AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Github Documentation

What Is Git?

Git is the most commonly used version control system. Git tracks the changes you make to files, so you have a record of what has been done, and you can revert to specific versions should you ever need to. Git also makes collaboration easier, allowing changes by multiple people to all be merged into one source.

Git is a software that runs locally. Your files and their history are stored on your computer. You can also use online hosts (such as **GitHub** or **Bitbucket**) to store a copy of the files and their revision history. Having a centrally located place where you can upload your changes and download changes from others, enable you to collaborate more easily with other developers.

What Is GitHub?

GitHub is designed as a **Git** repository hosting service. And what exactly is a Git repository hosting service? It’s an online database that allows you to keep track of and share your Git version control projects outside of your local computer/server. Unlike Git, GitHub is exclusively cloud-based. Through **GitHub**, you can share your code with others, giving them the power to make revisions or edits on your various **Git branches**. This makes it possible for entire teams to coordinate together on single projects in real-time. As changes are introduced, new branches are created, allowing the team to continue to revise the code without overwriting each other's work. These branches are like copies, and changes made on them do not reflect in the main directories on other users’ machines unless users choose to push/pull the changes to incorporate them.

Setting Up Git

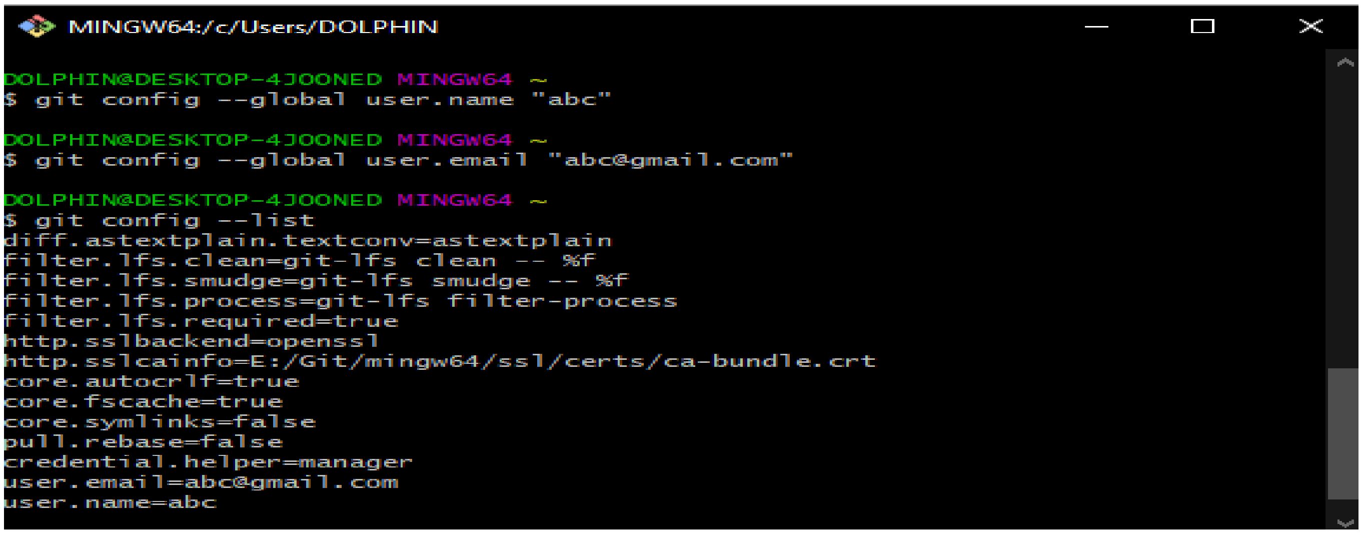
When using Git for the first time it can be a good idea to setup Git with the correct user information.

Open a terminal (or **Git Bash** (Windows)):

$ git config --global user.name" Your name"

$ git config --global user.email "your_email@example.com"

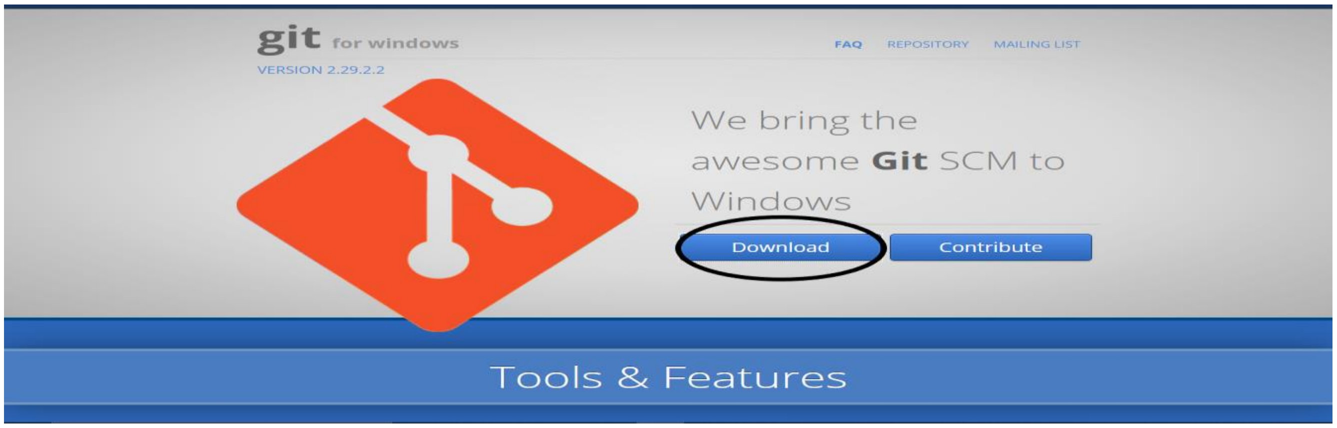
Check using: **git config –list**



```
DOLPHIN@DESKTOP-4JOONED MINGW64 ~
$ git config --global user.name "abc"

DOLPHIN@DESKTOP-4JOONED MINGW64 ~
$ git config --global user.email "abc@gmail.com"

DOLPHIN@DESKTOP-4JOONED MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=E:/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
user.email=abc@gmail.com
user.name=abc
```

