**Practical 16 : MVC Application using Entity Framework**

**Database Setup**

**🔹 Step 1: Create a Database in SQL Server**

Open SQL Server Management Studio (SSMS) or Visual Studio SQL Server Object Explorer.

**Create a Database** named StudentDB.

**Create a Table** using the following SQL: Create Database StudentDB

use StudentDB

CREATE TABLE Students (

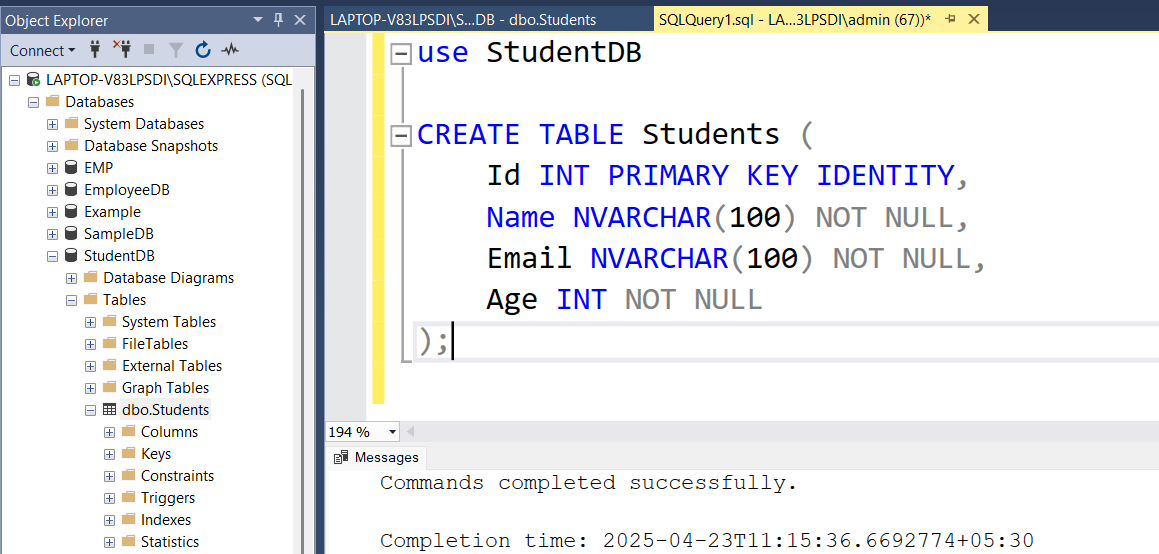
Id INT PRIMARY KEY IDENTITY,

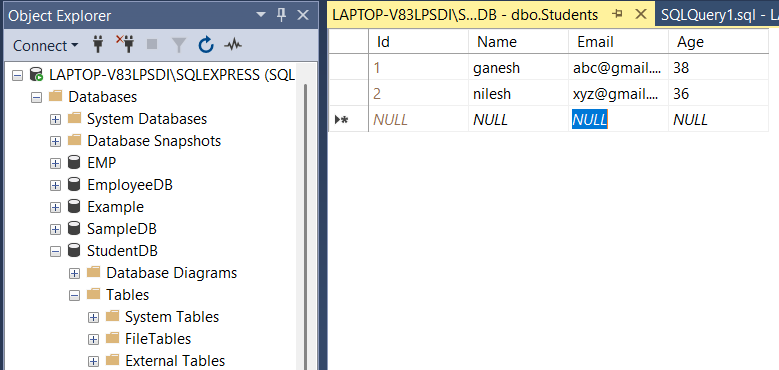
Name NVARCHAR(100) NOT NULL,

Email NVARCHAR(100) NOT NULL,

Age INT NOT NULL

);





