email = request.form['email']

address = request.form['address']

history = request.form['history']

cur = mysql.connection.cursor()

cur.execute("""

INSERT INTO patients (name, age, gender, phone, email, address, medical\_history)

VALUES (%s, %s, %s, %s, %s, %s, %s)

""", (name, age, gender, phone, email, address, history))

mysql.connection.commit()

return redirect(url\_for('patient\_list'))

**Sample Bootstrap Form:**

<form method="post" action="/add\_patient">

<input type="text" class="form-control" name="name" placeholder="Patient Name" required>

<input type="number" class="form-control" name="age" placeholder="Age" required>

<select class="form-select" name="gender">

<option>Male</option>

<option>Female</option>

<option>Other</option>

</select>

<button type="submit" class="btn btn-primary">Add Patient</button>

</form>

**System Workflow Summary**

* Admin logs in via the staff table.
* Admin/staff can add patients, doctors, and appointments.
* All appointments are linked via foreign keys.
* Views simplify data reporting.
* Frontend built using Bootstrap forms and tables.

**Conclusion**  
The Hospital Management System streamlines healthcare workflows by integrating patient care, staff administration, and appointment scheduling into a single platform. With Flask powering the backend and Bootstrap enhancing the frontend UI, this system is scalable, maintainable, and user-friendly for real-time hospital operations.