

SET-1

Title: Hospital Appointment Management System (Console-based Application)

Scenario:

A local hospital needs an appointment scheduling system to manage doctor-patient appointments through a console-based C# application. The hospital has different doctors specializing in various departments. Patients can book appointments with available doctors by choosing a date and time slot. Each doctor can attend only 5 patients per day.

You are hired as a C# developer to build a solution for this.

Tasks:

Task 1: Define Classes and Data Members

☐ **Create a class Doctor with properties:**

o int DoctorId

o string Name

o string Department

☐ **Create a class Appointment with properties:**

o int AppointmentId

o string PatientName

o int DoctorId

o DateTime AppointmentDate

☐ **Create a manager class HospitalSystem to handle logic and maintain lists of doctors and appointments.**

Task 2: Add Default Doctors

☐ **Inside the HospitalSystem class:**

o Add at least 3 predefined doctors.

o Store them using List<Doctor>.

Use a method AddDefaultDoctors() and call it in the constructor.

Task 3: Book an Appointment

□ Implement method `BookAppointment()` that:

- o Takes patient name, doctor ID, and date as input.
- o Validates if the doctor exists.
- o Checks if the doctor already has 5 appointments on that day.
- o If valid, adds the appointment to a `List<Appointment>`.

Task 4: Display Appointments

□ Implement method `ViewAppointments()` to:

- o Take a doctor ID and date.
- o Show all appointments booked with that doctor on that date.

Task 5: Input Validation and Constraints

□ Ensure:

- o Doctor ID entered exists.
- o Date is in correct format (use `DateTime.TryParse`).
- o No more than 5 appointments per doctor per day.

Task 6: User Interface

□ Create a menu-driven system with the following options:

1. View all available doctors
2. Book appointment
3. View appointments by doctor and date
4. Exit

Use `do-while` or `while(true)` with `switch-case` for menu logic.

Task 7: Output Sample

Provide formatted output such as:

Doctor ID: 101, Name: Dr. Asha, Department: Cardiology

Booking Successful! Appointment ID: 1, Patient: Ramesh, Doctor ID: 101, Date: 22-Jun-2025

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;

namespace HospitalAppointmentSystem
{
    // Doctor class
    public class Doctor
    {
        public int DoctorId { get; set; }
        public string Name { get; set; }
        public string Department { get; set; }
    }

    // Appointment class
    public class Appointment
    {
        public int AppointmentId { get; set; }
        public string PatientName { get; set; }
        public int DoctorId { get; set; }
        public DateTime AppointmentDate { get; set; }
    }

    // Hospital system manager class
    public class HospitalSystem
    {
        private List<Doctor> doctors = new List<Doctor>();
        private List<Appointment> appointments = new List<Appointment>();
        private int appointmentCounter = 1;

        public HospitalSystem()
        {
            AddDefaultDoctors();
        }

        private void AddDefaultDoctors()
        {
            doctors.Add(new Doctor { DoctorId = 101, Name = "Dr. Asha", Department =
"Cardiology" });
            doctors.Add(new Doctor { DoctorId = 102, Name = "Dr. Rakesh", Department =
"Neurology" });
            doctors.Add(new Doctor { DoctorId = 103, Name = "Dr. Meera", Department =
"Pediatrics" });
        }

        public void ViewDoctors()
        {

```

```

        Console.WriteLine("\nAvailable Doctors:");
        foreach (var doc in doctors)
        {
            Console.WriteLine($"Doctor ID: {doc.DoctorId}, Name: {doc.Name}, Department:
{doc.Department}");
        }
    }

    public void BookAppointment()
    {
        Console.Write("\nEnter Patient Name: ");
        string patientName = Console.ReadLine();

        Console.Write("Enter Doctor ID: ");
        if (!int.TryParse(Console.ReadLine(), out int doctorId) || !doctors.Any(d => d.DoctorId
== doctorId))
        {
            Console.WriteLine("Invalid Doctor ID.");
            return;
        }

