

PROJECT INTRODUCTION

Student Management MVC Application

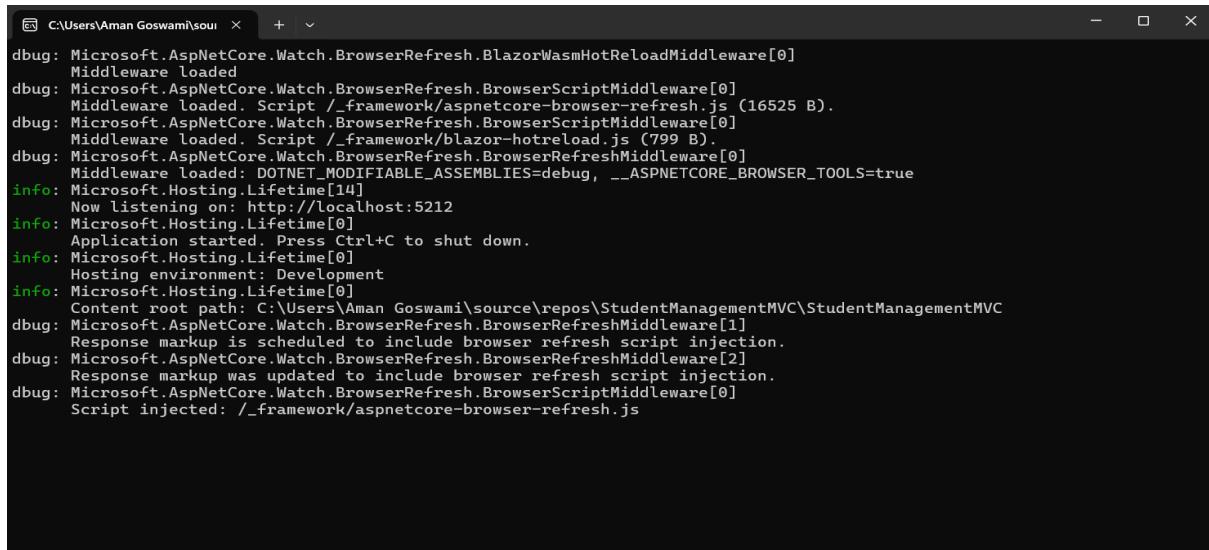
This project is an ASP.NET Core MVC web application used to manage student records.

It supports viewing, adding, editing, and deleting students, and viewing reports.

Technologies Used:

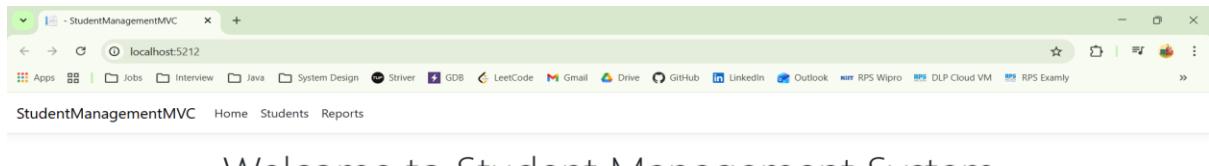
- ASP.NET Core MVC
- Entity Framework Core
- ADO.NET
- SQL Server

RUN PROJECT & LOCALHOST



```
C:\Users\Aman Goswami\source\repos\StudentManagementMVC\StudentManagementMVC\bin\Debug\net6.0\ StudentManagementMVC.csproj
```

```
dbug: Microsoft.AspNetCore.Watch.BrowserRefresh.BlazorWasmHotReloadMiddleware[0]
Middleware loaded
dbug: Microsoft.AspNetCore.Watch.BrowserRefresh.BrowserScriptMiddleware[0]
Middleware loaded. Script /_framework/aspnetcore-browser-refresh.js (16525 B).
dbug: Microsoft.AspNetCore.Watch.BrowserRefresh.BrowserScriptMiddleware[0]
Middleware loaded. Script /_framework/blazor-hotreload.js (799 B).
dbug: Microsoft.AspNetCore.Watch.BrowserRefresh.BrowserRefreshMiddleware[0]
Middleware loaded: DOTNET_MODIFIABLE_ASSEMBLIES=debug, __ASPNETCORE_BROWSER_TOOLS=true
info: Microsoft.Hosting.Lifetime[14]
Now listening on: http://localhost:5212
info: Microsoft.Hosting.Lifetime[0]
Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
Content root path: C:\Users\Aman Goswami\source\repos\StudentManagementMVC\StudentManagementMVC
dbug: Microsoft.AspNetCore.Watch.BrowserRefresh.BrowserRefreshMiddleware[1]
Response markup is scheduled to include browser refresh script injection.
dbug: Microsoft.AspNetCore.Watch.BrowserRefresh.BrowserRefreshMiddleware[2]
Response markup was updated to include browser refresh script injection.
dbug: Microsoft.AspNetCore.Watch.BrowserScriptMiddleware[0]
Script injected: /_framework/aspnetcore-browser-refresh.js
```



