Lab Manual for Cloud Computing

Lab No. 4

MVC Entity Framework and Query data

LAB 04: MVC ENTITY FRAMEWORK AND QUERY DATA

1. INTRODUCTION:

What is ADO .NET?

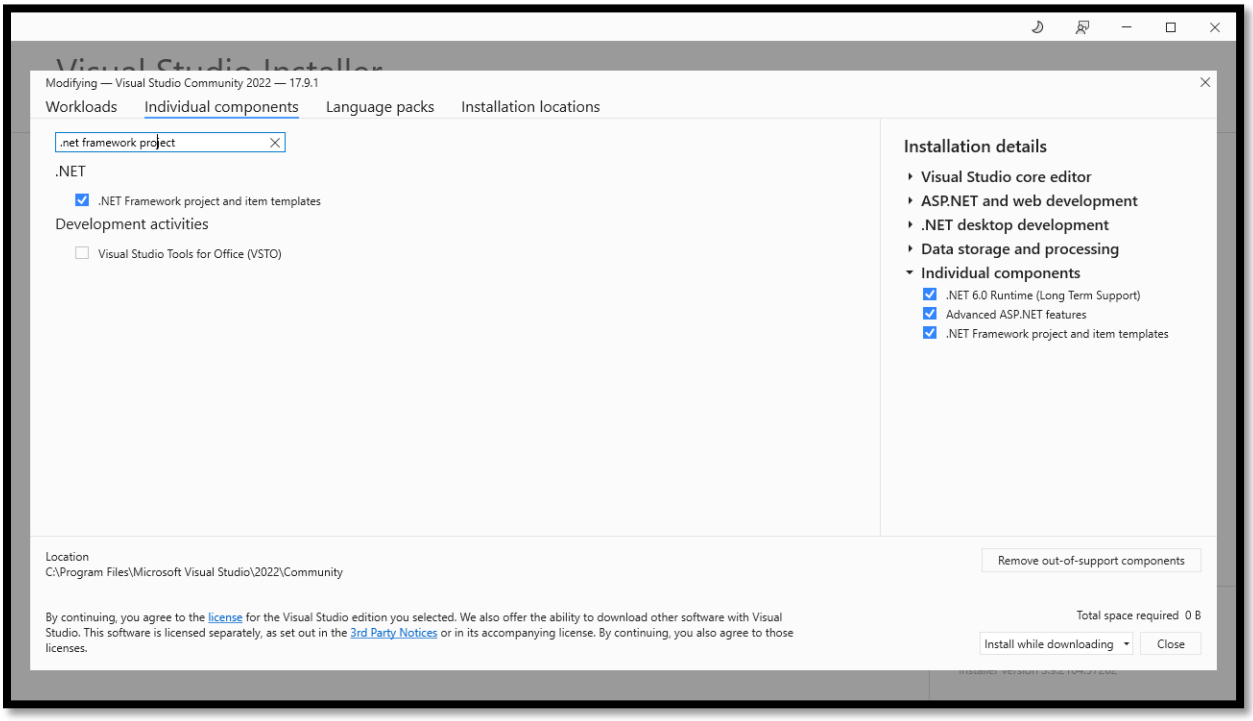
The integration of MVC (Model-View-Controller) architecture and Entity Framework Core provides a robust framework for developing data-centric applications within the .NET ecosystem. MVC's separation of an application into Model, View, and Controller components offers a modular structure, facilitating organized development and maintenance. On the other hand, Entity Framework Core, as a modern Object-Relational Mapping (ORM) framework, simplifies database access by allowing developers to work with strongly-typed objects and classes, abstracting the complexities of direct SQL interactions.

Within this framework, developers follow a structured approach to query and interact with data. They define model classes representing entities in the application, configure a DbContext class to handle database interactions, and utilize LINQ queries for expressing database queries in a familiar C# syntax. This tight integration between MVC and Entity Framework Core streamlines the development process, enabling developers to focus on application logic and user experience while efficiently managing data interactions with various databases. The combination of MVC architecture and Entity Framework Core not only enhances code maintainability but also promotes a scalable and standardized approach to building modern, data-driven applications in the .NET environment.

Tools Required:

1. **Visual studio 2022:** (make sure that .net frameworks projects and templates are installed in the software.

If they aren't installed, run the visual studio installer, and go to the option "modify". In the Individual Components, write .net framework project and then click option modify.



Getting Started Working with .Net Framework and Querying data in it

In this .NET Framework project, we aim to develop a student application capable of performing CRUD operations. To facilitate this, we have established a student database and created the necessary table through meticulous database and table creation steps as:

1. Creating database named "Dem".

```
create database Demo;
```

2. Using the created database and then design tabel named as Student.

```
use Demo;


CREATE TABLE students (
    student_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    age INT,
    grade VARCHAR(2)
);
```

3. Now inserting values into the created tabel.

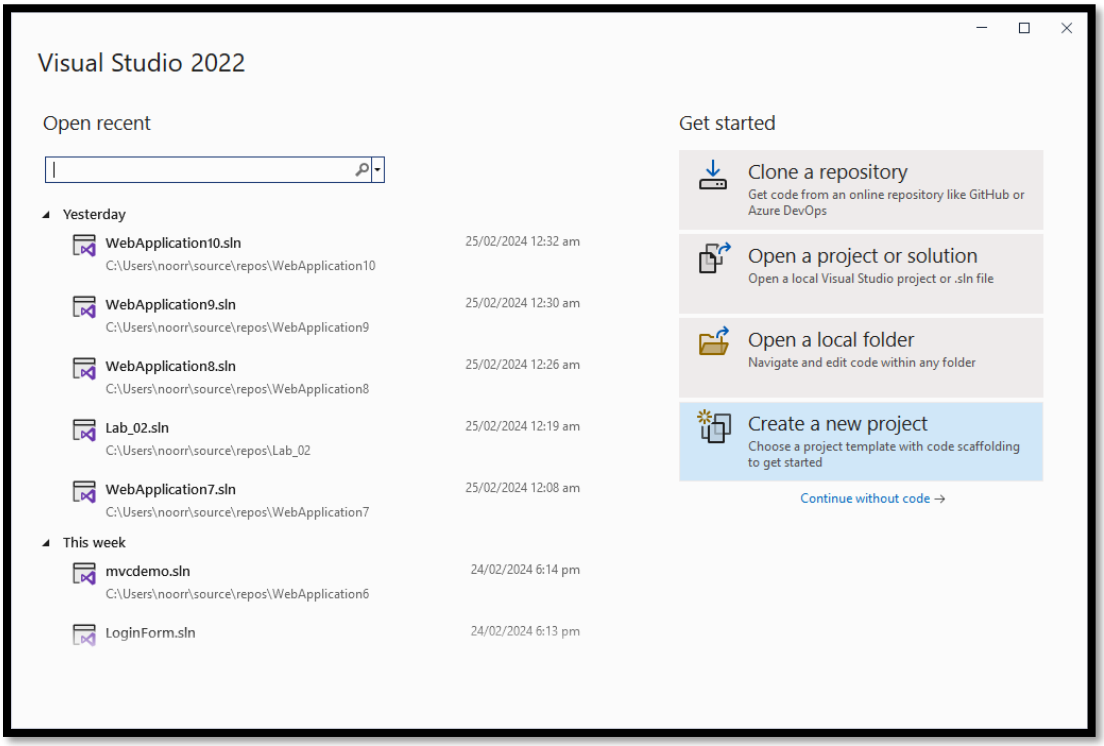
```
INSERT INTO students (student_id, first_name, last_name, age, grade) VALUES
(1, 'John', 'Doe', 20, 'A'),
(2, 'Jane', 'Smith', 22, 'B'),
(3, 'Mike', 'Johnson', 21, 'C'),
(4, 'Emily', 'Williams', 19, 'A'),
(5, 'Chris', 'Taylor', 23, 'B');
```

Now we'll move towards creating the project in visual studio.

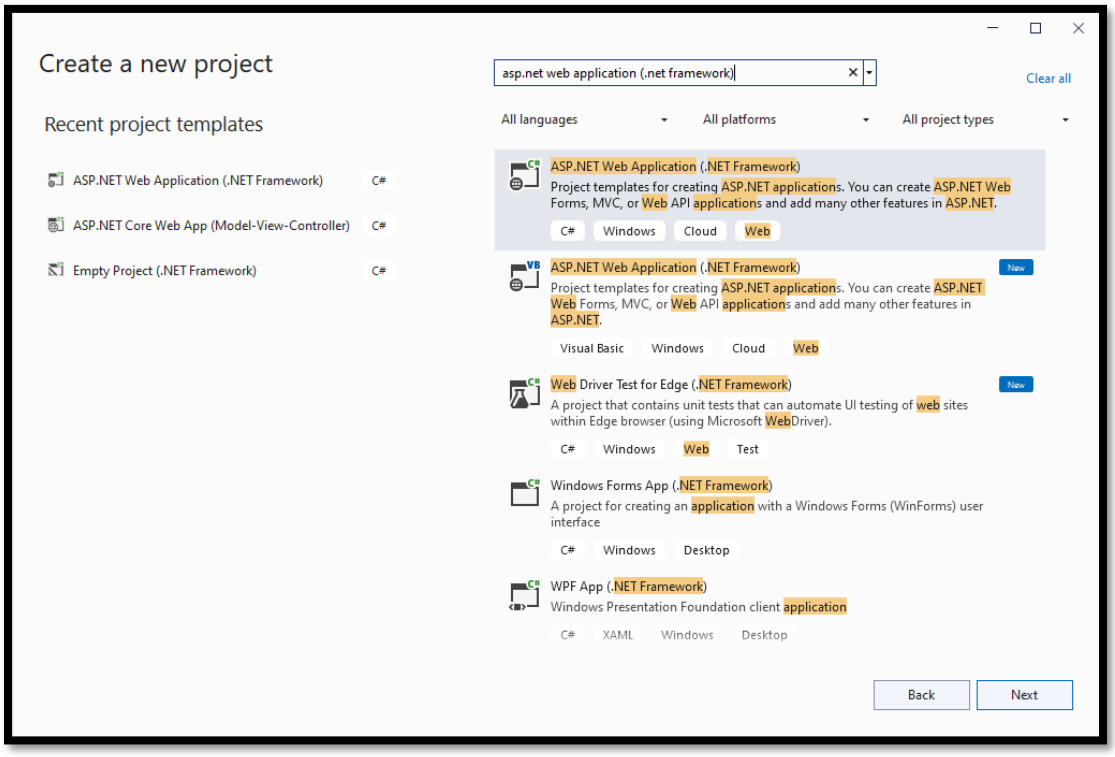
Step 1: Create a new project

- Open Visual Studio 2022.

