Lab Manual- Create Sample DotNet Console App and Store in Github

Prepared for:

Date: 18th Nov 2018

Prepared by: Aditi Shrivastava

Document Name: Lab Manual

Document Number SysOpsLab312

Contributor:

Table of Contents

1 OBJECTIVE

For various exercise we need to setup the local development environment where developer will create and update the code for application.

2 PRE-REQUISISTE

- Accounts in Azure
- A local Computer with 4 CPU, 16 GB RAM, 200 GB disk space

3 Create DotNet Core Console App

3.1 Create your app

A simple application written in C# that prints `Hello, World!` to the consol

- In your command prompt, run the following command to create your app:

  dotnet new console -o myApp

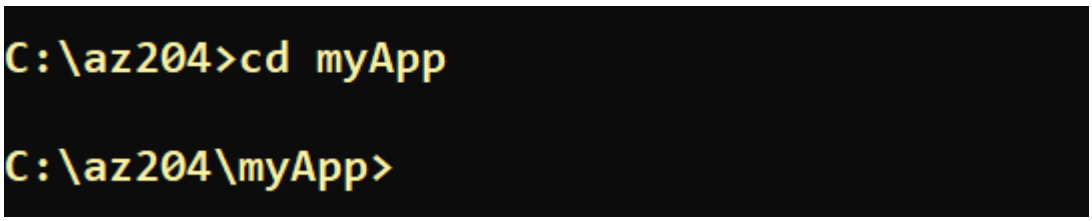
```
C:\az204>dotnet new console -o myApp

Welcome to .NET Core 3.1!
---------------------
SDK Version: 3.1.410

Telemetry
---------
The .NET Core tools collect usage data in order to help us improve your experience. It is
ou can opt-out of telemetry by setting the DOTNET_CLI_TELEMETRY_OPTOUT environment variab

Read more about .NET Core CLI Tools telemetry: https://aka.ms/dotnet-cli-telemetry

----------------
Explore documentation: https://aka.ms/dotnet-docs
Report issues and find source on GitHub: https://github.com/dotnet/core
Find out what's new: https://aka.ms/dotnet-whats-new
Learn about the installed HTTPS developer cert: https://aka.ms/aspnet-core-https
Use 'dotnet --help' to see available commands or visit: https://aka.ms/dotnet-cli-docs
Write your first app: https://aka.ms/first-net-core-app
--------------------------------------------------------------------------------------
Getting ready...
The template "Console Application" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on myApp\myApp.csproj...
  Determining projects to restore...
  Restored C:\az204\myApp\myApp.csproj (in 158 ms).

Restore succeeded.
```

[Note] What do these commands mean?

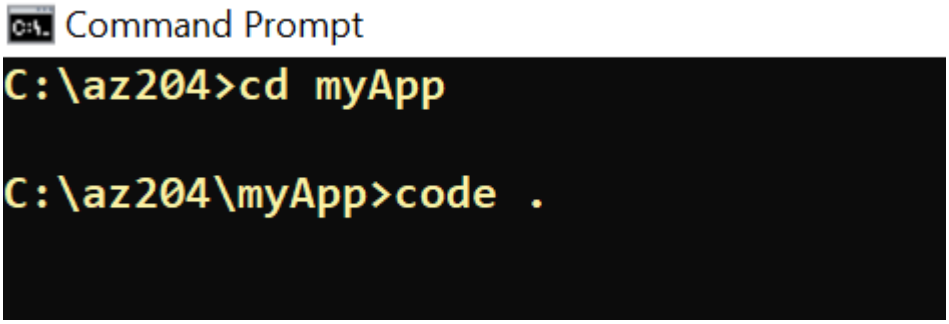
The `dotnet new console` command creates a new console app for you. The **`-o` parameter** creates a directory named **`myApp`** where your app is stored and populates it with the required files.

- Then, navigate to the new directory created by the previous command:

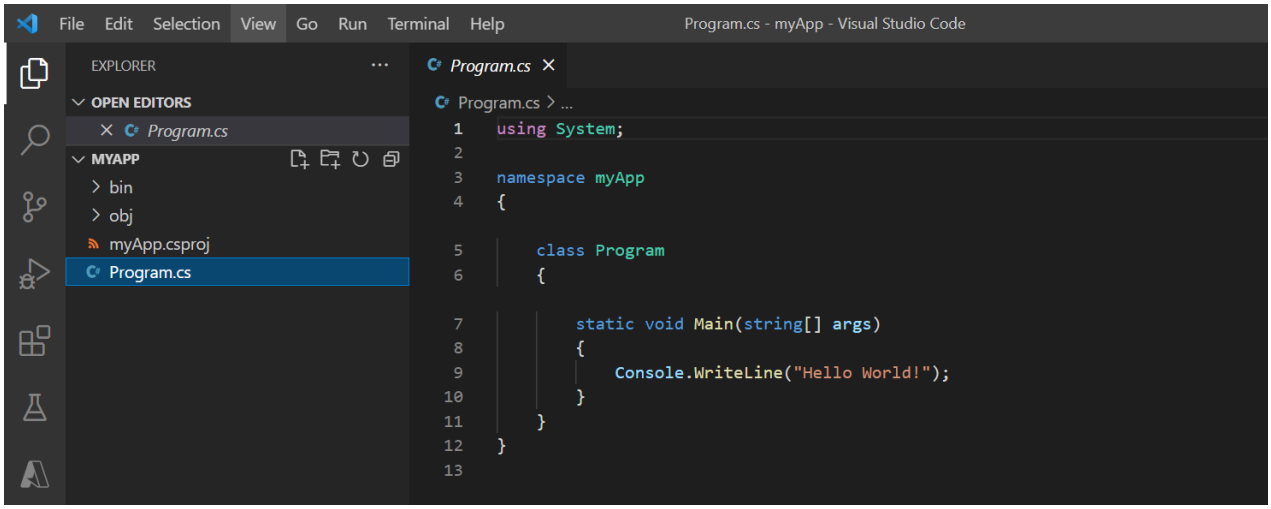
```
cd myApp
```



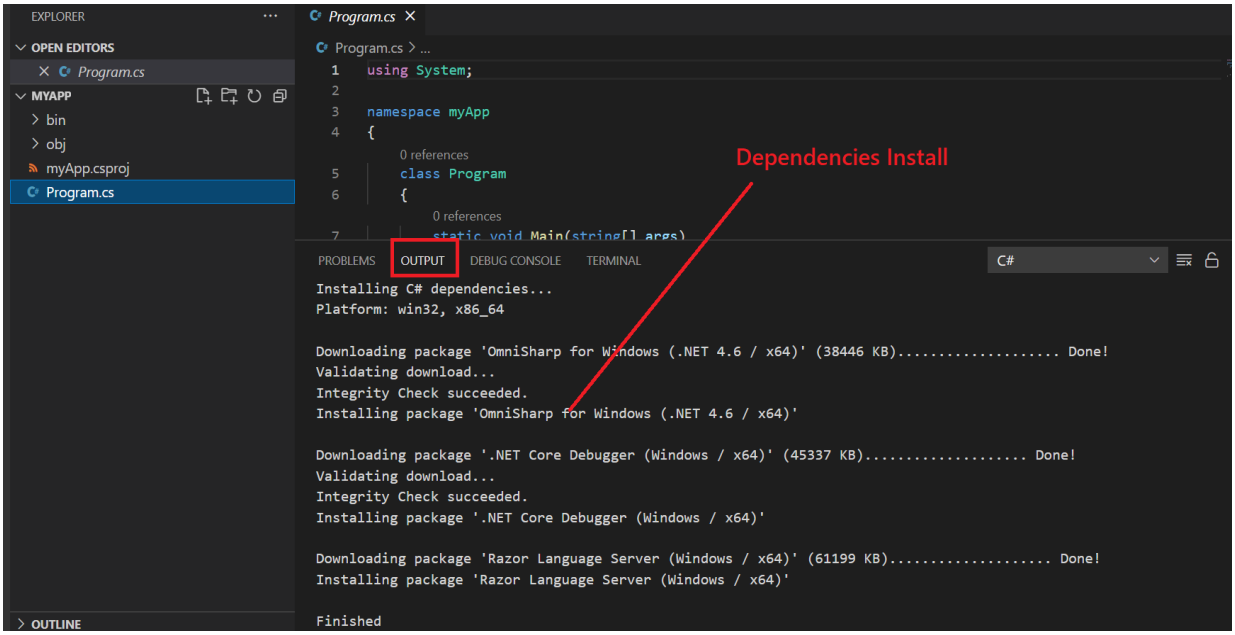
- Now open **Code .** to Launc VS Code Editor



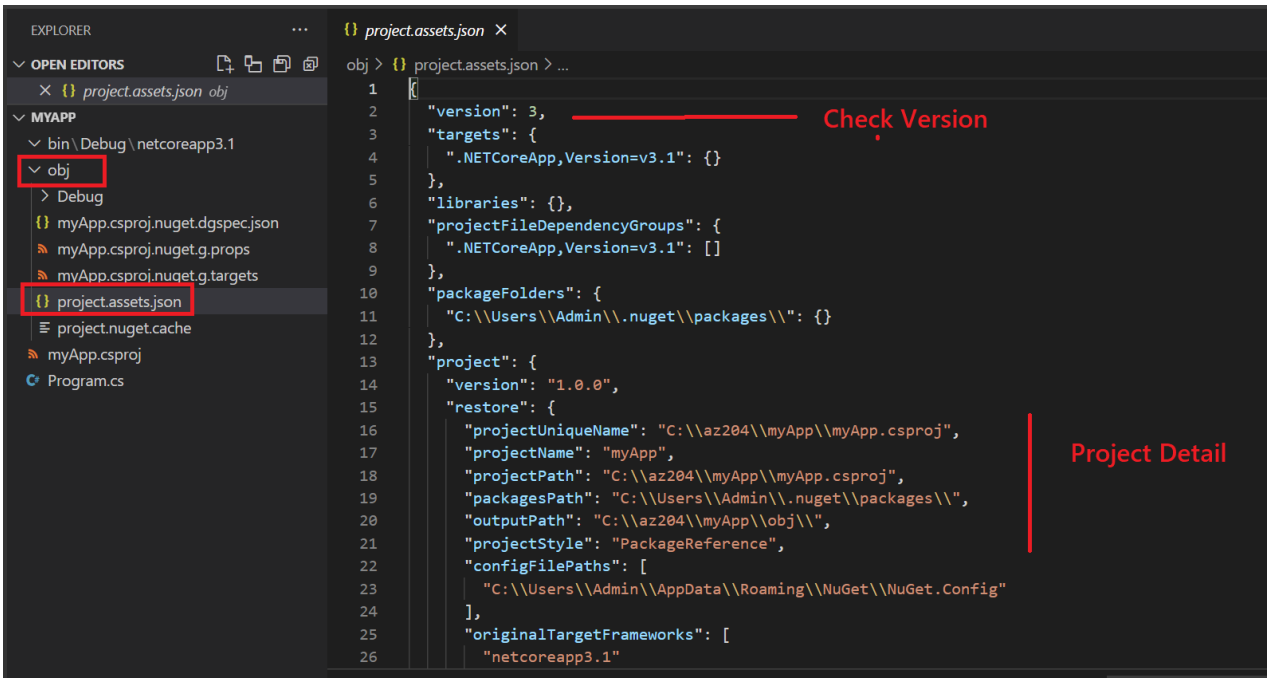
- It Open the project in VS Code Editor



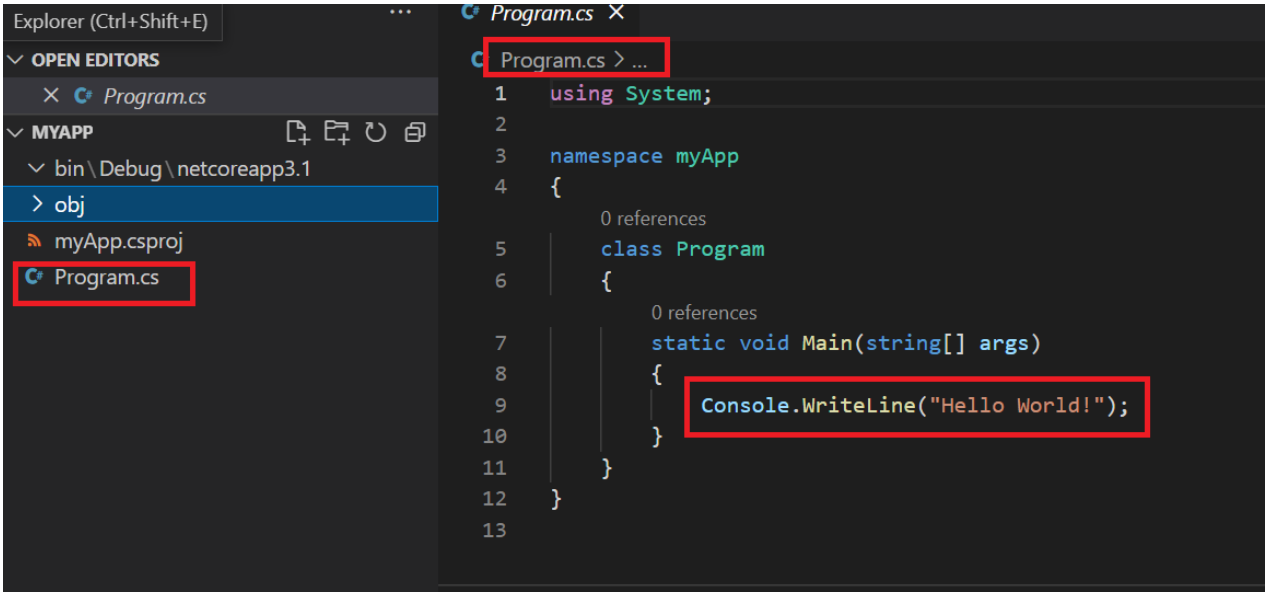
- When You create project it also install dependecies



