

Web Setup and First App View:

Terminal code :

ls : shows files and folders in directory

mkdir name: creates folder

cd name : change directory

code . : open in vs code

.NET CLI code:

dotnet new console

dotnet new console -o YourProjectName

```
PS C:\Users\mrn-cltp> cd .\practice-class\  
PS C:\Users\mrn-cltp\practice-class> code .  
PS C:\Users\mrn-cltp\practice-class> dotnet new list
```

Code
Share
Reference

Only
Backend

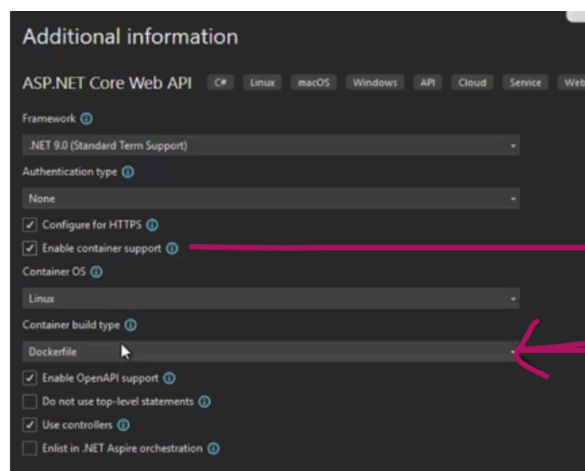
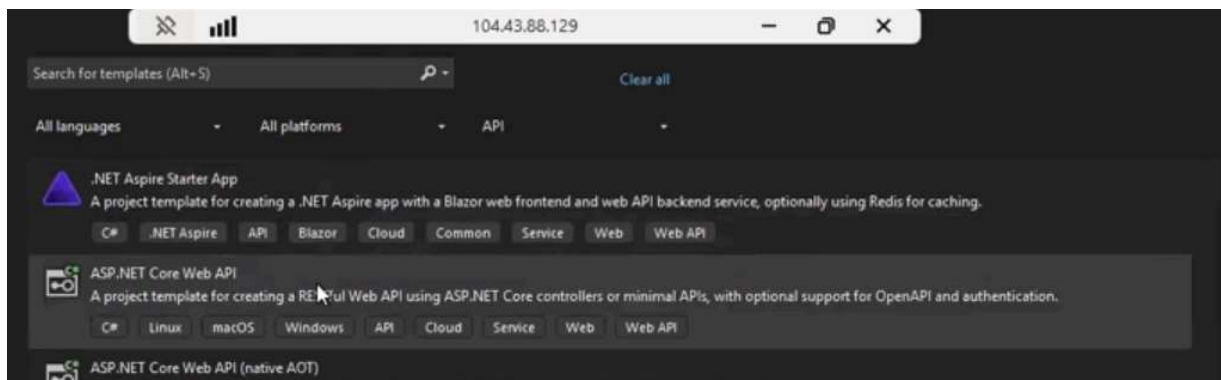
view
2/6

Dif &
version

These templates matched your input:

Template Name	Short Name	Language	Tags
API Controller	apicontroller	[C#]	Web/ASP.NET
ASP.NET Core Empty	web	[C#],F#	Web/Empty
ASP.NET Core gRPC Service	grpc	[C#]	Web/gRPC/API/Service
ASP.NET Core Web API	webapi	[C#],F#	Web/Web API/API/Service
ASP.NET Core Web API (native AOT)	webapiaot	[C#]	Web/Web API/API/Service
ASP.NET Core Web App (Model-View-Controller)	mvc	[C#],F#	Web/MVC
ASP.NET Core Web App (Razor Pages)	webapp,razor	[C#]	Web/MVC/Razor Pages
Blazor Web App	blazor	[C#]	Web/Blazor/WebAssembly
Blazor WebAssembly Standalone App	blazorwasm	[C#]	Web/Blazor/WebAssembly/PWA
Class	class	[C#],VB	Common
Class Library	classlib	[C#],F#,VB	Common/Library
Console App	console	[C#],F#,VB	Common/Console
dotnet gitignore file	gitignore,.gitignore		Config
Dotnet local tool manifest file	tool-manifest		Config
EditorConfig file	editorconfig,.editorconfig		Config
Enum	enum	[C#],VB	Common
global.json file	globaljson,global.json		Config
Interface	interface	[C#],VB	Common
MSBuild Directory.Build.props file	buildprops		MSBuild/props
MSBuild Directory.Build.targets file	buildtargets		MSBuild/props
MSBuild Directory.Packages.props file	packagesprops		MSBuild/packages/props/CPM
MSTest Playwright Test Project	mstest-playwright	[C#]	Test/MSTest/Playwright/Desktop/Web
MSTest Test Class	mstest-class	[C#],F#,VB	Test/MSTest
MSTest Test Project	mstest	[C#],F#,VB	Test/MSTest/Desktop/Web
MVC Controller	mvccontroller	[C#]	Web/ASP.NET
MVC ViewImports	viewimports	[C#]	Web/ASP.NET
MVC ViewStart	viewstart	[C#]	Web/ASP.NET
NuGet Config	nugetconfig,nuget.config		Config
NUnit 3 Test Item	nunit-test	[C#],F#,VB	Test/NUnit
NUnit 3 Test Project	nunit	[C#],F#,VB	Test/NUnit/Desktop/Web
NUnit Playwright Test Project	nunit-playwright	[C#]	Test/NUnit/Playwright/Desktop/Web
Protocol Buffer File	proto		Web/gRPC
Razor Class Library	razorclasslib	[C#]	Web/Razor/Library
Razor Component	razorcomponent	[C#]	Web/ASP.NET
Razor Page	page	[C#]	Web/ASP.NET
Razor View	view	[C#]	Web/ASP.NET
Record	record	[C#]	Common
Solution File	sln,solution		Solution
Struct	struct,structure	[C#],VB	Common

Choose ASP .Net core Web api: (only Backend)

