Exercise 05 Client Server Setup PART 1

Initial Setup

Create a new ASP.NET Web Application (.NET Framework) project.

To do this start Visual Studio 2019 and click the lower left button called “Create a new project”.

Use the filters to search for C#, Windows, Web, and then choose “ASP.NET Web Application (.NET Framework)”. Click on “Next”.

Name it and the solution “WebAppFSIS”. Click on “Create”.

Click on “WebForms” and uncheck the “Configure for HTTPS” option. Click on “Create”.

Update all Nu-Get Packages to their latest version.

Replace the navbar code in the site.master with the following (not the grey stuff, they are just boundary markers):

</asp:ScriptManager>

<div class="container">

<div class="navbar navbar-inverse">

<nav class="navbar navbar-expand-md navbar-light fixed-top">

<a class="navbar-brand" href="Default.aspx" runat="server">

<strong>Bootstrap 4+</strong>

</a>

<button class="navbar-toggler" type="button"

data-toggle="collapse" data-target="#navbarMenuName"

aria-controls="navbarMenuName" aria-expanded="false"

aria-label="Toggle Navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarMenuName">

<ul class="navbar-nav">

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#"

id="navbarSubMenu" data-toggle="dropdown"

runat="server" style="color:black"

aria-haspopup="true" aria-expanded="false">

ExercisePages

</a>

<div class="dropdown-menu" aria-labelledby="navbarSubMenu">

<a class="dropdown-item"

href="~/ExercisePages/ExerciseHome.aspx"

style="color:black" runat="server">

ExerciseHome

</a>

<a class="dropdown-item"

href="~/ExercisePages/SimpleQuery.aspx"

style="color:black" runat="server">

Simple PKey Query

</a>

</div>

</li>

<li class="nav-item">

<a class="nav-link" href="~/Default.aspx"

style="color:black" runat="server">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="~/About.aspx" runat="server"

style="color:black">About</a>

</li>

<li class="nav-item">

<a class="nav-link" href="~/Contact.aspx" runat="server"

style="color:black">Contact</a>

</li>

</ul>

</div>

</nav>

</div>

</div>

<div class="container body-content">

Add a folder to the WebApp project and name it “ExercisePages”.

Add two “WebForm with Master Page” pages to the folder and name them “ExerciseHome” and “SimpleQuery”, and select the site.master, master page for both.

In the ExerciseHome Form add a <h1>Hey Man</ h1>

Run without debugging to see if it works, if not make sure the replacement code in the site.master is exactly as above. Do not continue until the code runs.

Go to your GitHub account and create a new repo that is empty.

Copy the URL of the new repo into the clip board.

In Visual Studio go to the bottom right and click on “Add to Source Control” and click on Git. Save your work, then use the bottom most “Publish Git Repo” and copy the URL into the yellow area. Finally click “Publish”.

Add a “Class Library” project to the solution. To do this right click on the solution name and go to “Add” then “New Project”. Use the filters to search for C#, Windows, Library, and then choose “Class Library (.NET Framework)”. Name the project “FSISSystem”.

Add three folders to the FSISSSystem project called BLL, DAL, and ENTITIES.

Add a Class to each folder and call each one “Dummy.cs”. To do this right click on the folder name then “Add” then “New Item”. You may have to click on “Code” on the left side to see the “Class” option.

Go to the WebAppFSIS project’s reference area and add a project reference to the FSISSystem project. You may have to click on “Projects” on the left side to see the FSISSystem project.

Run the code again and make sure it works before moving on to PART 2. Save to your GitHub repo.

Exercise 05 Client Server Setup PART 2

ENTITIES Layer



In the ENTITIES folder create a class file and name it Team.cs

Add the following code:

using System.ComponentModel.DataAnnotations.Schema;

using System.ComponentModel.DataAnnotations;

namespace FSISSystem.ENTITIES

{

[Table("Team")]

public class Team

{

public int TeamID { get; set; }

public string TeamName { get; set; }

}

}

Entity Framework Package Download, DAL Layer, and web.config

Using Manage NuGet Packages, add *EntityFramework* to your “FSISSystem” project and “WebAppFSIS”web application project.

Add a reference to System.Data.Entity to both the “FSISSystem” project and “WebAppFSIS”web application project.

Add a Class to your DAL folder and call it FSISContext.cs

Add the following code:

using System.Data.Entity; //inheritance of DbContext from EntityFramework

using FSISSystem.ENTITIES;

namespace FSISSystem.DAL

{

internal class FSISContext : DbContext

{

public FSISContext() : base("FSIS\_db")

{

}

public DbSet<Team> Teams { get; set; }

}

}

Configure your solution’s knowledge of the database by making the following changes to your system.

Edit the web application’s *web.config* file to have the following for the <connectionStrings> tag

<connectionStrings configSource="WebConnectionStrings.config"/>

Add a new configuration file to the root of the web application and name it “*WebConnectionStrings.config*”; on the inside, put the following code:

<connectionStrings>  
 <add name="FSIS\_db"  
 connectionString="Data Source=.;Initial Catalog=FSIS\_Database;Integrated Security=True;"  
 providerName="System.Data.SqlClient"/>  
</connectionStrings>

Edit the web application’s *web.config* file to have the following tag <contexts> added to the <entityFramework> tag.

<contexts>

<context disableDatabaseInitialization="true"  
 type="FSISSystem.DAL.FSISContext,FSISSystem">

</context>

</contexts>

Run the code again and make sure it works before moving on to PART 3. Save to your GitHub repo.

Exercise 05 Client Server Setup PART 3

BLL Layer, Form Presentation Tags, Form Code Behind

Add a Class to your BLL folder and call it TeamController.cs

Add the following code:

using FSISSystem.DAL;

using FSISSystem.ENTITIES;

namespace FSISSystem.BLL

{

public class TeamController

{

public Team Teams\_FindByID(int teamid)

{

using (var context = new FSISContext())

{

return context.Teams.Find(teamid);

}

}

}

}

In the ExcercisePages/SimpleQuery Form add the tags necessary to present labels, a textbox, and a button, for a Simple Primary Key Query, similar to the Northwind Demo, but for the Team Table of the FSIS Database instead of the Region Table of the Northwind Database.

In the SimpleQuery code behind add the code necessary to make everything work when the button is pressed. This code will be similar to the code behind in the Northwind Demo. Remember to put in the proper using directives, so your code can access the BLL, and ENTITIES code.

Test your code and when it works, save it to your GitHub repo.

At this point you can demonstrate all of Exercise 05 to your instructor for evaluation.