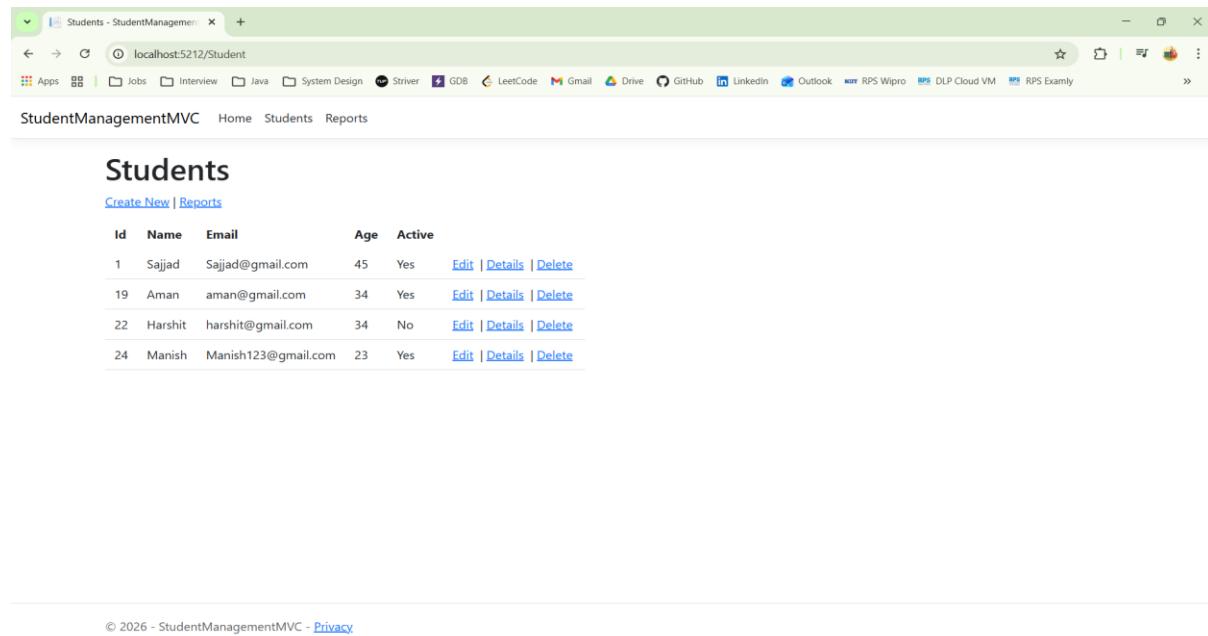
After clicking Run, the application starts from Program.cs.

The web server starts when app.Run() executes, and localhost opens automatically.

STUDENTS LIST PAGE

This page displays all students fetched from the database.

The request is handled by StudentController, and data is retrieved using StudentService and EF Core.



