**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 (

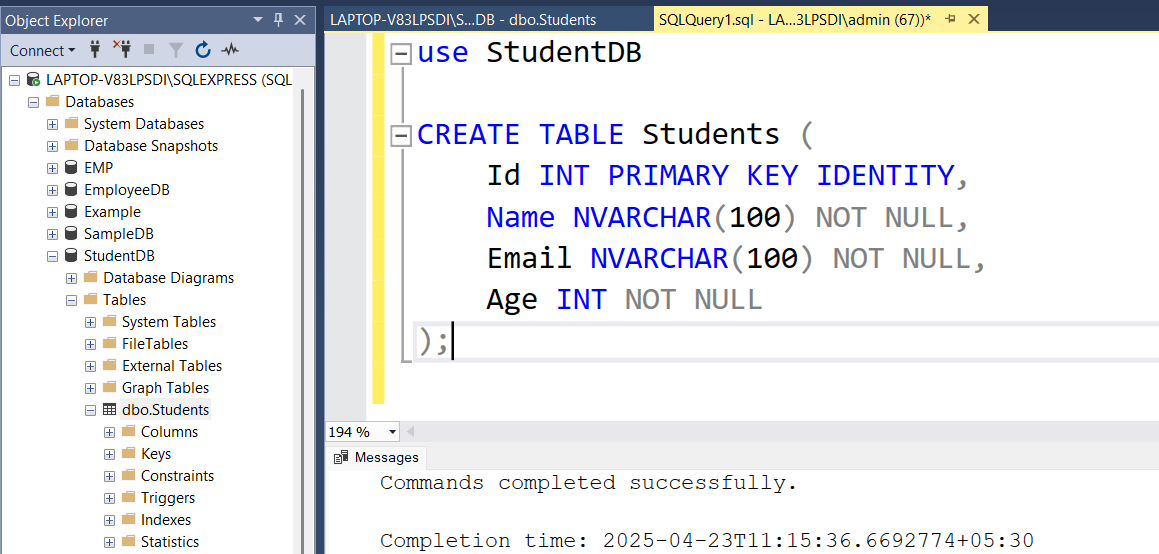
Id INT PRIMARY KEY IDENTITY,

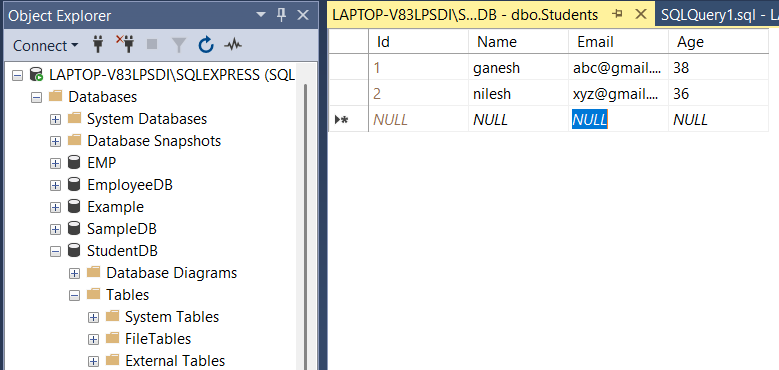
Name NVARCHAR(100) NOT NULL,

Email NVARCHAR(100) NOT NULL,

Age INT NOT NULL

);