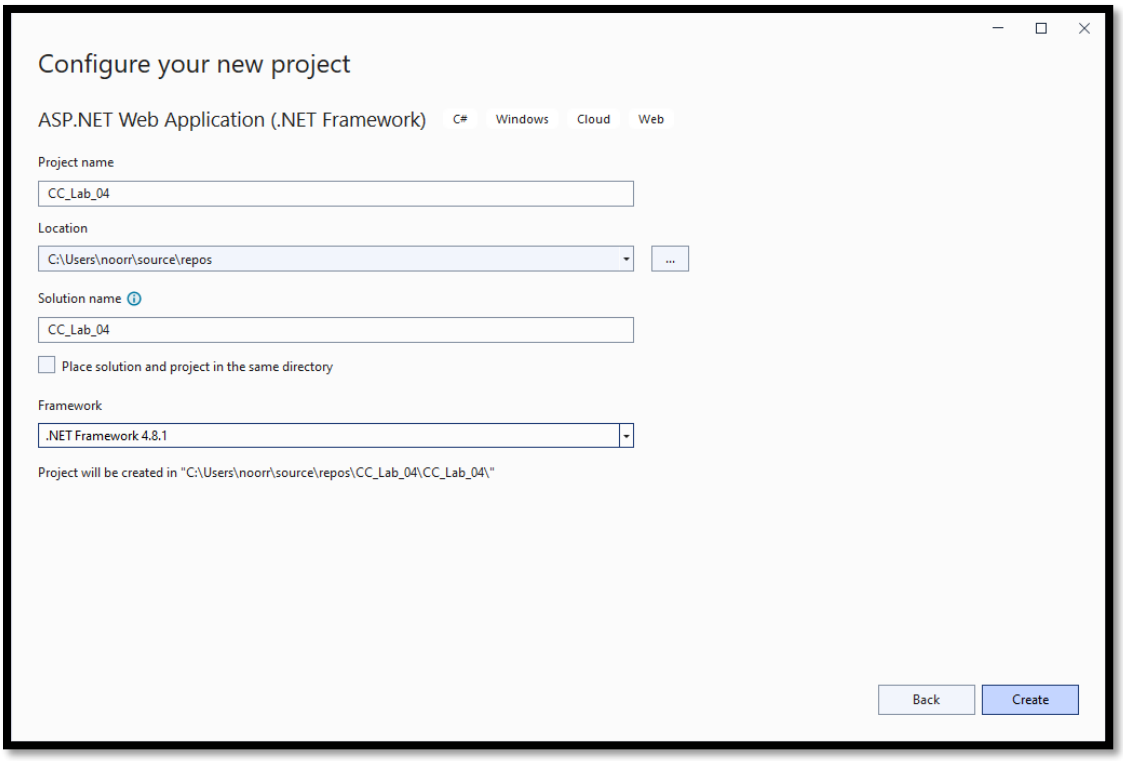
- Click on "Create a new project" in the start window.



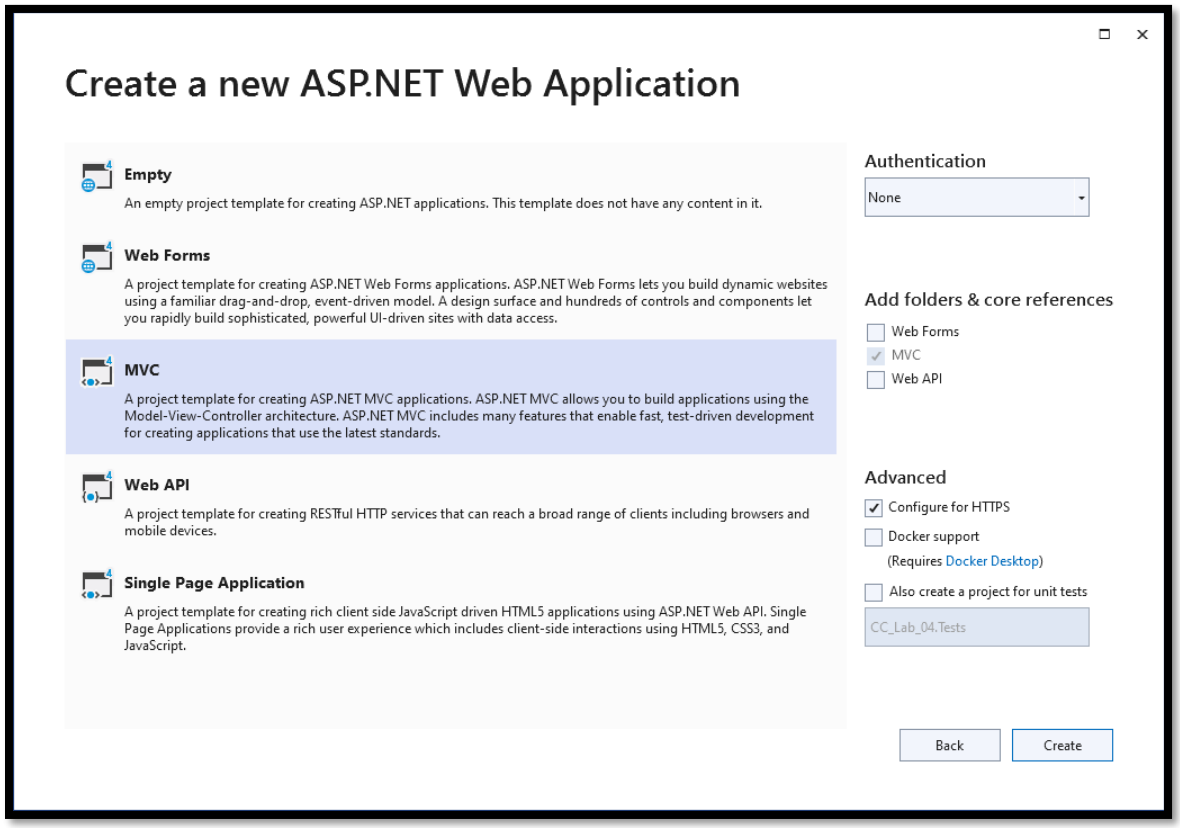
- In the "Create a new project" window, search for " ASP.NET Web Application (.Net Framework) in the search bar, and then click NEXT.

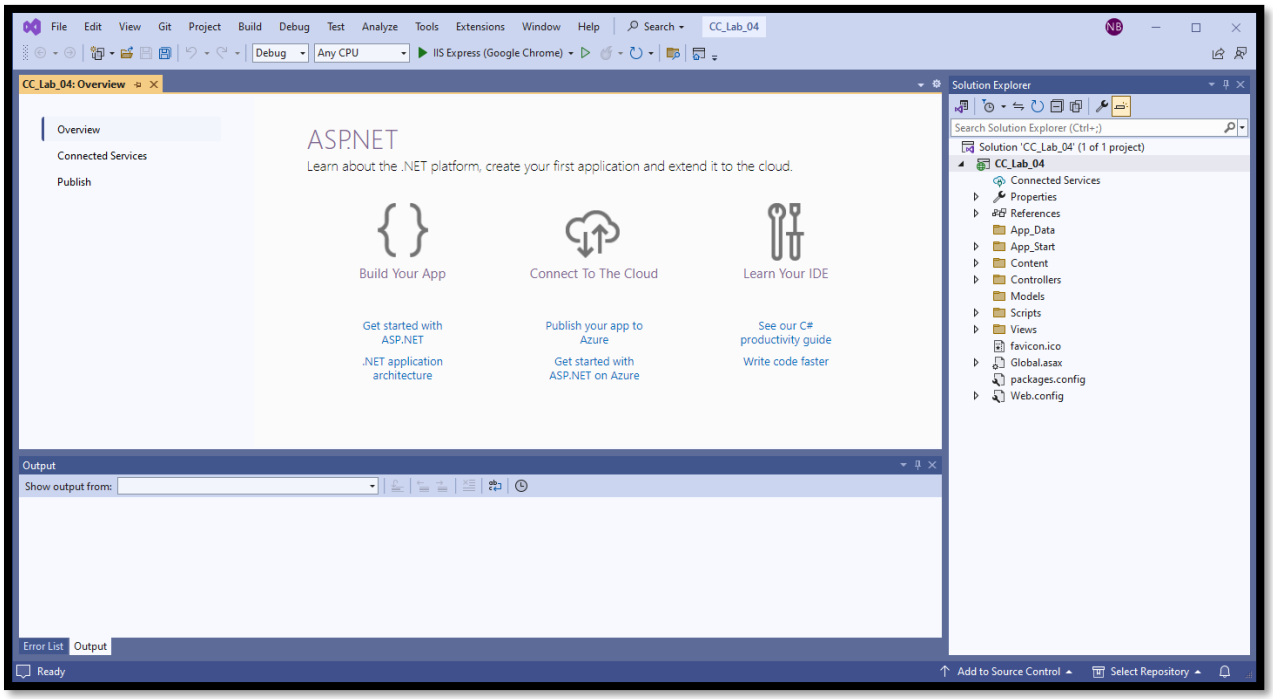


- Provide a name and location for your project and then Click NEXT.



