Lesson 5 – Modifying Data Part 1

# Setup DB

1. Open the window **SQL Server Object Explorer** by clicking **View** > **SQL Server Object Explorer**. The following window will appear

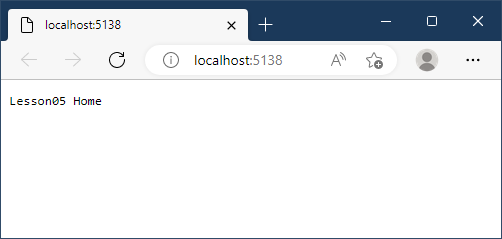
Graphical user interface, text, application, email

Description automatically generated

1. Expand **(localdb)\ProjectsModels**. Do **not** expand MSSQLLocalDB.
2. If you cannot find it, click on the icon **Add SQL Server…**  Expand **Local** and select **Project Models**. Click on the [Connect] button.
3. Right-click **Databases** and click **Add New Database**. The **Create Database** window will appear.
4. In the **Database Name** enter **DB 05**
5. In the **Database Location** , select the folder **Databases** in your C236 folder which you have created in Lesson 01.
6. Right-click on **DB05**, click **New Query.** The **SQLQuery1.sql** window will appear.
7. Copy the SQL statements given in **DBSetup.txt** and paste inside SQLQuery1.sql.
8. Click the **green** execute button at the top left corner of the SQLQuery1.sql window to execute the SQL statements.
9. Right-click on **DB05**, click **Refresh** . Expand the **Tables**. You will see the four tables: **BwBook, BwPublisher**, **OrgProduct** and **OrgSubscription.**
10. Close the **SQLQuery1.sql** window. You do not need to save any changes.

# Setup Application

1. In the Solution Explorer, right-click on the Project **Lesson05** > Set as Startup Project.
2. Right-click **Lesson05** and select **Build**
3. Run the project.



1. From the **Models** folder inthe lesson materials, **copy all files** into the **Models** folder in VS2022 Lesson05. Do not drag in the whole folder, or you will create Lesson05/Models/Models, which will not work.
2. From the **Models** folder in the lesson materials, copy **all the files** into the **Models** folder in VS2022 Lesson05.
3. From the **Views** folder in the lesson materials, **copy all the sub-folders and contents** into the **Views** folder in VS2022 Lesson05. Do not drag in the whole folder.
4. From the **Controllers** folder in the lesson materials, copy **all the files** into the **Controllers** folder in VS2022 Lesson05.
5. Drag the **images folder** from the lesson materials **onto the** **wwwroot** folder in VS2022 Lesson 05.
6. Drag the **utils folder** from the lesson materials **onto** **Lesson05** in VS2022 Lesson 05.
7. Drag the **libman.json file** from the lesson materials **onto** **Lesson05** in VS2022. Right-click **libman.json > Restore Client-Side Libraries**. This step creates **wwwroot/lib** folder structure containing the necessary bootstrap libraries.
8. Modify the **Program.cs** file by adding the following line to the very top of the file, then save the file.

global using Lesson05.Models;

global using System.Data;

global using RP.SOI.DotNet.Utils;

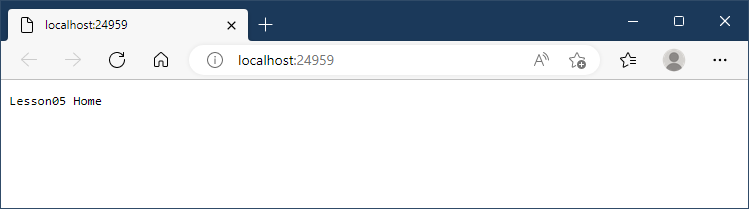
1. Double click Lesson05 to open the Lesson05.csproj file. Amend the file as shown.

|  |
| --- |
| <Project Sdk="Microsoft.NET.Sdk.Web">  <PropertyGroup>  <TargetFramework>net6.0</TargetFramework>  <Nullable>enable</Nullable>  <ImplicitUsings>enable</ImplicitUsings>  <AspNetCoreHostingModel>OutOfProcess</AspNetCoreHostingModel>  <AspNetCoreModuleName>AspNetCoreModule</AspNetCoreModuleName>  <TreatWarningsAsErrors>true</TreatWarningsAsErrors>  <RootNamespace>Lesson05</RootNamespace>  </PropertyGroup>  <ItemGroup>  <PackageReference Include="System.Data.SqlClient" Version="4.8.3" />  </ItemGroup>  </Project> |

1. Open **Appsettings.Development.json** (expand **Appsettings.json** to see the file). Amend the file as shown.

|  |
| --- |
| {  "Logging": {  "LogLevel": {  "Default": "Information",  "Microsoft.AspNetCore": "Warning"  }  },  "ConnectionStrings": {  "DefaultConnection": "Data Source=(localdb)\\ProjectModels;Initial Catalog=DB05;Integrated Security=True"  }  } |

1. Save all files (Ctrl-Shift-S).
2. Right-click the project **Lesson05** and select **Build**. Ensure there are no errors.
3. Run the project with debugging.