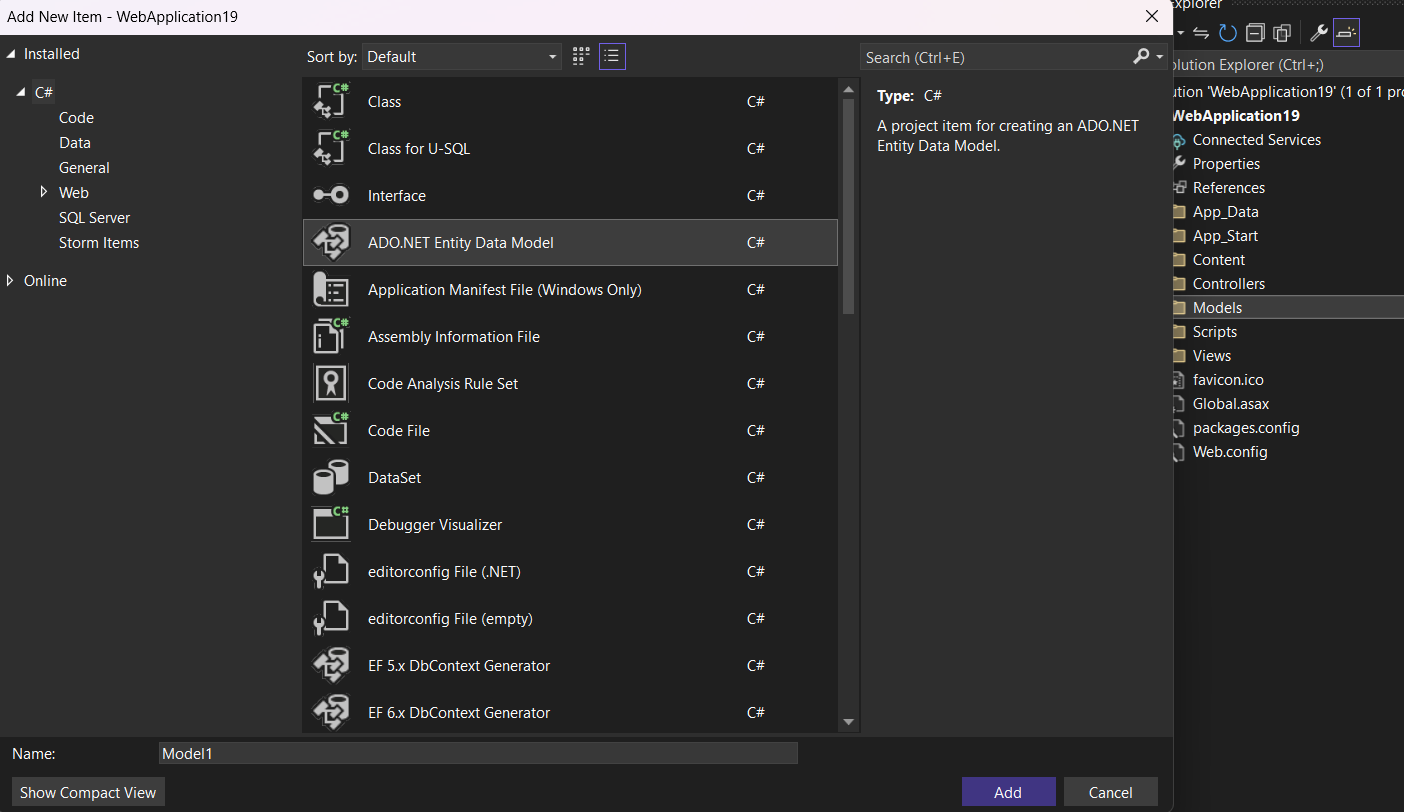
### Step 2: Create a New ASP.NET MVC Project

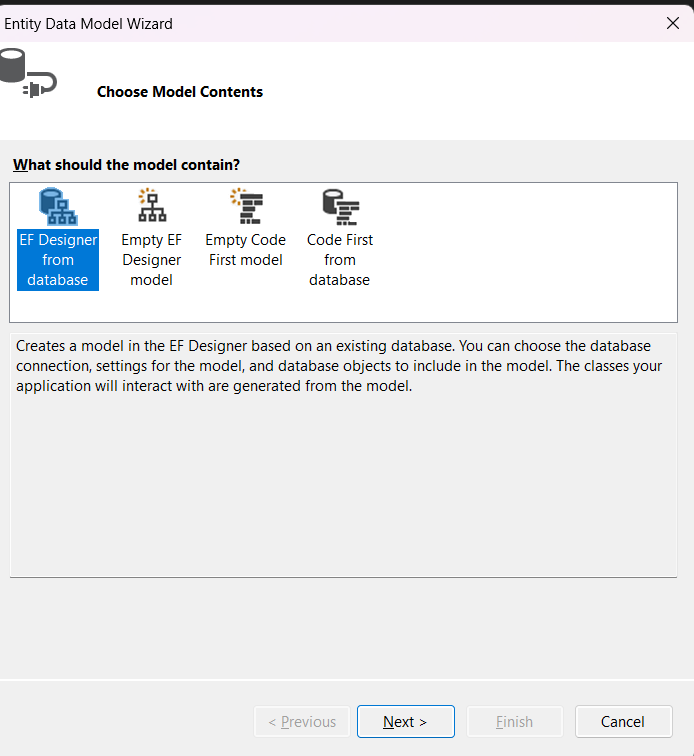
1. Open **Visual Studio**
2. Select **Create a new project**
3. Choose: **ASP.NET Web Application (.NET Framework)**
4. Name: StudentMVCApp
5. Choose **MVC** as the template
6. Click **Create**