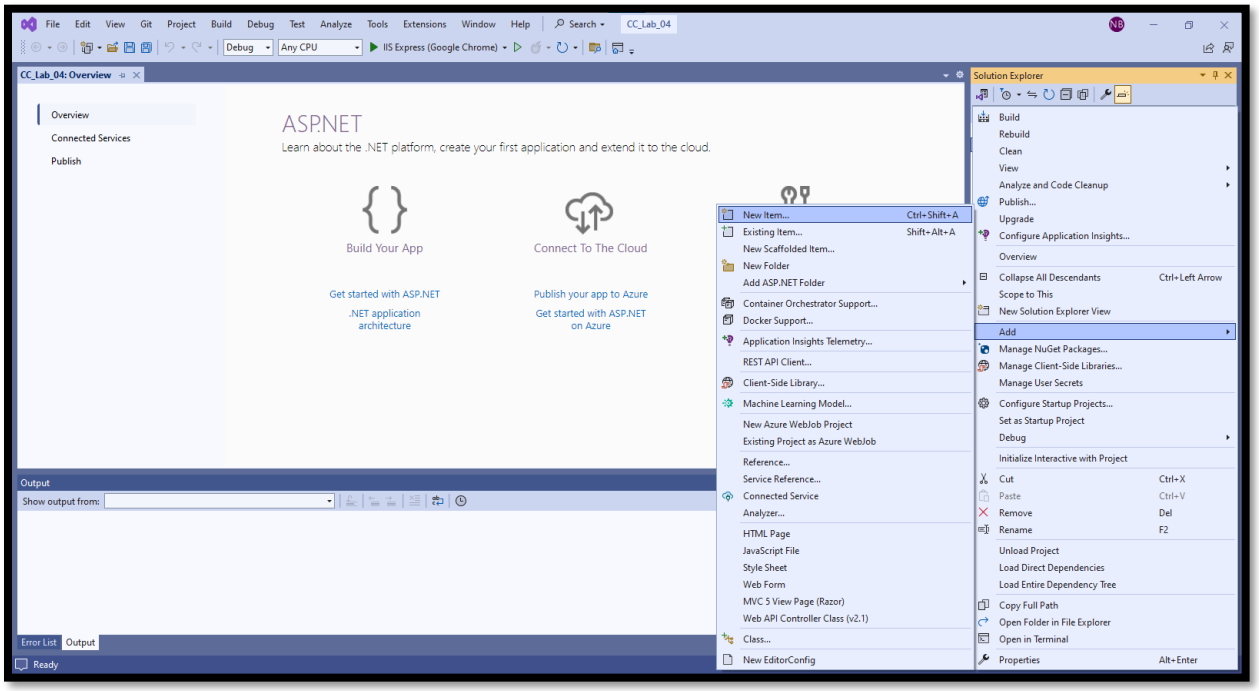
- In the next window, select "ASP.NET Core Web App (Model-View-Controller)" as the project template and click "Create."



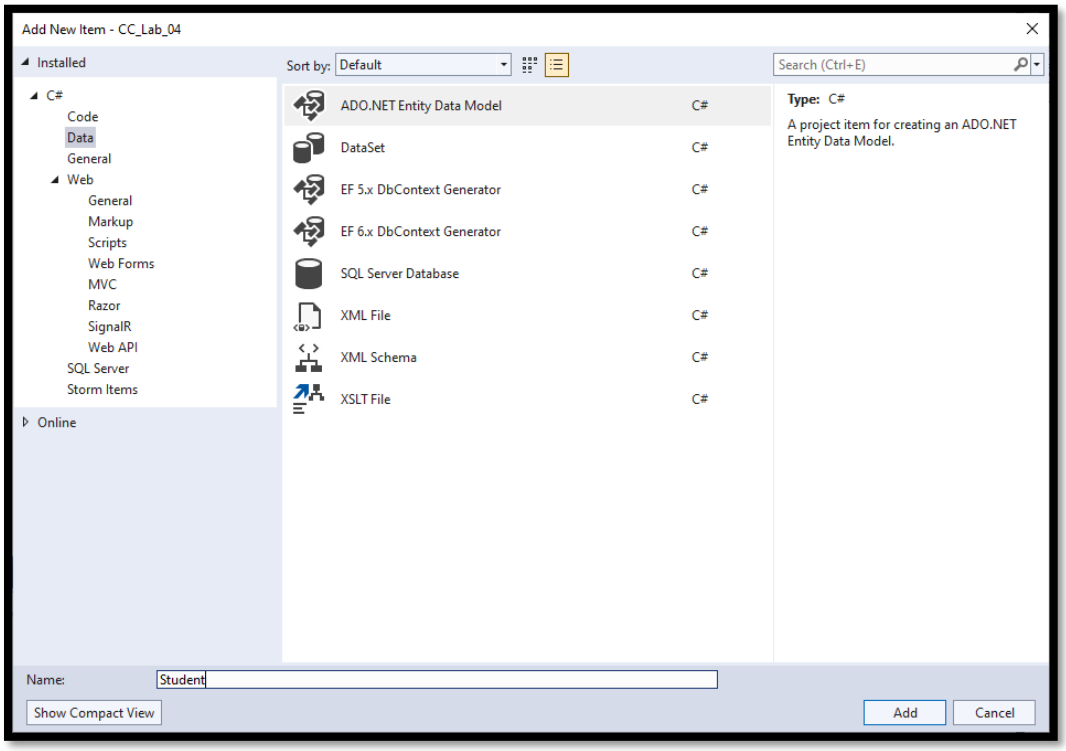


Step 2: Adding ADO.NET Entity Framework

- Right-click on your project in Solution Explorer, and then Choose "Add" > "New Item".

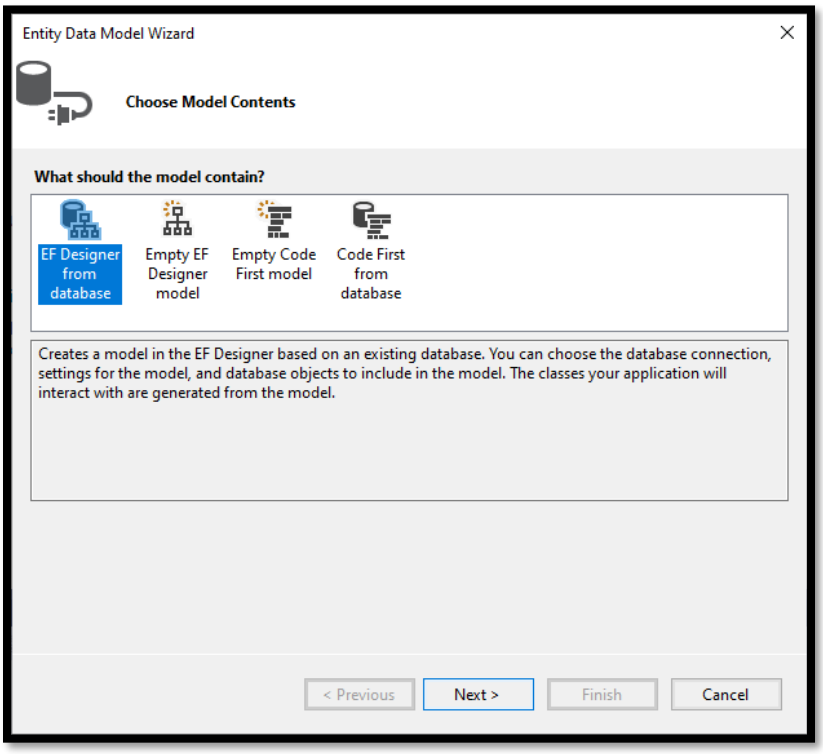


- In the "Add New Item" window, select "Data" from the left panel and then Choose "ADO.NET Entity Data Model", name it as Student and click "Add."