A screenshot of a web browser window titled "Students - StudentManagementMVC". The URL in the address bar is "localhost:5212/Student". The page content is titled "Students" and includes links for "Create New" and "Reports". Below is a table with columns: Id, Name, Email, Age, and Active. The data is as follows:

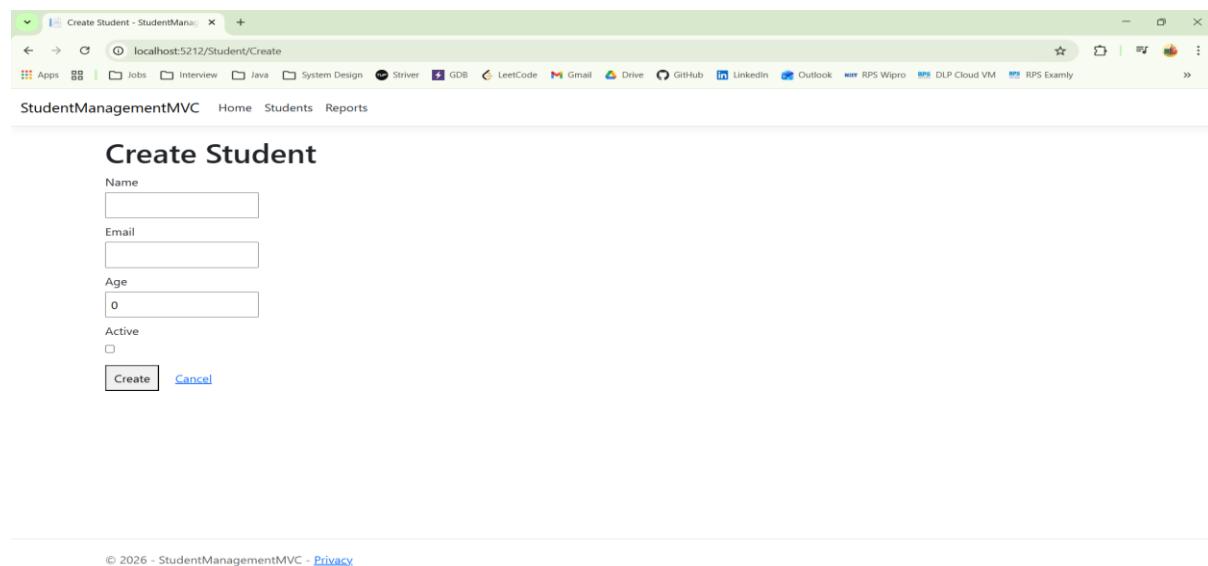
Id	Name	Email	Age	Active
1	Sajjad	Sajjad@gmail.com	45	Yes
19	Aman	aman@gmail.com	34	Yes
22	Harshit	harshit@gmail.com	34	No
24	Manish	Manish123@gmail.com	23	Yes

At the bottom, there is a copyright notice: "© 2026 - StudentManagementMVC - [Privacy](#)".

ADD STUDENT PAGE

This page allows adding a new student.

User input is validated, and data is saved to the database using Entity Framework Core.