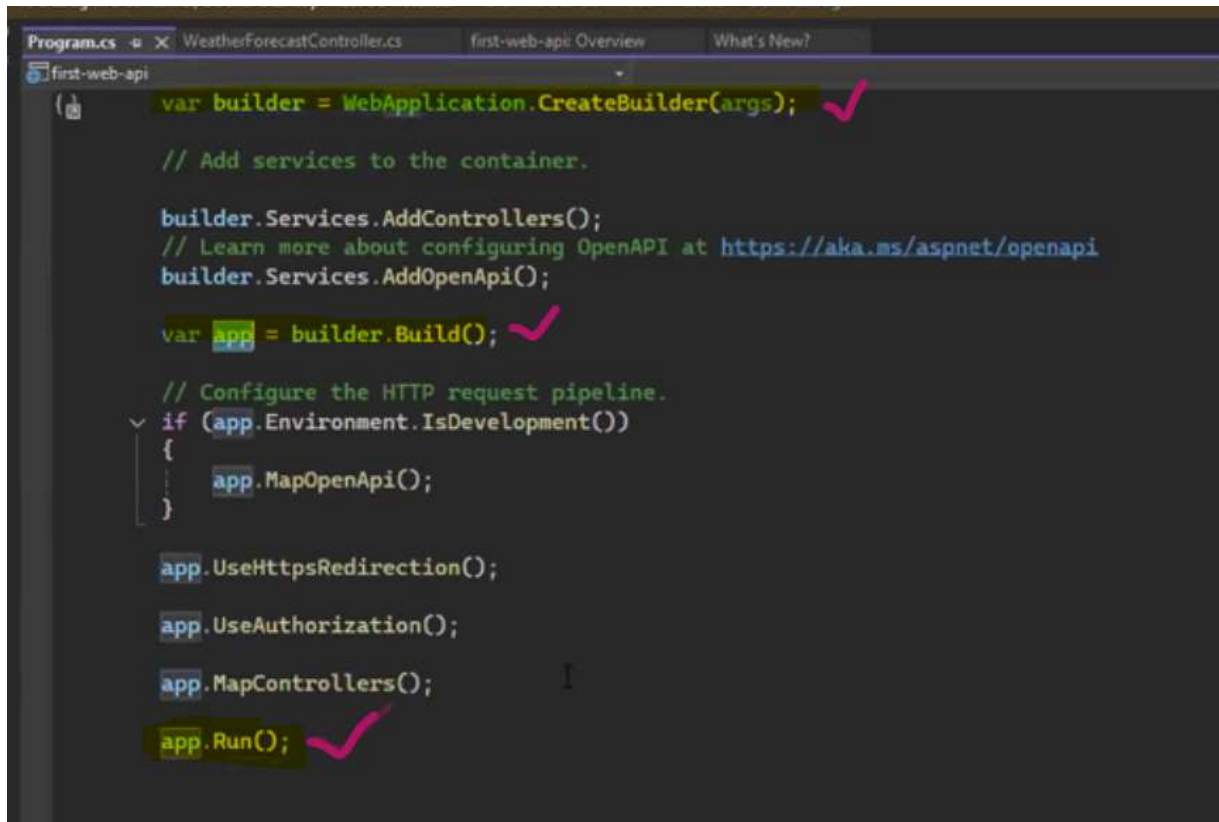


Enable this
Adds docker
container

Program.cs -> startup file:

Builder. Build () -> Builder build hole app create hobe.

app.run() -> app run hole program ta start hobe.



```
Program.cs x WeatherForecastController.cs first-web-api Overview What's New?
first-web-api
(
    var builder = WebApplication.CreateBuilder(args); ✓

    // Add services to the container.

    builder.Services.AddControllers();
    // Learn more about configuring OpenAPI at https://aka.ms/aspnet/openapi
    builder.Services.AddOpenApi();

    var app = builder.Build(); ✓

    // Configure the HTTP request pipeline.
    if (app.Environment.IsDevelopment())
    {
        app.MapOpenApi();
    }

    app.UseHttpsRedirection();

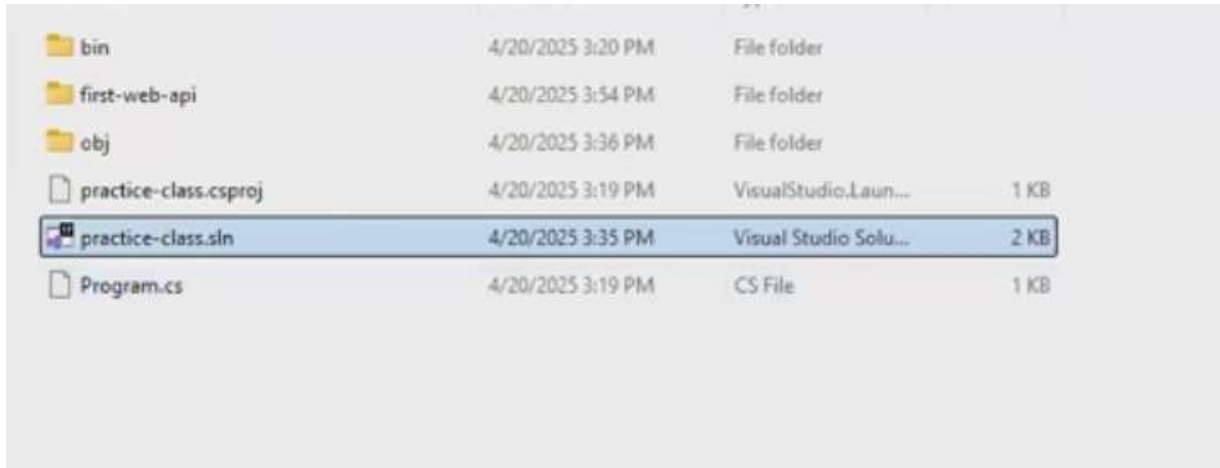
    app.UseAuthorization();

    app.MapControllers();

    app.Run(); ✓
```