CONTINUE ONLY IF YOUR PROJECT IS ERROR-FREE.

# Bootstrap Styling Exercise

1. With reference to the Organic Subscription view, copy **Views/Demo/StyleExercise.cshtml** and rename the copy **StyleExerciseCompleted.cshtml**. In the newly named view, add Bootstrap classes and any other HTML elements so that the form style reflects that illustrated below. Omit the navigation bar and message handling. Add the controller **demo**, and the action **styleexercisecompleted** and test data to the URL to see the result.

Graphical user interface, application

Description automatically generated

1. Place an image of your completed styling exercise below. **Please fill in your name for Field A.**

|  |
| --- |
| Replace this text with your image of the style exercise. Your name must be in in Field A. |

# Partial Views and Nav Bars

1. Double click on the file **launchSettings.json** in **Lesson05/Properties folder**. Add the following line after the line the contains "IISExpress" (Line 19). Save the file.

|  |
| --- |
| "IIS Express": {  "launchUrl": "organic/index",  "commandName": "IISExpress",  "launchBrowser": true,  "environmentVariables": {  "ASPNETCORE\_ENVIRONMENT": "Development"  } |

1. Run your project. Answer the following questions.

|  |  |  |
| --- | --- | --- |
| a. | Do you see any NavBar for the Organic Fruit app? | No |
| b. | What number of cshtml files are there in the Views/Organic folder? | 6 |
| c. | Which partial view that contains the NavBar? | Index |

1. Open all the Organic views. Locate **TODO: Lesson05 Task 2a – 2e**. Complete these tasks using the Razor code below.

|  |
| --- |
| @await Html.PartialAsync("*ViewName*") |

For **ViewName** refer to the previous question.

Stop and run your project again. Navigate using the menu and test whether all the items in the menu are working. Answer the following questions.

|  |  |  |
| --- | --- | --- |
| a. | Do you now see the NavBar in the Organic Fruit Web App? | Yes |
| b. | Which two menu items are not working? | Add and remove product |
| c. | Why are the two menu items not working? | Action methods for AddProduct and RemoveProduct are missing in OrganicController.cs  The ”href” attribute is not updated in “OrgNavBar.cshtml” |

# Publisher Add and Delete

Change the launchURL in the **launchSettings.json** to the following. Save the file.

|  |
| --- |
| "launchUrl": "bookworm/index", |

Stop and run your project again. Answer the following questions.

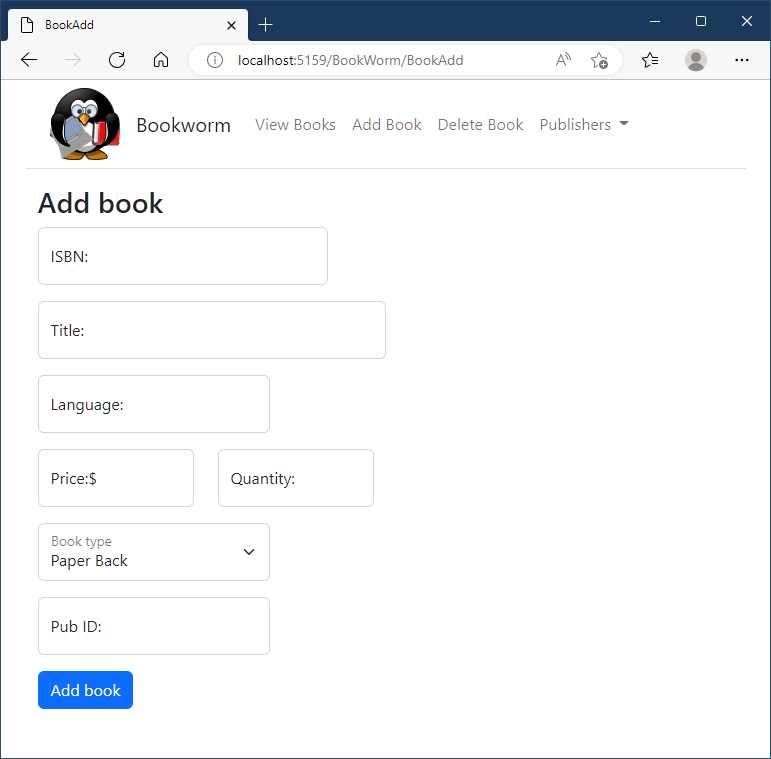
|  |  |  |
| --- | --- | --- |
| a. | Click **Publishers > View All.**  How many publishers are there? | 5 |
| b. | Click **Publishers** > **Add Publisher**.  Enter 3 as the Pub ID and any sensible input for Pub Name and Pub Address. Are you able to add the record? Why? | No. An error was triggered stating that 3 is a duplicate value upon being keyed in |
| c. | Click **Publishers** > **Delete Publisher**.  Enter 2 as the Pub ID. Are you able to delete the record? Why? | No. There is an error stating that there is a conflict upon deleting |
| d. | Click **Publishers** > **Add Publisher**.  Enter 100 as the Pub ID and and any sensible input for Pub Name and Pub Address.  How can you verify that the system added the record to the database? | It is successfully added as it shows the successfully added page and that there is a table displayed with the new details added in. |
| e. | Click **Publishers** > **Delete Publisher**. Enter 100 as the Pub ID. What is the message given after you click [Delete] | Deleted publishers record is not found in the list of Publishers view. |