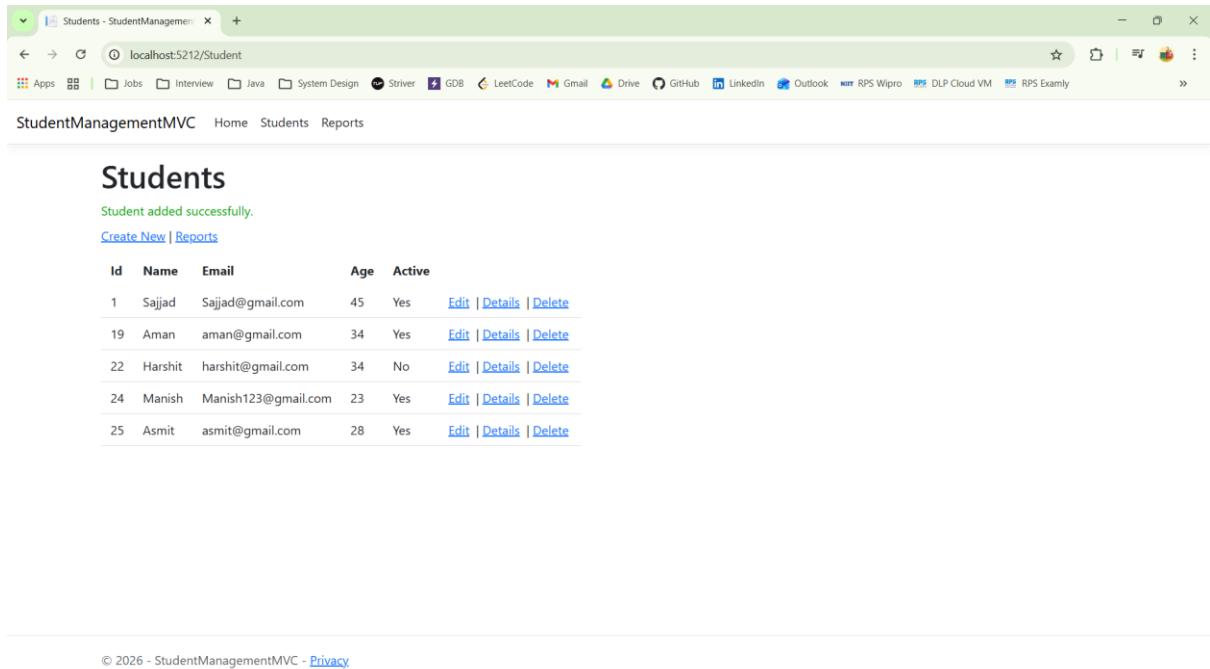


A screenshot of a web browser window titled "Create Student - StudentManagementMVC". The URL in the address bar is "localhost:5212/Student/Create". The page content is titled "Create Student" and includes fields for Name, Email, Age, and Active status, along with "Create" and "Cancel" buttons. At the bottom, there is a copyright notice: "© 2026 - StudentManagementMVC - [Privacy](#)".

ADD STUDENT SUCCESS

After submitting the form, the student is saved to the database.

A success message is displayed, and the updated list is shown.



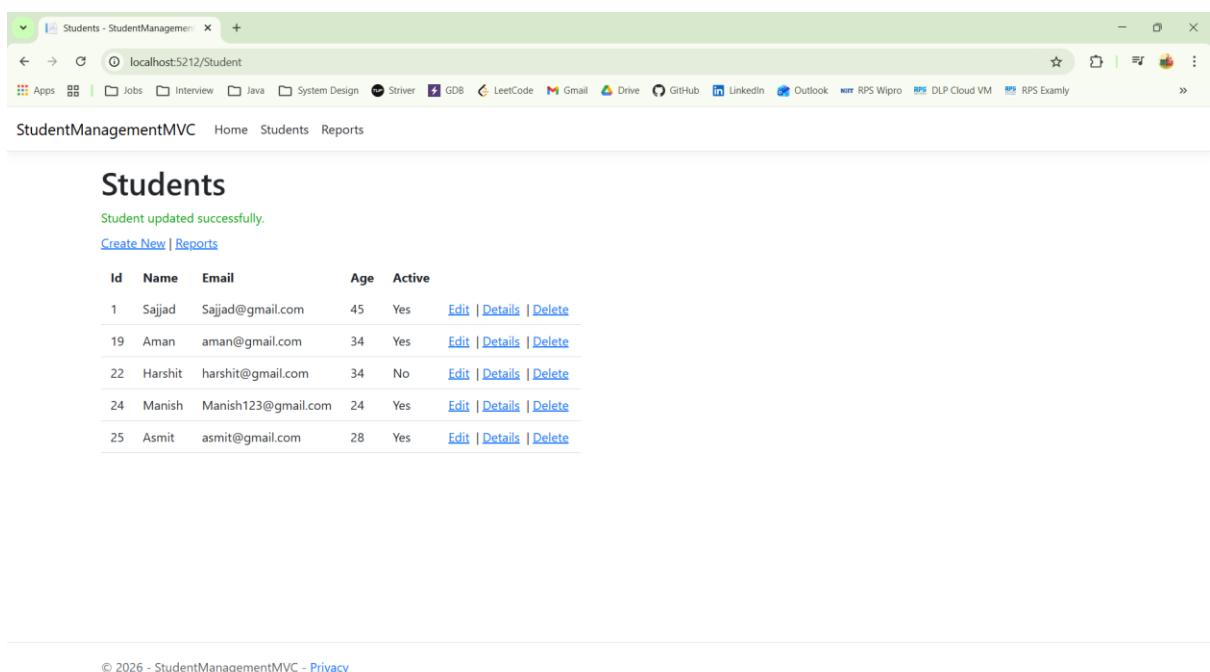
Id	Name	Email	Age	Active	
1	Sajjad	Sajjad@gmail.com	45	Yes	Edit Details Delete
19	Aman	aman@gmail.com	34	Yes	Edit Details Delete
22	Harshit	harshit@gmail.com	34	No	Edit Details Delete
24	Manish	Manish123@gmail.com	23	Yes	Edit Details Delete
25	Asmit	asmit@gmail.com	28	Yes	Edit Details Delete