Project File description:

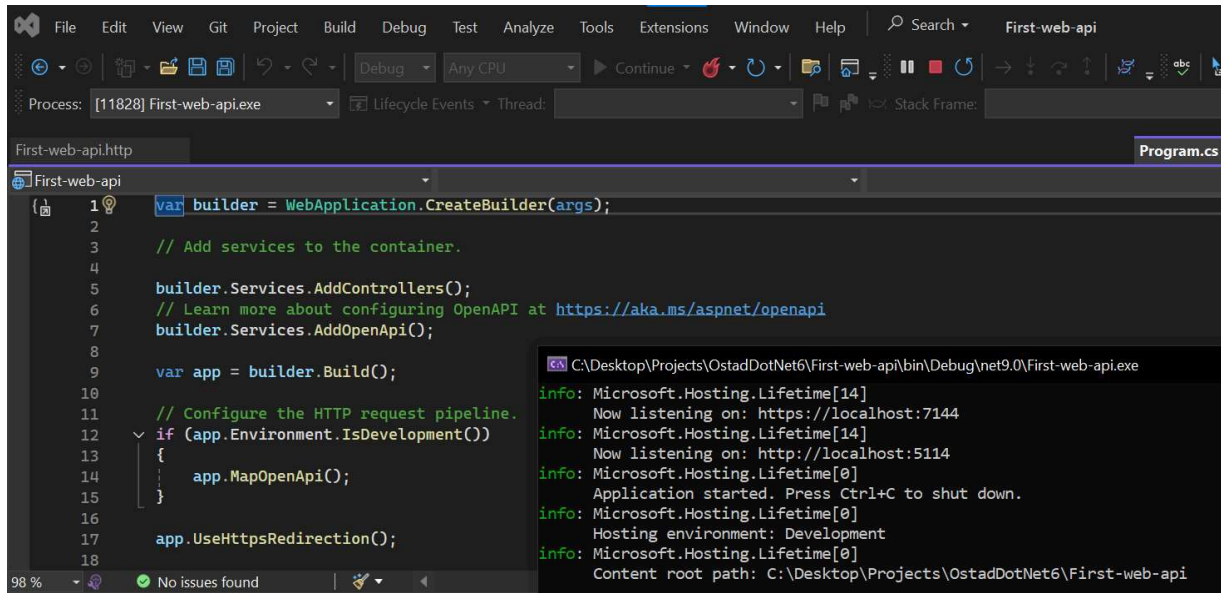
Practice-class.sln : Project gula solution file er under e thake. Ekta solution file er under multiple project file thake.

A screenshot of a file explorer window showing a list of files and folders. The list includes 'bin', 'first-web-api', 'obj', 'practice-class.csproj', 'practice-class.sln', and 'Program.cs'. The 'practice-class.sln' file is highlighted with a blue selection bar. The table has columns for the file name, its last modified date and time, its type, and its size.

bin	4/20/2025 3:20 PM	File folder	
first-web-api	4/20/2025 3:54 PM	File folder	
obj	4/20/2025 3:36 PM	File folder	
practice-class.csproj	4/20/2025 3:19 PM	VisualStudio.Laun...	1 KB
practice-class.sln	4/20/2025 3:35 PM	Visual Studio Solu...	2 KB
Program.cs	4/20/2025 3:19 PM	CS File	1 KB

Run API (Https):

port listen: Listens from two place



The screenshot shows the Visual Studio IDE with the 'First-web-api' project open. The code in 'Program.cs' is as follows:

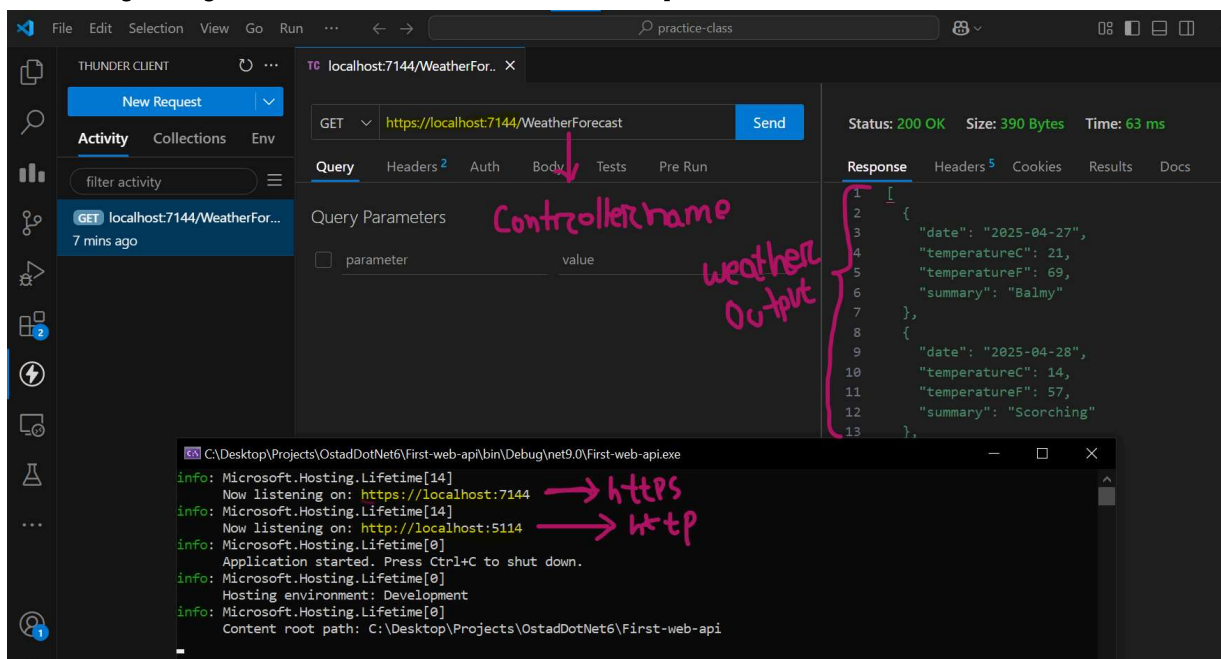
```
1 var builder = WebApplication.CreateBuilder(args);
2
3 // Add services to the container.
4
5 builder.Services.AddControllers();
6 // Learn more about configuring OpenAPI at https://aka.ms/aspnet/openapi
7 builder.Services.AddOpenApi();
8
9 var app = builder.Build();
10
11 // Configure the HTTP request pipeline.
12 if (app.Environment.IsDevelopment())
13 {
14     app.MapOpenApi();
15 }
16
17 app.UseHttpsRedirection();
18
```

The output console shows the following logs:

```
info: Microsoft.Hosting.Lifetime[14]
Now listening on: https://localhost:7144
info: Microsoft.Hosting.Lifetime[14]
Now listening on: http://localhost:5114
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:\Desktop\Projects\OstadDotNet6\First-web-api
```

We can open the link:

- postman/rest-api-client diye api er through controller e pathano jay. [thunder client : Lightweight rest-API-Client vscode extension]



The screenshot shows the Thunder Client extension in VS Code. A GET request is made to `https://localhost:7144/WeatherForecast`. The response is a JSON array of weather forecasts. Handwritten pink notes indicate that the path `/WeatherForecast` corresponds to the `WeatherForecastController` and that the port `7144` is for `https`.