- You Can also see the projectd details



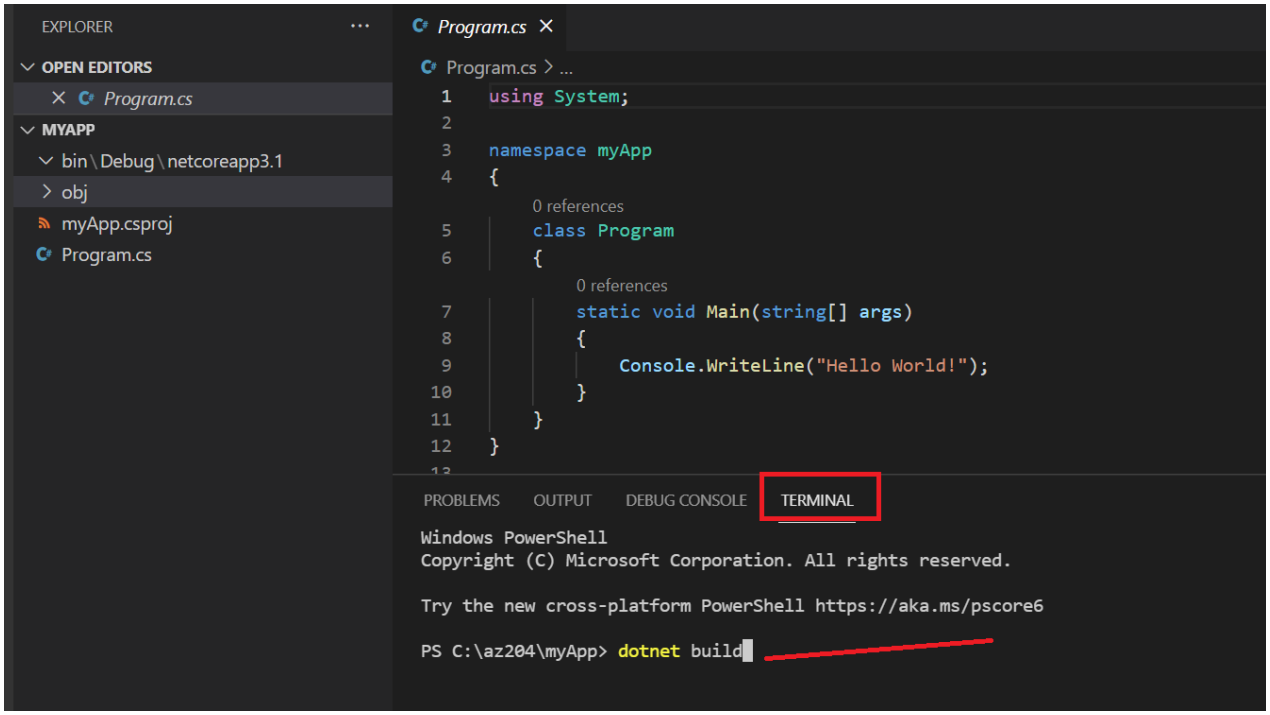
- The main file in the **myApp** folder is **Program.cs**. By default, it already contains the necessary code to write "**Hello World!**" to the Console.



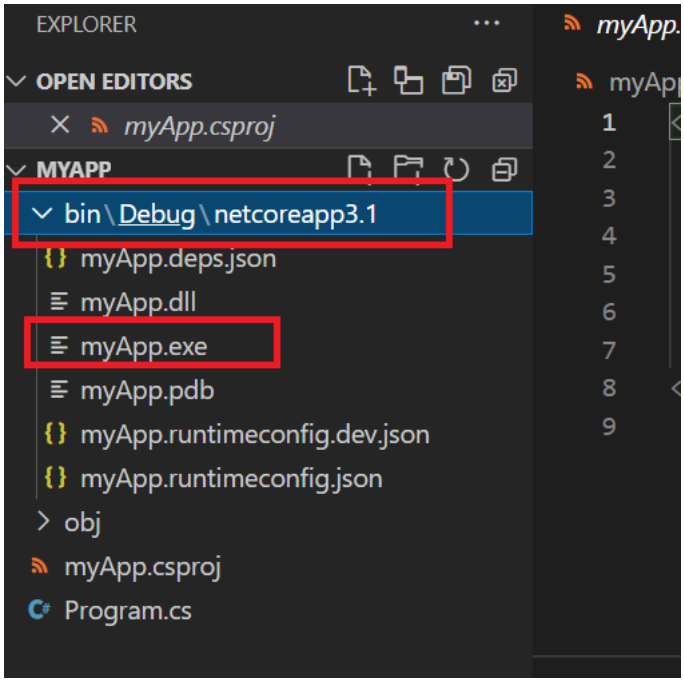
3.2 Build and Run Your App

- Now we are going to Build the program (Build compiles the source code into a (hopefully) runnable application)

Dotnet Build

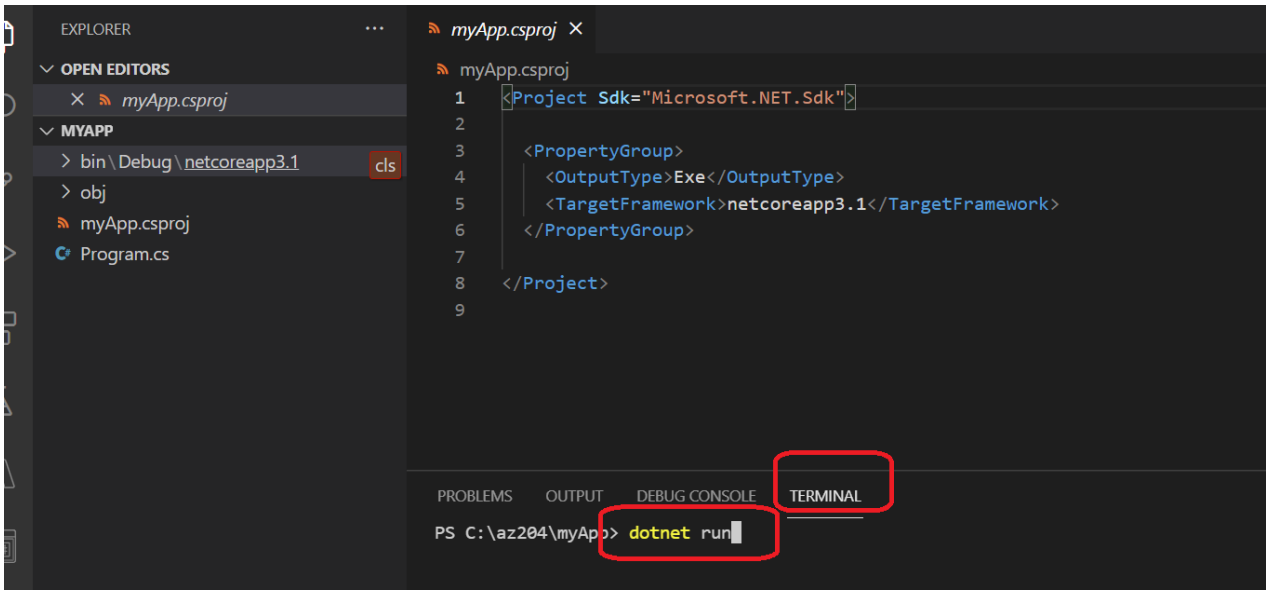


- When you Build the program compiles the source code into a (hopefully) runnable application) . You can check the output from bin directory and Notice **myapp.exe /Myapp.dll**

