2.14 CRUD Operations using Entity Framework



This section will guide you to:

* Implement CRUD operations using Entity Framework

This guide has eight subsections, namely:

2.14.1 Creating an ASP.NET MVC project

2.14.2 Creating a Student table in a database

2.14.3 Adding EntityFramework using Nuget

2.14.4 Using EntityFramework to generate models and views

2.14.5 Creating StudentController to auto-create the views for CRUD operations

2.14.6 Building the project

2.14.7 Publishing and running the project

2.14.8 Pushing the code to your GitHub repositories.

**Step 2.14.1:** Creating an ASP.NET MVC project

* Open Visual Studio.
* From the top menu, select **File->New->Project.**
* In **Create A New Project** screen, select **ASP.NET Web Application(.Net Framework)** from the list of available project types and click on **Next.**
* Please select the project type where the **language** of the project is **C#.**
* Enter **Project Name** as **Phase3Section2.20** and click on **Create.**
* From the list of project sub-types, choose **Web MVC** and uncheck **Configure for HTTPS.** Click on **Create.**
* This will create the files for an ASP.NET MVC project.

**Step 2.14.2:** Creating a Student table in a database

* Open SQL Server Management Studio. In the login screen, make a note of the **Server Name** value as you will need to put in your ASP.NET application.
* In **Object Explorer**,right click **Databases** and choose **New Database.**
* Enter **Database name** as **School1** and click **Ok.**
* In the **Object Explorer**,expand **School1->Tables.** Right click **Tables** and choose **New->Table.**
* For the first row enter ID as **Column Name**,int as **Data Type**, and uncheck **Allow Nulls.**
* In **Column Properties**,go to **Identity Specification** and expand it. Double click **Is Identity** to make it **Yes**.
* For the next row enter Name as **Column Name**,varchar(100) as **Data Type**, anduncheck **Allow Nulls.**
* For the next row enter Address as **Column Name**,varchar(100) as **Data Type**, anduncheck **Allow Nulls.**
* For the next row enter Email as **Column Name**,varchar(75) as **Data Type**, anduncheck **Allow Nulls.**
* For the next row enter Class as **Column Name,** varchar(5) as **Data Type**, anduncheck **Allow Nulls.**
* Click on the x icon to close the table grid tab. Click **Yes** on the save dialog.
* For **Enter a name** put Student and press **Ok.**
* In **Object Explorer**,expand **School1->Tables->Student.** Right click **Tables** and choose **Edit Top 200 Rows.**
* Add in a few rows of data with random values.
* Close the Management Studio

**Step 2.14.3:** Adding EntityFramework using NuGet

* In the top menu, go to **Tools->Nuget Package Manager->Package Manager Console.**
* In the console, type **install-package entityframework** and press **Enter.**

**Step 2.14.4:** Using EntityFramework to generate models and views

* In the **Solution Explorer**,right click **Phase3Section2.28** and choose **Add->New Item.**
* Under **Visual C#**,select **Data** and choose **ADO.NET Entity Data Model.** Enter **Name** as Student and click **Add.**
* In the next screen, click on **EF Designer from Database** and click **Next.**
* In the next screen, click on **New Connection.**
* This will open a popup window. In the **Server Name**,put the server name from the SQL Management Studio login screen.
* In **Select or enter a Database name,** choose **School1** and click **Ok.**
* This will close the popup window. In the current window, make sure **Save Connection settings in web.config** is checked and make a note of the value in the textbox eg.School1Entities.
* Click on **Next.**
* In the next screen, expand **Tables->dbo->Student** and check **Student.** Click **Finish.**
* This will create all the entity files for the Student table.

**Step 2.14.5:** Creating StudentController to auto-create the views for CRUD operations.

* In the **Solution Explorer**,expand **Controllers.** Right click **Controllers** and choose **Add->Controller**.
* From the list of types, choose **MVC5 Controller with views**, **using Entity Framework** and click **Add**.
* In the next screen, choose **Model Class** as Student, **Data Context Class** as School1Entities, **Controller Name** as StudentsController and click **Add**.
* This will generate the code for the Controller for all CRUD operations.

**Step 2.14.6:** Building the project

* From the top menu, choose **Build->Build Solution.**
* If any compile errors are shown, fix them as required.

**Step 2.14.7:** Publishing and running the project

* From the top menu, select **Debug->Start Without Debugging.**
* This will execute the program in the default browser.
* To see the student pages, go to the url : http://localhost:xxxx/students.

**Step 2.14.8:** Pushing the code to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add .

Commit the changes using the following command:

git commit -m “Changes have been committed.”

Push the files to the folder you created initially using the following command:

git push -u origin master