Query Parameters:

parameter	value

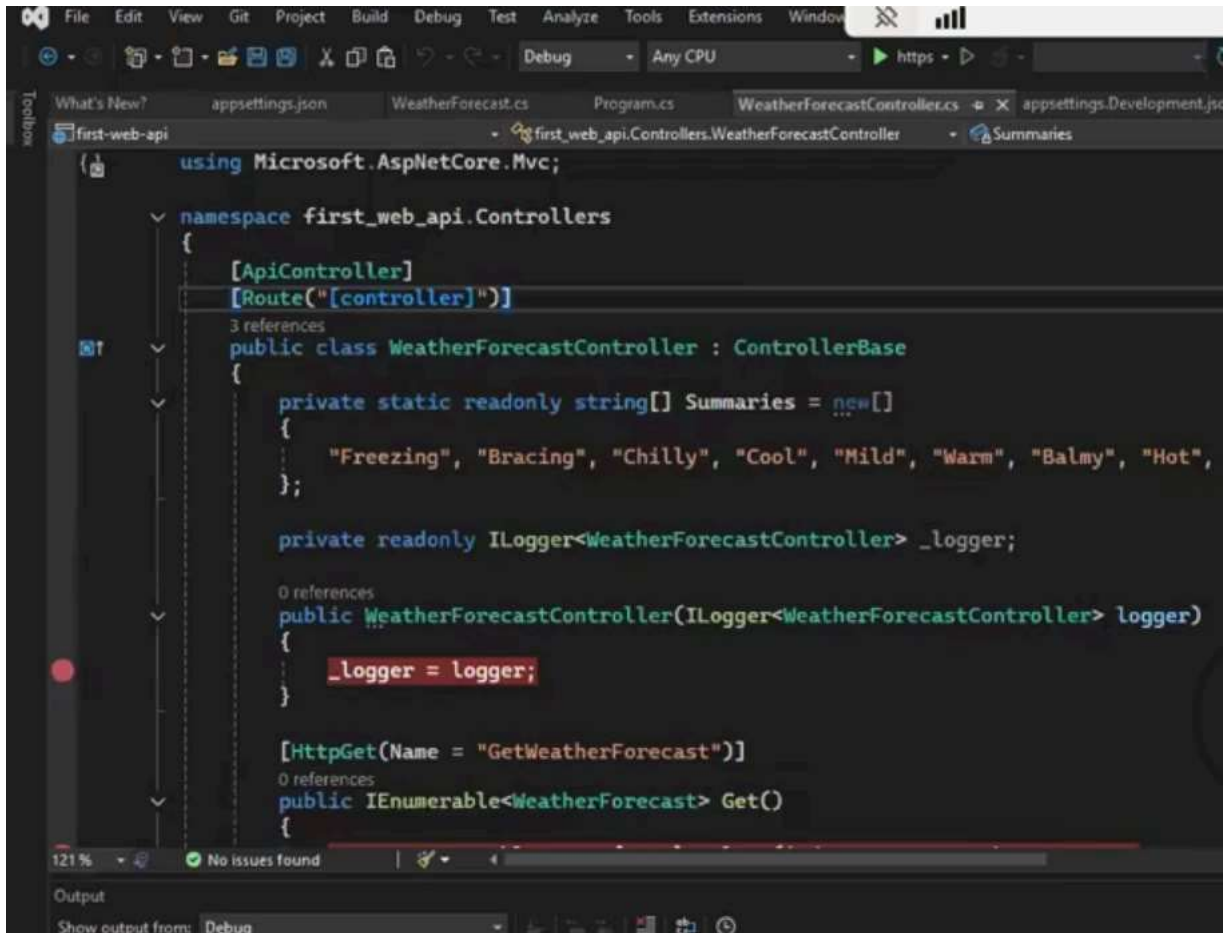
Response:

```
1 {
2   "date": "2025-04-27",
3   "temperatureC": 21,
4   "temperatureF": 69,
5   "summary": "Balmy"
6 },
7 {
8   "date": "2025-04-28",
9   "temperatureC": 14,
10  "temperatureF": 57,
11  "summary": "Scorching"
12 },
13 }
```

Output Console:

```
info: Microsoft.Hosting.Lifetime[14]
Now listening on: https://localhost:7144
info: Microsoft.Hosting.Lifetime[14]
Now listening on: http://localhost:5114
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:\Desktop\Projects\OstadDotNet6\First-web-api
```

Suggestion: Controller e debugging mode on kre api run kre dekha ki hocche.