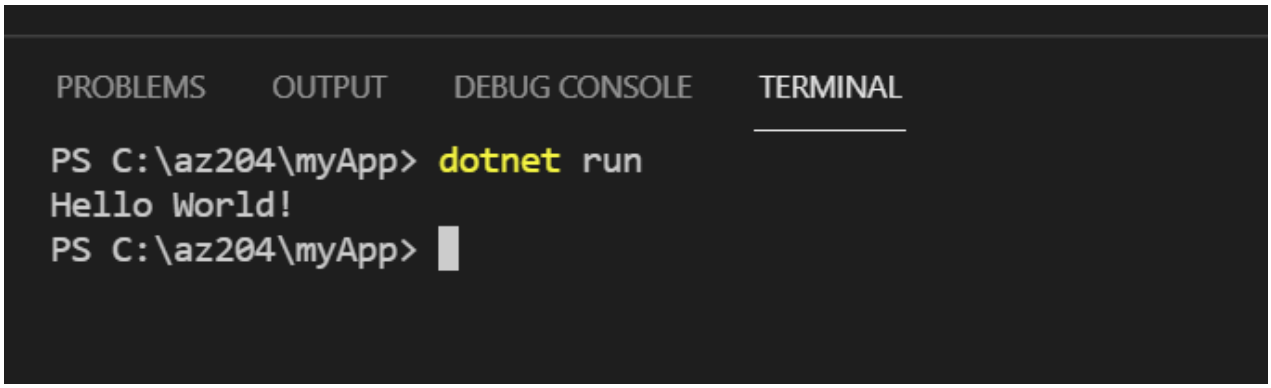


- In your command prompt, run the following command:

Dotnet Run

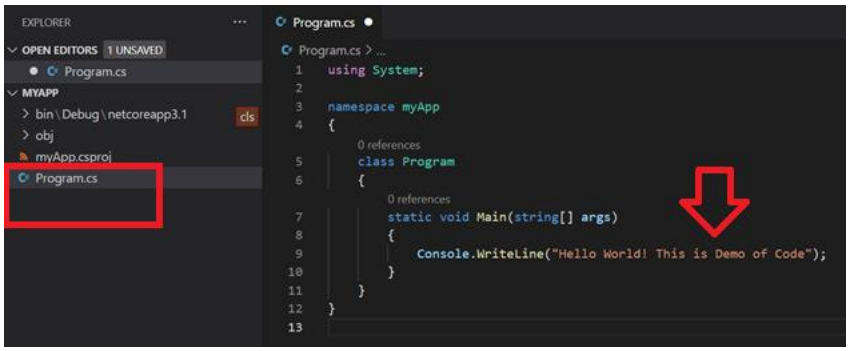


- Congratulations, you've built and run your first .NET app!

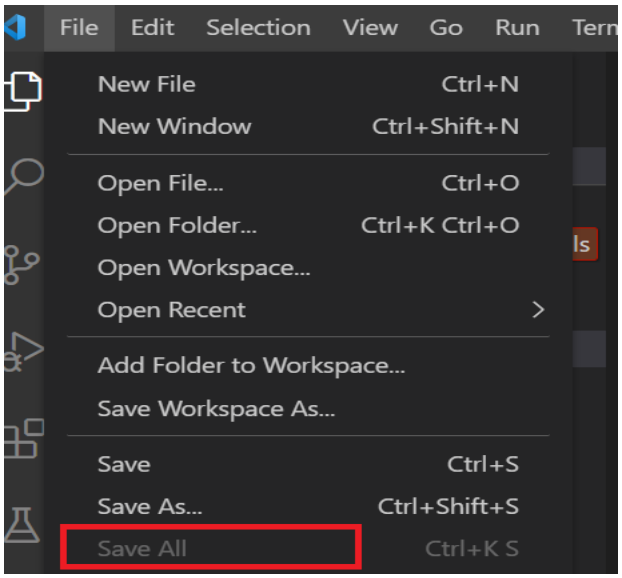


3.3 Update Your App

Now lets make small changes in code. For that we go to Program.cs and add some line in Println statement



Now save **– File → Save All**

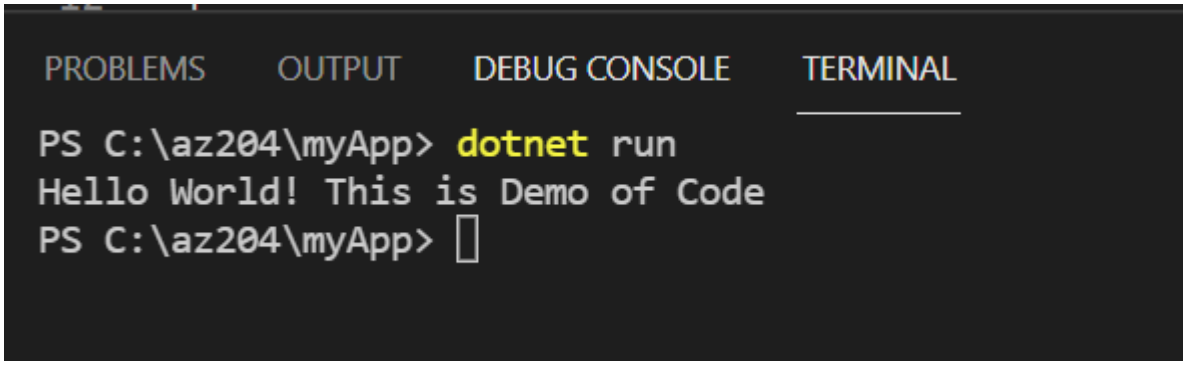


Now again Build and Run the Code.

DotNet Build

```
PS C:\az204\myApp> dotnet build
Microsoft (R) Build Engine version 16.7.2+b60ddb6f4 for .NET
Copyright (C) Microsoft Corporation. All rights reserved.

  Determining projects to restore...
  All projects are up-to-date for restore.
  myApp -> C:\az204\myApp\bin\Debug\netcoreapp3.1\myApp.dll

Build succeeded.
    0 Warning(s)
    0 Error(s)

Time Elapsed 00:00:04.09
```