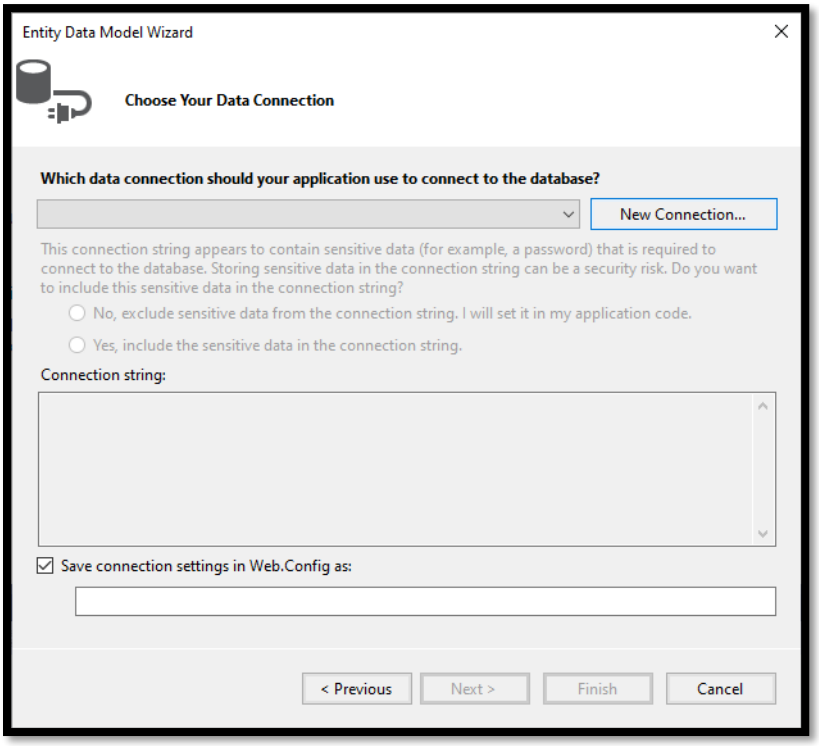


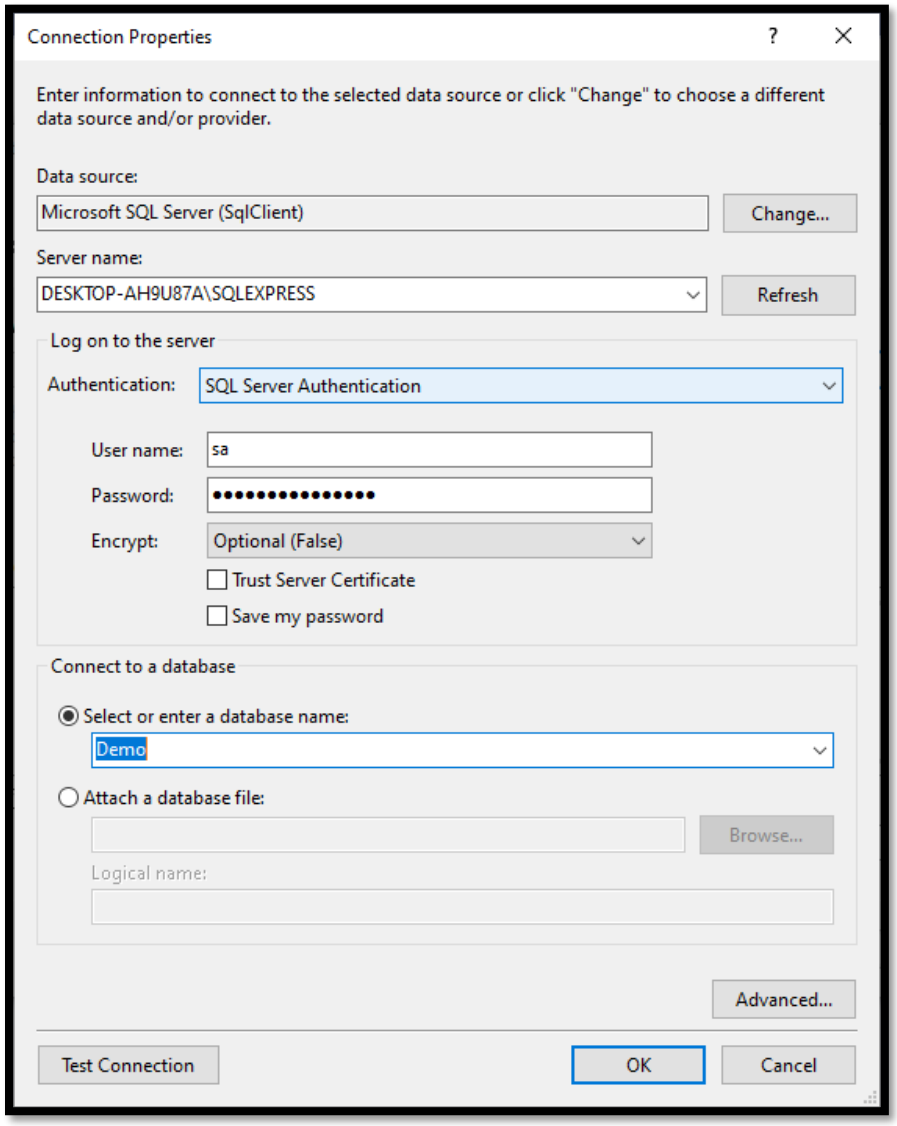
Step 3: Configuring ADO.NET Entity Data Model

- In the "Entity Data Model Wizard," choose the "EF Designer from database" option and click "Next."

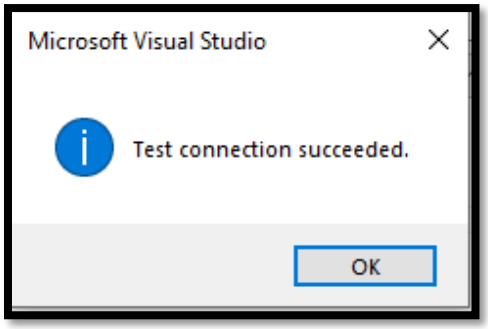


- Configure your database connection by clicking "New Connection" and entering your database details.



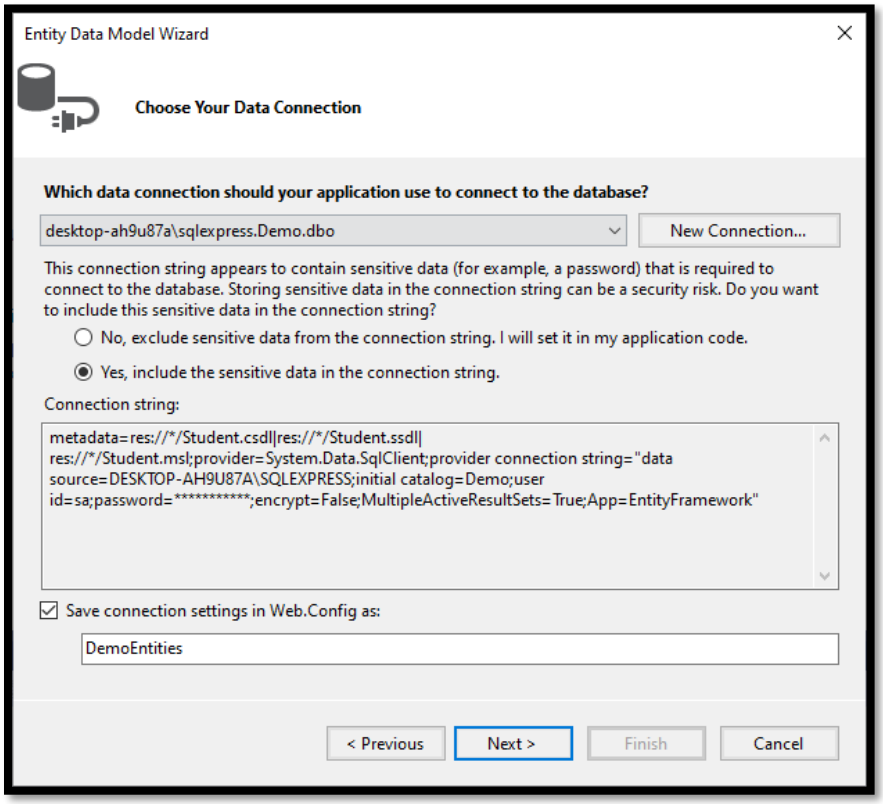


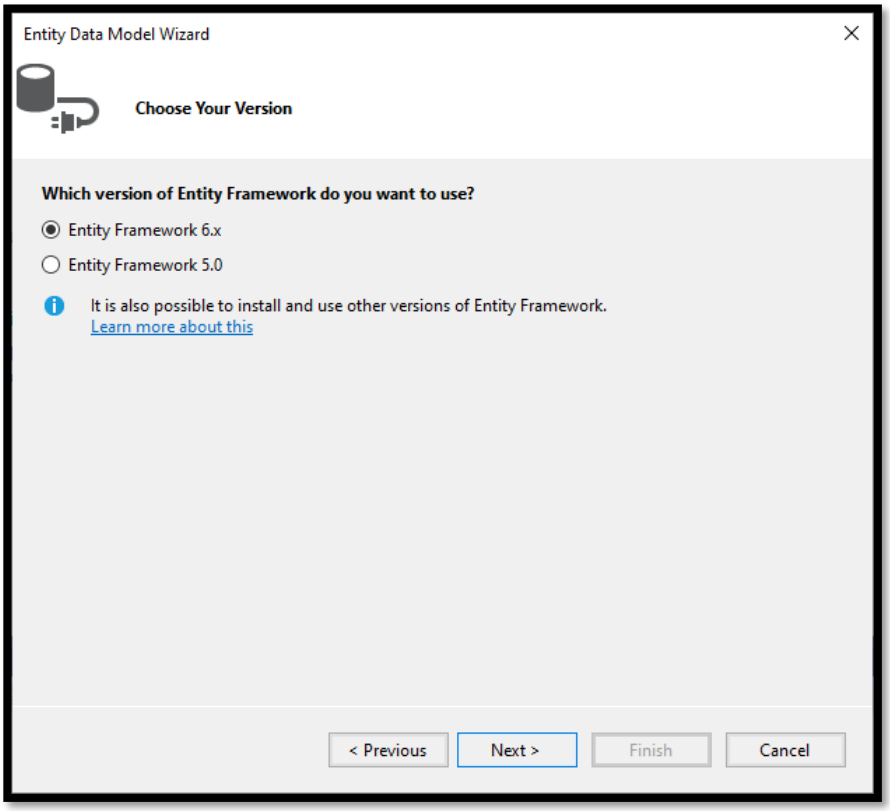
- After entering the connection details, click "Test Connection" to verify and then click "OK.", if the test connected succeeded, click on OK.



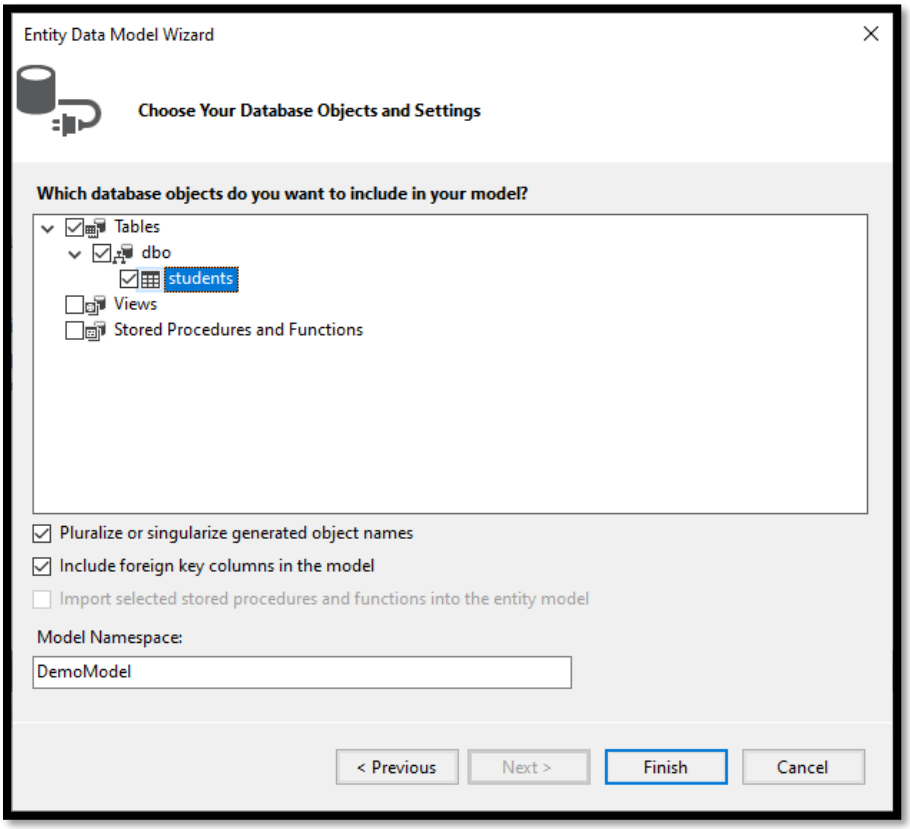
-

- Select the option for including sensitive data and then click NEXT and select the entity framework version.