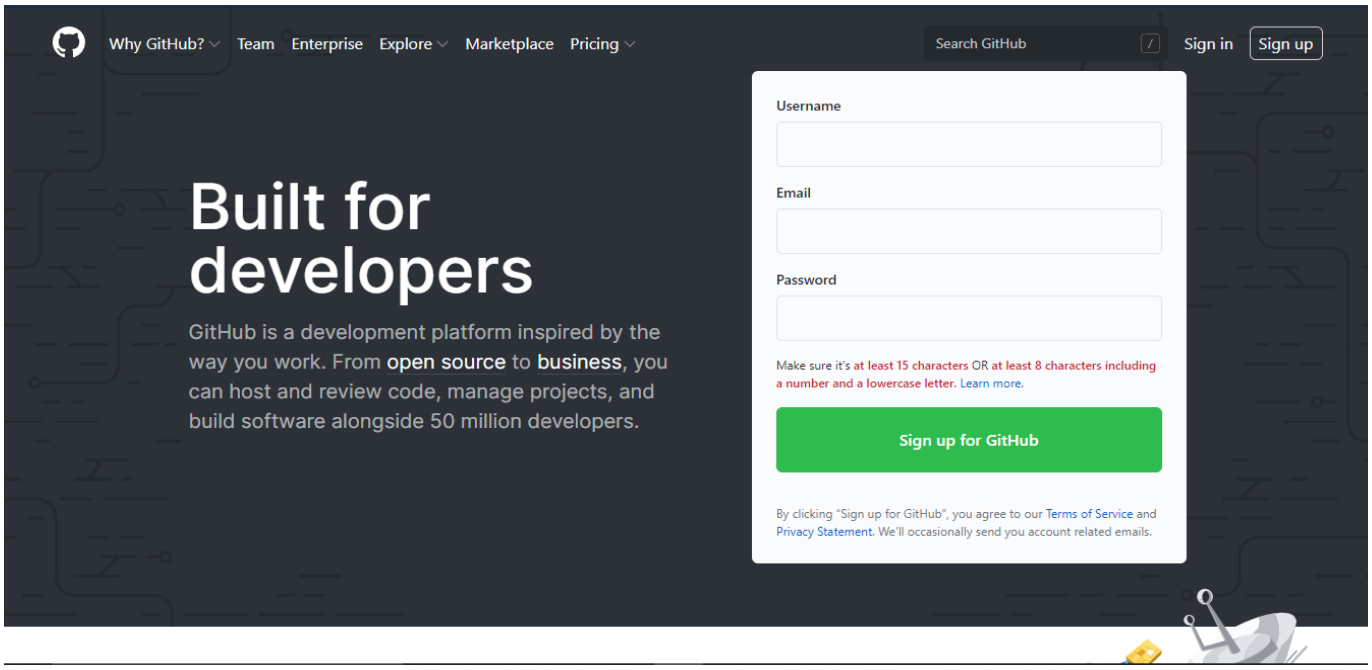
Git Bash

For using **Git Bash**(Windows), go to https://gitforwindows.org/ and download the latest version of **Git**.



Creating GitHub Repository

Go to https://github.com/ and sign up for **GitHub.**



Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 50 million developers.

Username

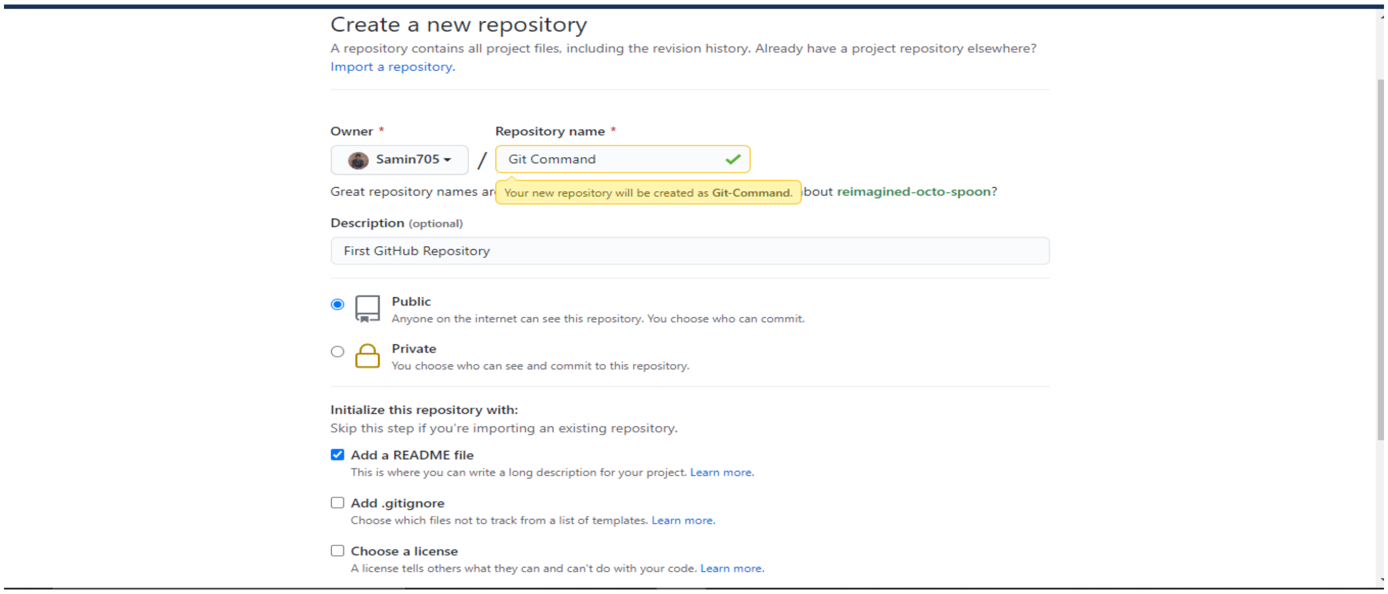
Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.

By clicking "Sign up for GitHub", you agree to our Terms of Service and Privacy Statement. We'll occasionally send you account related emails.

After successfully creating **GitHub** account, login to your account and click on **New** for creating **GitHub Repository.** Then fill out the necessary blanks.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * Samin705 / Repository name * Git Command

Great repository names ar Your new repository will be created as Git-Command. bout reimagined-octo-spoon?

Description (optional)

First GitHub Repository

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

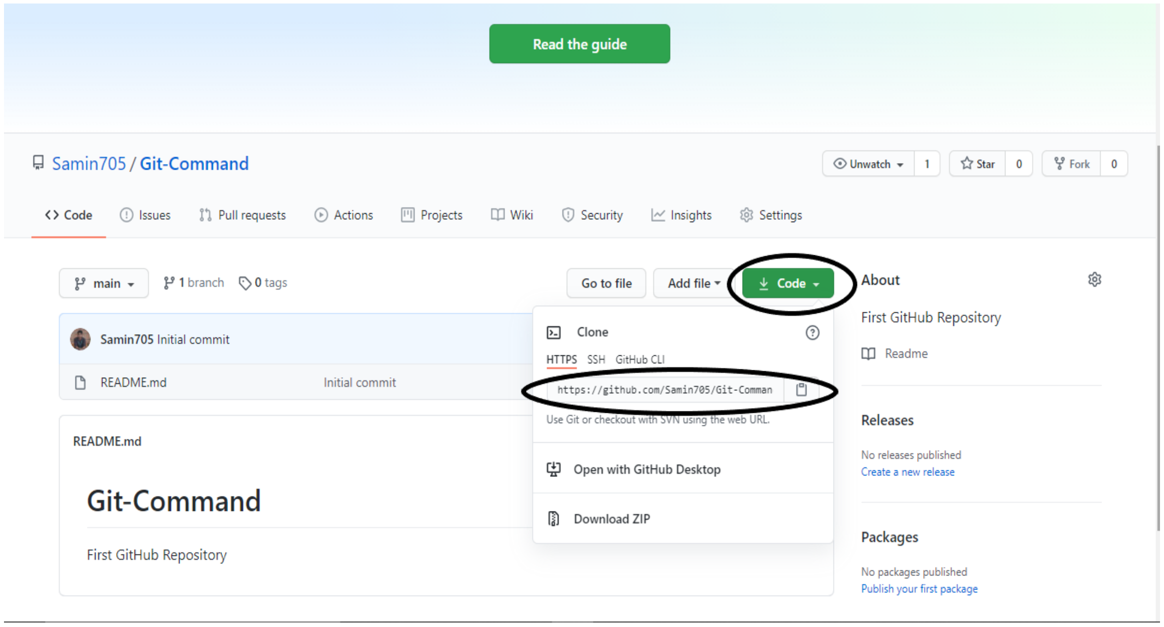
☑ Add a README file
This is where you can write a long description for your project. Learn more.