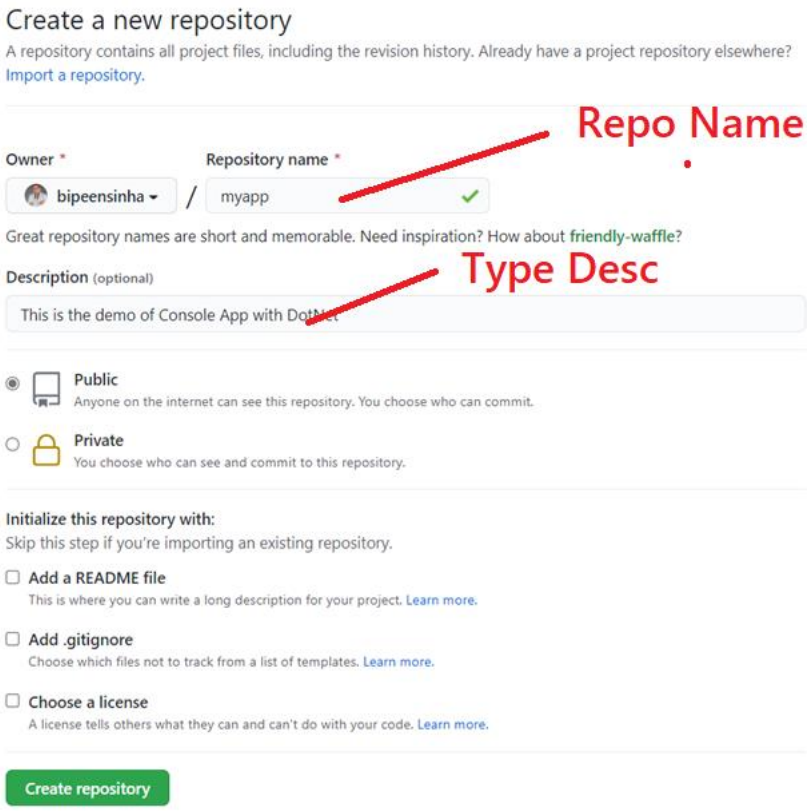
DotNet Run



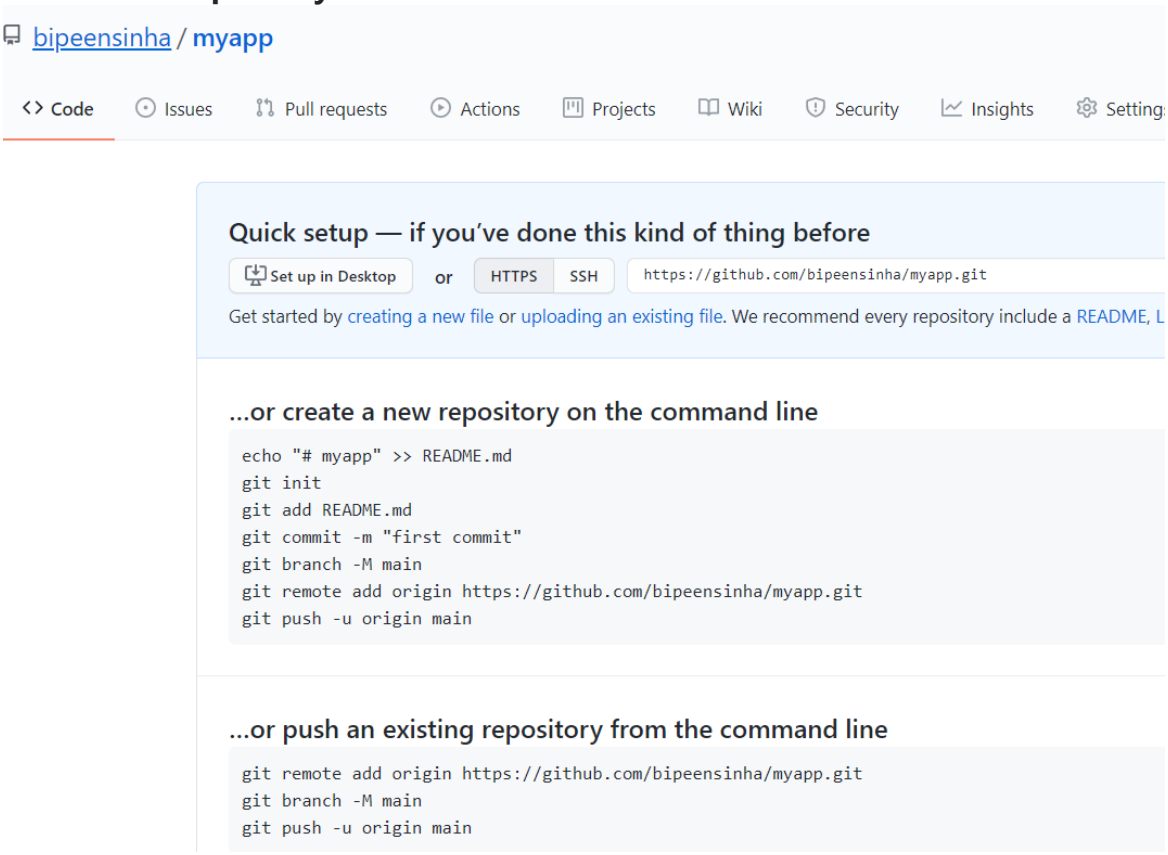
4 Push the Code to Source Code Repository (Github)

4.1 Create a Repo in Github

- Goto Github and click New Repo
- Type the Reponame as „myapp"
- Type some description
- Without clicking the any other check box



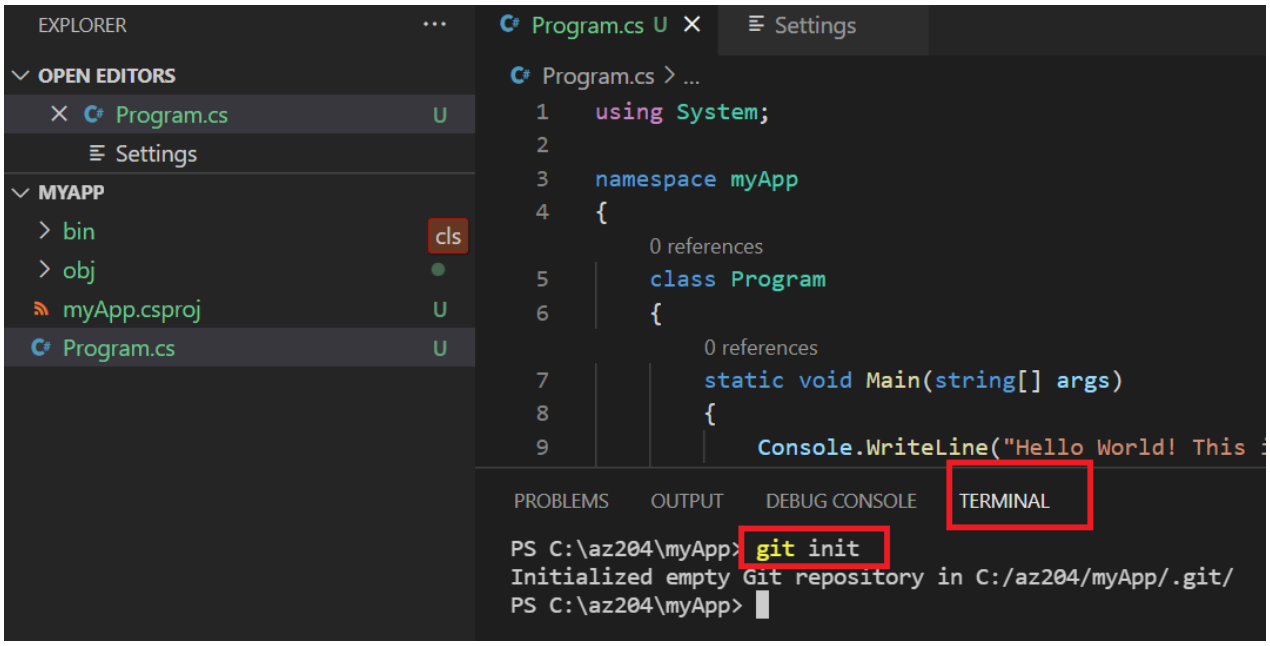
- Click **Create Repository**



4.2 Initialize Local Repo and Push the code to Github

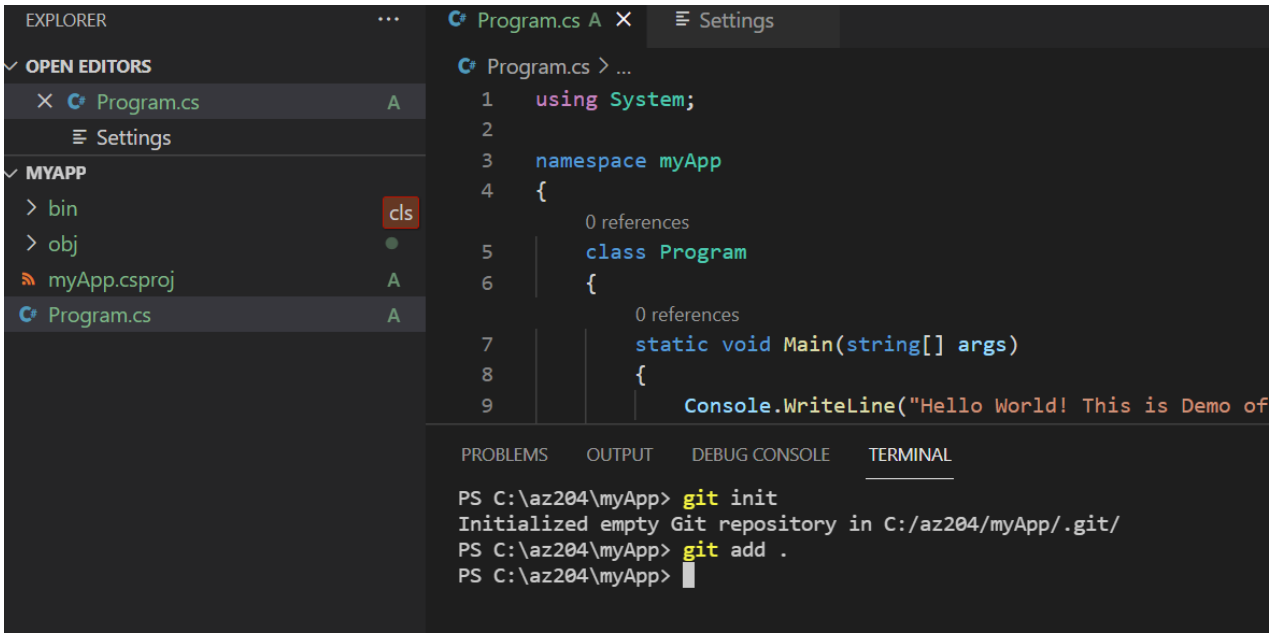
- On the local VS Code terminal type below command

Git Init



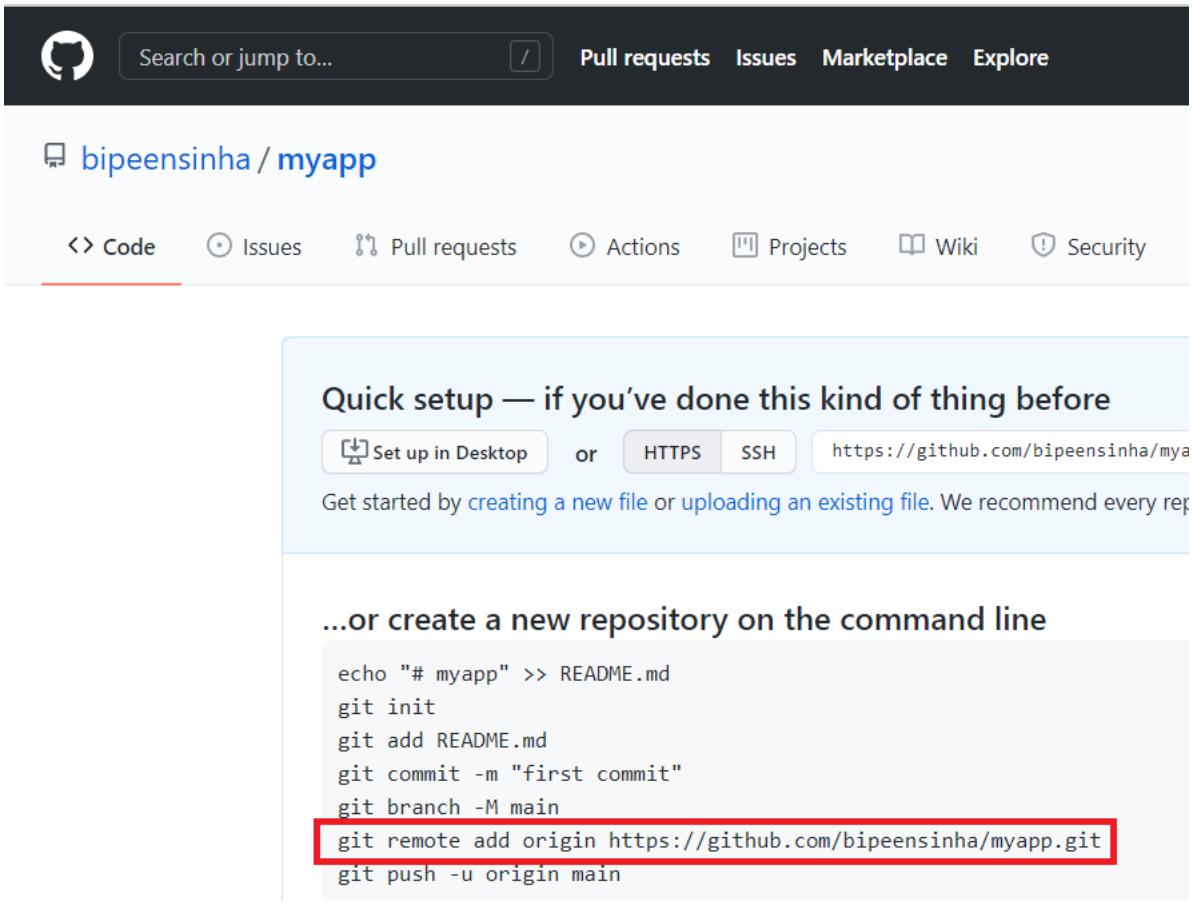
- Now add the code to git Quue with git add . (where . (dot) represent everything in current directory)

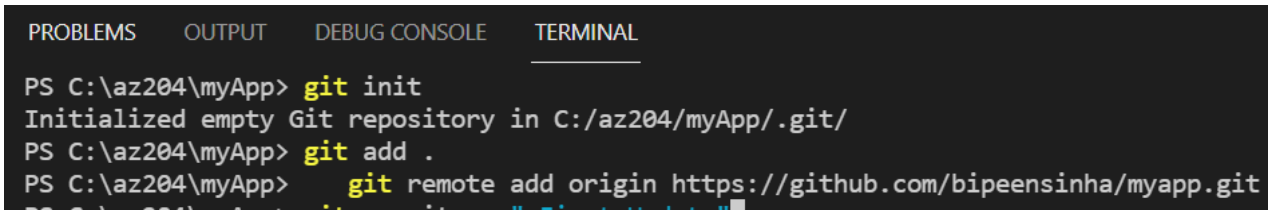
Git add .



- Now copy the git remote orgin command from github and past it here. It is going to tel your git agent installed your laptop that which repo to push the code

```
git remote add origin https://github.com/bipeensinha/myapp.git
```





- Now commit the code to Github queue

  ```
  git commit -m "first commit"
  ```

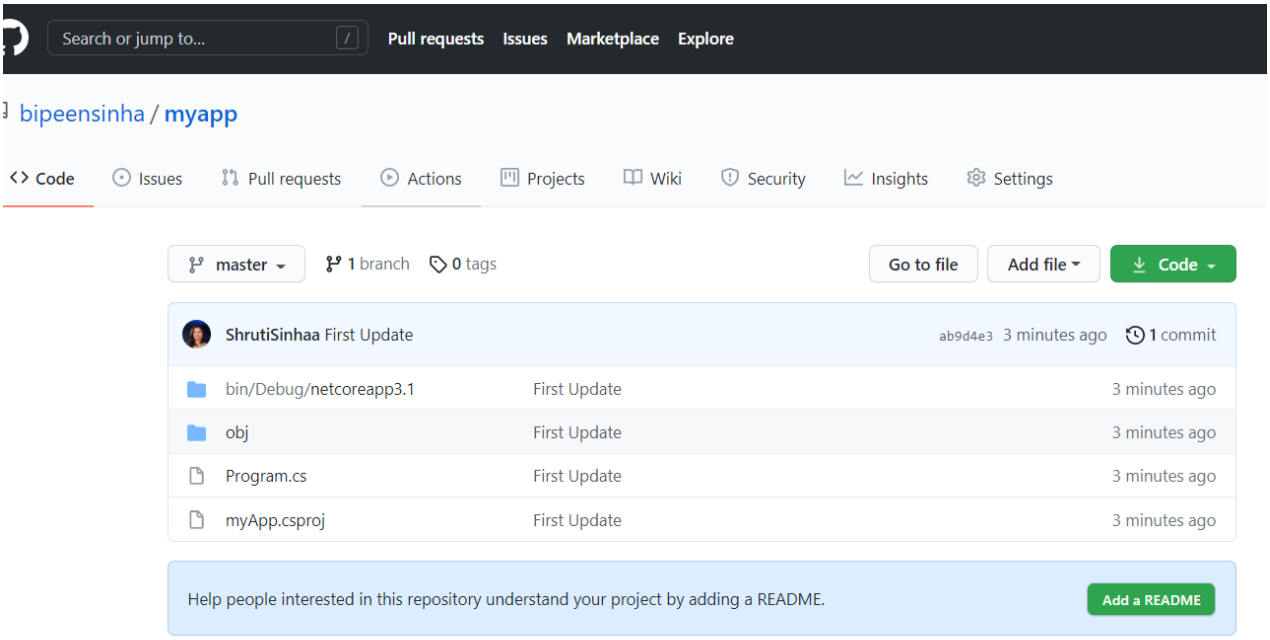
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS C:\az204\myApp> git commit -m " First Update"
[master (root-commit) ab9d4e3]  First Update
 25 files changed, 256 insertions(+)
 create mode 100644 Program.cs
 create mode 100644 bin/Debug/netcoreapp3.1/myApp.deps.json
 create mode 100644 bin/Debug/netcoreapp3.1/myApp.dll
 create mode 100644 bin/Debug/netcoreapp3.1/myApp.exe
 create mode 100644 bin/Debug/netcoreapp3.1/myApp.pdb
 create mode 100644 bin/Debug/netcoreapp3.1/myApp.runtimeconfig.dev.json
 create mode 100644 bin/Debug/netcoreapp3.1/myApp.runtimeconfig.json
 create mode 100644 myApp.csproj
 create mode 100644 obj/Debug/netcoreapp3.1/.NETCoreApp,Version=v3.1.Assembly
 create mode 100644 obj/Debug/netcoreapp3.1/myApp.AssemblyInfo.cs
 create mode 100644 obj/Debug/netcoreapp3.1/myApp.AssemblyInfoInputs.cache
 create mode 100644 obj/Debug/netcoreapp3.1/myApp.assets.cache
 create mode 100644 obj/Debug/netcoreapp3.1/myApp.csproj.AssemblyReference.ca
```

- Now Push the code to Github Repository

  `git push –all`

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS C:\az204\myApp> git push --all
Enumerating objects: 30, done.
Counting objects: 100% (30/30), done.
Delta compression using up to 4 threads
Compressing objects: 100% (24/24), done.
Writing objects: 100% (30/30), 102.39 KiB | 3.41 MiB/s, done.
Total 30 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/bipeensinha/myapp.git
 * [new branch]      master -> master
PS C:\az204\myApp>
```

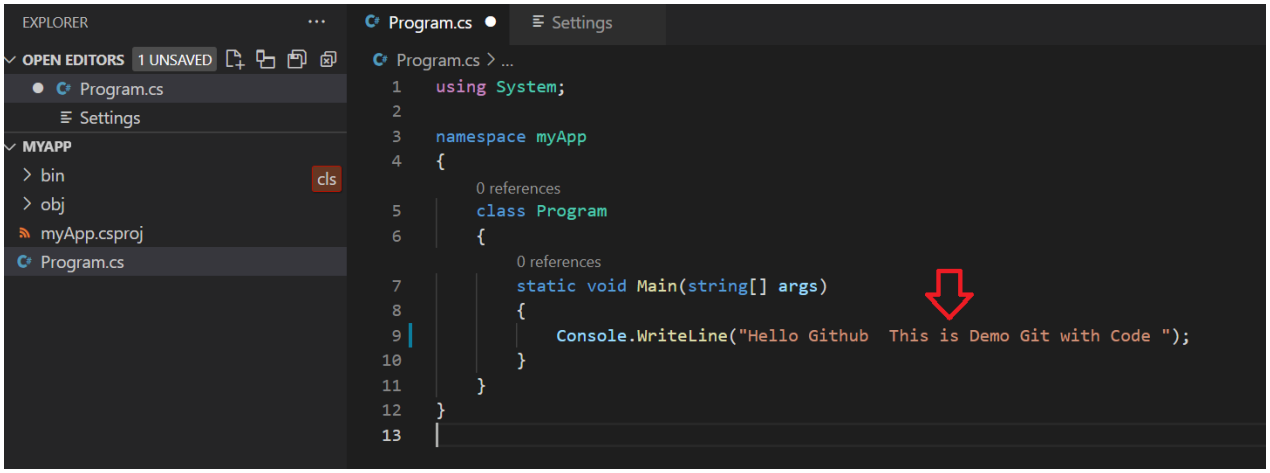
- Now go to github Portal , Refresh it and check you have your code there.



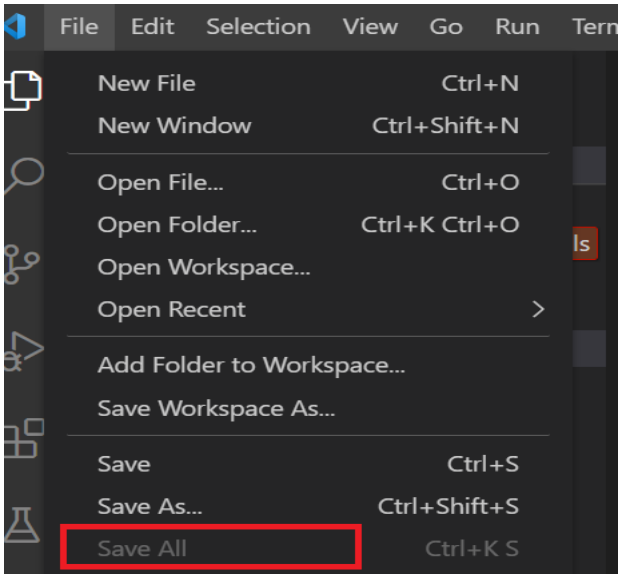
5 Update the Code and Push again to Repository (Github)

5.1 Update Your App

Now lets make small changes in code. For that we go to Program.cs and add some line in **PrintIn statement**



Now save **– File → Save All**



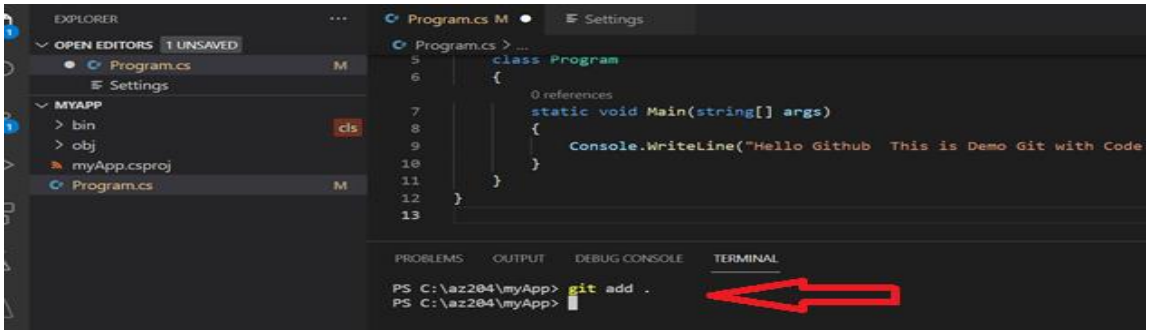
Now again Build Code.

DotNet Build

```
PS C:\az204\myApp> dotnet build
Microsoft (R) Build Engine version 16.7.2+b60ddb6f4 for .NET
Copyright (C) Microsoft Corporation. All rights reserved.

  Determining projects to restore...
  All projects are up-to-date for restore.
  myApp -> C:\az204\myApp\bin\Debug\netcoreapp3.1\myApp.dll

Build succeeded.
    0 Warning(s)
    0 Error(s)

Time Elapsed 00:00:04.09
```

- Now add the code to git Quue with git add . (where . (dot) represent everything in current directory)

Git add .



- Now commit the code to Github queue

  `git commit -m "Second commit"`

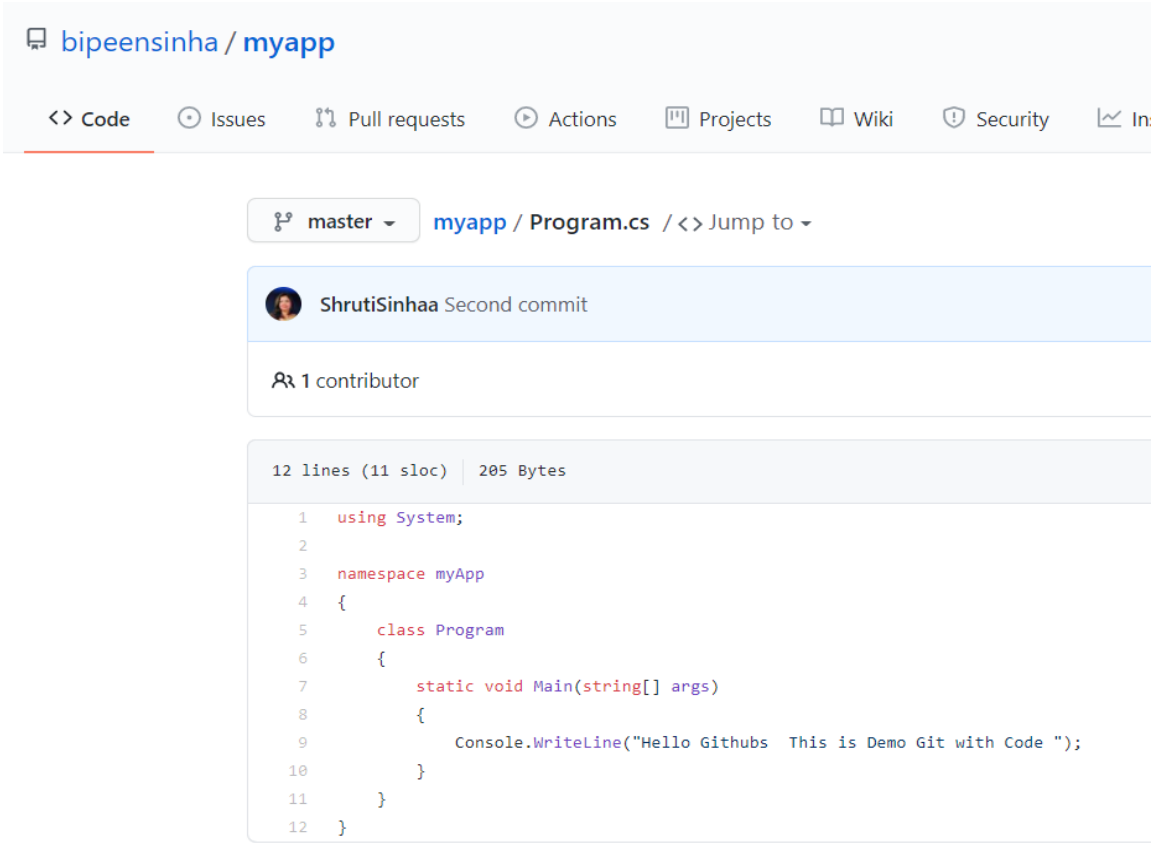
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\az204\myApp> git add .
PS C:\az204\myApp> git commit -m "Second commit"
[master 8fd51b0] Second commit
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\az204\myApp>
```

- Now Push the code to Github Repository

  `Git push --all`

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\az204\myApp> git add .
PS C:\az204\myApp> git commit -m "Second commit"
[master 8fd51b0] Second commit
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\az204\myApp> git push --all
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 714 bytes | 357.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/bipeensinha/myapp.git
   ab9d4e3..8fd51b0  master -> master
PS C:\az204\myApp>
```

- Now go to github Portal, Refresh it and check you have your code there.


bipeensinha / myapp
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
master
myapp / Program.cs
Jump to
ShrutiSinhaa Second commit
1 contributor
12 lines (11 sloc)
205 Bytes
using System;
namespace myApp
{
class Program
{
static void Main(string[] args)
{
Console.WriteLine("Hello Githubs This is Demo Git with Code ");
}
}
}