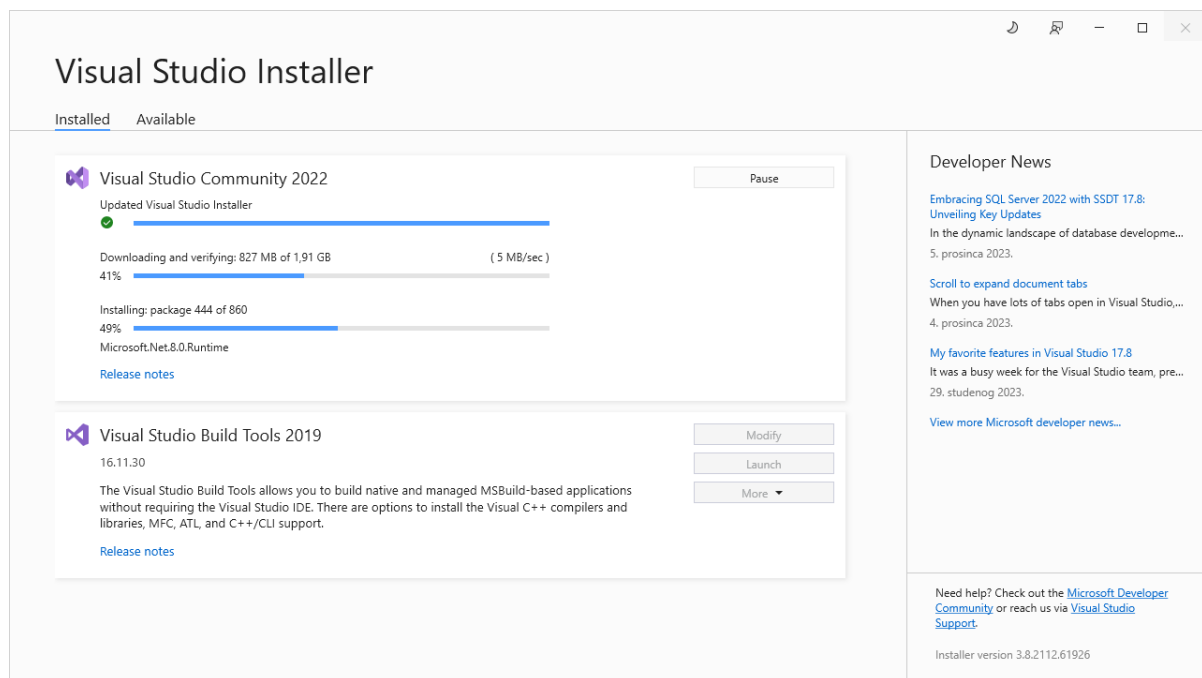


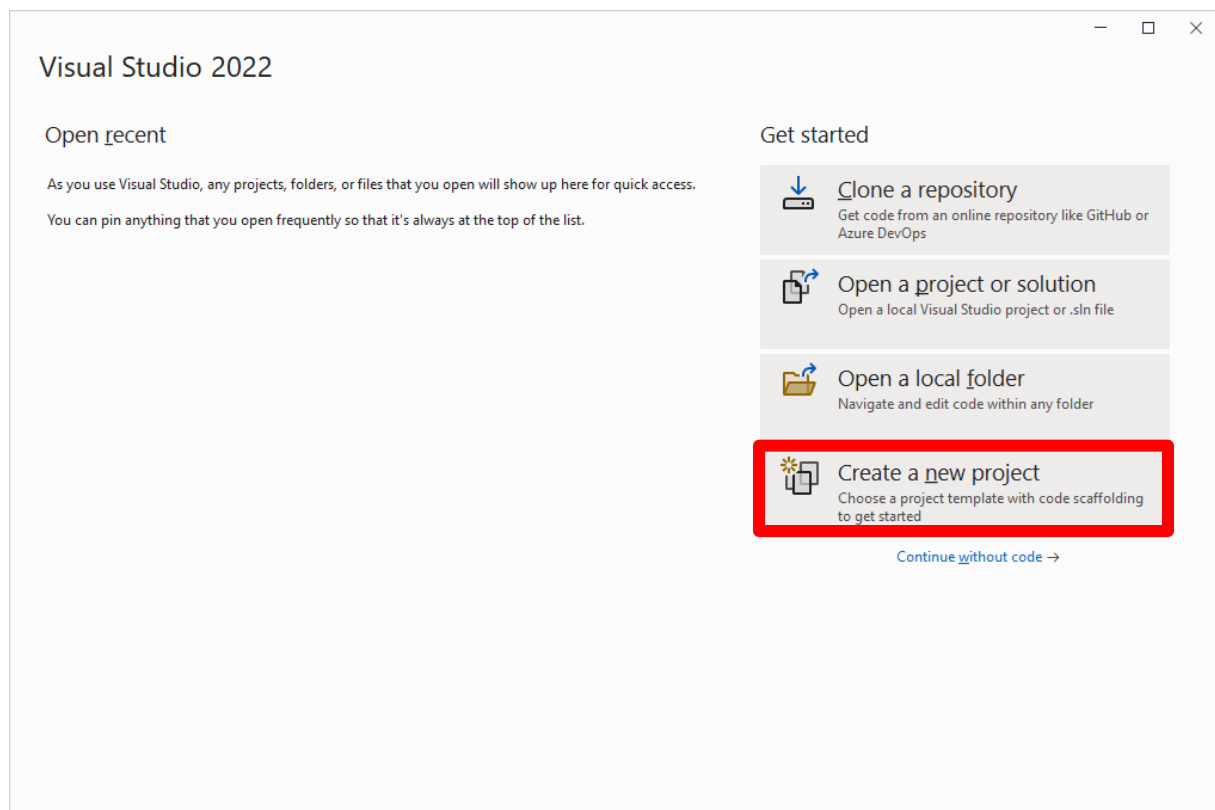
Po potrebi prvo napravite UPDATE MS Visual studio. Ja sam zadnji puta napravio 07. prosinca 2023. n verziju 17.8.3.