☐ Add .gitignore
Choose which files not to track from a list of templates. Learn more.

☐ Choose a license
A license tells others what they can and can't do with your code. Learn more.

Cloning Repository

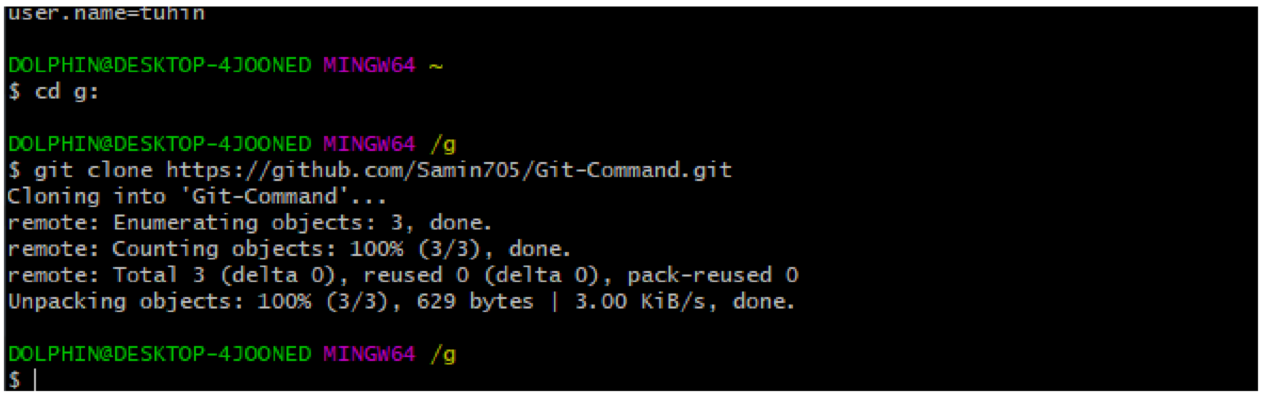
To get the copy of your repository in your desktop, click on **Code** and copy the **URL** https://github.com/<" your github user name"> /Git-Command.git



Then open the **Git Bash** terminal and write some commands:

cd <working-directory>

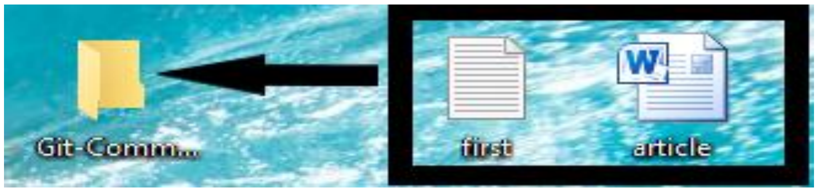
git clone https://github.com/<"your github user name"> /Git-Command.git



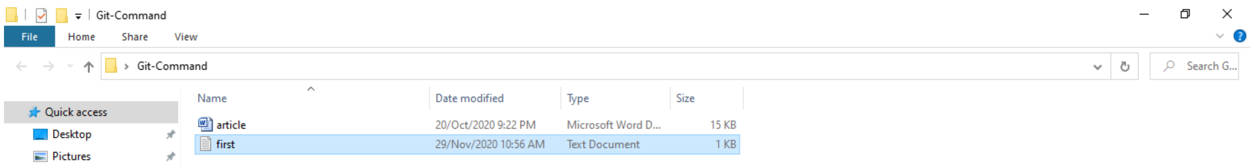
```
user.name=tuhin

DOLPHIN@DESKTOP-4JOONED MINGW64 ~
$ cd g:

DOLPHIN@DESKTOP-4JOONED MINGW64 /g
$ git clone https://github.com/Samin705/Git-Command.git
Cloning into 'Git-Command'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 629 bytes | 3.00 KiB/s, done.

DOLPHIN@DESKTOP-4JOONED MINGW64 /g
$ |
```

Add changes into Git Repository

As we successfully created a clone folder of our repository into our local computer. Now our task is to add our work into the repository. For example, we have a text file (**first.txt**) and a word file (**article.docx**). We want to add this work into our Git Hub account at (**Git Command**) repositories.

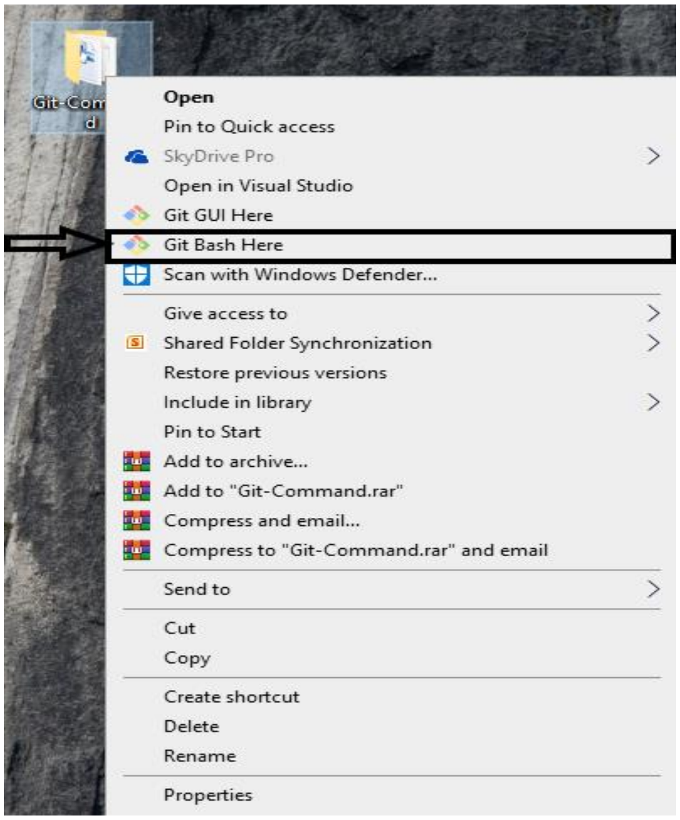


Now, add both folder into the clone **Git Command** folder into our local computer.

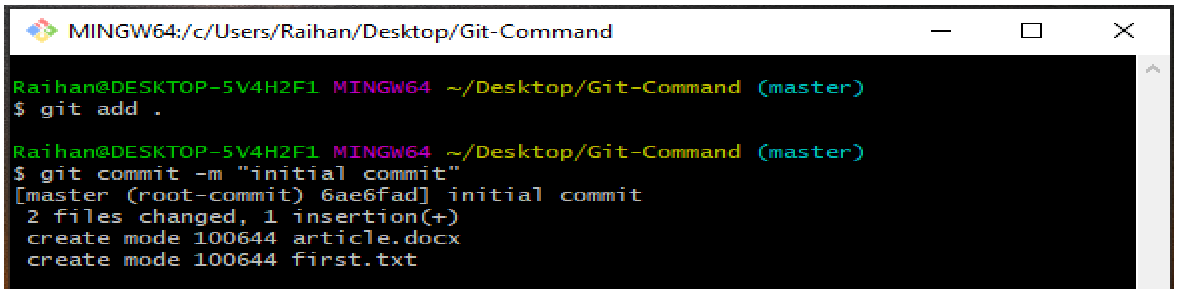


Name	Date modified	Type	Size
article	20/Oct/2020 9:22 PM	Microsoft Word D...	15 KB
first	29/Nov/2020 10:56 AM	Text Document	1 KB

After adding both file start git bash command prompt with Git Command folder.