© 2026 - StudentManagementMVC - [Privacy](#)

EDIT STUDENT

This page allows editing an existing student.

The data is loaded from the database and updated using Entity Framework Core.



Id	Name	Email	Age	Active	
1	Sajjad	Sajjad@gmail.com	45	Yes	Edit Details Delete
19	Aman	aman@gmail.com	34	Yes	Edit Details Delete
22	Harshit	harshit@gmail.com	34	No	Edit Details Delete
24	Manish	Manish123@gmail.com	24	Yes	Edit Details Delete
25	Asmit	asmit@gmail.com	28	Yes	Edit Details Delete

© 2026 - StudentManagementMVC - [Privacy](#)

DELETE STUDENT

This page confirms student deletion.

After confirmation, the record is removed from the database.

The screenshot shows a browser window titled "Students - StudentManagementMVC". The address bar says "localhost:5212/Student". The page header includes "StudentManagementMVC" and navigation links for "Home", "Students", and "Reports". Below the header, a success message "Student deleted successfully." is displayed in green. There is a link "Create New | Reports". A table lists four student records:

ID	Name	Email	Age	Active	Action
1	Sajjad	Sajjad@gmail.com	45	Yes	Edit Details Delete
19	Aman	aman@gmail.com	34	Yes	Edit Details Delete
24	Manish	Manish123@gmail.com	24	Yes	Edit Details Delete
25	Asmit	asmit@gmail.com	28	Yes	Edit Details Delete

At the bottom, there is a copyright notice: "© 2026 - StudentManagementMVC - [Privacy](#)".

REPORTS PAGE

This page displays student reports.

ADO.NET is used here to fetch total and active students directly from the database.

The screenshot shows a browser window titled "Reports - StudentManagementMVC". The address bar says "localhost:5212/Student/Reports". The page header includes "StudentManagementMVC" and navigation links for "Home", "Students", and "Reports". Below the header, it displays "Total Students: 4" and "Active Students: 4". There is a link "Back to List". At the bottom, there is a copyright notice: "© 2026 - StudentManagementMVC - [Privacy](#)".

DATABASE

This screenshot shows the Students table stored in SQL Server.

The database structure is created using the provided SQL script.

The screenshot displays the SQL Server Management Studio (SSMS) interface. On the left, the Object Explorer shows the database structure under 'AMAN-PC\SQLEXPRESS (SQL Server 17.0.1000 - AMAN)'. The 'StudentDB' database is selected, revealing its tables, views, and other objects. In the center, the 'SQLQuery1 S...swami (67)' window contains the following SQL script:

```
CREATE TABLE Students (
    StudentId INT IDENTITY(1,1) PRIMARY KEY,
    Name NVARCHAR(100) NOT NULL,
    Email NVARCHAR(100) NOT NULL,
    Age INT NOT NULL,
    IsActive BIT NOT NULL
);
SELECT * FROM Students;
```

Below the script, the results pane shows a table with four rows of data:

StudentId	Name	Email	Age	IsActive
1	Sajid	Sajid@gmail.com	45	1
2	Aman	aman@gmail.com	34	1
3	Manish	Manish123@gmail.com	24	1
4	Amit	amit@gmail.com	28	1

The status bar at the bottom indicates 'Query executed successfully.' and provides session details: AMAN-PC\SQLEXPRESS (17.0 RTM) | AMAN-PC\Aman Goswami (67) | StudentDB | 00:00:00 | Row: 1, Col: 1 | 4 rows.