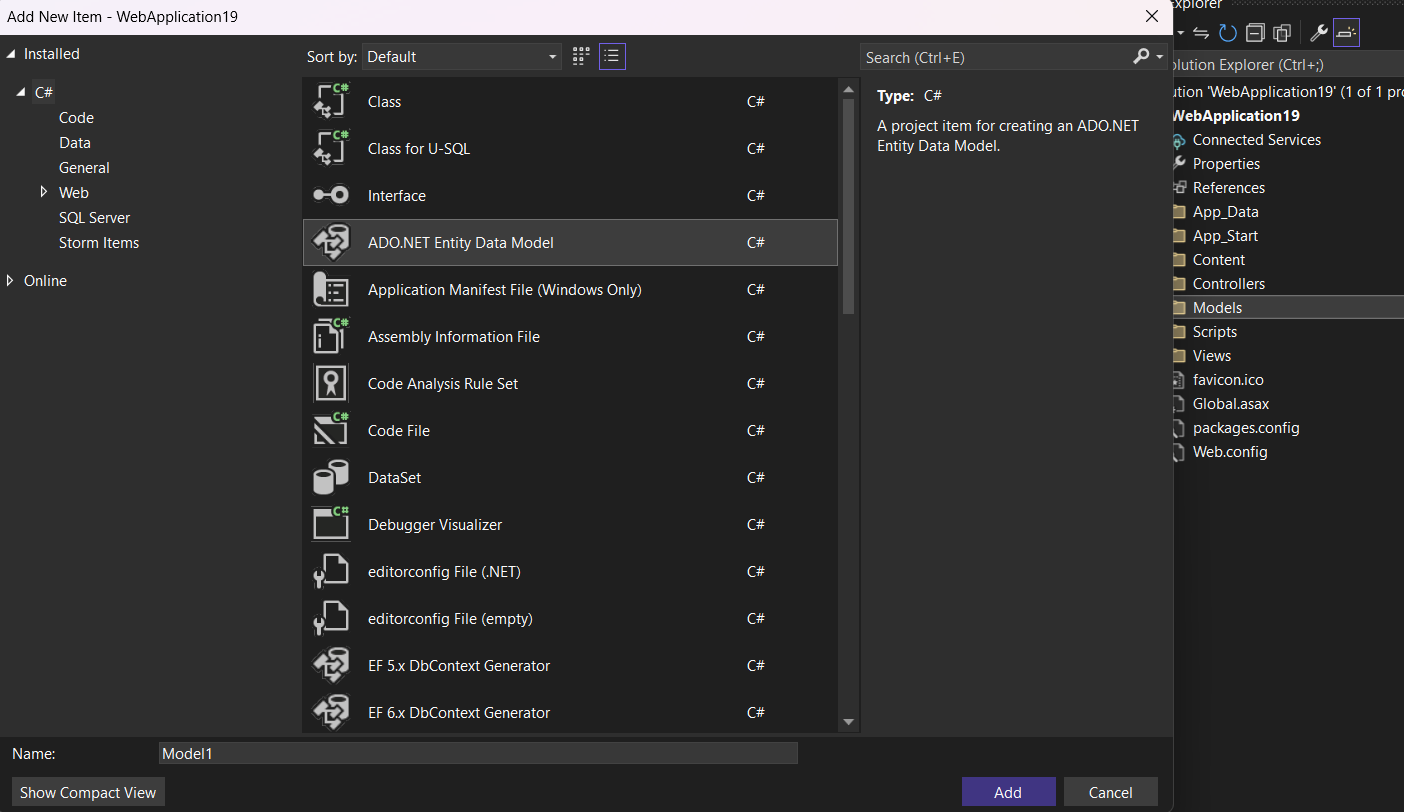
### Step 2: Create a New ASP.NET MVC Project

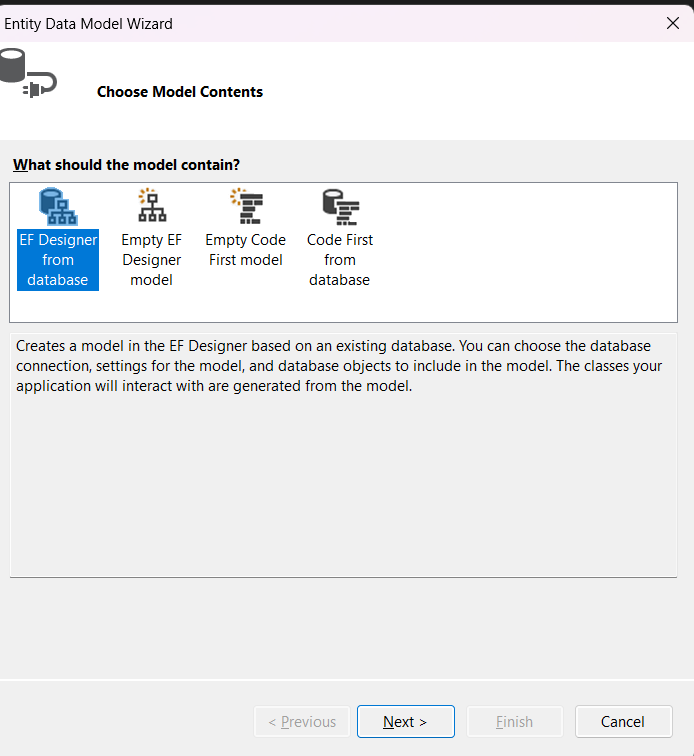
1. Open **Visual Studio**
2. Select **Create a new project**
3. Choose: **ASP.NET Web Application (.NET Framework)**
4. Name: StudentMVCApp
5. Choose **MVC** as the template
6. Click **Create**