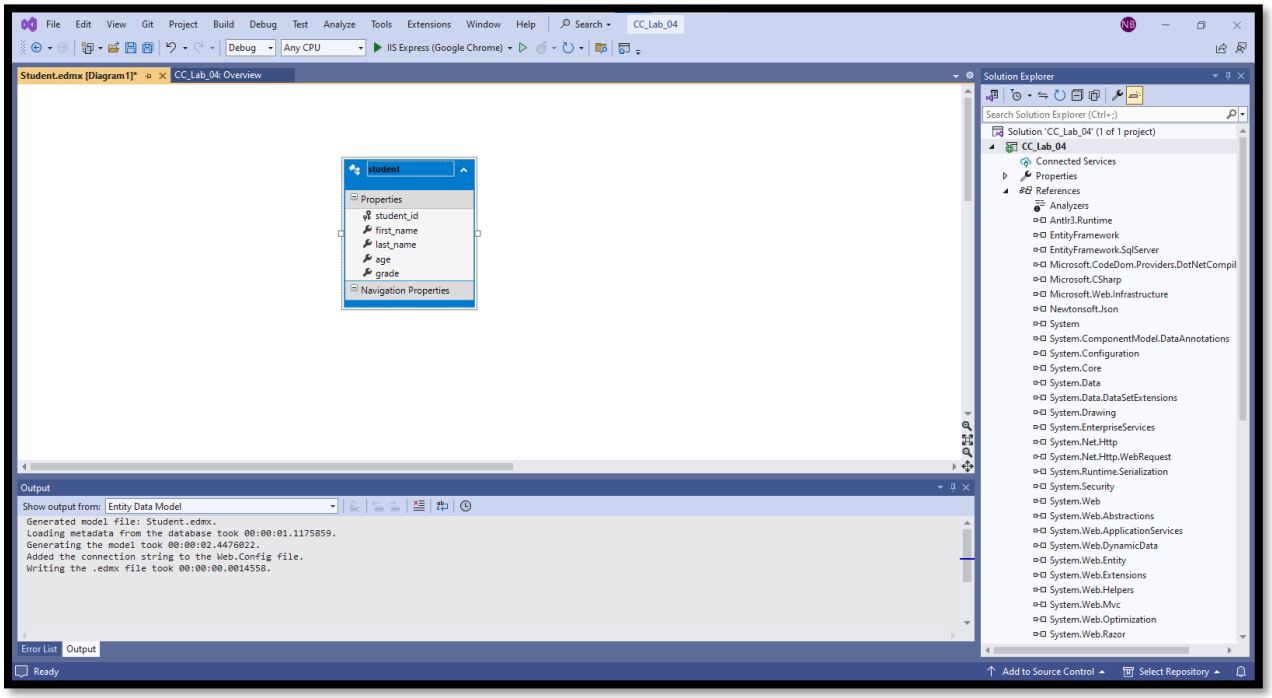




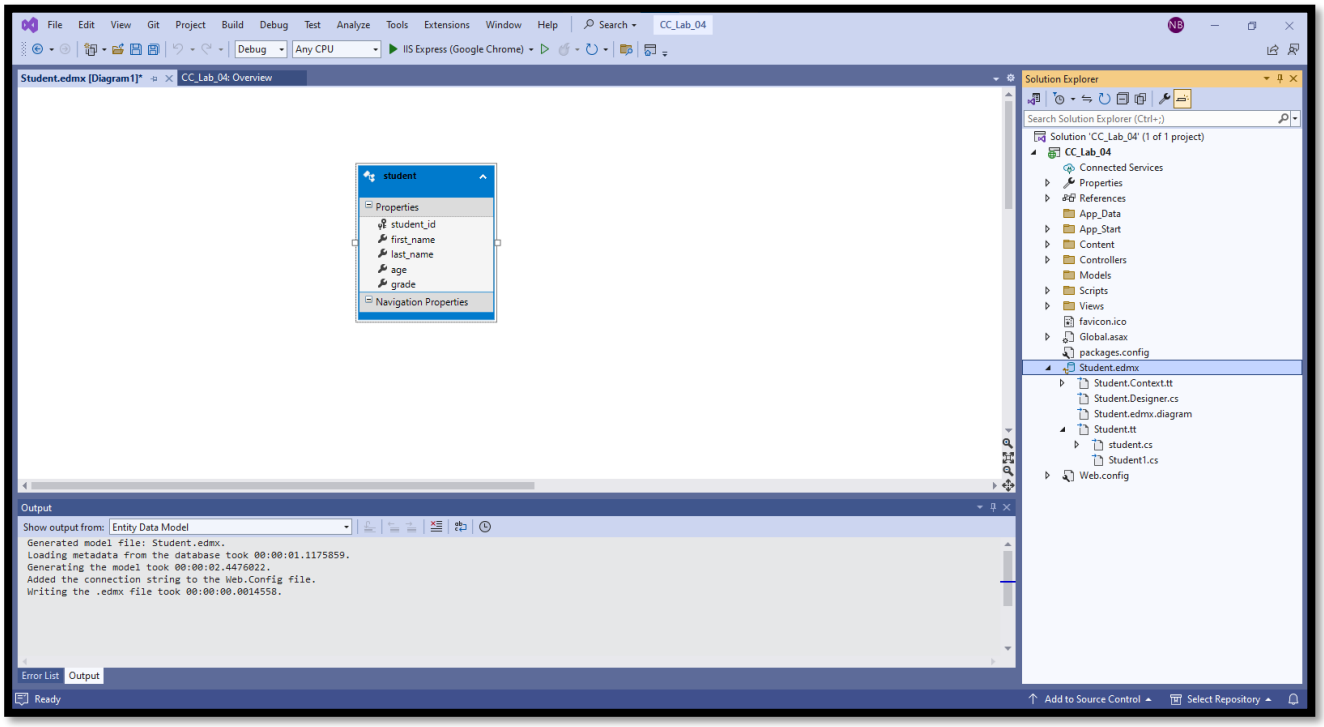
- In the next window, choose the tables you want to include in the model (For now as we have only Student tabel, s select it") and click "Finish."

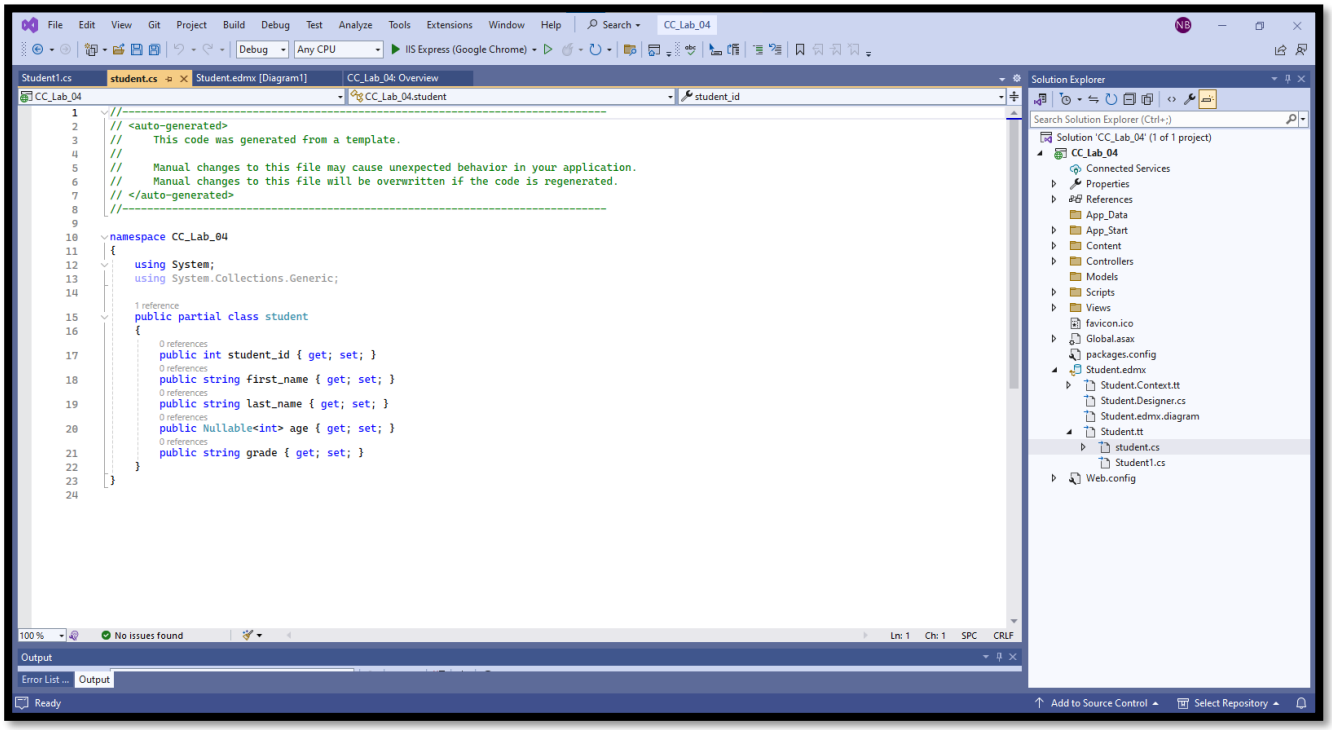


- After this Visual Studio will generate a database diagram for the table that we chose.