Open BookWormController.cs, examine codes and complete the following table.

|  |  |
| --- | --- |
| **Action** | **Purpose** |
| Index | Returns the view of BookWorm |
| Books | Selects the book that is entered by the user |
| Publishers | Selects the publisher that is entered in by the user |
| PubAdd | Returns the view of adding the publisher |
| PubAddPost | Returns the view depending on the context of whether it passes the adding phase, then it will depend on if the publisher s successfully entered. |
| PubDelete | Returns the PubDelete view |
| PubDeletePost | Returns the view depending on whether the publisher id is found, then it depends on whether it passes or fails when deleting |

# Solving the Problem

Create a new view, **BookAdd.cshtml** in the folder Views/BookWorm. When complete, the view will appear as follows. Try to complete the layout by yourself. If you get stuck, you can use the C236 web site to assist.

****

Ignore the generated javascript and browser links. Refer to the BookWorm sample for error handling code.

Place a copy of your completed Book add screen in the table below. If time permits, demonstrate the completed form to your lecturer. The **Title** of your book is to be

"***Your name* - A biography**".

Make sure top substitute **your own name** for Your name in the title.

|  |
| --- |
| Replace this text with your images of the Add book view. Fill in the form. **Your name** must be in in **Title**. Add both medium and small viewports. |

Perform Task 3. Refer to the code in **PubAdd** action to help.

Perform Task 4. Refer to the code in **PubAddPost** action to help.

Perform Task 5. Refer to the code in PubDelete action to help.

Perform Task 6. Refer to the code in PubDeletePost action to help. Write the template SQL string for Task 6 in the space below. The title of your book is to be "***Your name* - A biography**". Make sure top substitute your own name for Your name in the title.

|  |
| --- |
| string sql = @"SELECT \* FROM BwPublisher WHERE PubID={0}";  string select = String.Format(sql, pubid);  DataTable dt = DBUtl.GetTable(select);  if (dt.Rows.Count == 0)  {  ViewData["Message"] = "Publisher ID Not Found.";  ViewData["MsgType"] = "warning";  return View("PubDelete");  } |

Perform the following tests on your complete Web App.

* + - 1. Add two or three books
      2. Use the **/BookWorm/Books** to confirm the system saved the books.
      3. Try adding a book with a **PubID** that does not exist
      4. Delete those books which you have added.
      5. Use the **/BookWorm/Books** to confirm the system deleted the books.
      6. Navigate using the NavBar and ensure all links are working.

# String.Format Gym

Navigate to: [C236 (c236dotnet.azurewebsites.net)](http://c236dotnet.azurewebsites.net/)

* + - 1. Click  on the right-hand side to login.
      2. Use your **Student ID** (e.g., 19001234) and the last four characters (e.g., 789X) of your **NRIC** to login.
      3. Under the menu **Practice**, click **String Format Gym**.
      4. Reach **Level 20** **before 23:59 hrs**. Your lecturer will confirm level 20 you have reached as part of your self-directed learning score.

Chart

Description automatically generated with low confidence

*— End of Worksheet —*