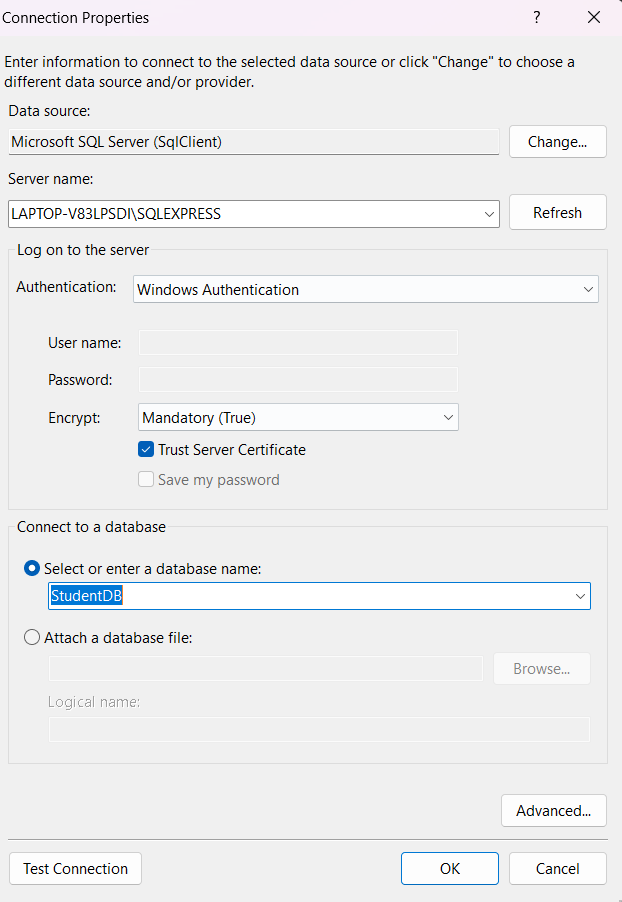
**Entity Framework Model Setup**

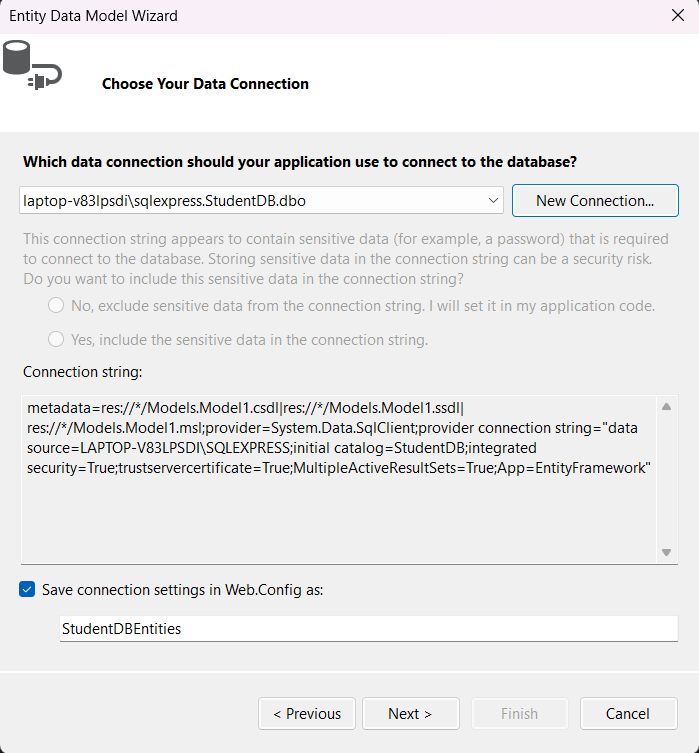
**🔹 Step 3: Add Entity Framework Model**

