



Hospital Management System



A Command-Line System to Manage Patients, Doctors, and Operations Efficiently

💡 Project Overview

This project is a simple yet powerful **Hospital Management System** developed using Python. It demonstrates how to structure and implement core functionalities of a hospital's operational workflow in a scalable and interactive way.

PYTHON CODE :

```
In [9]: # Hospital Management System

class Patient:
    def __init__(self, name, age, disease):
        self.name = name
        self.age = age
        self.disease = disease
        self.is_discharged = False
        self.assigned_doctor = None

    def assign_doctor(self, doctor):
        self.assigned_doctor = doctor

    def discharge(self):
        self.is_discharged = True
        print(f"{self.name} has been discharged successfully.")

class Doctor:
    def __init__(self, name, specialization):
        self.name = name
        self.specialization = specialization

    def display_doctor_info(self):
        print(f"Doctor: {self.name} | Specialization: {self.specialization}")

class Hospital:
    def __init__(self):
        self.patients = {}
        self.doctors = {}

    def add_patient(self, patient):
        self.patients[patient.name] = patient
```

```

def add_doctor(self, doctor):
    self.doctors[doctor.name] = doctor

def view_patients(self):
    if not self.patients:
        print("No patients in the hospital.")
    else:
        for patient in self.patients.values():
            print(f"Patient: {patient.name}, Age: {patient.age}, Disease: {patient.disease}, Discharged: {patient.is_discharged}")
            if patient.assigned_doctor:
                print(f"Assigned Doctor: {patient.assigned_doctor.name}")

def assign_doctor_to_patient(self, patient_name, doctor_name):
    if patient_name in self.patients and doctor_name in self.doctors:
        patient = self.patients[patient_name]
        doctor = self.doctors[doctor_name]
        patient.assign_doctor(doctor)
        print(f"Doctor {doctor_name} has been assigned to {patient_name}.")
    else:
        print("Invalid patient or doctor name.")

def discharge_patient(self, patient_name):
    if patient_name in self.patients:
        patient = self.patients[patient_name]
        if not patient.is_discharged:
            patient.discharge()
        else:
            print(f"{patient_name} is already discharged.")
    else:
        print("Patient not found.")

```

```

def menu():
    print("\n== Hospital Management System ==")
    print("1. Add Patient")
    print("2. Add Doctor")
    print("3. View Patients")
    print("4. Assign Doctor to Patient")
    print("5. Discharge Patient")
    print("6. Exit")

def main():
    hospital = Hospital()

    while True:
        menu()
        choice = input("Enter your choice (1-6): ")

        if choice == "1":
            name = input("Enter patient name: ")
            age = int(input("Enter patient age: "))
            disease = input("Enter patient disease: ")
            patient = Patient(name, age, disease)
            hospital.add_patient(patient)
            print(f"Patient {name} added successfully.")

        elif choice == "2":
            name = input("Enter doctor name: ")
            specialization = input("Enter doctor's specialization: ")
            doctor = Doctor(name, specialization)
            hospital.add_doctor(doctor)
            print(f"Doctor {name} added successfully.")

```

```
        elif choice == "3":  
            hospital.view_patients()  
  
        elif choice == "4":  
            patient_name = input("Enter patient name to assign doctor: ")  
            doctor_name = input("Enter doctor name to assign: ")  
            hospital.assign_doctor_to_patient(patient_name, doctor_name)  
  
        elif choice == "5":  
            patient_name = input("Enter patient name to discharge: ")  
            hospital.discharge_patient(patient_name)  
  
        elif choice == "6":  
            print("Exiting the Hospital Management System. Goodbye!")  
            break  
  
    else:  
        print("Invalid choice! Please try again.")  
  
if __name__ == "__main__":  
    main()
```

⚙️ Features

1. **Add New Patients:** Register patient details including name, age, and disease.
2. **Add New Doctors:** Record doctors' names and their specializations (e.g., Cardiologist, Surgeon).
3. **Assign Doctors:** Seamlessly link doctors to patients based on their specialization.
4. **View Patients:** View a list of all registered patients, their diseases, and assigned doctors.
5. **Discharge Patients:** Track patient discharges and update their status in real-time.
6. **Interactive Menu:** A user-friendly menu system for ease of navigation.

🛠️ Technologies Used

- **Programming Language:** Python
- **Concepts Applied:** Object-Oriented Programming (OOP), Data Management, User Input Handling

OUTPUT :

```
==> Hospital Management System ==>
1. Add Patient
2. Add Doctor
3. View Patients
4. Assign Doctor to Patient
5. Discharge Patient
6. Exit
Enter your choice (1-6): 2
Enter doctor name: Praveen Kumar
Enter doctor's specialization: Cardiologist
Doctor Praveen Kumar added successfully.

==> Hospital Management System ==>
1. Add Patient
2. Add Doctor
3. View Patients
4. Assign Doctor to Patient
5. Discharge Patient
6. Exit
Enter your choice (1-6): 1
Enter patient name: Chris Lyon
Enter patient age: 29
Enter patient disease: Heart
Patient Chris Lyon added successfully.

==> Hospital Management System ==>
1. Add Patient
2. Add Doctor
3. View Patients
4. Assign Doctor to Patient
5. Discharge Patient
6. Exit
Enter your choice (1-6): 4
Enter patient name to assign doctor: Chris Lyon
Enter doctor name to assign: Praveen Kumar
Doctor Praveen Kumar has been assigned to Chris Lyon.
```

```
==> Hospital Management System ==>
1. Add Patient
2. Add Doctor
3. View Patients
4. Assign Doctor to Patient
5. Discharge Patient
6. Exit
Enter your choice (1-6): 3
Patient: Chris Lyon, Age: 29, Disease: Heart, Discharged: False
Assigned Doctor: Praveen Kumar

==> Hospital Management System ==>
1. Add Patient
2. Add Doctor
3. View Patients
4. Assign Doctor to Patient
5. Discharge Patient
6. Exit
Enter your choice (1-6): 5
Enter patient name to discharge: Chris Lyon
Chris Lyon has been discharged successfully.

==> Hospital Management System ==>
1. Add Patient
2. Add Doctor
3. View Patients
4. Assign Doctor to Patient
5. Discharge Patient
6. Exit
Enter your choice (1-6): 3
Patient: Chris Lyon, Age: 29, Disease: Heart, Discharged: True
Assigned Doctor: Praveen Kumar
```

Real-World Applications

This project models the foundation of a **Hospital Management Software** that can:

- Optimize patient and doctor management.
 - Serve as a learning resource for building real-world management systems.
 - Be extended with advanced features like database integration, GUI, or web support.
-

 Explore the Code on GitHub : <https://github.com/praveenk20104>

 Let's Connect on LinkedIn : <https://www.linkedin.com/in/praveen-kumar-590517277/>