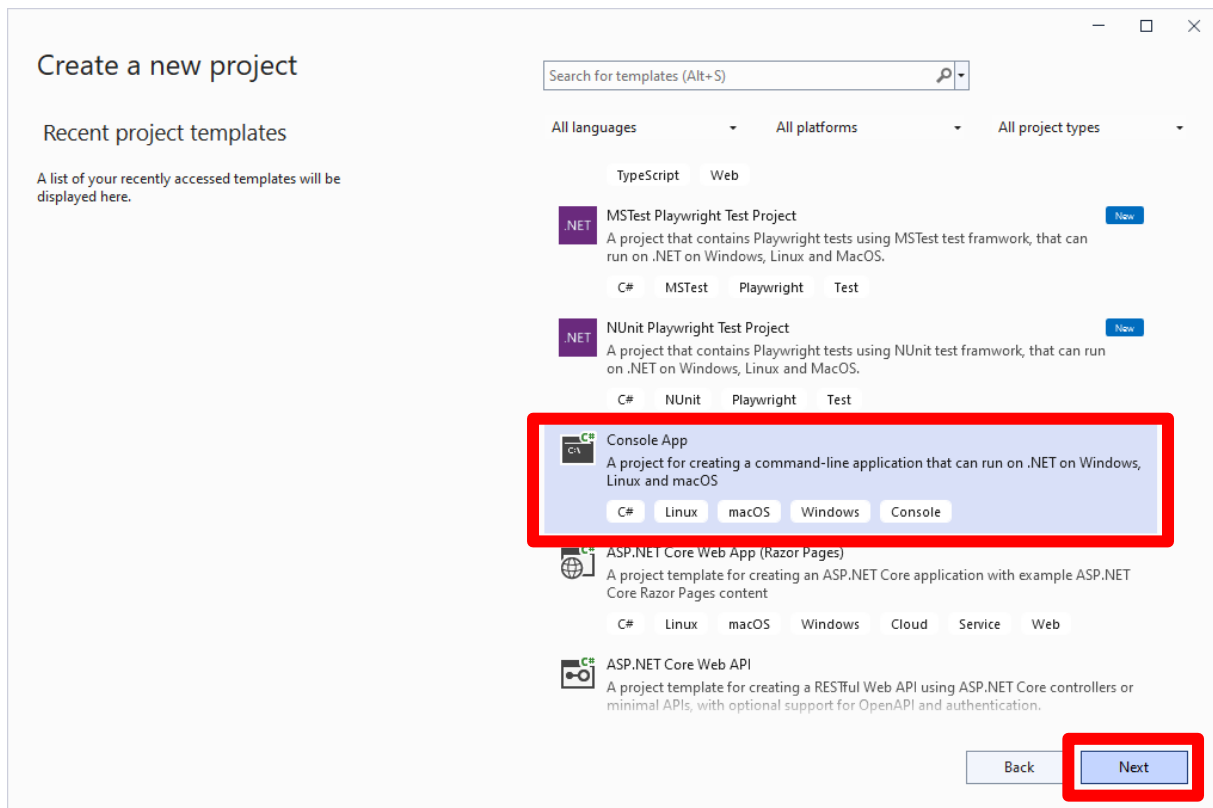
Učenje

Raditi ćemo paralelno na dva projekta(Project): **UčenjeCS** i **WebAPI** u sklopu rješenja (Solution) .

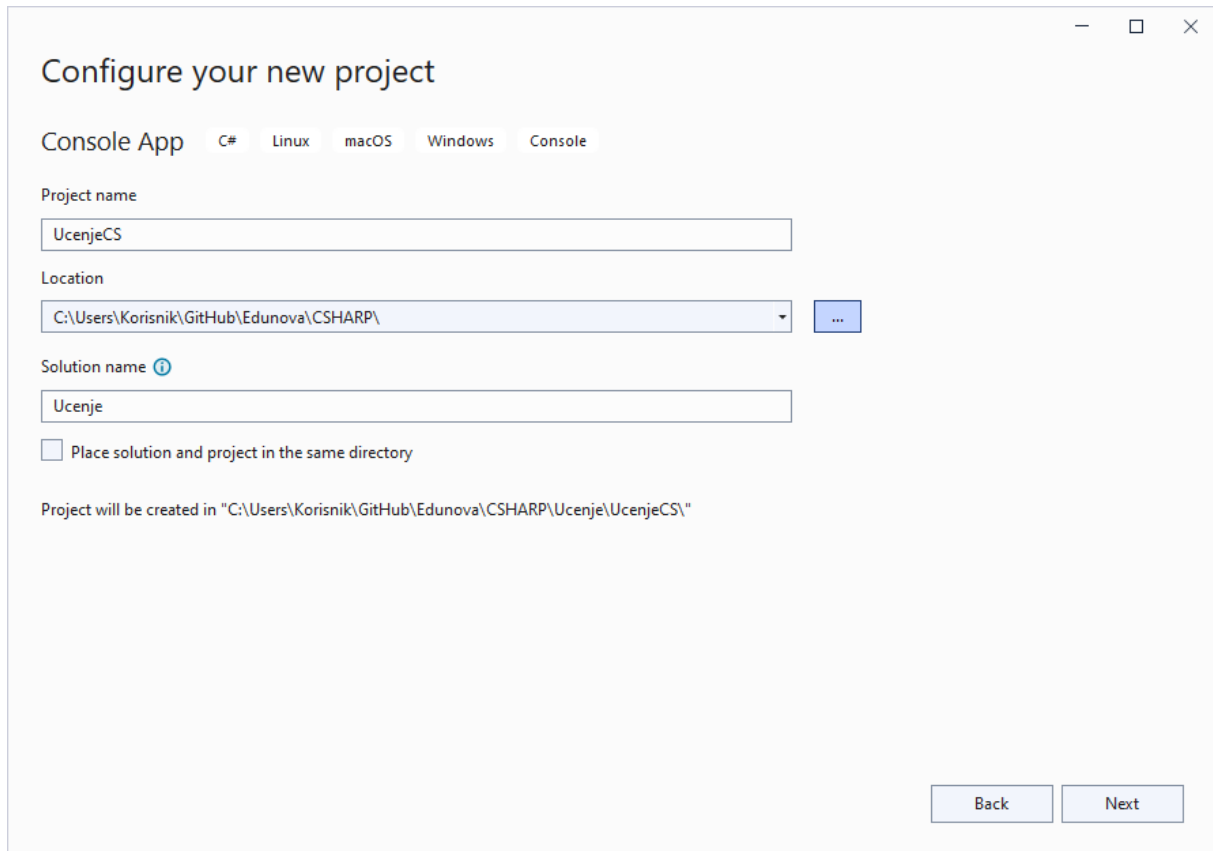
Prvo odaberemo Create a new project.