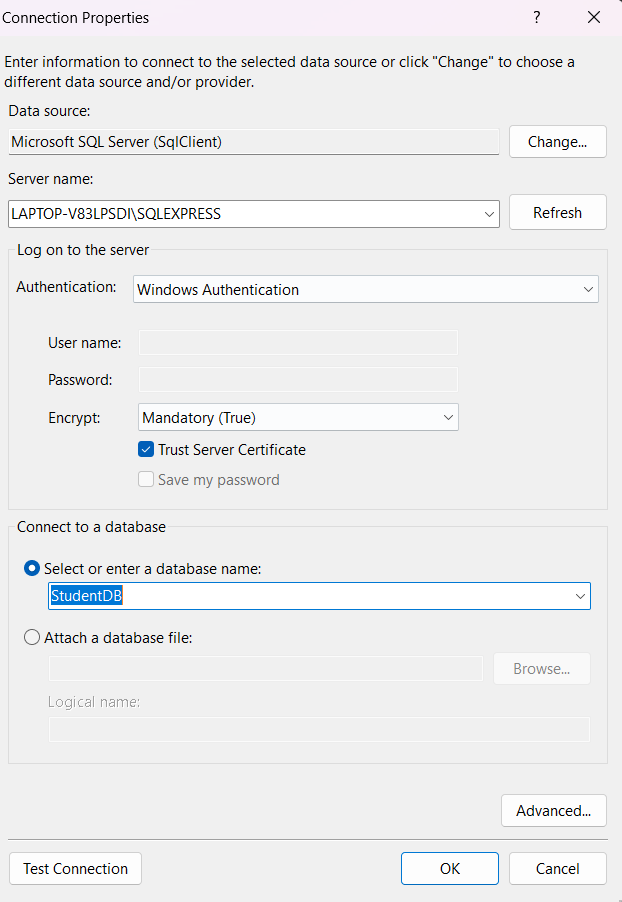
**Entity Framework Model Setup**

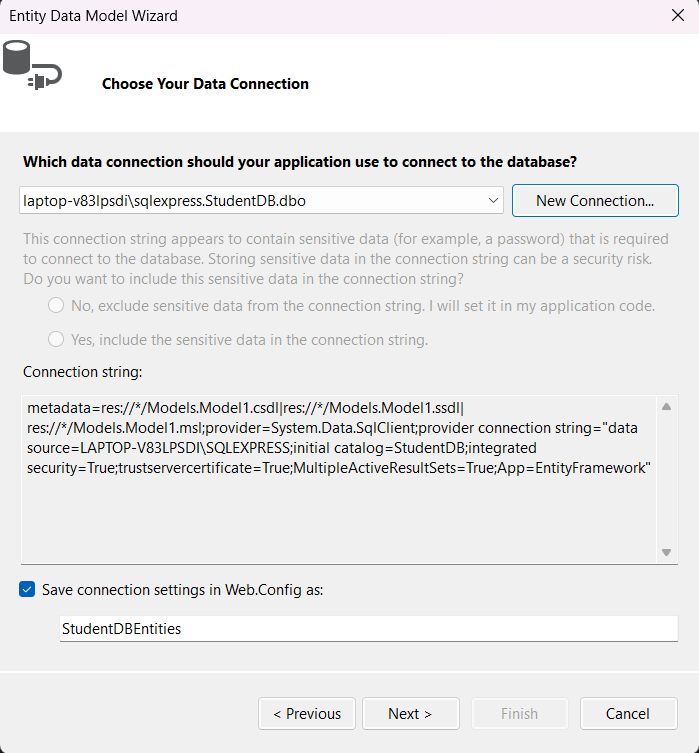
**🔹 Step 3: Add Entity Framework Model**

