Question 2

Problem Statement: Bookstore Inventory Management System

Objective:

Develop a **console-based C# application** using **ADO.NET** to perform **CRUD (Create, Read, Delete)** operations on a BooksInventory table in a **SQL Server** database. The system should allow users to **add new books**, **delete books by genre**, and **list all available books**. The application must follow a **disconnected architecture** using SqlConnection, SqlDataAdapter, and DataTable. All classes, properties, and methods should be declared as **public**.

Database Details

- **Database Name**: appdb
- **Table Name**: BooksInventory

Columns:

- BookID – int, auto-incremented, NOT NULL, Primary Key
- Title – varchar(100)
- Author – varchar(100)
- Genre – varchar(50) (e.g., Fiction, Non-Fiction, Thriller, Romance)
- Price – decimal(10,2)
- StockCount – int
- PublishedDate – varchar(50) (Format: "yyyy-mm-dd")

Connection String

private string connectionString = "User ID=sa;password=examlyMssql@123; server=localhost;Database=appdb;trusted_connection=false;Persist Security Info=False;Encrypt=False";

Folder Structure

dotnetapp/
├── Models/
| └── Book.cs
└── Program.cs

Classes and Properties

- ***Book Class (in Models folder)***

```
public class Book
{
   public int BookID { get; set; }
   public string Title { get; set; }
   public string Author { get; set; }
   public string Genre { get; set; }
   public decimal Price { get; set; }
   public int StockCount { get; set; }
   public string PublishedDate { get; set; }
}
```

Program Class

Methods:

1. AddBook(Book book)

- Adds a new book to the inventory.
- **Access**: public static void
- **Console Output**:
  "Book added to inventory successfully."

2. DeleteBooksByGenre(string genre)

- Deletes all books of the given genre.
- **Access**: public static void
- **Console Output**:
  - If deleted: "Books of genre '{genre}' deleted successfully."
  - If none: "No books found with genre '{genre}'."

3. DisplayAllBooks()

- Displays all books in the inventory.
- **Access**: public static void
- **Console Output**:

BookID: {row["BookID"]} Title: {row["Title"]} Author: {row["Author"]} Genre: {row["Genre"]} Price: {row["Price"]} Stock: {row["StockCount"]} Published:

{row["PublishedDate"]}

- If no records: "No books available in inventory."

Main Menu

Bookstore Inventory Management Menu
1. Add New Book
2. Delete Books by Genre
3. Display All Books
4. Exit

Invalid Input Handling

Invalid choice, please try again.

SQL Table Setup (Using sqlcmd)

```
sqlcmd -U sa
-- password: examlyMssql@123

> use appdb
> go

> create table BooksInventory (
   BookID int IDENTITY(1,1) PRIMARY KEY,
   Title varchar(100),
   Author varchar(100),
   Genre varchar(50),
   Price decimal(10,2),
   StockCount int,
   PublishedDate varchar(50)
)
> go
```

Run Commands

bash

cd dotnetapp
dotnet restore
dotnet build
dotnet run

Question 3:

Problem Statement: Hospital Appointment Scheduling System

Objective:

Develop a **console-based C# application** using **ADO.NET (disconnected architecture)** to perform **CRUD (Create, Read, Delete)** operations on a Appointments table in a **SQL Server** database. The application should allow users to **schedule new appointments**, **cancel appointments by doctor name**, and **list all scheduled appointments**.

Use SqlConnection, SqlDataAdapter, and DataTable. All classes, properties, and methods must be declared **public**.

Database Details

- **Database Name**: appdb
- **Table Name**: Appointments

Columns:

- AppointmentID – int, auto-incremented, NOT NULL, Primary key
- PatientName – varchar(255)
- DoctorName – varchar(255)
- Department – varchar(100)
- AppointmentDate – varchar(50) (Format: "yyyy-mm-dd")
- Status – varchar(50) (e.g., Scheduled, Cancelled, Completed)

Connection String

private string connectionString = "User ID=sa;password=examlyMssql@123;
server=localhost;Database=appdb;trusted_connection=false;Persist Security

Info=False;Encrypt=False";

🗀 Folder Structure

```
├── Models/
│   └── Appointment.cs
└── Program.cs
```

Classes and Properties

- ***Appointment Class (in Models folder)***

```
public class Appointment
{
   public int AppointmentID { get; set; }
   public string PatientName { get; set; }
   public string DoctorName { get; set; }
   public string Department { get; set; }
   public string AppointmentDate { get; set; }
   public string Status { get; set; }
}
```

Methods in Program.cs

1. AddAppointment(Appointment appointment)

- **Purpose**: Schedules a new appointment.
- **Access**: public static void
- **Console Message**: "Appointment scheduled successfully."

2. DeleteAppointmentsByDoctor(string doctorName)

- **Purpose**: Deletes all appointments for the given doctor.
- **Access**: public static void
- **Console Messages**:
    - Success: "Appointment(s) cancelled successfully."
    - If no matches: "No appointments found for the specified doctor."

3. DisplayAllAppointments()

- **Purpose**: Lists all appointments.
- **Access**: public static void
- **Console Message**:
    - For each appointment:

AppointmentID: {row["AppointmentID"]} Patient: {row["PatientName"]} Doctor: {row["DoctorName"]} Department: {row["Department"]} Date: {row["AppointmentDate"]} Status: {row["Status"]}

    - If none: "No appointments found."

Main Menu

Hospital Appointment Scheduling Menu

1. Schedule Appointment
2. Cancel Appointments by Doctor Name
3. Display All Appointments
4. Exit

Invalid Input Handling:

Invalid choice, please try again.

SQL Table Setup

sqlcmd -U sa

-- password: examlyMssql@123

> use appdb

> go

```
> create table Appointments (
   AppointmentID int IDENTITY(1,1) PRIMARY KEY,
   PatientName varchar(255),
   DoctorName varchar(255),
```

```
    Department varchar(100),
    AppointmentDate varchar(50),
    Status varchar(50)
)
> go
```

Run Commands

```
cd dotnetapp
dotnet restore
dotnet build
dotnet run
```