Zatim odaberemo Console App i odaberemo Next



Projekt nazovemo **UcenjeCS**, postavimo ga na lokaciju **CSHARP** (direktorij koji smo kreirali u našem git repozitoriju – pokraj direktorija SQL) te postavimo ime rješenja **Ucenje** (Checkbox Place solution and project into sam directory **JE OTKAČEN**) te odaberemo Next.



Configure your new project

Console App C# Linux macOS Windows Console

Project name

UcenjeCS

Location

C:\Users\Korisnik\GitHub\Edunova\CSHARP\

Solution name ⓘ

Ucenje

☐ Place solution and project in the same directory

Project will be created in "C:\Users\Korisnik\GitHub\Edunova\CSHARP\Ucenje\UcenjeCS\"

Back Next

Odaberemo radnu okolinu (Framework) 7.0 – ne moramo ići na zadnji dostupni jer nam možda produkcija neće imati podršku za njega.

– □ ×

Additional information

Console App C# Linux macOS Windows Console

Framework ⓘ

.NET 8.0 (Long Term Support)	▼
.NET 6.0 (Long Term Support)	
.NET 7.0 (Standard Term Support)	
.NET 8.0 (Long Term Support)	

Back Create

Isto tako **NE OZNAČIMO** Checkbox Do not use top-level statements te odaberemo Create.