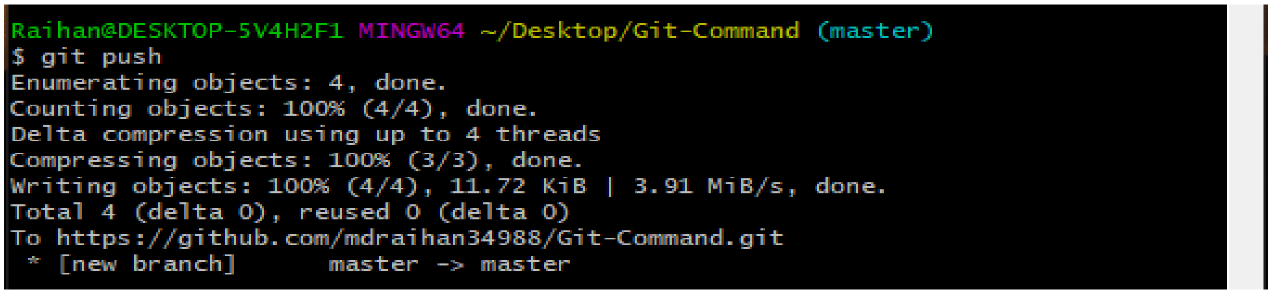


After Starting the Git bash command prompt we have to provide ($ **git add .**) command to add all the files and folder into **Git Command** folder. Then we have to commit our tasks with a message ($ **git commit –m “your commit message”**).



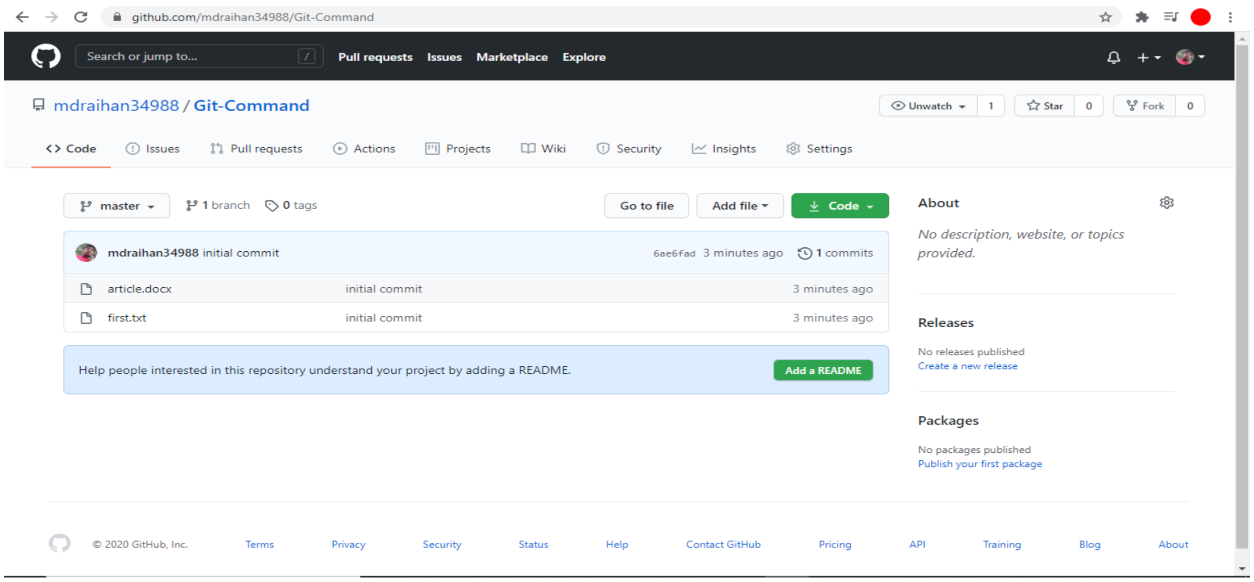
```
MINGW64:/c/Users/Raihan/Desktop/Git-Command

Raihan@DESKTOP-5V4H2F1 MINGW64 ~/Desktop/Git-Command (master)
$ git add .

Raihan@DESKTOP-5V4H2F1 MINGW64 ~/Desktop/Git-Command (master)
$ git commit -m "initial commit"
[master (root-commit) 6ae6fad] initial commit
 2 files changed, 1 insertion(+)
 create mode 100644 article.docx
 create mode 100644 first.txt
```

Still, the files are not added into our git hub account online repositories. Finally, we have to give push command to successfully upload both files into our github account online repositories ($ **git push / $ git push –u origin master**).



```
Raihan@DESKTOP-5V4H2F1 MINGW64 ~/Desktop/Git-Command (master)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 11.72 KiB | 3.91 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/mdraihan34988/Git-Command.git
 * [new branch]      master -> master
```

Now, we can see that both files are successfully added into our online github repository.



mdraihan34988 initial commit 6ae6fad 3 minutes ago 1 commits

article.docx	initial commit	3 minutes ago
first.txt	initial commit	3 minutes ago

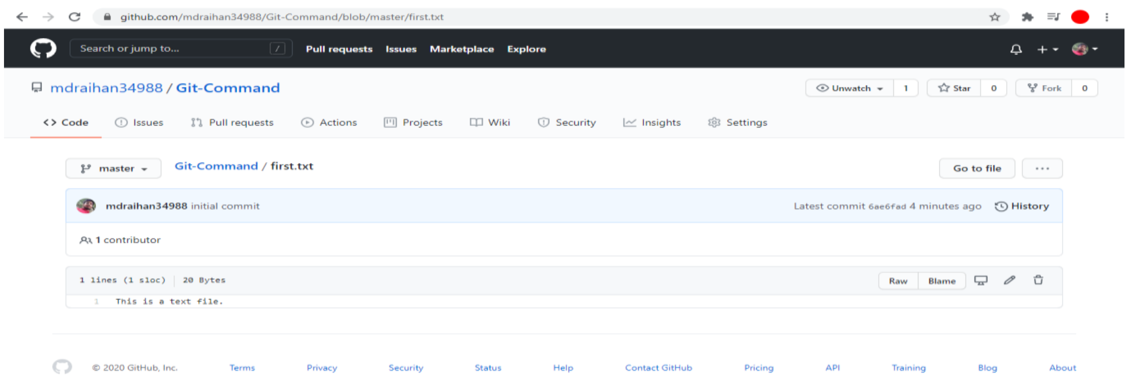
Help people interested in this repository understand your project by adding a README.

No description, website, or topics provided.

No releases published
Create a new release

No packages published
Publish your first package

Now, we can see our **first.txt** file content**.**

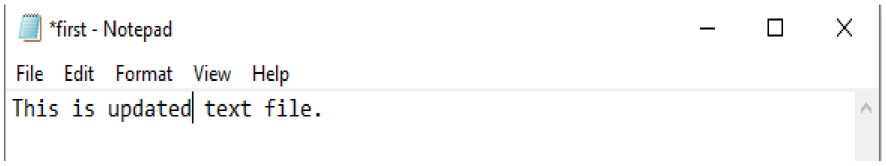


mdraihan34988 initial commit   Latest commit 6ae6fad 4 minutes ago

1 contributor

```
1 lines (1 sloc) | 20 Bytes
This is a text file.
```

Now, make change into that first.txt file. We have also make the changes into our online git repository. To, make this changes we have to again open the Git Command folder with Git Bash command prompt.



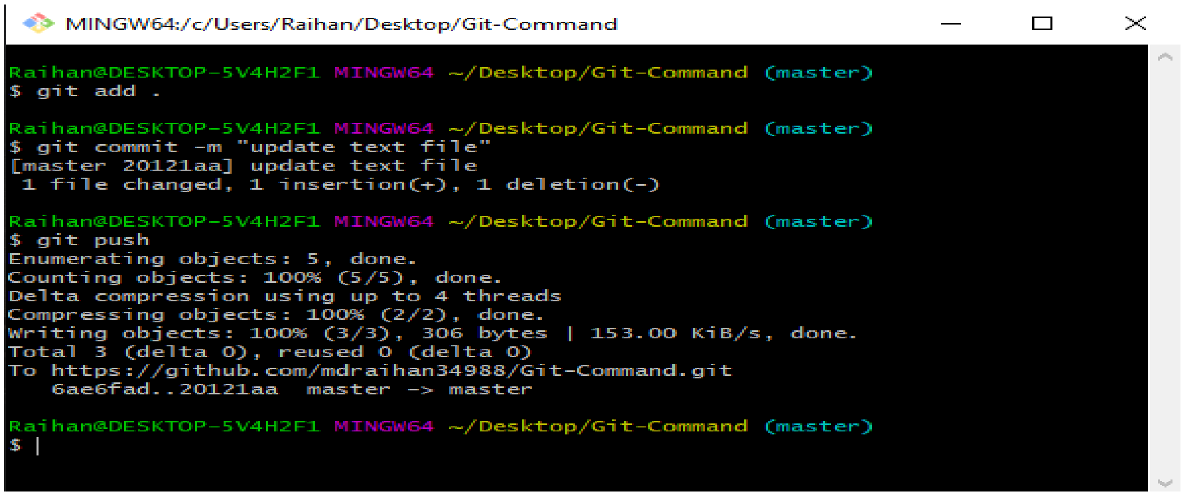
```
This is updated text file.
```

After opening the command prompt we have to give following command as we used before.

$ git add .

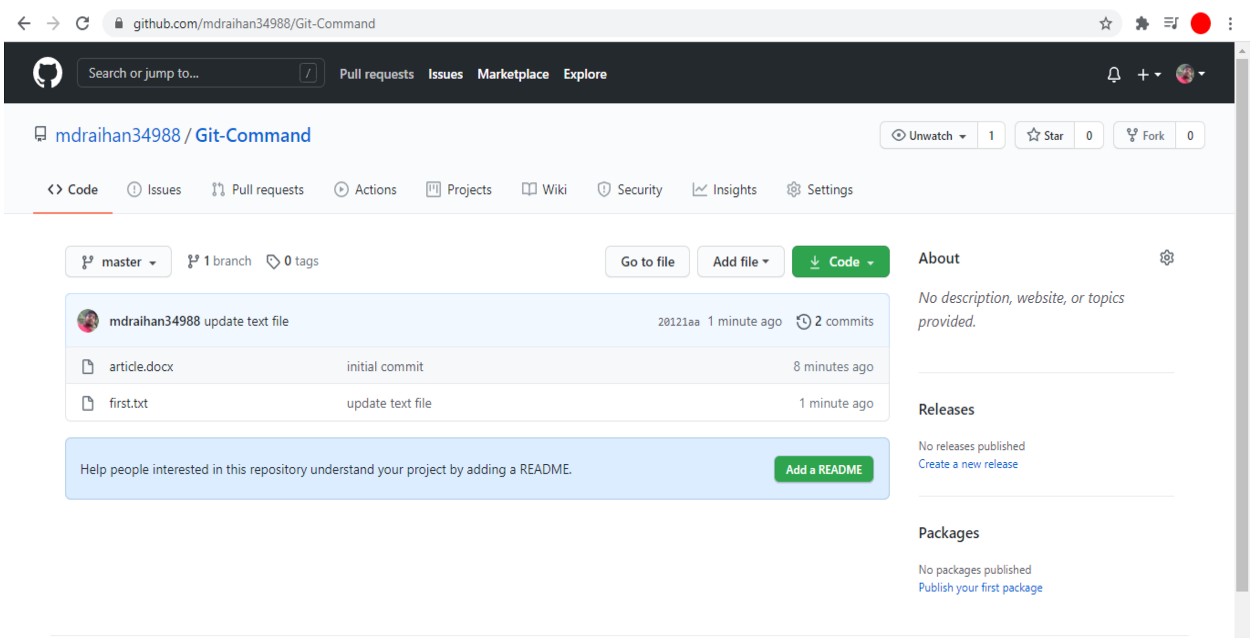
$ git commit –m "your commit message"

$ git push



```
Raihan@DESKTOP-5V4H2F1 MINGW64 ~/Desktop/Git-Command (master)
$ git add .

Raihan@DESKTOP-5V4H2F1 MINGW64 ~/Desktop/Git-Command (master)
$ git commit -m "update text file"
[master 20121aa] update text file
 1 file changed, 1 insertion(+), 1 deletion(-)

Raihan@DESKTOP-5V4H2F1 MINGW64 ~/Desktop/Git-Command (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 306 bytes | 153.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/mdraihan34988/Git-Command.git
   6ae6fad..20121aa  master -> master

Raihan@DESKTOP-5V4H2F1 MINGW64 ~/Desktop/Git-Command (master)
$ |
```

The changes now appears into our github online repository. The commit for first.txt is changed and the content also changed**.**


github.com/mdraihan34988/Git-Command

Search or jump to... Pull requests Issues Marketplace Explore

mdraihan34988 / Git-Command

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags Go to file Add file Code

mdraihan34988 update text file 20121aa 1 minute ago 2 commits

article.docx	initial commit	8 minutes ago
first.txt	update text file	1 minute ago

Help people interested in this repository understand your project by adding a README. Add a README

About

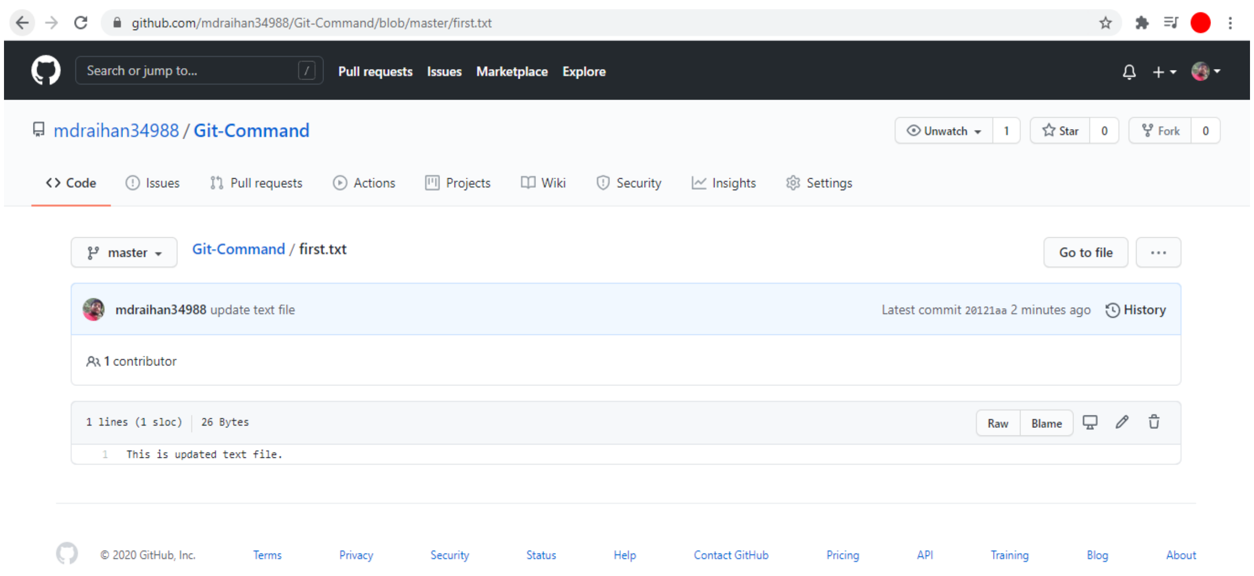
No description, website, or topics provided.

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package


github.com/mdraihan34988/Git-Command/blob/master/first.txt

Search or jump to... Pull requests Issues Marketplace Explore

mdraihan34988 / Git-Command

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master Git-Command / first.txt Go to file

mdraihan34988 update text file Latest commit 20121aa 2 minutes ago History

1 contributor

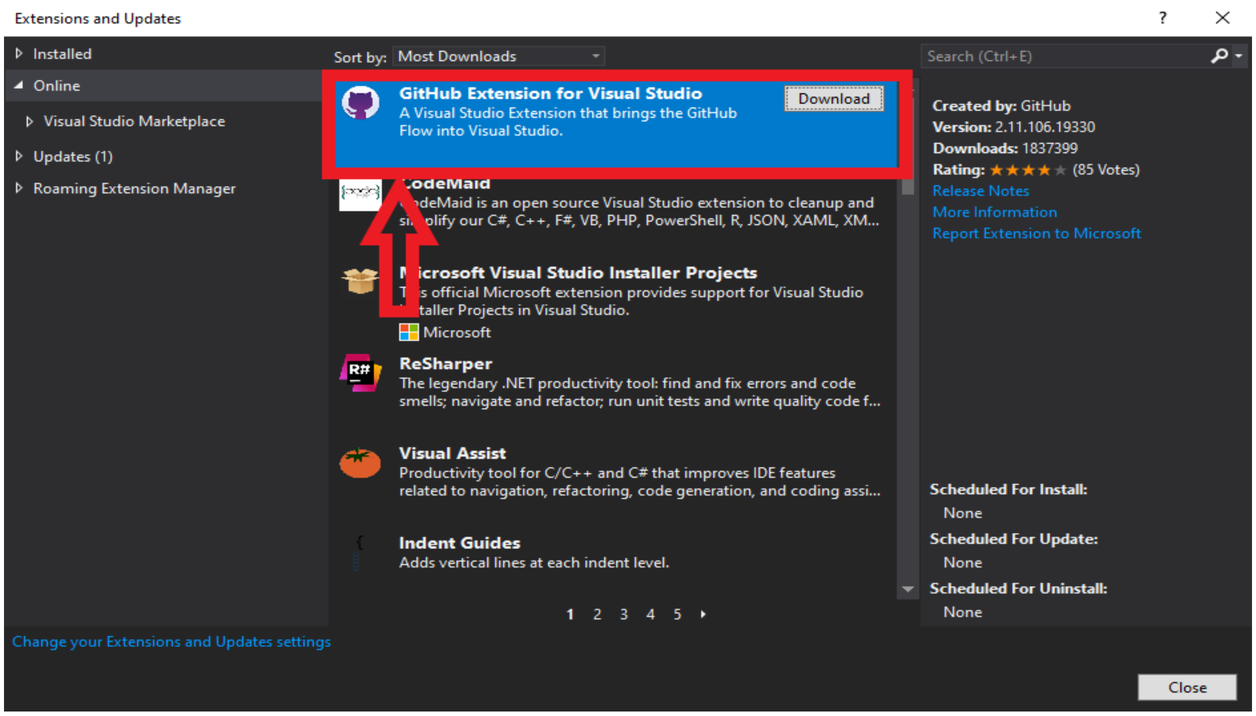
1 lines (1 sloc) | 26 Bytes Raw Blame

```
This is updated text file.
```

© 2020 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

Github with Visual Studio 2017

To connect github account with Visual studio 2017 we need to add an extension with visual studio. **Open visual studio –> Tools –> Extensions and Updates –> Online –> Search Github –> Download Github Extension for visual studio.**



Extensions and Updates

- Installed
- Online
  - Visual Studio Marketplace
- Updates (1)
- Roaming Extension Manager

Sort by: Most Downloads

GitHub Extension for Visual Studio — Download
A Visual Studio Extension that brings the GitHub Flow into Visual Studio.

CodeMaid
CodeMaid is an open source Visual Studio extension to cleanup and simplify our C#, C++, F#, VB, PHP, PowerShell, R, JSON, XAML, XM...

Microsoft Visual Studio Installer Projects
This official Microsoft extension provides support for Visual Studio Installer Projects in Visual Studio.
Microsoft

ReSharper
The legendary .NET productivity tool: find and fix errors and code smells; navigate and refactor; run unit tests and write quality code f...

Visual Assist
Productivity tool for C/C++ and C# that improves IDE features related to navigation, refactoring, code generation, and coding assi...

Indent Guides
Adds vertical lines at each indent level.

1 2 3 4 5

Search (Ctrl+E)

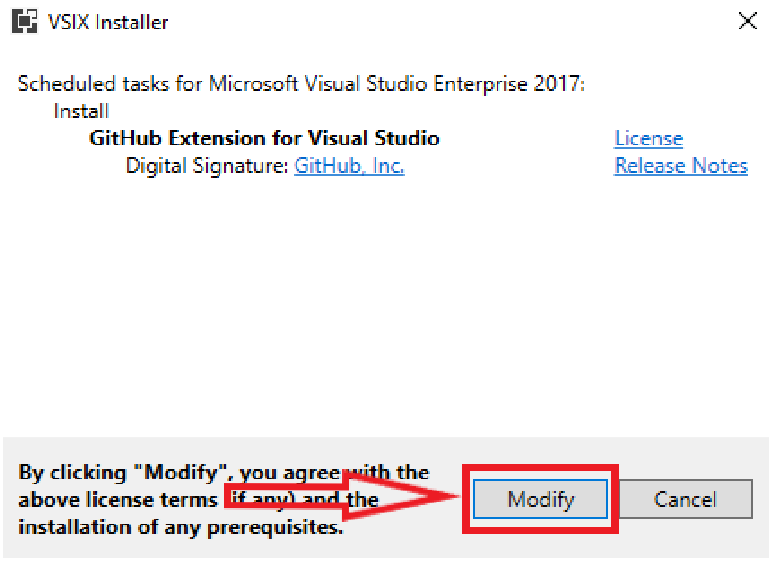
Created by: GitHub
Version: 2.11.106.19330
Downloads: 1837399
Rating: ★★★★★ (85 Votes)
Release Notes
More Information
Report Extension to Microsoft

Scheduled For Install:
None
Scheduled For Update:
None
Scheduled For Uninstall:
None

Change your Extensions and Updates settings

Close

After download the extension close the visual studio and modify VSIX installer.



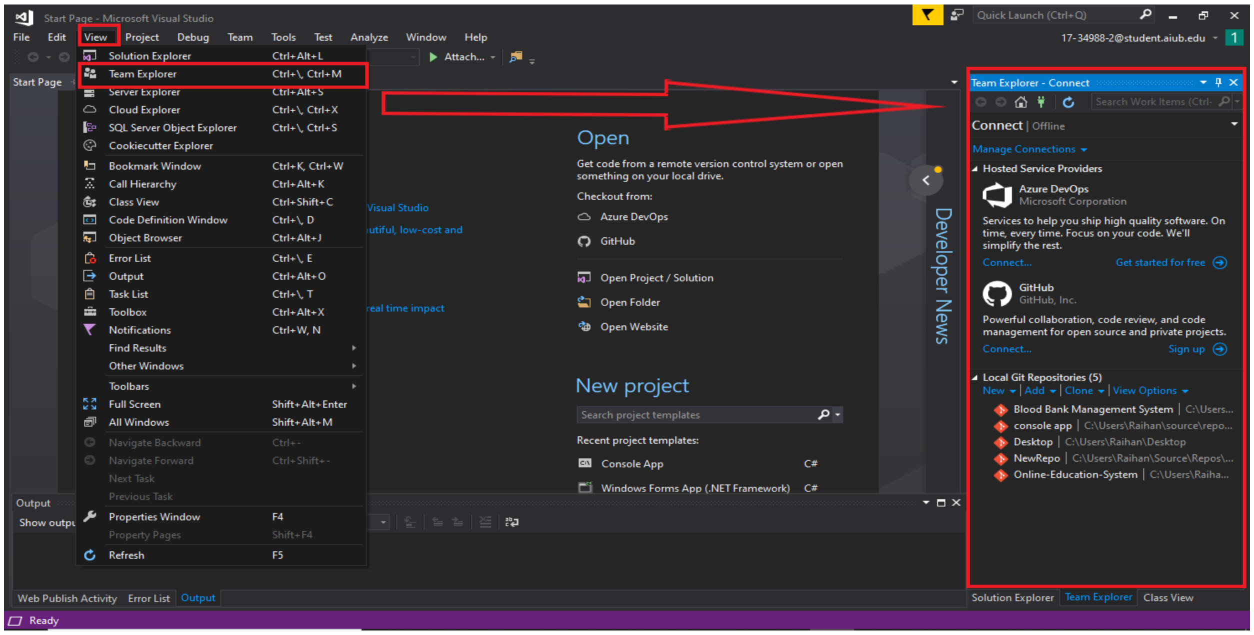
VSIX Installer

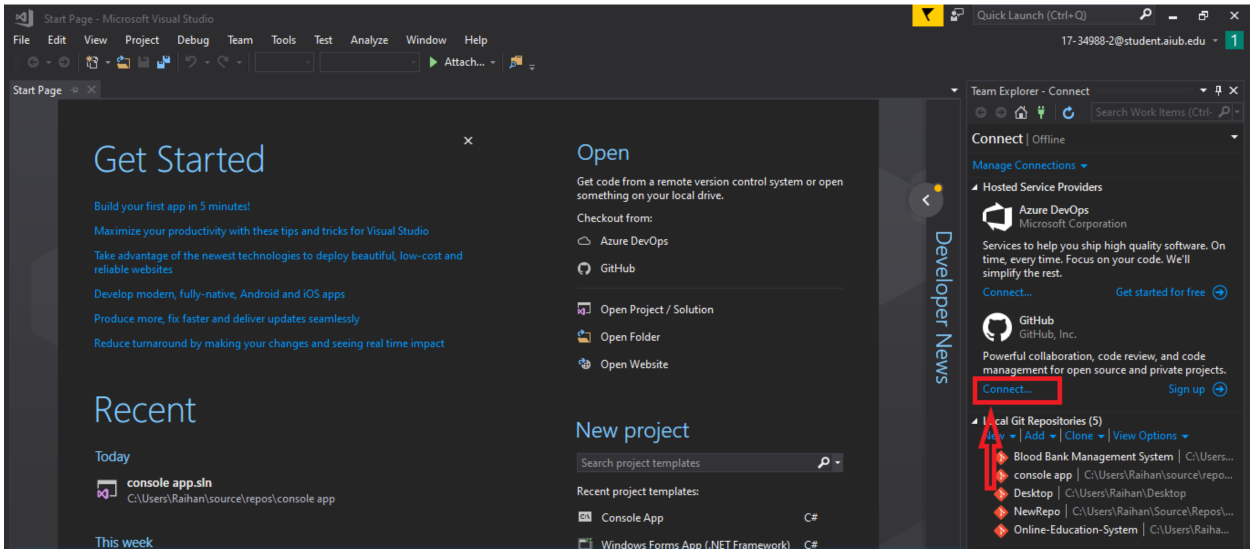
Scheduled tasks for Microsoft Visual Studio Enterprise 2017:
Install
GitHub Extension for Visual Studio — License
Digital Signature: GitHub, Inc. — Release Notes

By clicking "Modify", you agree with the above license terms (if any) and the installation of any prerequisites.

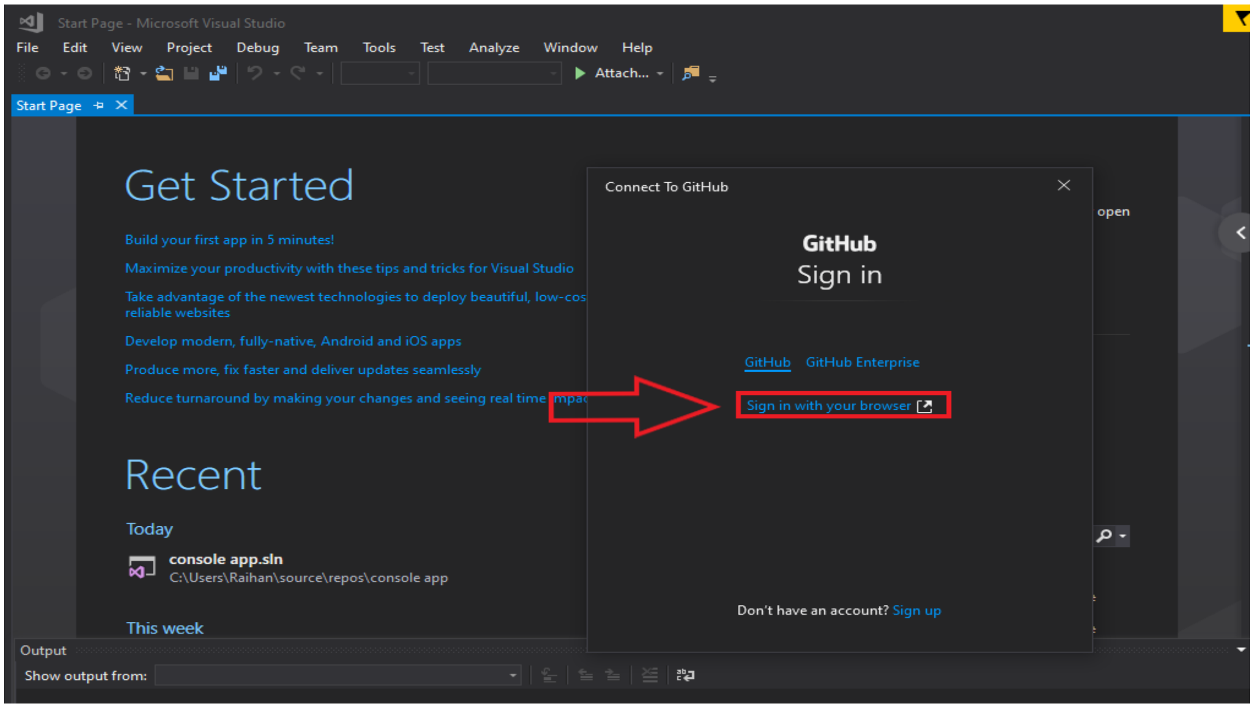
Modify | Cancel

After complete modifing the VSIX installer, **open visual studio -> View -> Team Explorer -> Github -> Connect.**