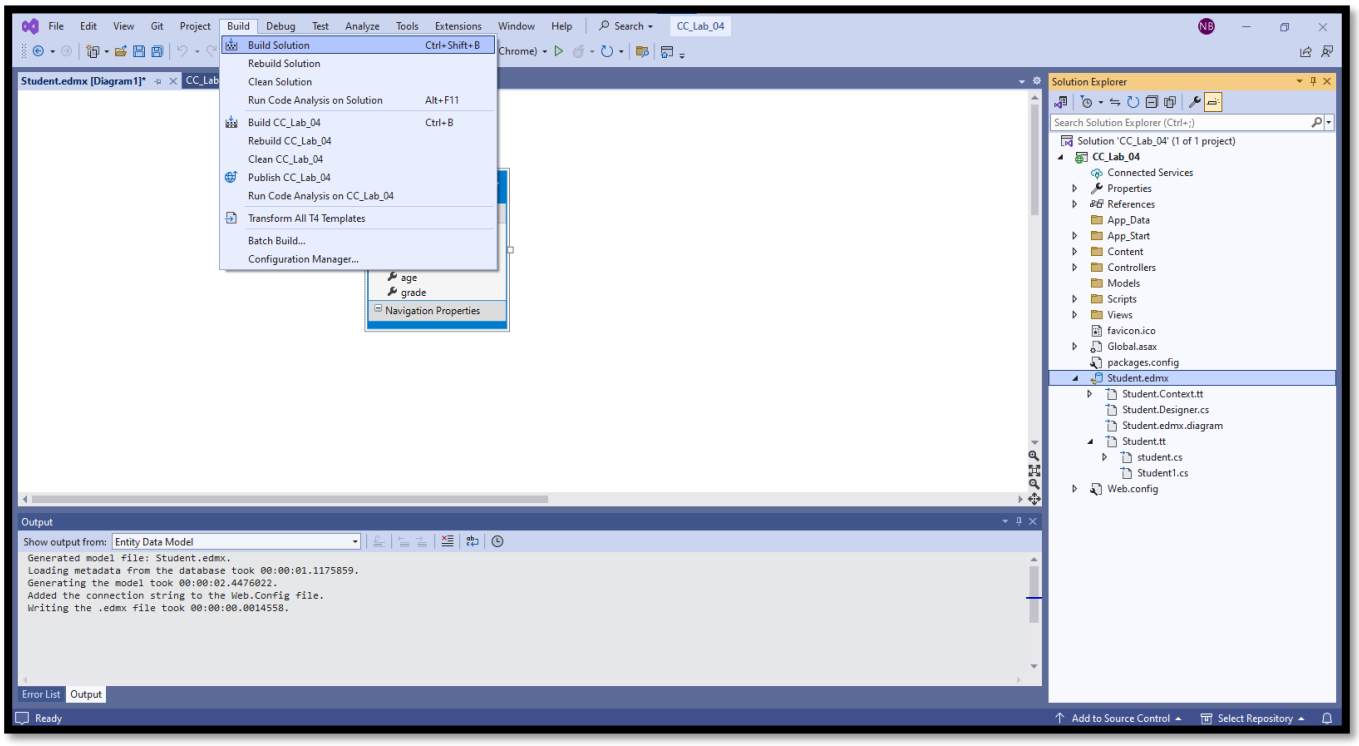
- In Solution Explorer you will find the model and your tables.



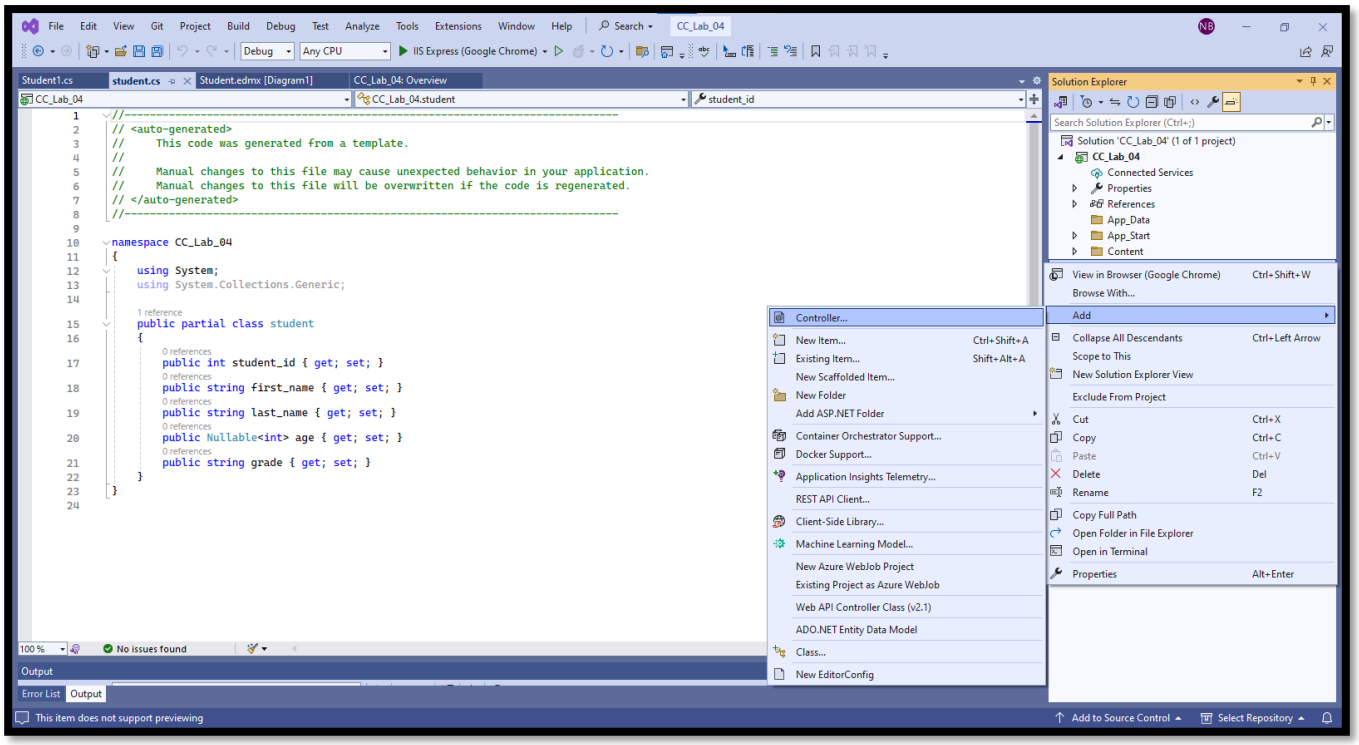


Step 5: Build the project and add the Controller

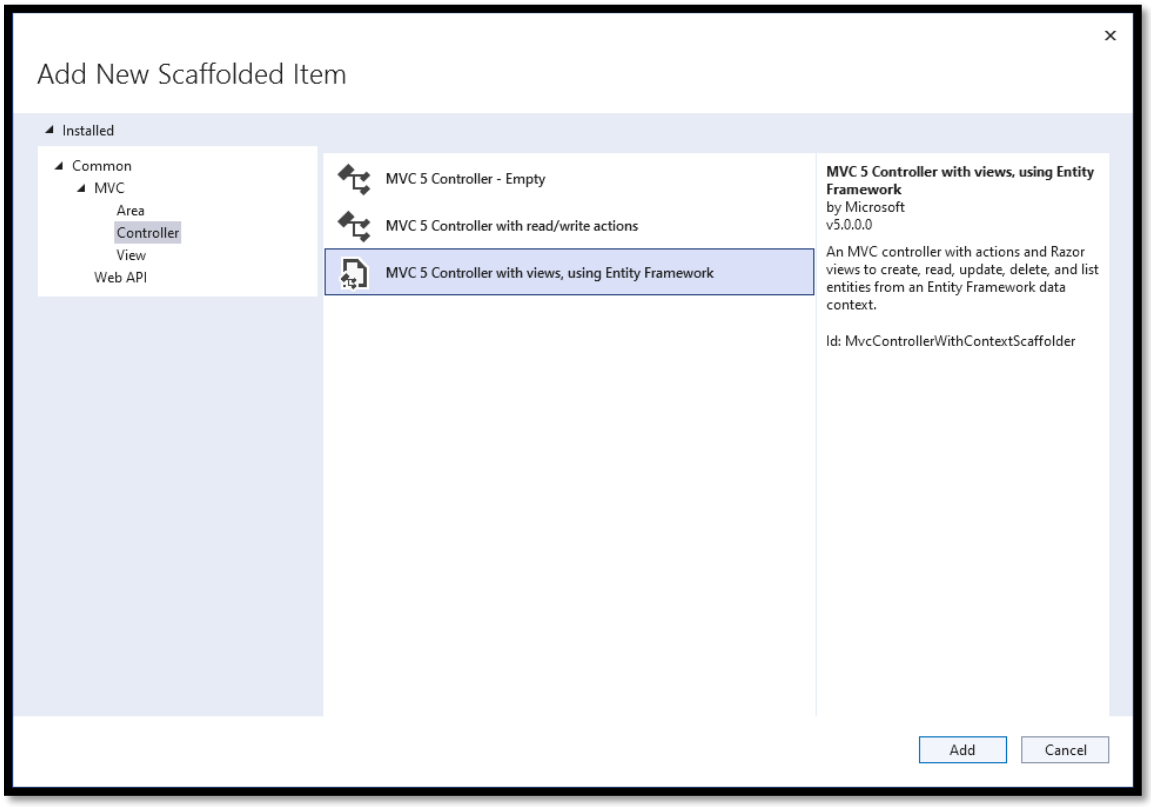
- Save your project and then build your project.



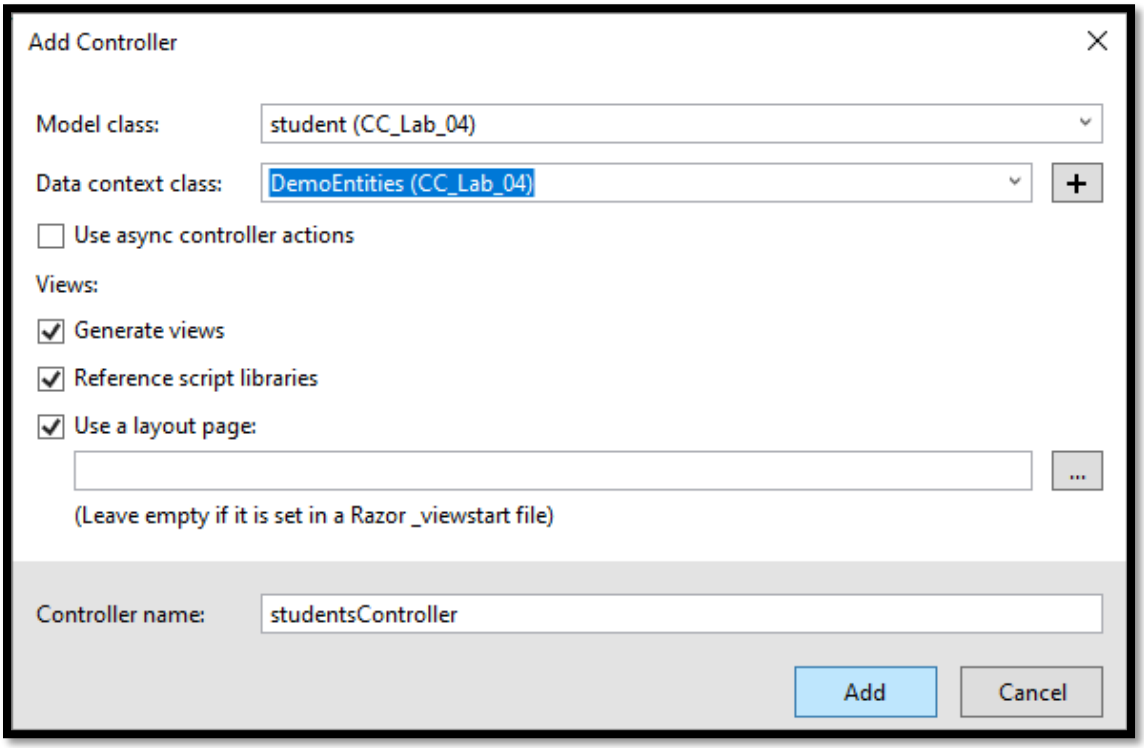
- Now Right-click on the project in Solution Explorer. Choose "Add" > "Controller..."