□

×

Additional information

Console App C# Linux macOS Windows Console

Framework ⓘ

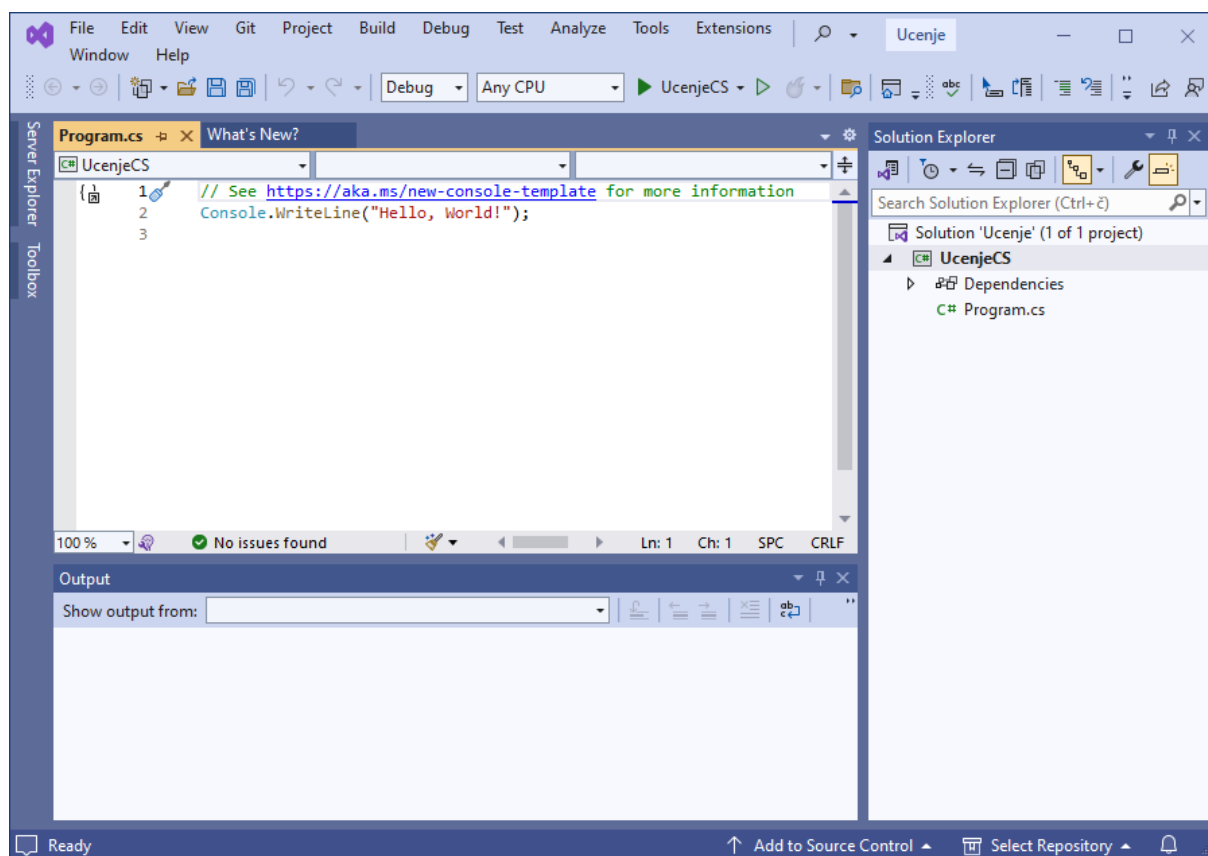
.NET 7.0 (Standard Term Support) ▾

☐ Do not use top-level statements ⓘ

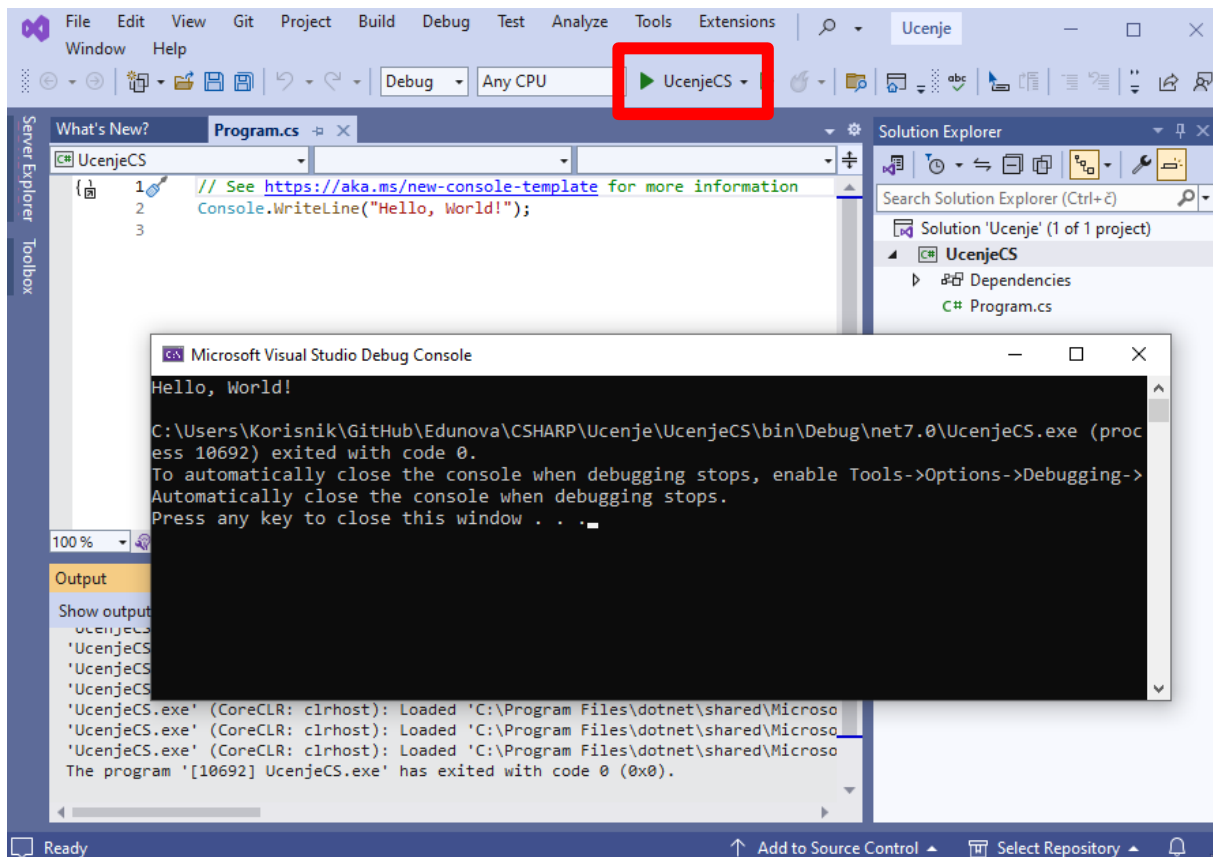
Back

Create

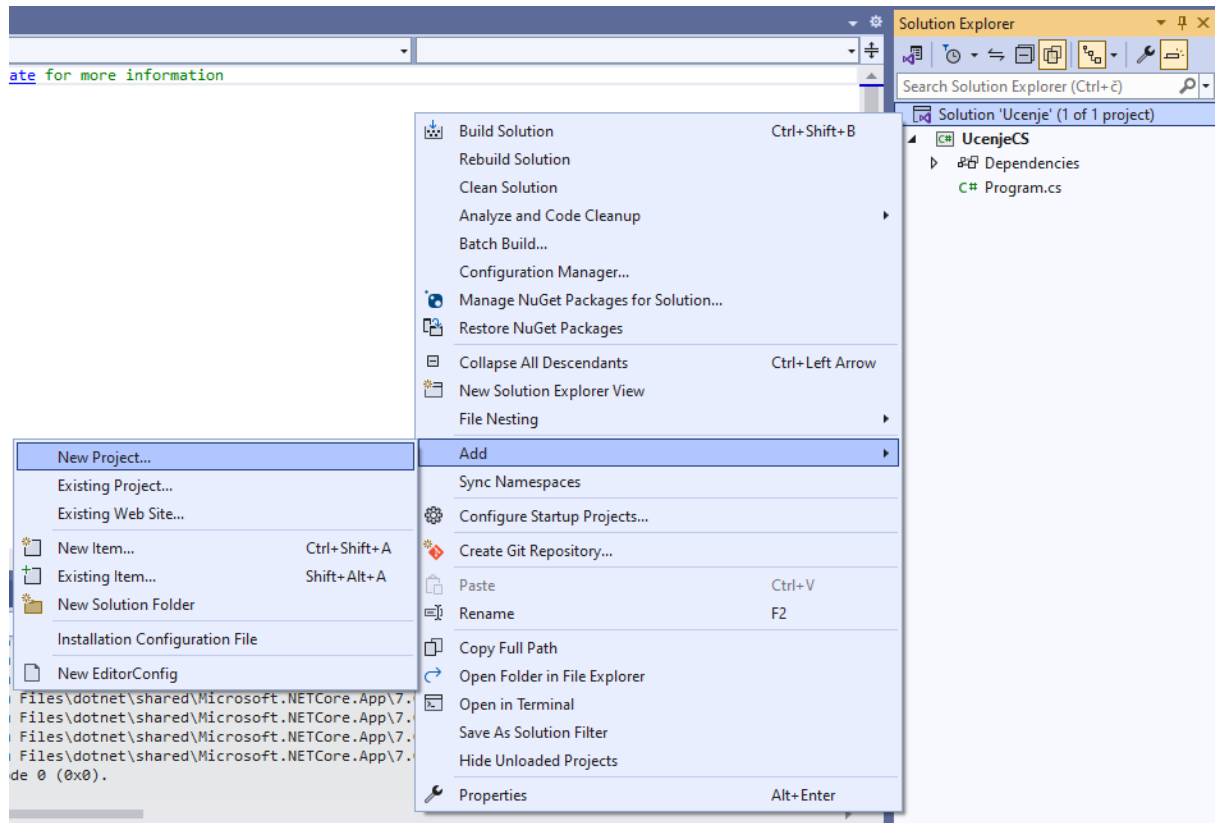
Ovo je naš Konzolni (za učenje) Hello, world program.