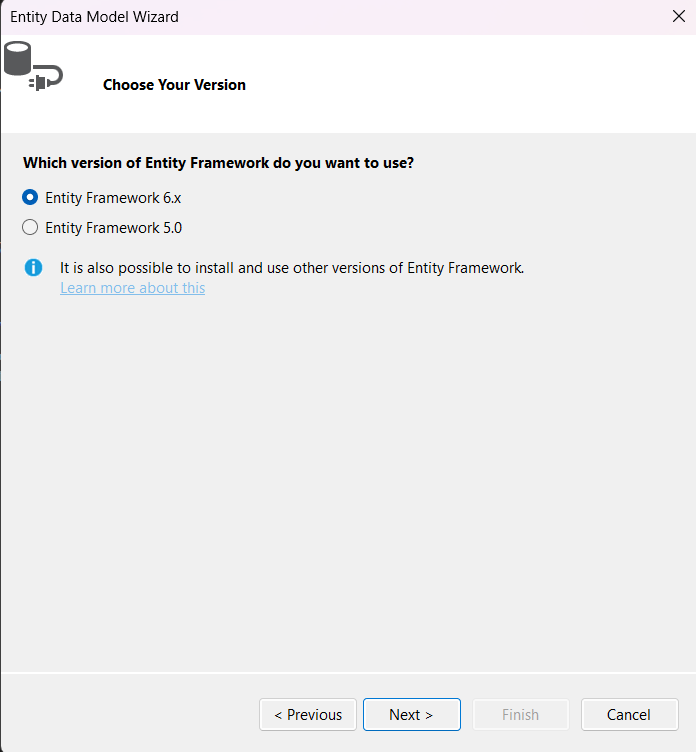
1. Right-click the **Models** folder → Add → New Item
2. Choose **ADO.NET Entity Data Model**
3. Name it: StudentModel.edmx
4. Choose: "EF Designer from database"
5. Select your SQL Server database (StudentDB)
6. Select the Students table
7. Finish to generate model classes