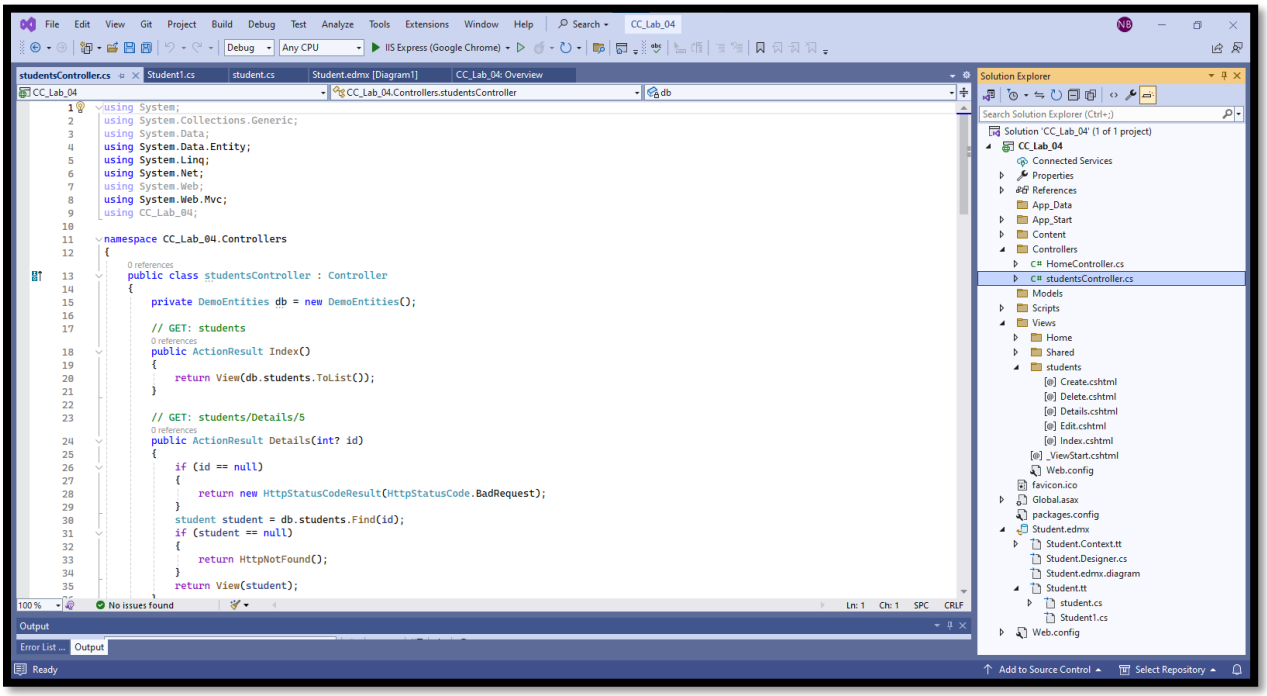
- In the "Add Scaffold" window, choose "MVC Controller with views, using Entity Framework" template.



- Select the Model class (e.g., "Student") and Data context class. Click "Add."

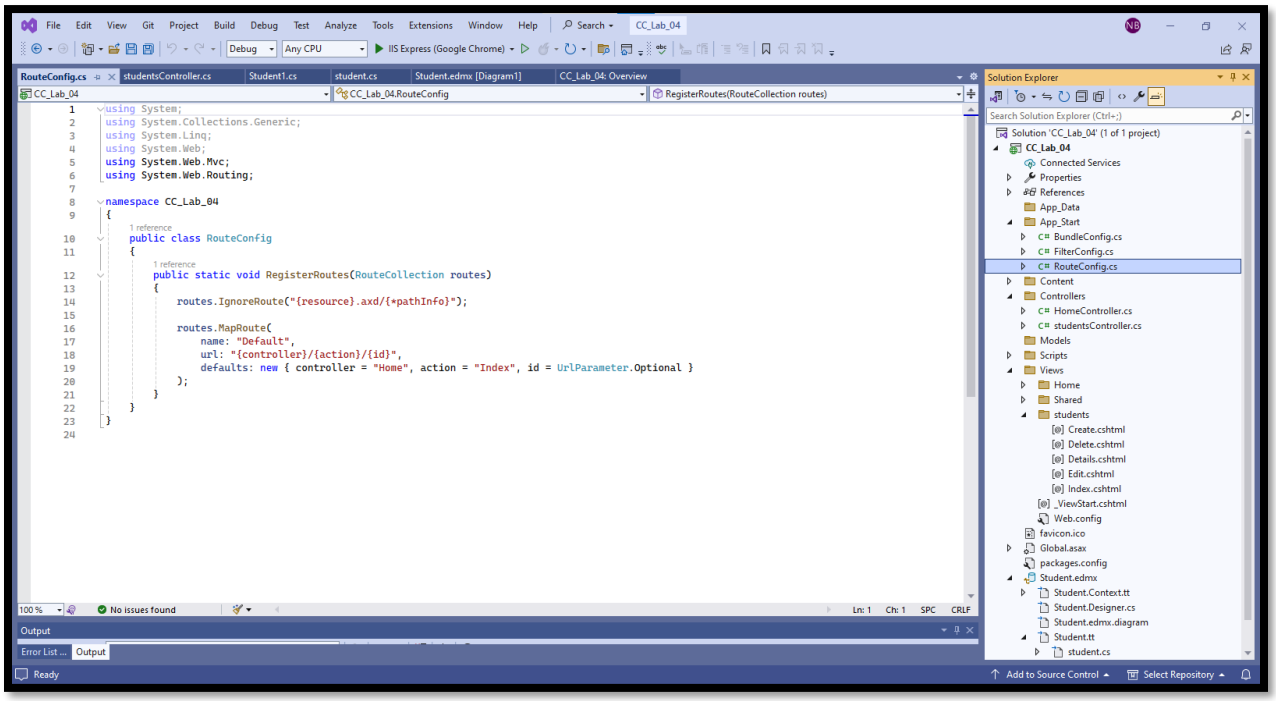


- The controller as well as views will be generated for student.

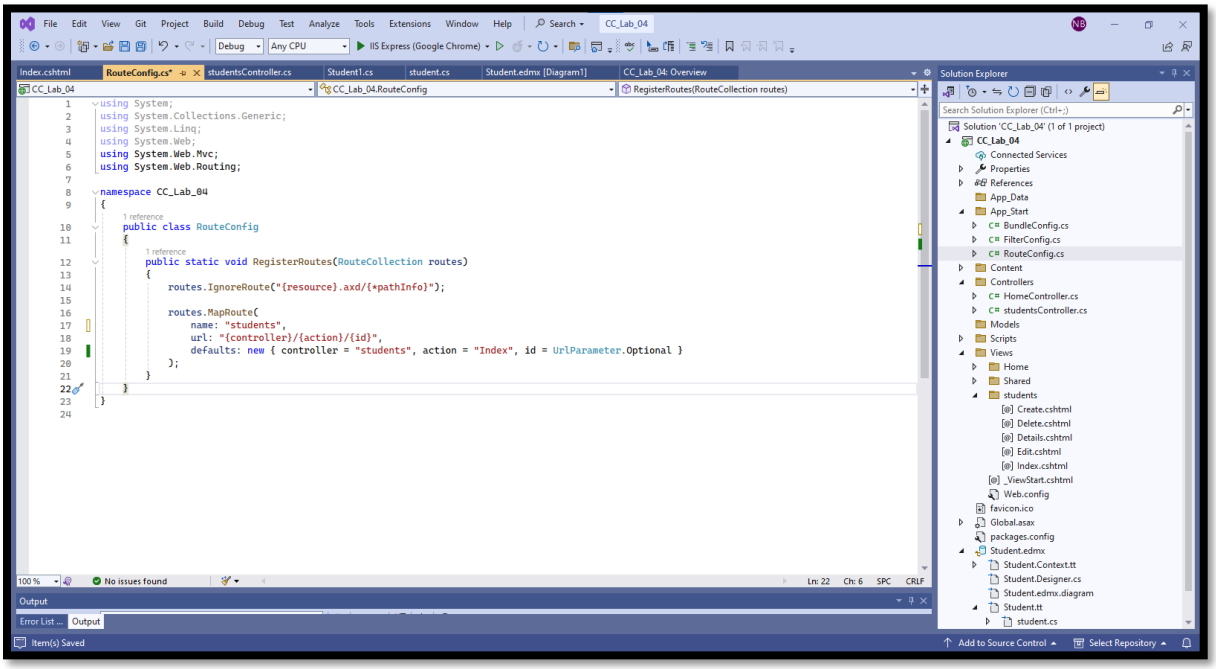


Step 6: Configure the route.

- Locate the RouteConfig.cs file in the App_Start folder.