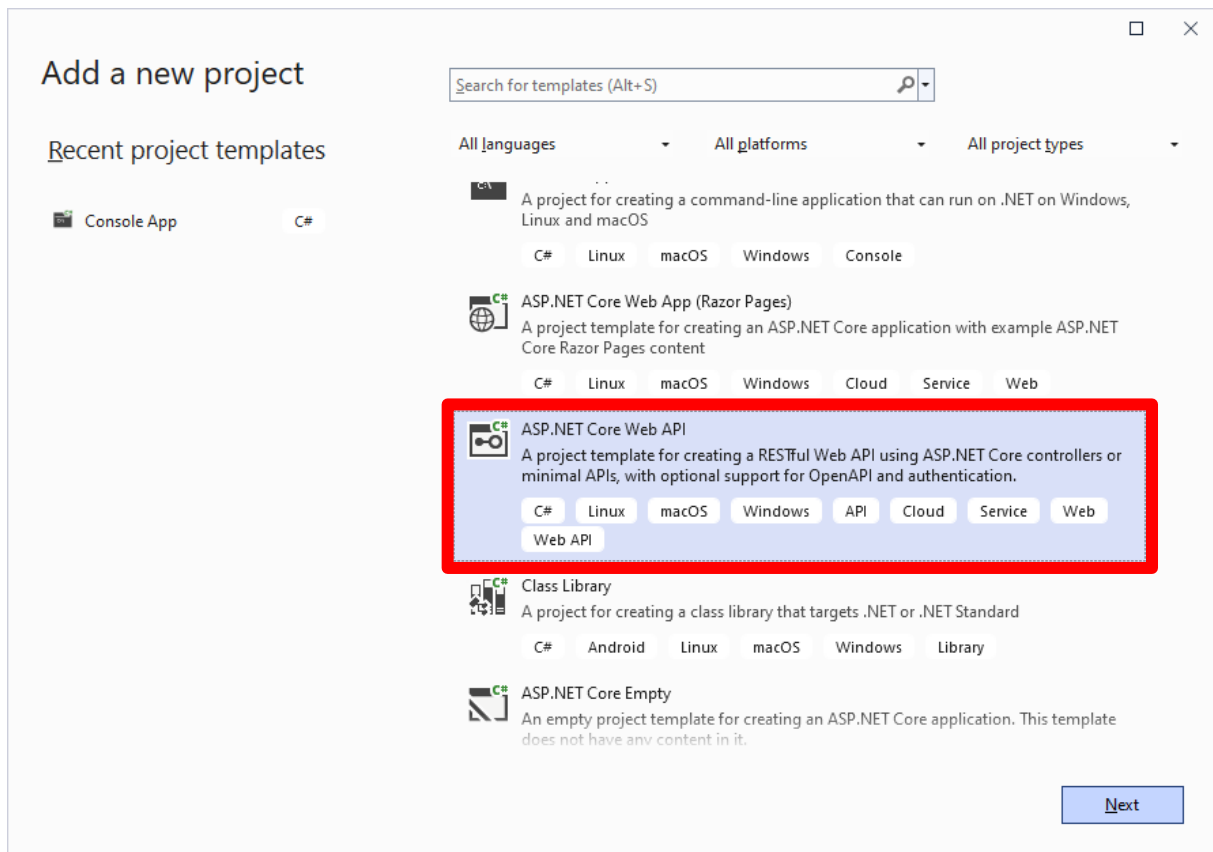
Program pokrećemo s klikom na zelenu PLAY ikonu te rezultat izvođenja programa bude vidljiv u CMD prozoru koji se otvori po uspješnom kreiranju (Build) izvršne datoteke. Pritiskom na bilo koju tipku u tom CMD prozoru, prozor se zatvara.



Desni klik miša na Solution Učenje u dijelu Solution explorer (desni gornji dio zaslona) idemo na Addpa na New Project... kako bi kreirali i ASP .NET Core Web API projekt.



U predloščima pronađemo ASP .NET Core Web API te odaberemo Next



Projekt nazovemo WebAPI te postavimo lokaciju kako stoji jer je to drugi projekt u istom rješenju i odaberemo Next.

□

×

Configure your new project

ASP.NET Core Web API C# Linux macOS Windows API Cloud Service Web Web API

Project name

Location

...

Project will be created in "C:\Users\Korisnik\GitHub\Edunova\CSHARP\Ucenje\WebAPI\."