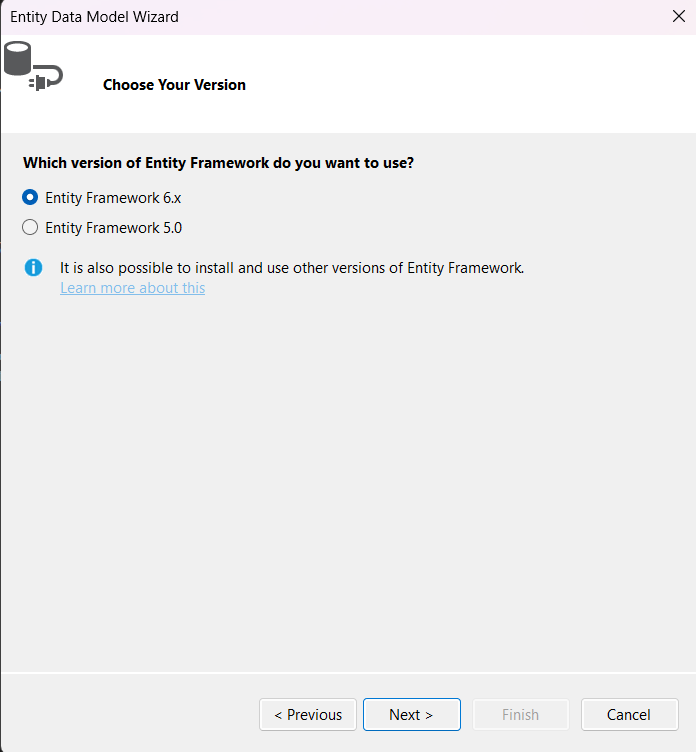
1. Right-click the **Models** folder → Add → New Item
2. Choose **ADO.NET Entity Data Model**
3. Name it: StudentModel.edmx
4. Choose: "EF Designer from database"
5. Select your SQL Server database (StudentDB)
6. Select the Students table
7. Finish to generate model classes

