

ASP.NET (.Net Framework) MVC – Empty Project

Step 1: Set Up Your Project.....	2
Step 2: Create the Model.....	2
Step 3: Create the Controller.....	2
Step 4: Create the Views.....	4
Step 5: Create the Student Table in SQL Server.....	6
Step 6: Update the Connection String	6
Step 7: Run Your Application.....	7

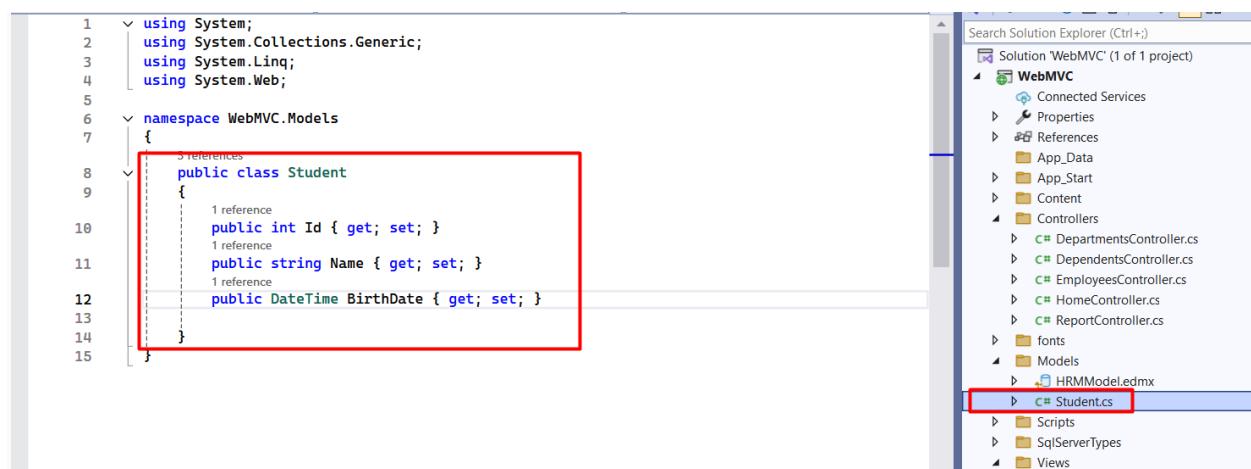
This project aims to create a web application using ASP.NET MVC (.NET Framework) that allows users to filter student data based on birth dates and display the results in a report. The application is connected to an SQL Server database to store and retrieve student information. By following this project, you will learn how to build a form, handle form submissions, query a database, and display data in a structured format using MVC views.

Step 1: Set Up Your Project

1. Create a new ASP.NET MVC project in Visual Studio.
 - Open Visual Studio.
 - Select **File > New > Project**.
 - Choose **ASP.NET Web Application (.NET Framework)**.
 - Select **MVC** template and click **Create**.

Step 2: Create the Model

1. Add a new class for the student model.
 - Right-click on the **Models** folder and select **Add > Class**.
 - Name it **Student.cs**.



```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5
6  namespace WebMVC.Models
7  {
8      public class Student
9      {
10         public int Id { get; set; }
11         public string Name { get; set; }
12         public DateTime BirthDate { get; set; }
13     }
14 }
15
```

```
public class Student
{
    public int Id { get; set; }
    public string Name { get; set; }
    public DateTime BirthDate { get; set; }
}
```

Step 3: Create the Controller

1. Add a new controller to handle the form submission and report generation.
 - Right-click on the **Controllers** folder and select **Add > Controller**.
 - Choose **MVC Controller - Empty** and name it **ReportController**.

```

4  using System.Data.SqlClient;
5  using System.Linq;
6  using System.Web;
7  using System.Web.Mvc;
8  using WebMVC.Models;
9
10 namespace WebMVC.Controllers
11 {
12     public class ReportController : Controller
13     {
14         private readonly string connectionString = ConfigurationManager.ConnectionStrings["MyRpt"].ConnectionString;
15         // GET: Report
16         public ActionResult Index()
17         {
18             return View();
19         }
20         [HttpPost]
21         public ActionResult GenerateReport(DateTime fromDate, DateTime toDate)
22         {
23             List<Student> students = new List<Student>();
24
25             using (SqlConnection conn = new SqlConnection(connectionString))
26             {
27                 string query = "SELECT * FROM Student WHERE BirthDate BETWEEN @FromDate AND @ToDate";
28                 SqlCommand cmd = new SqlCommand(query, conn);
29                 cmd.Parameters.AddWithValue("@FromDate", fromDate);
30                 cmd.Parameters.AddWithValue("@ToDate", toDate);
31
32                 conn.Open();
33                 SqlDataReader reader = cmd.ExecuteReader();
34                 while (reader.Read())
35                 {
36                     students.Add(new Student
37                     {
38                         Id = (int)reader["Id"],
39                         Name = (string)reader["Name"],
40                         BirthDate = (DateTime)reader["BirthDate"]
41                     });
42                 }
43             }
44             return View("ReportView", students);
45         }
46     }
47 }