        Console.Write("Enter Appointment Date (dd-MM-yyyy): ");
        if (!DateTime.TryParse(Console.ReadLine(), out DateTime date))
        {
            Console.WriteLine("Invalid date format.");
            return;
        }

        int count = appointments.Count(a => a.DoctorId == doctorId &&
a.AppointmentDate.Date == date.Date);
        if (count >= 5)
        {
            Console.WriteLine("This doctor already has 5 appointments on this date.");
            return;
        }

        appointments.Add(new Appointment
        {
            AppointmentId = appointmentCounter++,
            PatientName = patientName,
            DoctorId = doctorId,
            AppointmentDate = date
        });

        Console.WriteLine("\nBooking Successful!");
    }

    public void ViewAppointments()
    {
        Console.Write("\nEnter Doctor ID: ");
    }

```

```

        if (!int.TryParse(Console.ReadLine(), out int doctorId) || !doctors.Any(d => d.DoctorId
== doctorId))
        {
            Console.WriteLine("Invalid Doctor ID.");
            return;
        }

        Console.Write("Enter Date (dd-MM-yyyy): ");
        if (!DateTime.TryParse(Console.ReadLine(), out DateTime date))
        {
            Console.WriteLine("Invalid date format.");
            return;
        }

        var results = appointments.Where(a => a.DoctorId == doctorId &&
a.AppointmentDate.Date == date.Date).ToList();

        if (results.Count == 0)
        {
            Console.WriteLine("No appointments found.");
        }
        else
        {
            Console.WriteLine($"
Appointments for Doctor ID {doctorId} on {date:dd-MMM-
yyyy}:");
            foreach (var app in results)
            {
                Console.WriteLine($"Appointment ID: {app.AppointmentId}, Patient:
{app.PatientName}, Date: {app.AppointmentDate:dd-MMM-yyyy}");
            }
        }
    }
}

// Main Program
class Program
{
    static void Main(string[] args)
    {
        HospitalSystem system = new HospitalSystem();
        bool exit = false;

        while (!exit)
        {
            Console.WriteLine("
--- Hospital Appointment System ---");
            Console.WriteLine("1. View all available doctors");
            Console.WriteLine("2. Book appointment");
            Console.WriteLine("3. View appointments by doctor and date");
            Console.WriteLine("4. Exit");
            Console.Write("Choose an option: ");

```

```

switch (Console.ReadLine())
{
    case "1":
        system.ViewDoctors();
        break;
    case "2":
        system.BookAppointment();
        break;
    case "3":
        system.ViewAppointments();
        break;
    case "4":
        exit = true;
        Console.WriteLine("Thank you! Exiting...");
        break;
    default:
        Console.WriteLine("Invalid choice, try again.");
        break;
}
}
}
}
}

```

OUTPUT:

```

--- Hospital Appointment System ---
1. View all available doctors
2. Book appointment
3. View appointments by doctor and date
4. Exit
Choose an option: 1

Available Doctors:
Doctor ID: 101, Name: Dr. Asha, Department: Cardiology
Doctor ID: 102, Name: Dr. Rakesh, Department: Neurology
Doctor ID: 103, Name: Dr. Meera, Department: Pediatrics

```

```
--- Hospital Appointment System ---
1. View all available doctors
2. Book appointment
3. View appointments by doctor and date
4. Exit
Choose an option: 2

Enter Patient Name: prinsa
Enter Doctor ID: 102
Enter Appointment Date (dd-MM-yyyy): 2005-12-23

Booking Successful!
```

```
--- Hospital Appointment System ---
1. View all available doctors
2. Book appointment
3. View appointments by doctor and date
4. Exit
Choose an option: 3

Enter Doctor ID: 102
Enter Date (dd-MM-yyyy): 2005-12-23

Appointments for Doctor ID 102 on 23-Dec-2005:
Appointment ID: 1, Patient: prinsa, Date: 23-Dec-2005

--- Hospital Appointment System ---
1. View all available doctors
2. Book appointment
3. View appointments by doctor and date
4. Exit
Choose an option: 4
Thank you! Exiting...
```