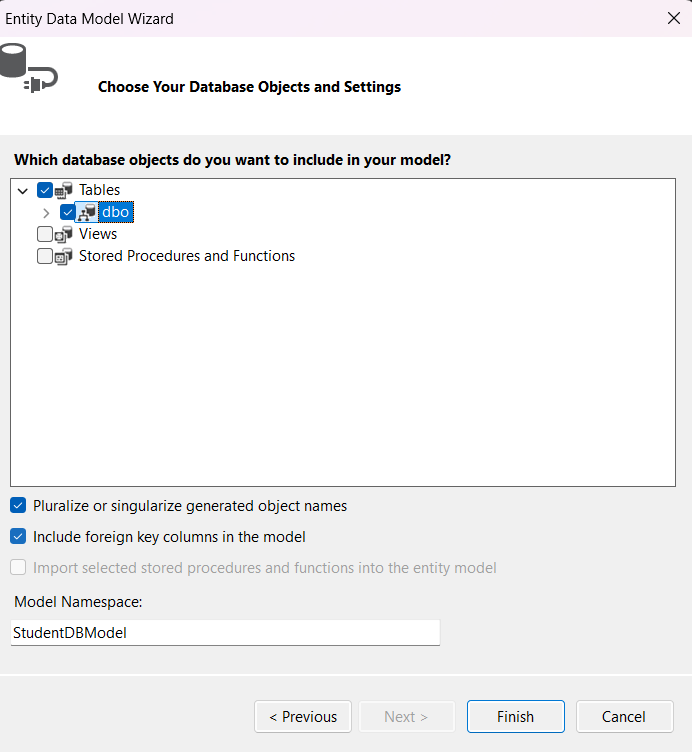


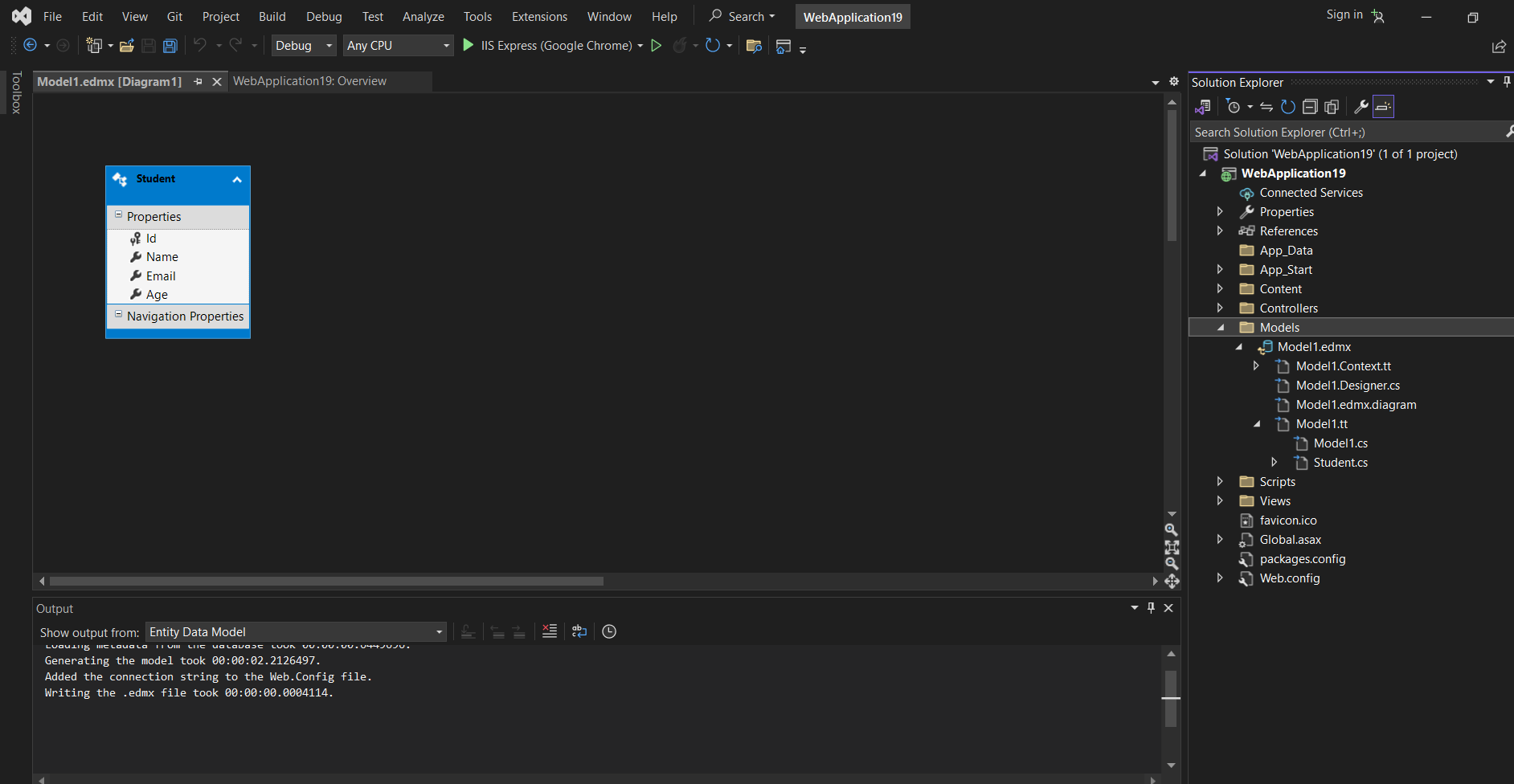






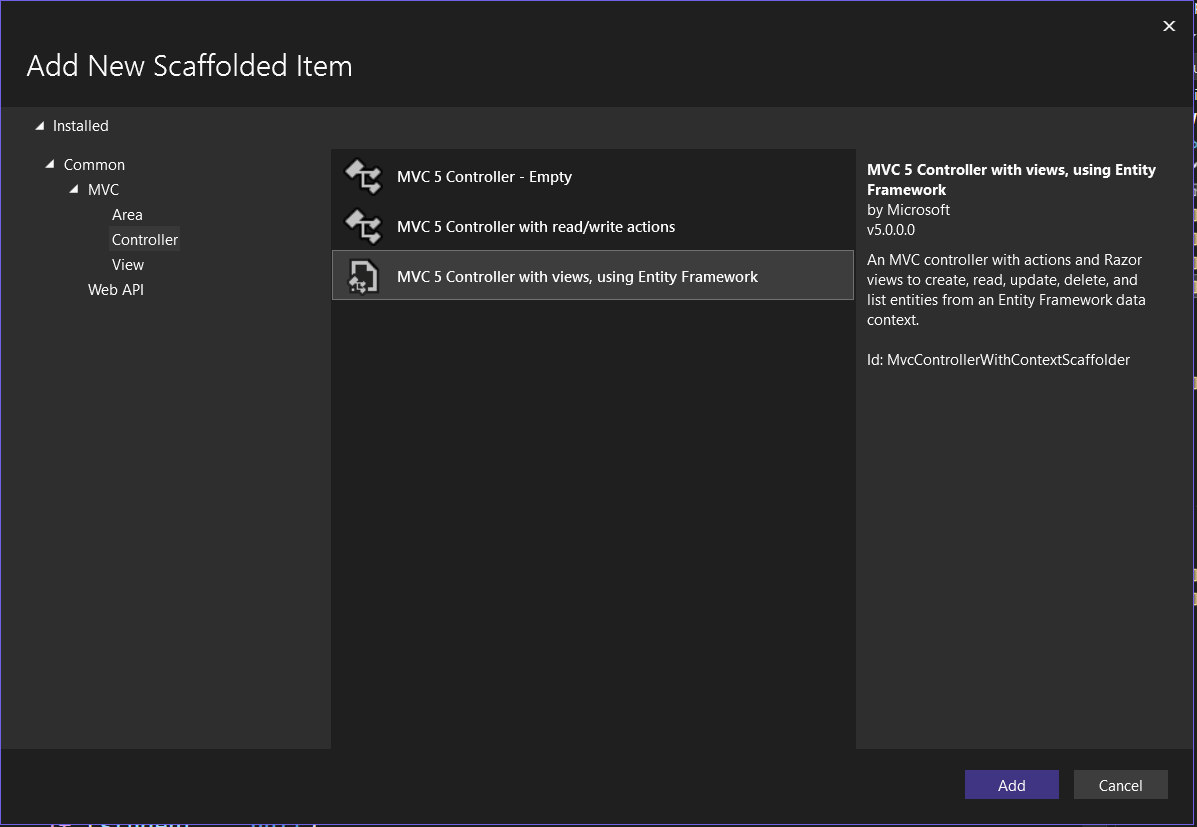


