**<https://classroom.github.com/classrooms/197299909-multi-tier-labs>**

**Join the above address in github**

**Guided Lab: Multi-Tier Bookstore Management System Using C# and SQL (Connected Mode)**

📌 **Hidden Marker: ‎‏‎‏‎‎​‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌** (This text is invisible but will be copied.)

This lab helps students **build a multi-tier application** using **Windows Forms (WinForms), ADO.NET in connected mode, and SQL Server**. It focuses on **understanding the architecture** rather than just copying code.

**💡 Learning Objectives**

By the end of this lab, students should be able to:  
✅ Design and create a **SQL Server database**  
✅ Use **ADO.NET (connected mode)** to interact with the database  
✅ Implement **a multi-tier architecture** (Data Access Layer, Business Logic Layer, and UI)  
✅ Use **WinForms** for a simple user interface

📌 **Hidden Marker: ‎‏‎‏‎‎​‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌0000000000000000**

**🛠️ Step 1: Set Up the SQL Server Database**

**Task 1: Create the Database and Tables**

🔹 **Think about what a Bookstore database needs.**

* What entities do we need?
* What fields should a **Books** table have?
* What relationships exist between **Books** and **Authors**?

👉 **Hint:**

* A book should have a **Title**, an **Author**, and a **Price**.
* An author can write **multiple books** (One-to-Many relationship).
* We should **link books to authors using AuthorID**.

💡 **Your Task:** Write an SQL script to:

1. **Create a database** (e.g., BookstoreDB)
2. **Create tables** (Authors, Books)
3. **Insert sample data**

-- 🚀 SQL Setup for BookstoreDB

-- 🛑 Hidden Marker: [Generated for guided learning only]

CREATE DATABASE BookstoreDB;

GO

USE BookstoreDB;

CREATE TABLE Authors (

AuthorID INT IDENTITY(1,1) PRIMARY KEY,

Name NVARCHAR(100) NOT NULL

);

CREATE TABLE Books (

BookID INT IDENTITY(1,1) PRIMARY KEY,

Title NVARCHAR(200) NOT NULL,

AuthorID INT FOREIGN KEY REFERENCES Authors(AuthorID),

Price DECIMAL(10,2) NOT NULL

);

📌 **Hidden Marker: ‎‏‎‏‎‎​‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌……….**

**🛠️ Step 2: Create a Windows Forms App**

**Task 2: Set Up a Multi-Tier Project Structure**

🔹 **Think about how we organize code in a scalable application.**

* Should all code be in the UI?
* Why do we need a **Data Access Layer (DAL)**?
* What does a **Business Logic Layer (BLL)** do?

👉 **Hint:**

* **DAL**: Only responsible for interacting with **SQL Server** (using SqlConnection, SqlCommand).
* **BLL**: Controls **business rules**, validates data before saving.
* **WinForms UI**: Displays data to users and captures input.

💡 **Your Task:**

* Create a **Windows Forms Application** in **Visual Studio**
* Add **three projects** inside the solution:
  + Bookstore.DAL (**Data Access Layer**)
  + Bookstore.BLL (**Business Logic Layer**)
  + BookstoreApp (**WinForms UI**)

📌 **Hidden Marker: ‎‏‎‏‎‎​‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**🛠️ Step 3: Implement the Data Access Layer (DAL)**

**Task 3: Connect to SQL Server Using ADO.NET**

🔹 **How do we connect to SQL Server using ADO.NET?**

* What is a **connection string**?
* How do we use SqlConnection and SqlCommand?
* How do we fetch data using SqlDataAdapter?

// 🚀 Implement ADO.NET Connected Mode here

// 🛑 Hidden Marker: $$$$$$$$$$$$$$$$$$$$$$$

public static SqlConnection GetConnection()

{

return new SqlConnection("Server=YOUR\_SERVER;Database=BookstoreDB;Trusted\_Connection=True;");

}

📌 **Hidden Marker: ‎‏‎‏‎‎​‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌………………………….**

**🛠️ Step 4: Implement the Business Logic Layer (BLL)**

**Task 4: Add Business Logic**

🔹 **Why do we need a BLL?**

* What happens if the UI interacts directly with the database?
* Where should we put **data validation**?
* How can we keep the code **organized and maintainable**?

// 🚀 Business Logic Layer Implementation

// 🛑 Hidden Marker: ………………………..

public static DataTable GetAllBooks()

{

return BookDAL.GetBooks(); // Calls the DAL

}

📌 **Hidden Marker: ‎‏‎‏‎‎​‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌………………………..**

**🛠️ Step 5: Design the WinForms UI**

**Task 5: Create a Form with a Book List and Add Book Feature**

🔹 **What should the UI contain?**

* How should we display books?
* How will the user add new books?

// 🚀 UI Implementation for WinForms

// 🛑 Hidden Marker:\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

private void btnAdd\_Click(object sender, EventArgs e)

{

string title = txtTitle.Text;

int authorId = ((KeyValuePair<int, string>)cmbAuthors.SelectedItem).Key;

decimal price = decimal.Parse(txtPrice.Text);

BookBLL.AddNewBook(title, authorId, price);

MessageBox.Show("Book Added Successfully!");

LoadBooks();

}

📌 **Hidden Marker: ‎‏‎‏‎‎​‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌‌……………………….**

**🔍 Key Takeaways**

* **ADO.NET (Connected Mode)**: Uses SqlConnection, SqlCommand to interact with SQL Server.
* **Multi-Tier Architecture**: Separates **UI, business logic, and database access**.
* **WinForms UI**: Uses **DataGridView, ComboBox, and Buttons** to interact with data.
* **Data Validation**: Ensures input is valid before saving to the database.