```

No issues found | L: 33 Ch: 17 SPC CRLF

Search Solution Explorer (Ctrl+.)

WebMVC

- Connected Services
- Properties
- References
- App_Data
- App_Start
- Content
- Controllers
 - DepartmentsController.cs
 - DependentsController.cs
 - EmployeesController.cs
 - HomeController.cs
 - ReportController.cs**
- fonts
- Models
 - HRMModel.edmx
 - Student.cs
- Scripts
- SqlServerTypes
- Views
 - Departments
 - Dependents
 - Employees
 - Home
 - Report
 - Index.cshtml
 - report.cshml
 - ReportView.cshtml
 - StudentReport.rdlc
 - Shared
 - ViewStart.cshtml
- Web.config

MyRpt in Web.Config to <ConnectionStrings>

<ConnectionStrings>

<add name="MyRpt" connectionString="Data Source=(local)\SQLEXPRESS; Initial Catalog=LabDB; Integrated Security=True" providerName="System.Data.SqlClient"/>

```

</compilers>
<connectionStrings>
<add name="MyConn" connectionString="metadata=res://*/Models.HRMModel.csdl|res://*/Models.HRMModel.ssdl|res://*/Models.HRMModel.msl;provider=System.Data.SqlClient;providerName=System.Data.SqlClient" />
<add name="MyRpt" connectionString="Data Source=(local)\SQL2017;Initial Catalog=LabDB; Integrated Security=True" providerName="System.Data.SqlClient"/>
</connectionStrings>
<entityFramework>
<defaultConnectionFactory type="System.Data.Entity.Infrastructure.SqlConnectionFactory, EntityFramework" />
<providers>
<provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />
</providers>
</entityFramework>
<system.webServer>
<validation validateIntegratedModeConfiguration="false" />
<modules runAllManagedModulesForAllRequests="true" />
<handlers>
<add name="ReportViewerWebControlHandler" verb="*" path="Reserved.ReportViewerWebControl.axd" preCondition="integratedMode" type="Microsoft.Reporting.WebForms.HttpHandler" />
</handlers>
</system.webServer>
</configuration>

```

```

public class ReportController : Controller
{
    private readonly string connectionString =
    ConfigurationManager.ConnectionStrings["MyRpt"].ConnectionString;

    public ActionResult Index()
    {
        return View();
    }

    [HttpPost]
    public ActionResult GenerateReport(DateTime fromDate, DateTime toDate)
    {
        List<Student> students = new List<Student>();

        using (SqlConnection conn = new SqlConnection(connectionString))
        {

```

```

        string query = "SELECT * FROM Student WHERE BirthDate BETWEEN
@FromDate AND @ToDate";
        SqlCommand cmd = new SqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@FromDate", fromDate);
        cmd.Parameters.AddWithValue("@ToDate", toDate);

        conn.Open();
        SqlDataReader reader = cmd.ExecuteReader();
        while (reader.Read())
        {
            students.Add(new Student
            {
                Id = (int)reader["Id"],
                Name = (string)reader["Name"],
                BirthDate = (DateTime)reader["BirthDate"]
            });
        }
    }

    return View("ReportView", students);
}
}

```

Step 4: Create the Views

1. Create the Index view to display the form.

- Right-click on the **Views\Report** folder and select **Add > View**.
- Name it **Index.cshtml**.

```

@{
    ViewBag.Title = "Student Report";
}

<h2>Enter Date Range</h2>

<form method="post" action="/Report/GenerateReport">
    <div>
        <label for="fromDate">From Date:</label>
        <input type="date" id="fromDate" name="fromDate" required />
    </div>
    <div>
        <label for="toDate">To Date:</label>
        <input type="date" id="toDate" name="toDate" required />
    </div>
    <button type="submit">Submit</button>
</form>