The screenshot shows the Visual Studio IDE with the `WeatherForecastController.cs` file open. The file is in debug mode, as indicated by the `Debug` button in the top toolbar. The code is as follows:

```
using Microsoft.AspNetCore.Mvc;

namespace first_web_api.Controllers
{
    [ApiController]
    [Route("[controller]")]
    public class WeatherForecastController : ControllerBase
    {
        private static readonly string[] Summaries = new[]
        {
            "Freezing", "Bracing", "Chilly", "Cool", "Mild", "Warm", "Balmy", "Hot",
        };

        private readonly ILogger<WeatherForecastController> _logger;

        public WeatherForecastController(ILogger<WeatherForecastController> logger)
        {
            _logger = logger;
        }

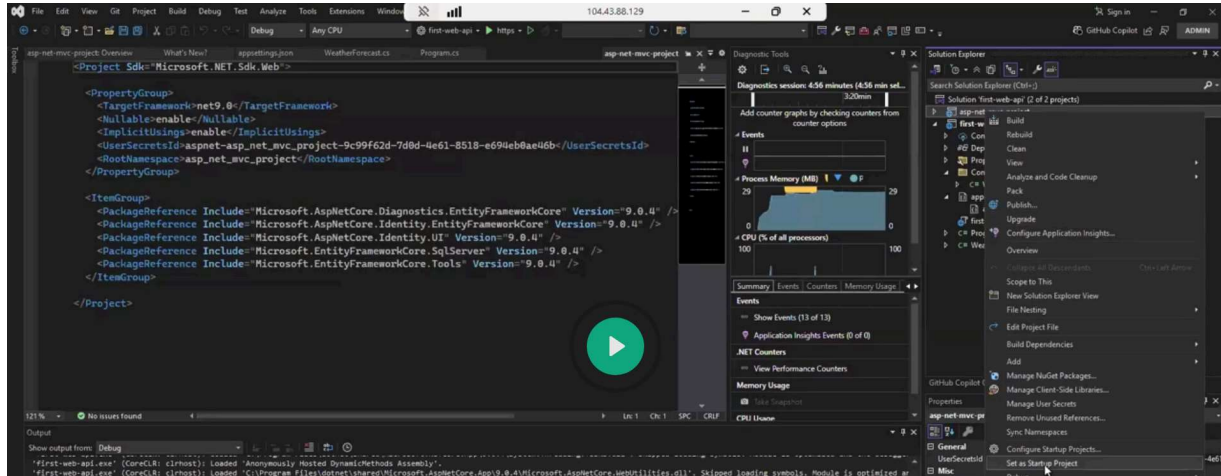
        [HttpGet(Name = "GetWeatherForecast")]
        public IEnumerable<WeatherForecast> Get()
        {
        }
    }
}
```

The `Debugger` window is visible on the left, showing the `first_web_api` project. The `Output` window at the bottom shows `Show output from: Debug`.

Some Basics:

VS studio te multiple project theke kono ekta project run krte chaile :

Solution expolrer -> right click project name -> set as setup project



Bin folder:

Program build krle eta binary code generate kre. Binary code ta Bin folder e save hoy. Build e click kre clean solution dile binary code gulo delete hoye jabe.

Rebuild -> clean + build

Multiple way te build kra jay. Debug mode/ release mode.

Bin folder -> debug/release folder -> binary code