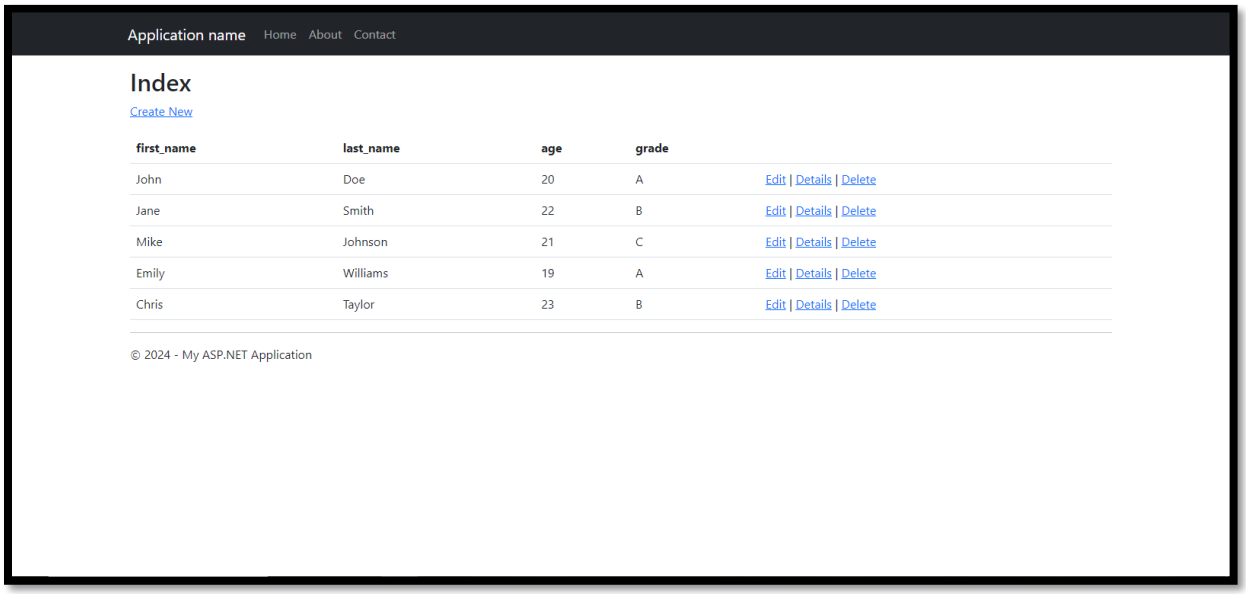


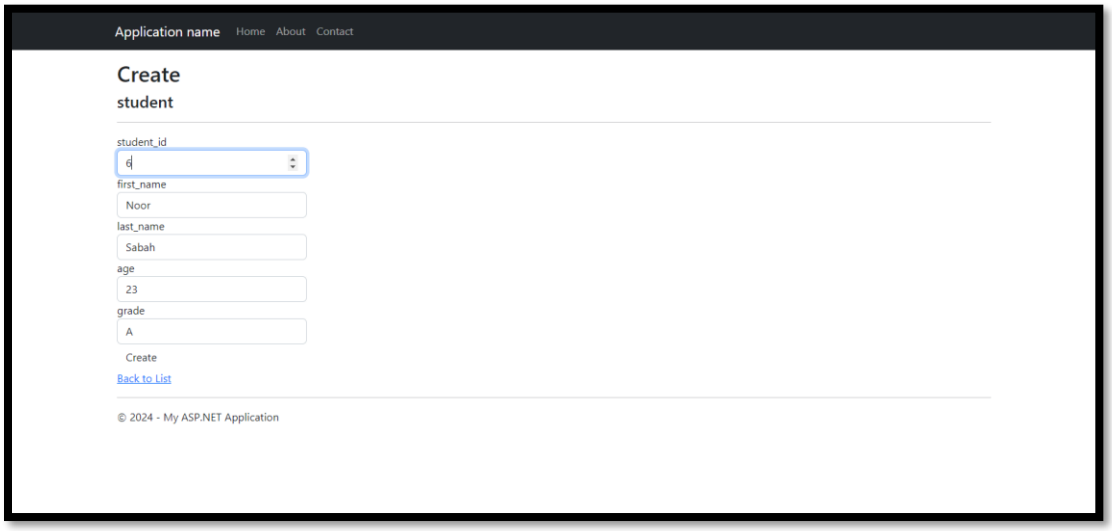
- Open the RouteConfig.cs file, and you'll find a class named RouteConfig. This class contains a static method called RegisterRoutes. Inside this method, you can configure your routes.

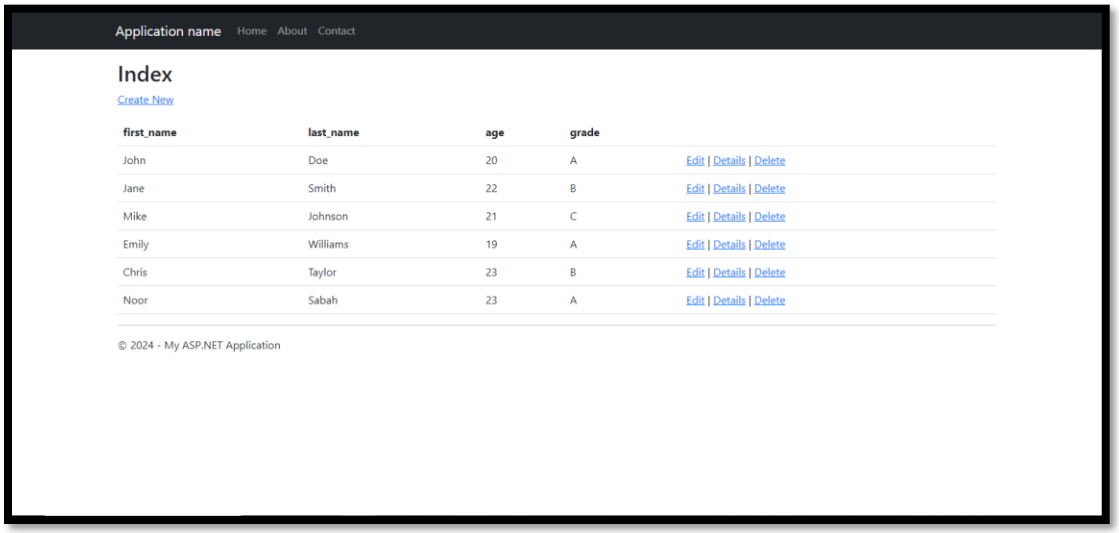


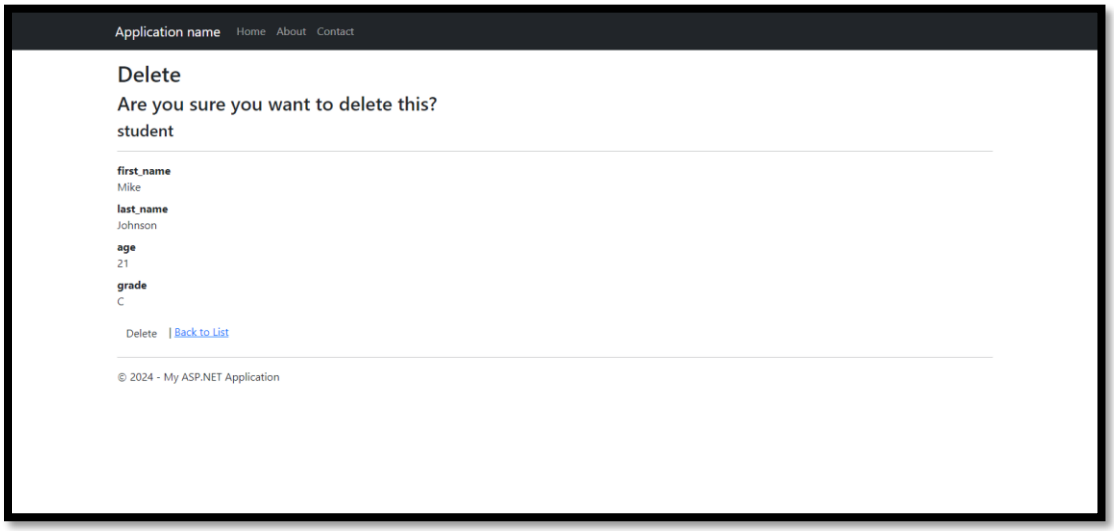
Step 6: Run the Project

- Build your project and then Press F5 or click on the "Run" button to launch your application.




Application name
Home
About
Contact
Create
student
student_id
6
first_name
Noor
last_name
Sabah
age
23
grade
A
Create
Back to List
© 2024 - My ASP.NET Application


Application name
Home
About
Contact
Index
Create New
first_name
last_name
age
grade
John
Doe
20
A
Edit | Details | Delete
Jane
Smith
22
B
Edit | Details | Delete
Mike
Johnson
21
C
Edit | Details | Delete
Emily
Williams
19
A
Edit | Details | Delete
Chris
Taylor
23
B
Edit | Details | Delete
Noor
Sabah
23
A
Edit | Details | Delete
© 2024 - My ASP.NET Application


Application name
Home
About
Contact
Delete
Are you sure you want to delete this?
student
first_name
Mike
last_name
Johnson
age
21
grade
C
Delete | Back to List
© 2024 - My ASP.NET Application

2. Time Boxing

Activity Name	Activity Time	Total Time
Login Systems + Setting up Visual studio Environment	3 mints + 5 mints	8 mints
Walk through Theory & Tasks	60 mints	60 mints
Implement Tasks	80 mints	80 mints
Evaluation Time	30 mints	30 mints
	Total Duration	178 mints

3. Objectives

After completing this lab the student should be able to:

a. *Master MVC architecture, Entity Framework, and ADO.NET for organized .NET applications.*

b. *Integrate Entity Framework and ADO.NET into MVC for efficient data querying and scalability.*

c. *Apply knowledge to develop a .NET app, utilizing MVC, Entity Framework, and ADO.NET for advanced database interactions.*

4. Lab Tasks/Practical Work

1. Develop an employee profile within the MVC .NET web application. Design a user interface that allows users to view, create, update, and delete employee records using ADO .Net Entity Framework.

2. Implement a product catalog in the MVC .NET web application utilizing ADO.NET Entity Framework. Design a user interface that enables users to browse, add, modify, and remove product records