Back

Next

Opcije ostavio kako je prikazano te odaberemo Create

□

×

Additional information

ASP.NET Core Web API C# Linux macOS Windows API Cloud Service Web Web API

Framework ⓘ

.NET 7.0 (Standard Term Support) ▾

Authentication type ⓘ

None ▾

☒ Configure for HTTPS ⓘ

☐ Enable Docker ⓘ

Docker OS ⓘ

Linux ▾

☒ Enable OpenAPI support ⓘ

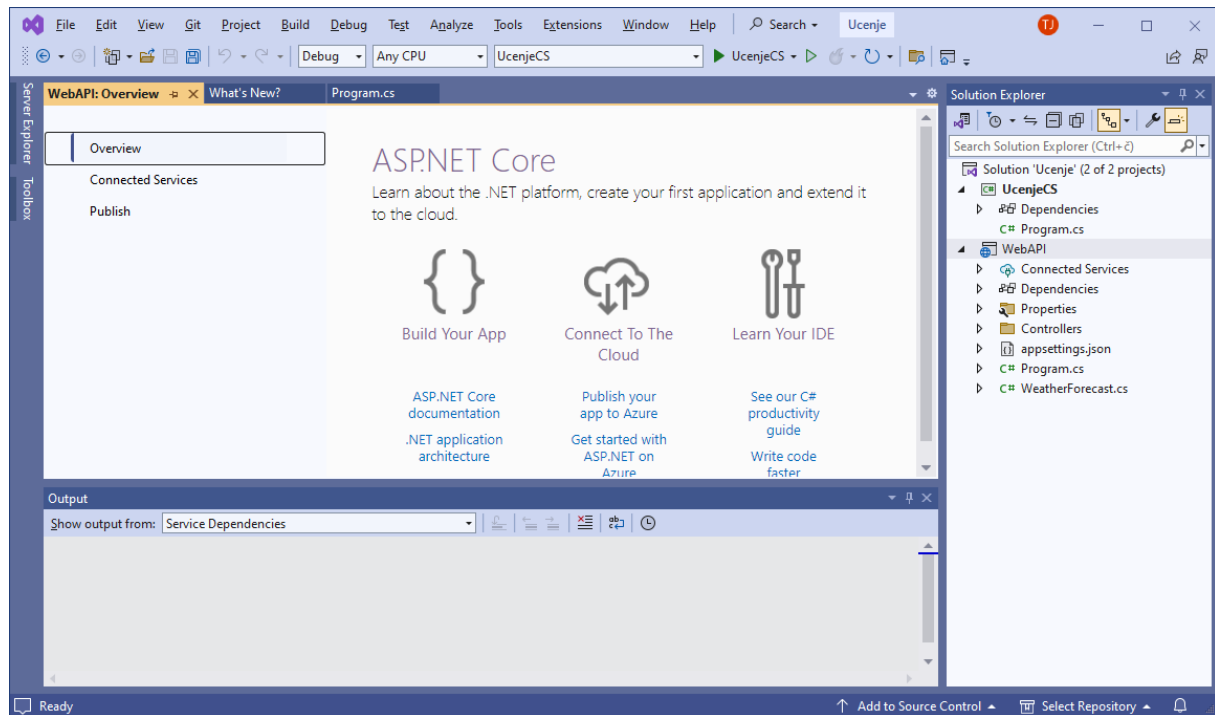
☐ Do not use top-level statements ⓘ

☒ Use controllers ⓘ

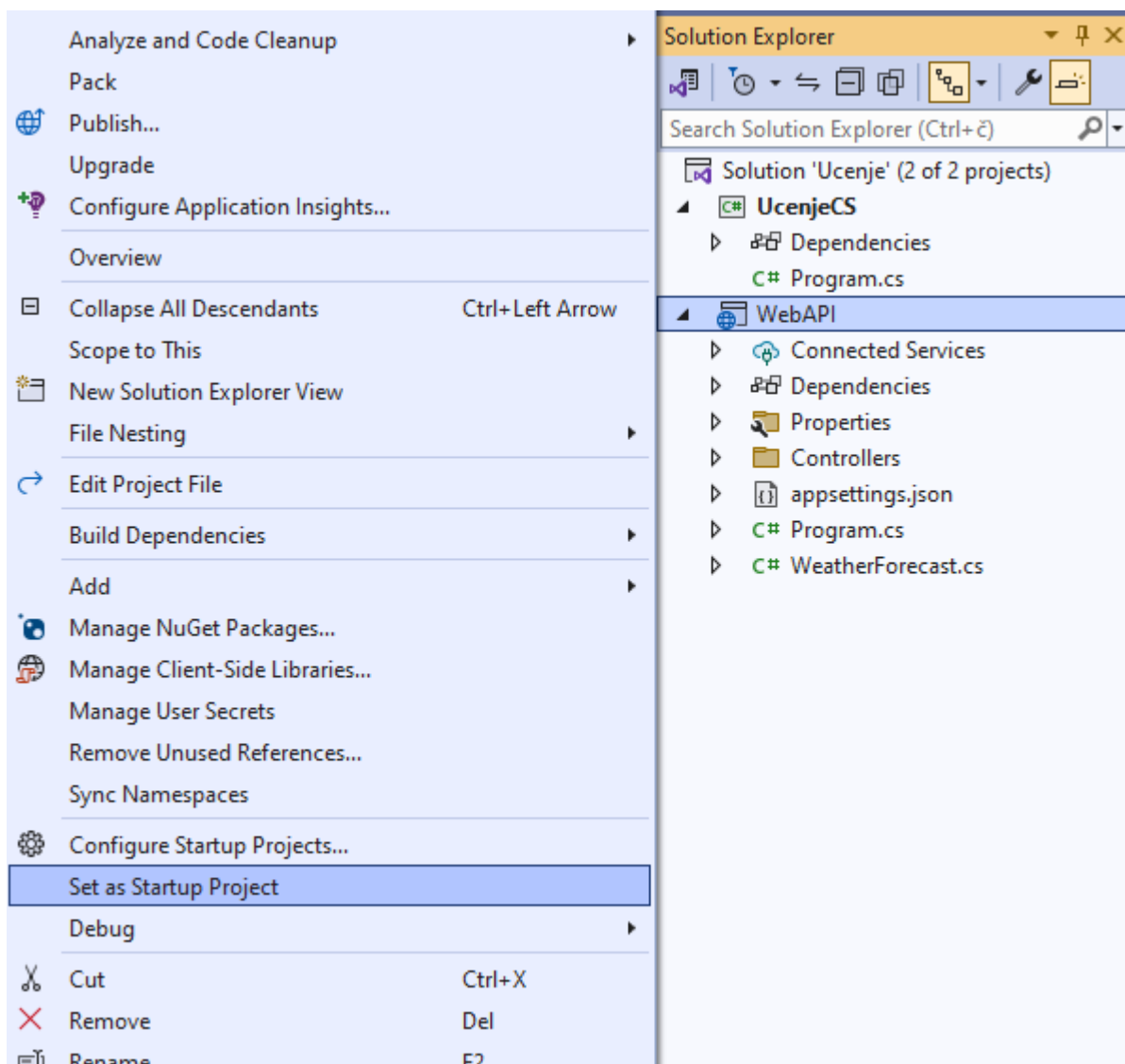