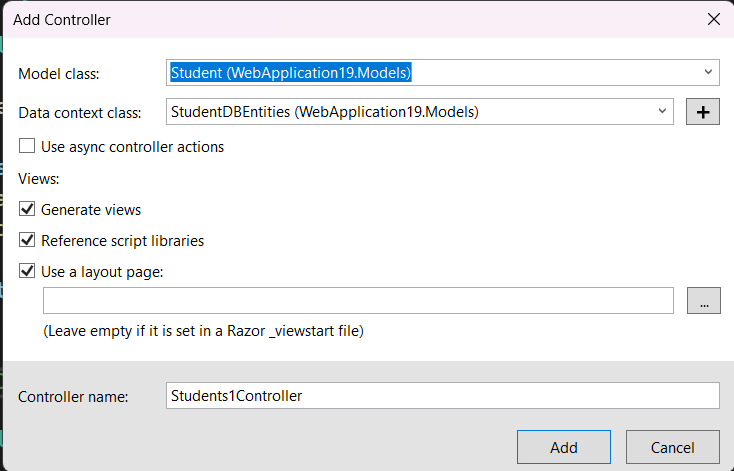


++

**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

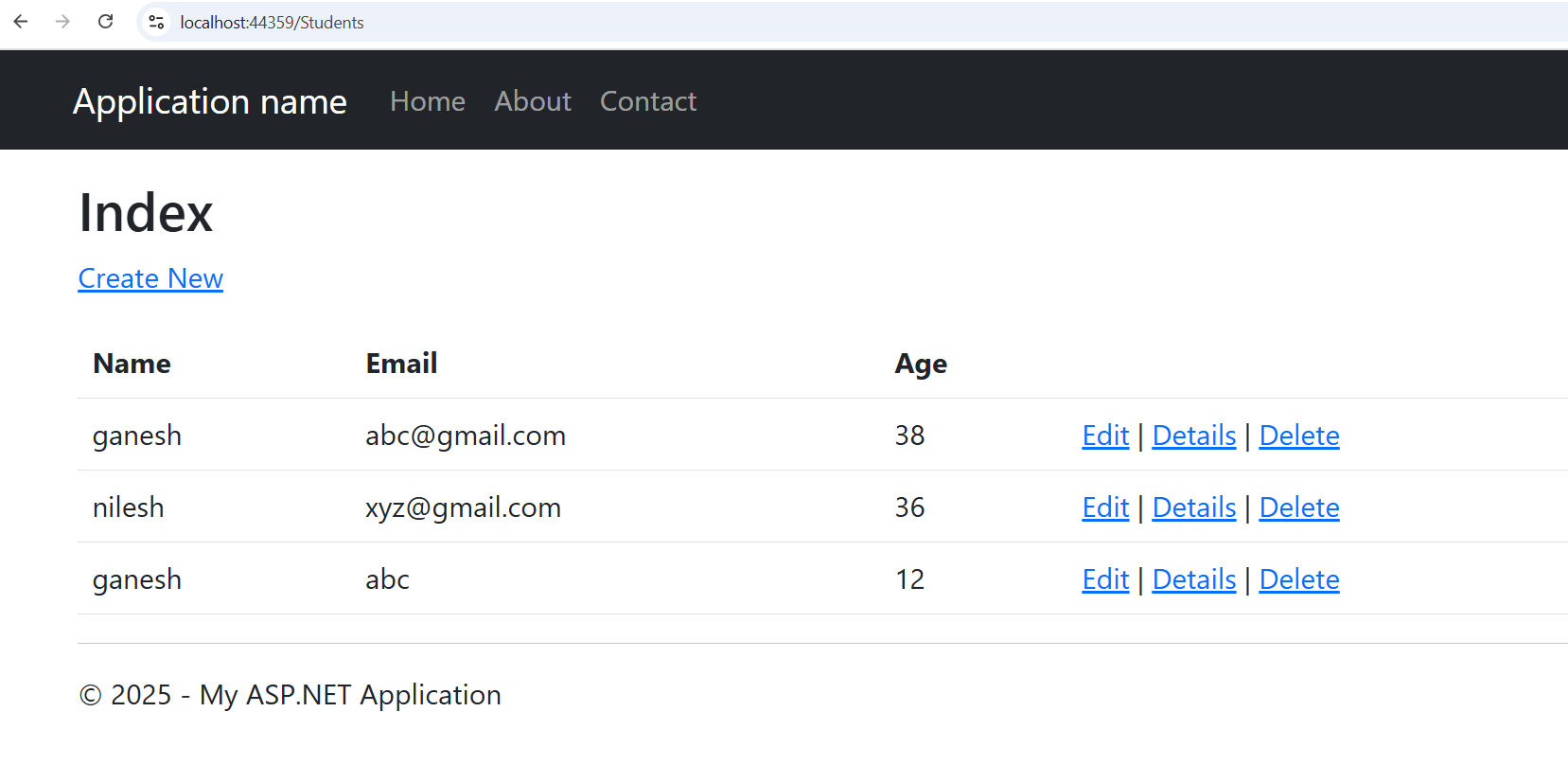
In App\_Start/RouteConfig.cs, change