```

2. Create the Report view to display the filtered student information.

- Right-click on the **Views\Report** folder and select **Add > View**.
- Name it **ReportView.cshtml**.

```

1 @model List<WebMVC.Models.Student>
2
3 @{
4     ViewBag.Title = "Student Report";
5 }
6
7 <h2>Student Report</h2>
8
9 <style>
10    table {
11        width: 100%;
12        border-collapse: collapse;
13    }
14
15    th, td {
16        padding: 10px;
17        text-align: left;
18        border-bottom: 1px solid #ddd;
19    }
20
21    th {
22        background-color: #f2f2f2;
23    }
24 </style>
25
26 <table>

```

No issues found

Ln: 16 Ch: 23 SPC CRLF

List

0 Errors | 5 Warnings | 0 Messages | Build + IntelliSense | Search Error List

Code Description Project File Line Suppression State

```

@model List<YourProjectName.Models.Student>

{
    ViewBag.Title = "Student Report";
}

<h2>Student Report</h2>

<style>
    table {
        width: 100%;
        border-collapse: collapse;
    }
    th, td {
        padding: 10px;
        text-align: left;
        border-bottom: 1px solid #ddd;
    }
    th {
        background-color: #f2f2f2;
    }
</style>

<table>
    <thead>
        <tr>
            <th>ID</th>
            <th>Name</th>
            <th>Birth Date</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var student in Model)

```

```

    {
        <tr>
            <td>@student.Id</td>
            <td>@student.Name</td>
            <td>@student.BirthDate.ToShortDateString()</td>
        </tr>
    }
</tbody>
</table>

```

Step 5: Create the Student Table in SQL Server

1. Create the **student table** in your SQL Server database.

```

CREATE TABLE Student (
    Id INT PRIMARY KEY IDENTITY(1,1),
    Name NVARCHAR(100),
    BirthDate DATE
);

```

2. Insert sample records into the Student table.

```

INSERT INTO Student (Name, BirthDate) VALUES
('John Doe', '2000-01-15'),
('Jane Smith', '1998-05-23'),
('Alice Johnson', '1999-03-12'),
('Bob Brown', '2001-07-19'),
('Charlie Davis', '1997-11-30'),
('Diana Evans', '2002-02-25'),
('Ethan Foster', '1996-09-10'),
('Fiona Green', '2000-12-05'),
('George Harris', '1998-04-18'),
('Hannah King', '1999-06-22');

```

Step 6: Update the Connection String

1. Update the connection string in `Web.config`.

```

<connectionStrings>
    <add name="DefaultConnection" connectionString="Data
Source=YOUR_SERVER_NAME; Initial Catalog=YOUR_DATABASE_NAME; Integrated
Security=True" providerName="System.Data.SqlClient" />
</connectionStrings>

```

Same with Step 3 add name="MyRpt" or MyConn

Step 7: Run Your Application

1. **Build and run** your application.
 - Press **F5** to start debugging.
2. **Navigate** to the URL where your application is hosted
(<https://localhost:xxxx/Report/Index>)



Enter Date Range

From Date:

To Date:

© 2025 - Developed by P.T.K.Trung

Click Submit then it will show Student Report

Student Report

ID	Name	Birth Date
1	John Doe	15/01/2000
2	Jane Smith	23/05/1998
3	Alice Johnson	12/03/1999
4	Bob Brown	19/07/2001
6	Diana Evans	25/02/2002
8	Fiona Green	05/12/2000
9	George Harris	18/04/1998
10	Hannah King	22/06/1999

© 2025 - Developed by P.T.K.Trung

Lessons Learned

1. **Understanding MVC Architecture:** This project reinforces the principles of Model-View-Controller architecture, helping you understand how to separate concerns and manage data flow in a web application.
2. **Database Integration:** You will gain practical experience in connecting an ASP.NET MVC application to a SQL Server database, including creating tables and executing queries.
3. **Form Handling:** Learn how to create and handle forms in MVC, including validating user input and processing form submissions.
4. **Data Presentation:** Discover techniques for presenting data in a structured and visually appealing manner using HTML and CSS within MVC views.
5. **Debugging and Troubleshooting:** Enhance your problem-solving skills by debugging and resolving common issues that arise during development

Enjoy 😊

Email: tg_phamthaikytrung@tdtu.edu.vn