Back

Create

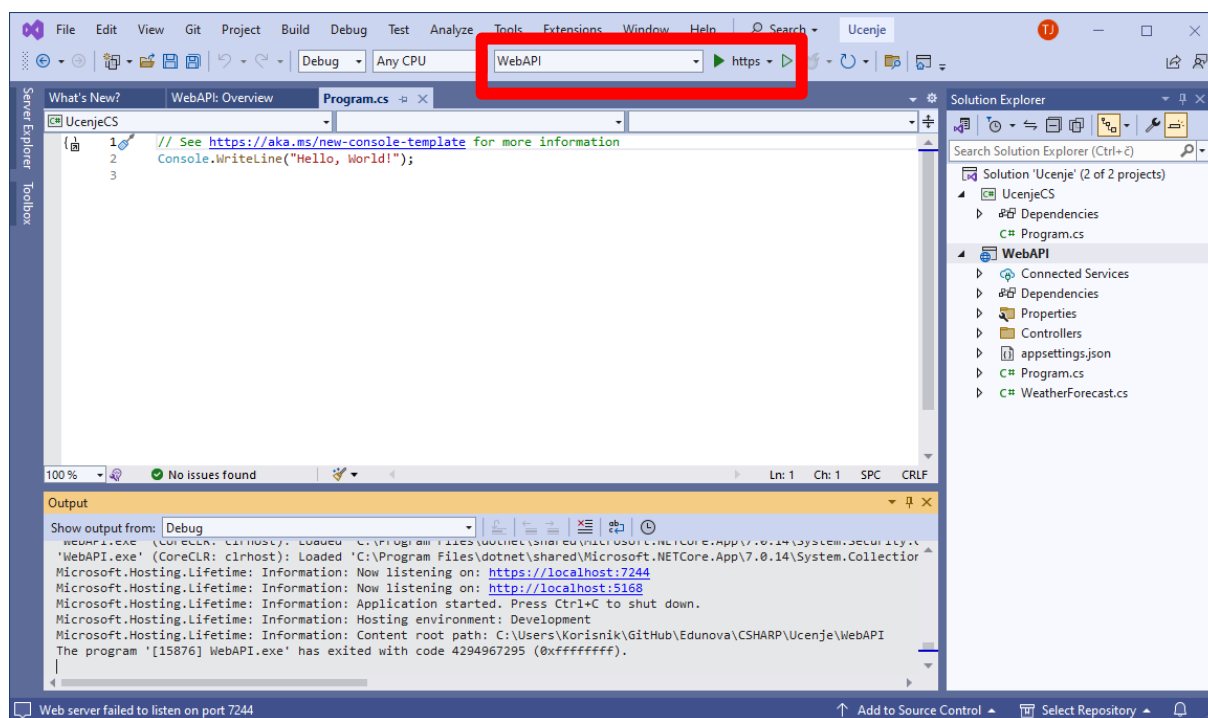
Ovaj drugi projekt je po strukturi puno kompleksniji ali to nas za sada ne zanima – ta kompleksnost je predmet učenja.



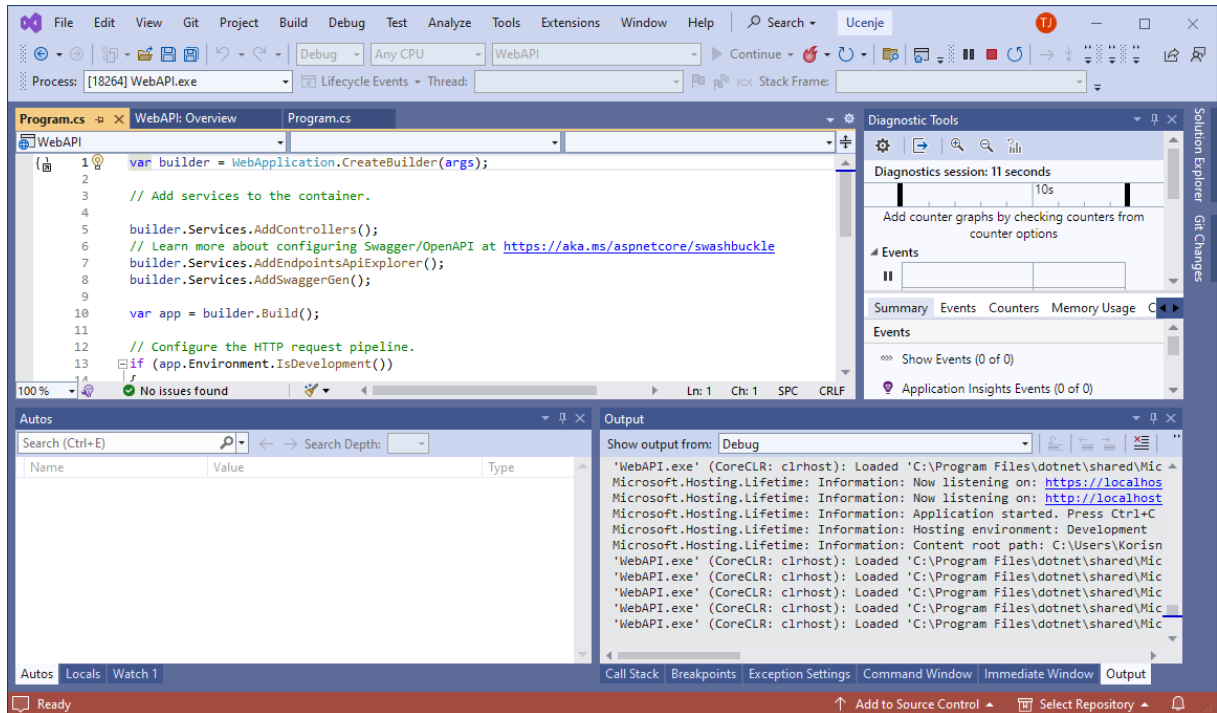
Kako sada u rješenju imamo dva projekta pokretanje je ostalo na projektu UcenjeCS. Da bi omogućili pokretanje WebAPI projekta moramo ga postaviti kao Start Up projekt. Stoga radimo desni klik na naziv projekta i odaberemo Set as Startup Project



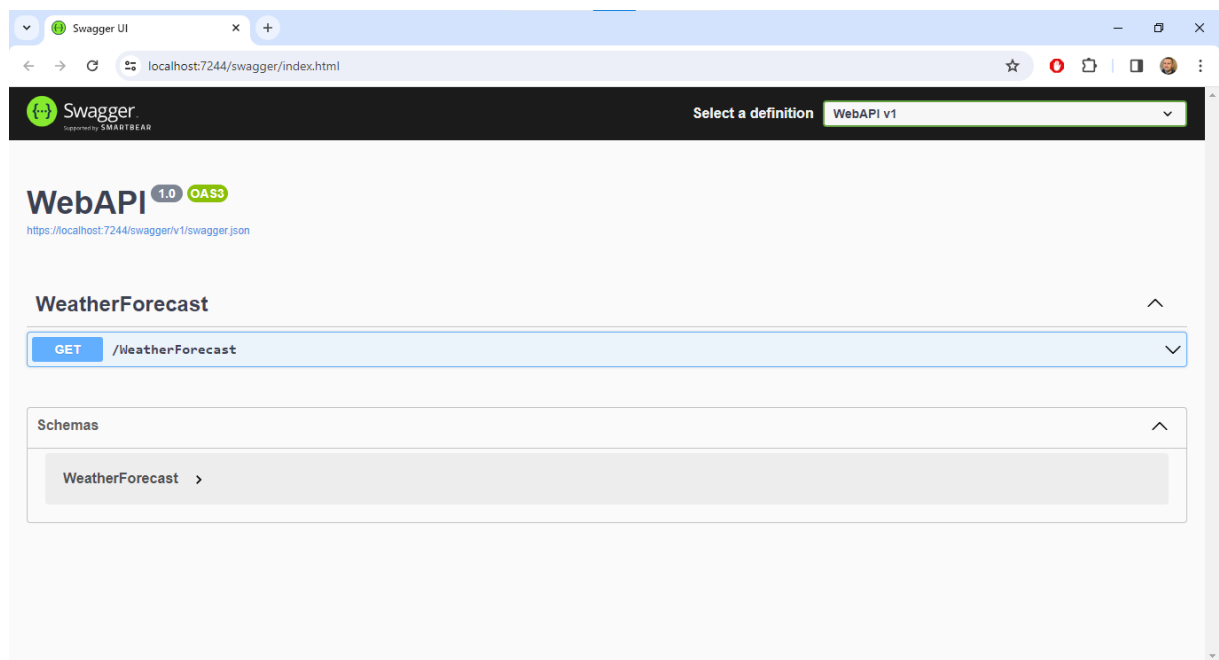
Sada možemo ponovo na zelenu PLAY ikonu pokrenuti projekt WebAPI.



Sada dobivamo drugačije pokretanje u odnosu Console aplikaciju (UcenjeCS) jer će se sada prilikom pokretanja podići i web server.



Te će se pokrenuti preglednik (Chrome ili Edge) s prikazom Swagger sučelja.



Swagger je alat koji nam omogućuje vizualni pregled i korištenje dostupnih ruta. Kliknemo na GET /EatherForecast te klik na opciju Try it out (gore desno) pa Execute i dobijemo rezultat izvođenja te rute.

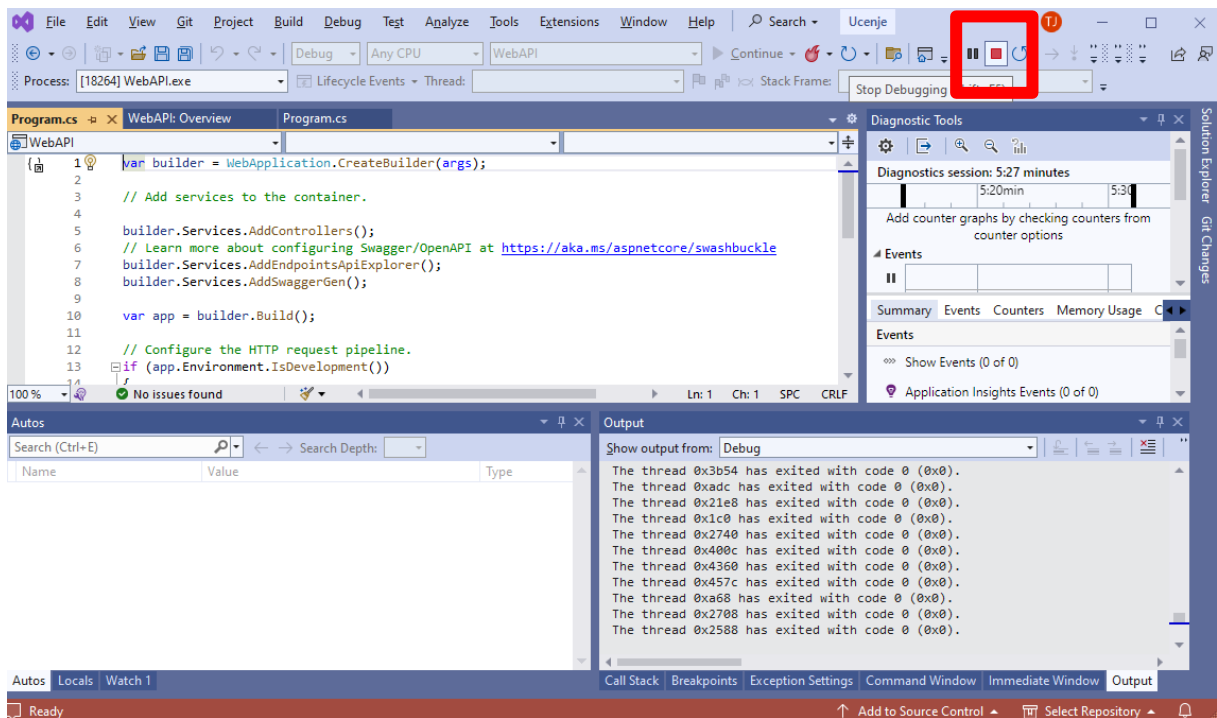
The screenshot displays the Swagger UI for an API named "WeatherForecast". The interface is divided into several sections:

- GET /WeatherForecast**: The selected endpoint.
- Parameters**: A section indicating "No parameters" with a "Cancel" button.
- Execute**: A blue button to run the request, followed by a "Clear" button.
- Responses**: A section showing the results of the request.
 - Curl**: A text box containing the command:

```
curl -X 'GET' \
'https://localhost:7244/WeatherForecast' \
-H 'accept: text/plain'
```
 - Request URL**: A text box showing `https://localhost:7244/WeatherForecast`.
 - Server response**: A section showing the response details.
 - Code**: 200.
 - Response body**: A large text box containing a JSON array of weather forecast data. The data includes dates, temperatures, and summaries for different times of the day.
 - Response headers**: A text box showing headers: `content-type: application/json; charset=utf-8`, `date: Thu, 07 Dec 2023 14:14:16 GMT`, and `server: Kestrel`.
- Responses**: A table at the bottom showing the response code (200) and description (Success).

Ovo su nam polazišne točke.

WebAPI projekt se zaustavlja pomoću crvene stop ikone.



Happy coding!