default route to:

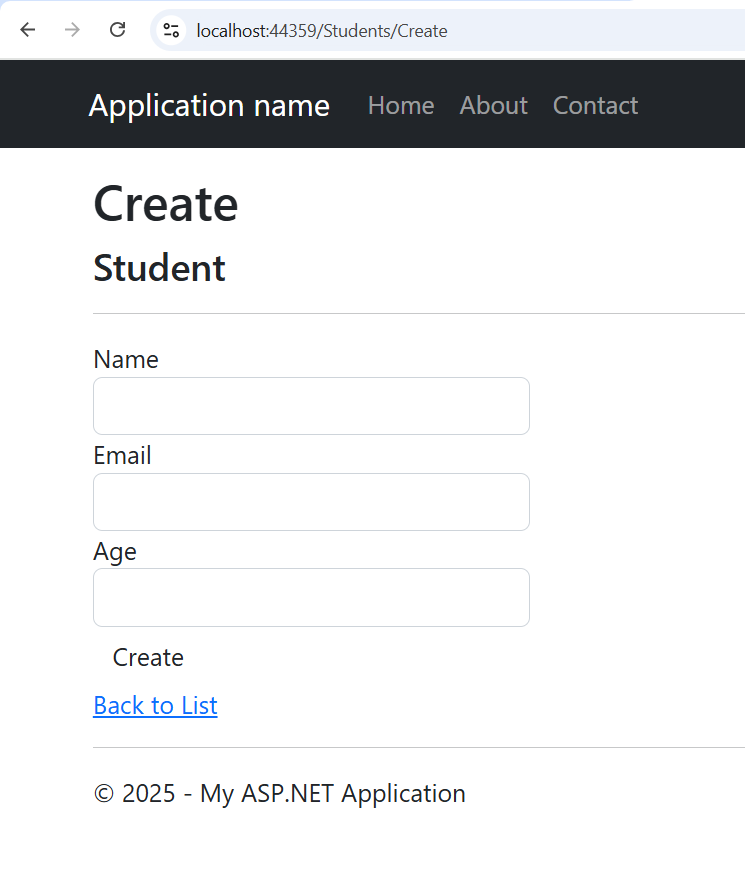
csharp

CopyEdit

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

OUTPUT:





Here’s a full step-by-step walkthrough of **Practical 16: MVC Application using Entity Framework** with detailed **where to click, how to click, and what to do at each step**.

**✅ STEP 1: Create the Database in SQL Server**

**Option 1: Using SQL Server Management Studio (SSMS)**

1. Open **SQL Server Management Studio**.
2. Click **Connect** to connect to your server.
3. In the toolbar, click **New Query**.
4. Copy and paste the following SQL code:
5. CREATE DATABASE StudentDB;
6. USE StudentDB;
7. CREATE TABLE Students (
8. Id INT PRIMARY KEY IDENTITY,
9. Name NVARCHAR(100) NOT NULL,
10. Email NVARCHAR(100) NOT NULL,
11. Age INT NOT NULL
12. );
13. Press **F5** or click the **Execute** button.

**Option 2: Using Visual Studio (SQL Server Object Explorer)**

1. Open **Visual Studio**.
2. View → **SQL Server Object Explorer**.
3. Right-click **SQL Server** → **Add SQL Server** → Connect.
4. Right-click **Databases** → **Add New Database** → Name it StudentDB.
5. After creation, right-click StudentDB → **New Query** → Paste the same SQL code above (starting from USE StudentDB) and **Execute**.

**✅ STEP 2: Create a New ASP.NET MVC Project**

1. Open **Visual Studio**.
2. Go to **File** → **New** → **Project**.
3. In the dialog, choose:
   * Project type: **ASP.NET Web Application (.NET Framework)**
   * Name: StudentMVCApp
   * Click **Create**.
4. In the template selection, choose **MVC** and click **Create**.

**✅ STEP 3: Add Entity Framework Model (EDMX)**

1. In **Solution Explorer**, right-click the **Models** folder → **Add** → **New Item**.
2. Select **Data** → Choose **ADO.NET Entity Data Model**.
3. Name it: StudentModel.edmx → Click **Add**.
4. In the Wizard:
   * Select **EF Designer from Database** → Click **Next**.
   * Click **New Connection** → Choose your **server name**, database: StudentDB.
   * Test connection → Click **OK**.
5. Choose **Yes, include sensitive data** → Next.
6. Select **Students** table → Finish.

✅ A new model StudentModel.edmx will be generated with Student class.

**✅ STEP 4: Create Controller**

1. In **Solution Explorer**, right-click the **Controllers** folder → **Add** → **Controller**.
2. Choose: **MVC 5 Controller with views, using Entity Framework** → Click **Add**.
3. In the Add Controller dialog:
   * **Model class**: Select Student (StudentMVCApp.Models)
   * **Data context class**: Select StudentDBEntities
   * Click **Add**
4. Visual Studio will auto-generate **CRUD actions and views**.

**✅ STEP 5: Views Setup (Auto-generated)**

* By default, all views (Index, Create, Edit, Details, Delete) are auto-created in:
* Views → Students
* To manually check:
  + Expand Views/Students
  + Open **Index.cshtml** to verify the list view

**✅ STEP 6: Set Default Route**

1. Go to **App\_Start** → Open RouteConfig.cs.
2. Find the line:
3. defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
4. Change it to:
5. defaults: new { controller = "Students", action = "Index", id = UrlParameter.Optional }

**✅ FINAL STEP: Run the Project**

1. Press **Ctrl + F5** or click the **Start button (▶)**.
2. The browser should open with the Student List.

**✅ OUTPUT PREVIEW**

**Student List Page:**

| **Name** | **Email** | **Age** | **Actions** |
| --- | --- | --- | --- |
| John Smith | [john@example.com](mailto:john@example.com) | 21 | Edit |

Would you like the **viva questions** or **code for customizing the views** next?