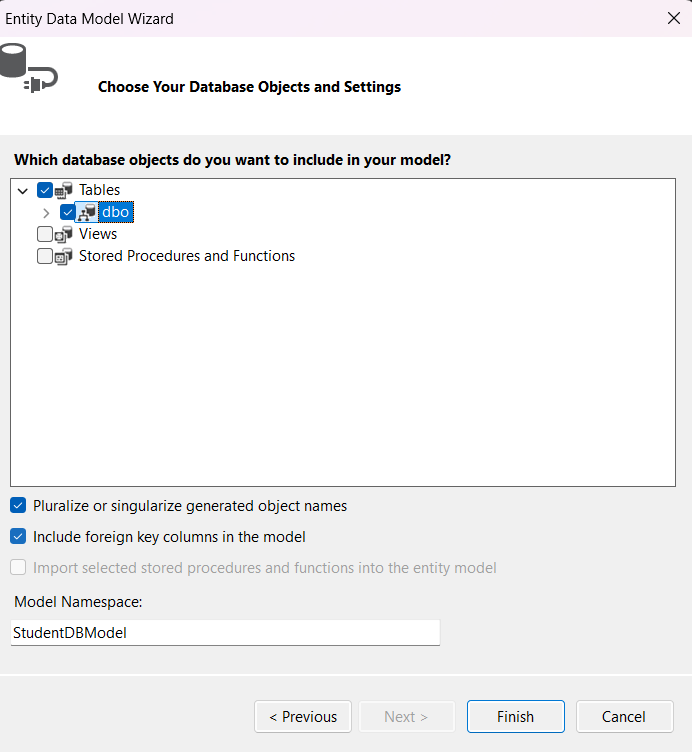


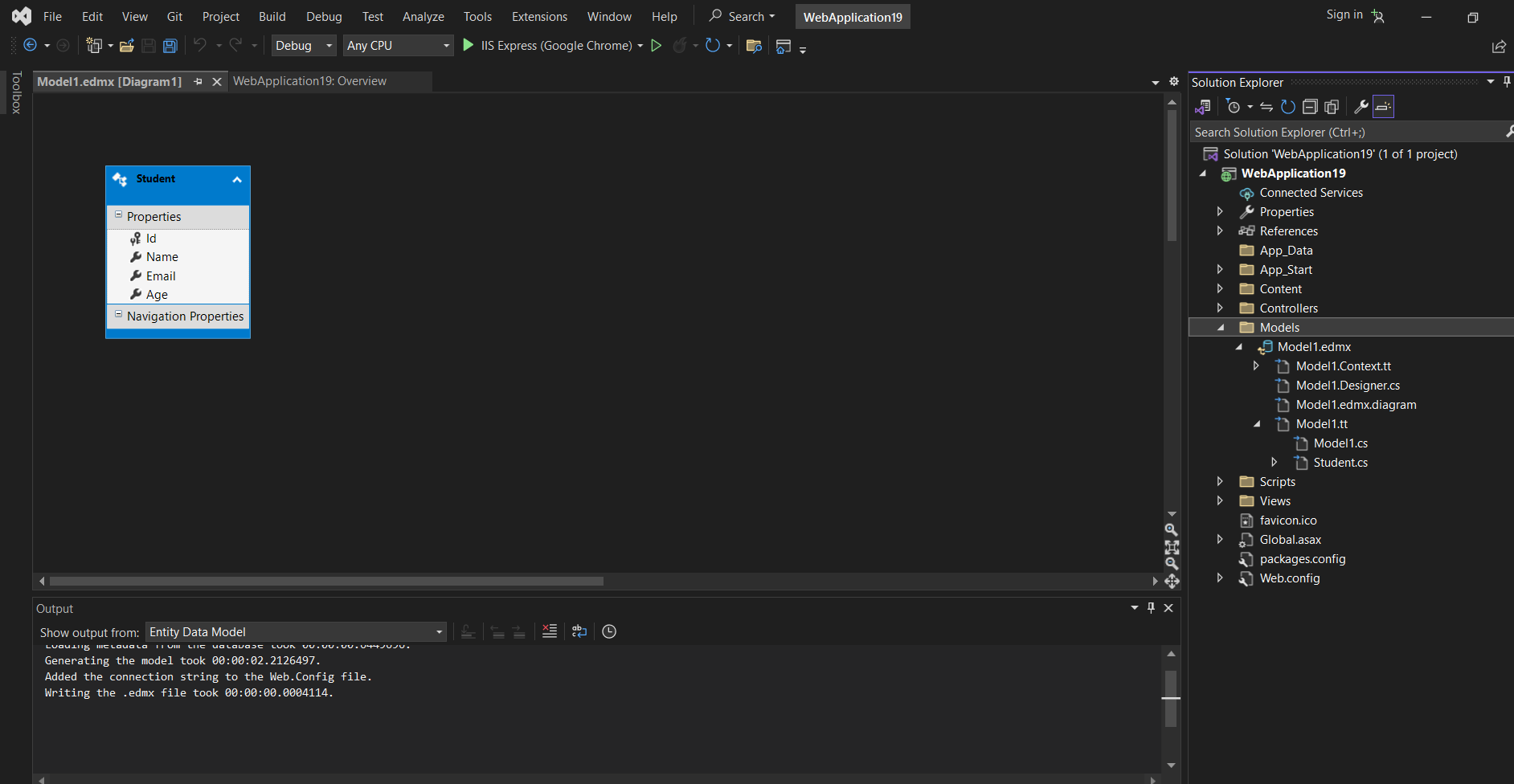






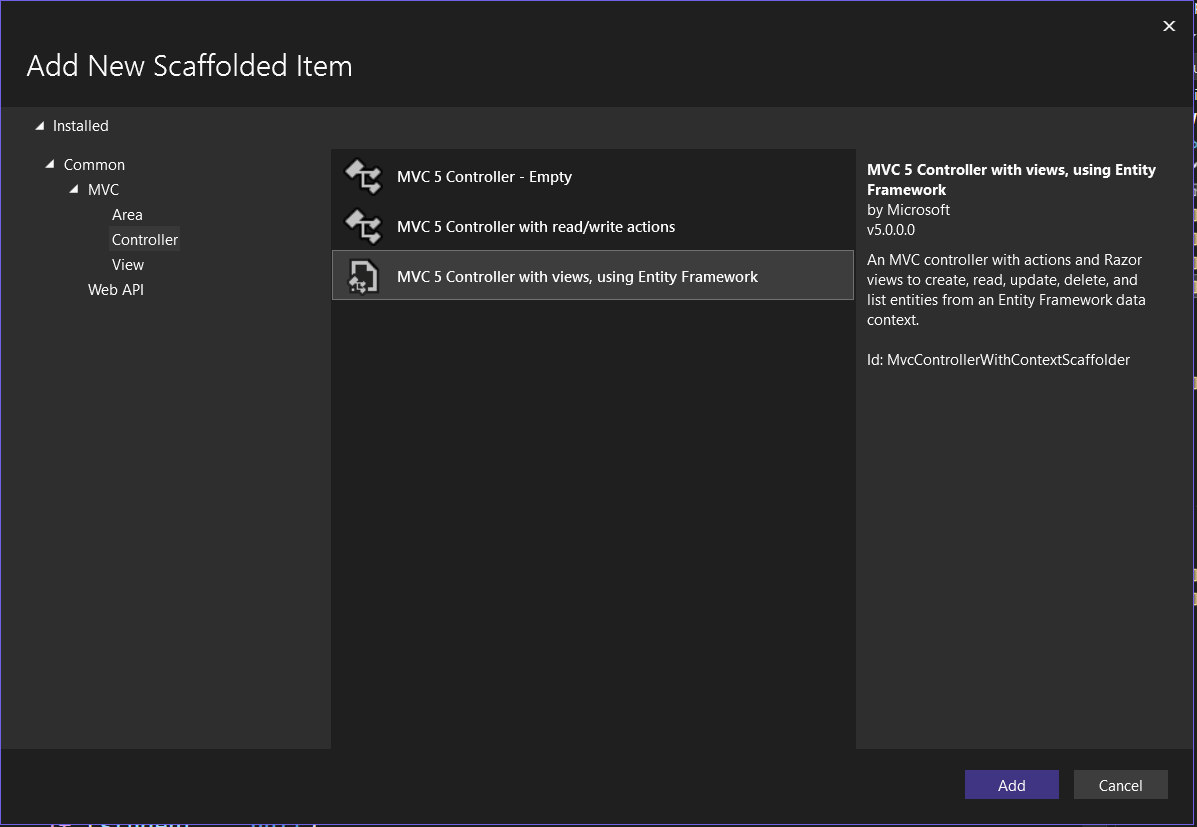


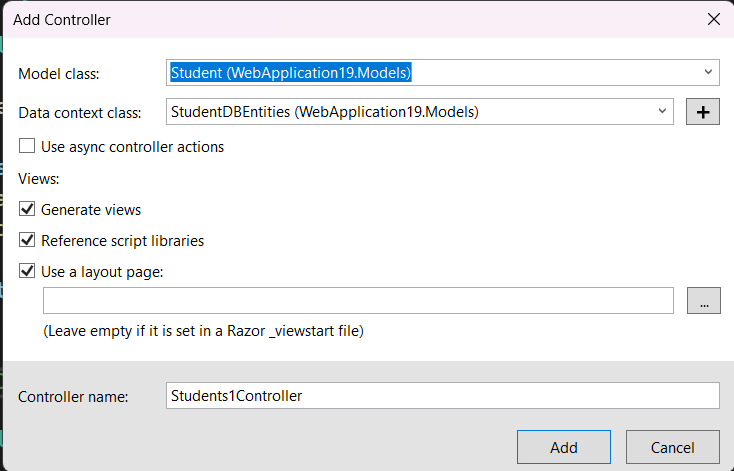


++

**Step 4: Create Controller**

1. Right-click **Controllers** → Add → Controller
2. Choose: **MVC 5 Controller with views, using Entity Framework**
3. Model class: Student
4. Data context: StudentDBEntities (if using .edmx) or StudentDBContext
5. Click **Add**





## Step 5: Create Views

You can right-click on each controller action and choose **Add View**, or use the auto-generated ones.

Example: Views/Students/Index.cshtml

@model IEnumerable<StudentMVCApp.Models.Student>

@{

ViewBag.Title = "Student List";

}

<h2>Student List</h2>

<p>@Html.ActionLink("Create New", "Create")</p>

<table class="table">

<tr>

<th>@Html.DisplayNameFor(model => model.Name)</th>

<th>@Html.DisplayNameFor(model => model.Email)</th>

<th>@Html.DisplayNameFor(model => model.Age)</th>

<th>Actions</th>

</tr>

@foreach (var item in Model) {

<tr>

<td>@item.Name</td>

<td>@item.Email</td>

<td>@item.Age</td>

<td>

@Html.ActionLink("Edit", "Edit", new { id = item.Id }) |

@Html.ActionLink("Details", "Details", new { id = item.Id }) |

@Html.ActionLink("Delete", "Delete", new { id = item.Id })

</td>

</tr>

}

</table>

## step 6: Set Default Route

In App\_Start/RouteConfig.cs, change default route to:

csharp

CopyEdit

defaults: new { controller = "Students", action = "Index", id = UrlParameter.Optional }

OUTPUT:

