3. Configure global settings for the name and email address to be used when committing in this Git repository:

   Cmd

   ```
   git config --global user.name "John Doe"
   git config --global user.email "john.doe@contoso.com"
   ```

   If you are working behind an enterprise proxy, you can make your Git repository proxy-aware by adding the proxy details in the Git global configuration file.

   Different variations of this command will allow you to set up an HTTP/HTTPS proxy (with username/password) and optionally bypass SSL verification.

   Run the below command to configure a proxy in your global git config.

   Cmd

   ```
   git config --global http.proxy
   http://proxyUsername:proxyPassword@proxy.server.com:port
   ```

4. Create a new ASP.NET core application. The new command offers a collection of switches that can be used for language, authentication, and framework selection. More details can be found on Microsoft docs.

   Cmd

   ```
   dotnet new mvc
   ```

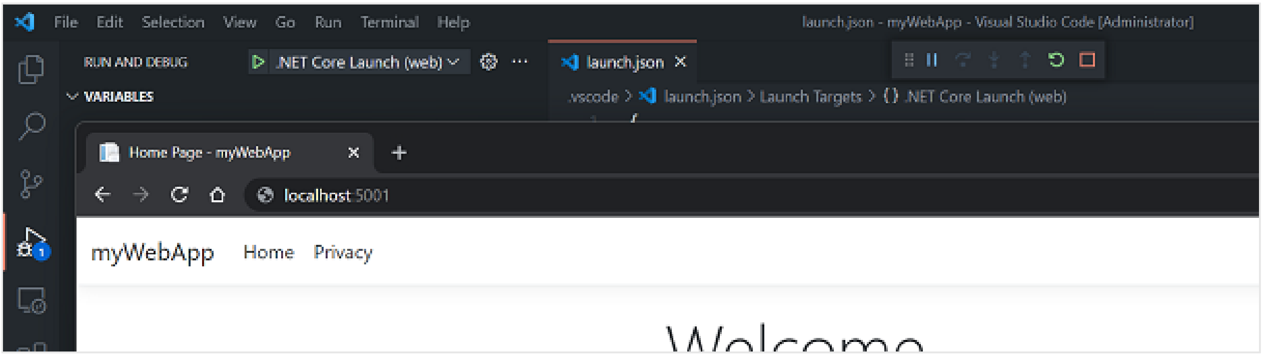
   Launch Visual Studio Code in the context of the current-working folder:

   Cmd

   ```
   code .
   ```

5. When the project opens in Visual Studio Code, select **Yes** for the **Required assets to build and debug are missing from 'myWebApp.' Add them?** Warning message. Select

Restore for the **There are unresolved dependencies** info message. Hit **F5** to debug the application, then myWebApp will load in the browser, as shown in the following screenshot:

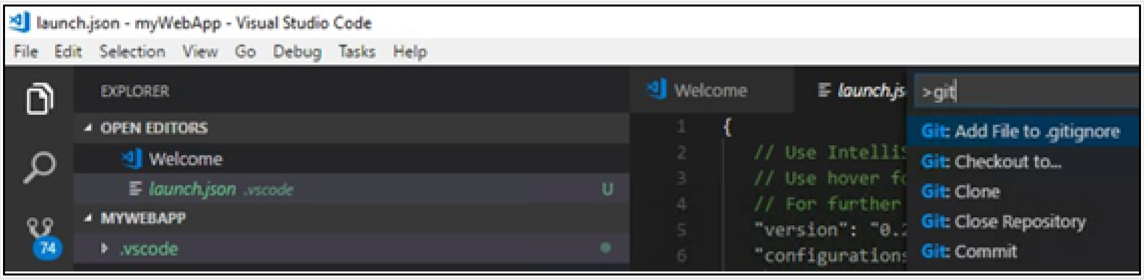

File Edit Selection View Go Run Terminal Help
launch.json - myWebApp - Visual Studio Code [Administrator]
RUN AND DEBUG
.NET Core Launch (web)
launch.json
VARIABLES
.vscode > launch.json > Launch Targets > {} .NET Core Launch (web)
Home Page - myWebApp
localhost:5001
myWebApp
Home
Privacy
Welcome

If you prefer to use the command line, you can run the following commands in the context of the git repository to run the web application.

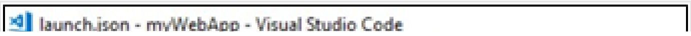
Cmd

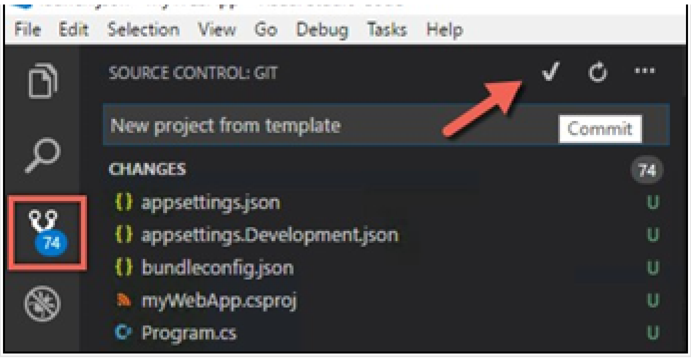
```
dotnet build
dotnet run
```

You will notice the ".vscode" folder is added to your working folder. To avoid committing this folder into your Git repository, you can include it in the .gitignore file. With the ".vscode" folder selected, hit F1 to launch the command window in Visual Studio Code, type gitIgnore, and accept the option to include the selected folder in the .gitIgnore file:



6. To stage and commit the newly created myWebApp project to your Git repository from Visual Studio Code, navigate the Git icon from the left panel. Add a commit comment and commit the changes by clicking the checkmark icon. It will stage and commit the changes in one operation:

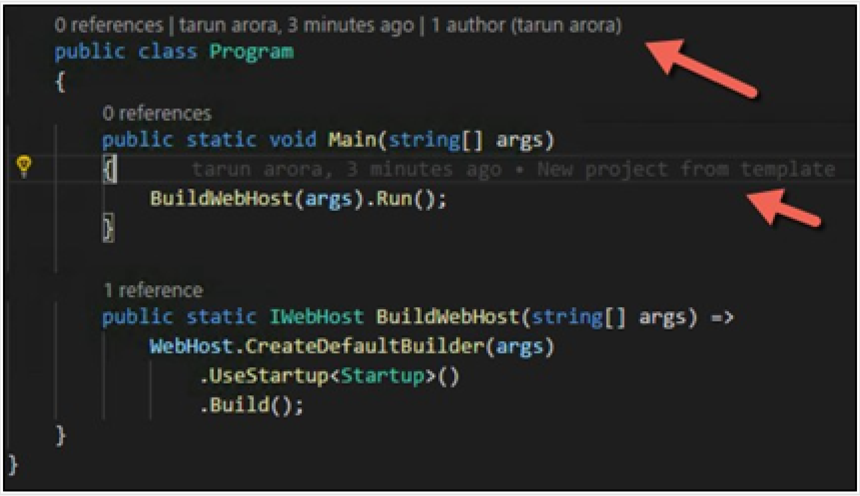




New project from template

```
{} appsettings.json                   U
{} appsettings.Development.json       U
{} bundleconfig.json                  U
   myWebApp.csproj                    U
C# Program.cs                         U
```

Open Program.cs, you will notice Git lens decorates the classes and functions with the commit history and brings this information inline to every line of code:



```
    0 references | tarun arora, 3 minutes ago | 1 author (tarun arora)
    public class Program
    {
        0 references
        public static void Main(string[] args)
        {        tarun arora, 3 minutes ago • New project from template
            BuildWebHost(args).Run();
        }

        1 reference
        public static IWebHost BuildWebHost(string[] args) =>
            WebHost.CreateDefaultBuilder(args)
                .UseStartup<Startup>()
                .Build();
    }
}
```

7. Now launch cmd in the context of the git repository and run `git branch --list`. It will show you that currently, only the `main` branch exists in this repository. Now run the following command to create a new branch called `feature-devops-home-page`.

Cmd

```
git branch feature-devops-home-page
git checkout feature-devops-home-page
git branch --list
```

With these commands, you have created a new branch, checked it out. The `--list` keyword shows you a list of all branches in your repository. The green color represents the branch that is currently checked out.

8. Now navigate to the file `~\Views\Home\Index.cshtml` and replace the contents with the text below.

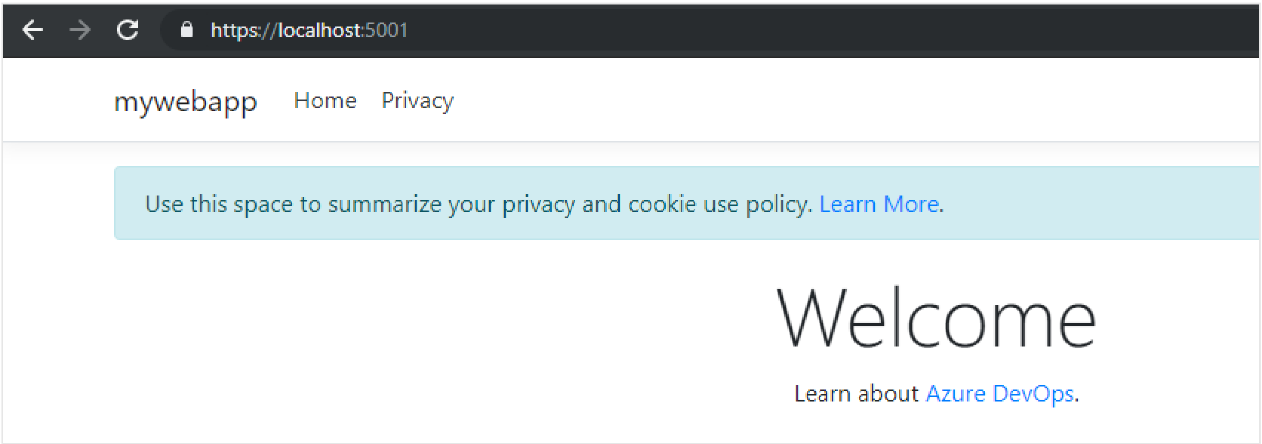
C#

```
@{
    ViewData["Title"] = "Home Page";
}
```

```
}

<div class="text-center">
    <h1 class="display-4">Welcome</h1>
    <p>Learn about <a
href="https://azure.microsoft.com/services/devops/">Azure DevOps</a>.</p>
</div>
```

9. Refresh the web app in the browser to see the changes.



Use this space to summarize your privacy and cookie use policy. Learn More.

Welcome

Learn about Azure DevOps.

10. In the context of the git repository, execute the following commands. These commands will stage the changes in the branch and then commit them.

Cmd

```
git status

git add .

git commit -m "updated welcome page."

git status
```

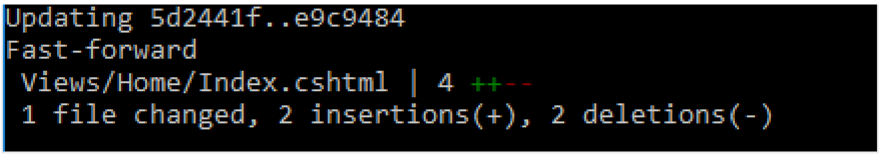
11. To merge the changes from the feature-devops-home-page into main, run the following commands in the context of the git repository.

Cmd

```

git checkout main

git merge feature-devops-home-page
```



```
Updating 5d2441f..e9c9484
Fast-forward
 Views/Home/Index.cshtml | 4 ++--
 1 file changed, 2 insertions(+), 2 deletions(-)
```

12. Run the below command to delete the feature branch.

Cmd

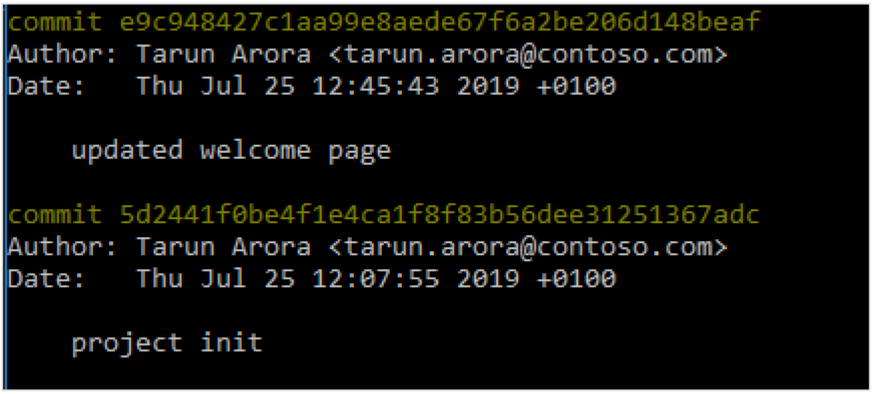
```
git branch --delete feature-devops-home-page
```

How it works

The easiest way to understand the outcome of the steps done earlier is to check the history of the operation. Let us have a look at how to do it.

1. In Git, committing changes to a repository is a two-step process. Running: `add .` The changes are staged but not committed. Finally, running commit promotes the staged changes into the repository.

2. To see the history of changes in the main branch, run the command `git log -v`



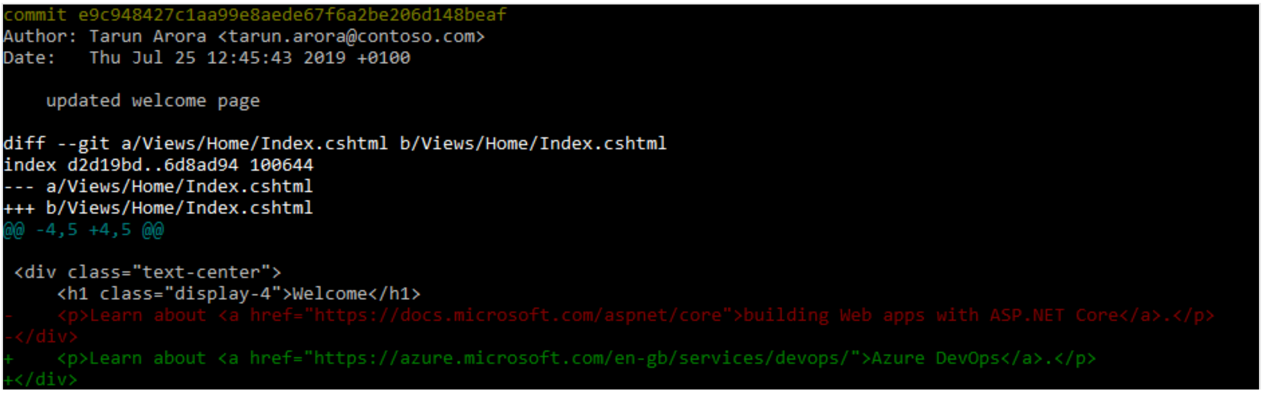
```
commit e9c948427c1aa99e8aede67f6a2be206d148beaf
Author: Tarun Arora <tarun.arora@contoso.com>
Date:   Thu Jul 25 12:45:43 2019 +0100

    updated welcome page

commit 5d2441f0be4f1e4ca1f8f83b56dee31251367adc
Author: Tarun Arora <tarun.arora@contoso.com>
Date:   Thu Jul 25 12:07:55 2019 +0100

    project init
```

3. To investigate the actual changes in the commit, you can run the command `git log -p`



```
commit e9c948427c1aa99e8aede67f6a2be206d148beaf
Author: Tarun Arora <tarun.arora@contoso.com>
Date:   Thu Jul 25 12:45:43 2019 +0100

    updated welcome page

diff --git a/Views/Home/Index.cshtml b/Views/Home/Index.cshtml
index d2d19bd..6d8ad94 100644
--- a/Views/Home/Index.cshtml
+++ b/Views/Home/Index.cshtml
@@ -4,5 +4,5 @@

 <div class="text-center">
     <h1 class="display-4">Welcome</h1>
-    <p>Learn about <a href="https://docs.microsoft.com/aspnet/core">building Web apps with ASP.NET Core</a>.</p>
-</div>
+    <p>Learn about <a href="https://azure.microsoft.com/en-gb/services/devops/">Azure DevOps</a>.</p>
+</div>
```

```
\ No newline at end of file
```

There is more

Git makes it easy to back out changes. Following our example, if you want to take out the changes made to the welcome page.

You can do It hard resetting the main branch to a previous version of the commit using the following command.

Cmd

```
git reset --hard 5d2441f0be4f1e4ca1f8f83b56dee31251367adc
```

Running the above command would reset the branch to the project init change.

If you run `git log -v`, you will see that the changes done to the welcome page are removed from the repository.

Next unit: Knowledge check

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