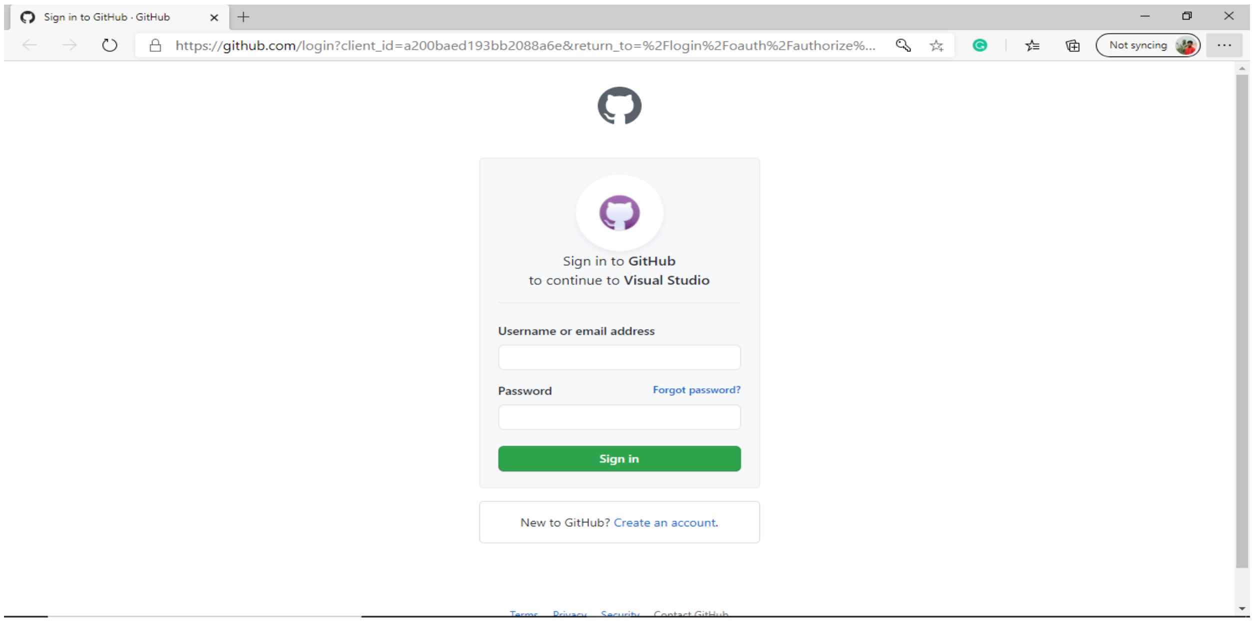




Open browser for sign in your github account.


Start Page - Microsoft Visual Studio
File Edit View Project Debug Team Tools Test Analyze Window Help
Attach...
Start Page
Get Started
Build your first app in 5 minutes!
Maximize your productivity with these tips and tricks for Visual Studio
Take advantage of the newest technologies to deploy beautiful, low-cos reliable websites
Develop modern, fully-native, Android and iOS apps
Produce more, fix faster and deliver updates seamlessly
Reduce turnaround by making your changes and seeing real time impac
Recent
Today
console app.sln
C:\Users\Raihan\source\repos\console app
This week
Output
Show output from:
Connect To GitHub
GitHub
Sign in
GitHub GitHub Enterprise
Sign in with your browser
Don't have an account? Sign up
open

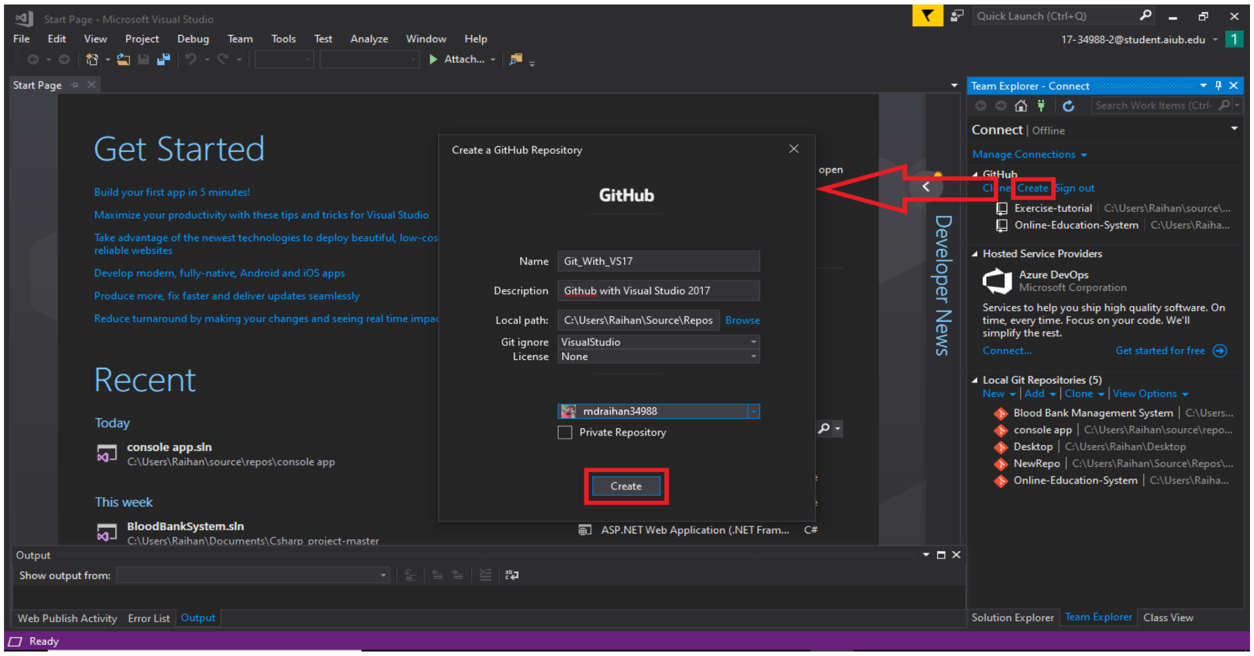
Sign in into your github account to connect with visual studio 2017.


Sign in to GitHub · GitHub
https://github.com/login?client_id=a200baed193bb2088a6e&return_to=%2Flogin%2Foauth%2Fauthorize%...
Not syncing
Sign in to GitHub
to continue to Visual Studio
Username or email address
Password
Forgot password?
Sign in
New to GitHub? Create an account.
Terms Privacy Security Contact GitHub

In this way, we successfully connect GitHub with our visual studio 2017.

Use Git repository with Visual Studio 2017

Now create a repository (**Git_With_VS17**) through visual studio 2017. It creates a local repository on local computer and a repository into GitHub online account.

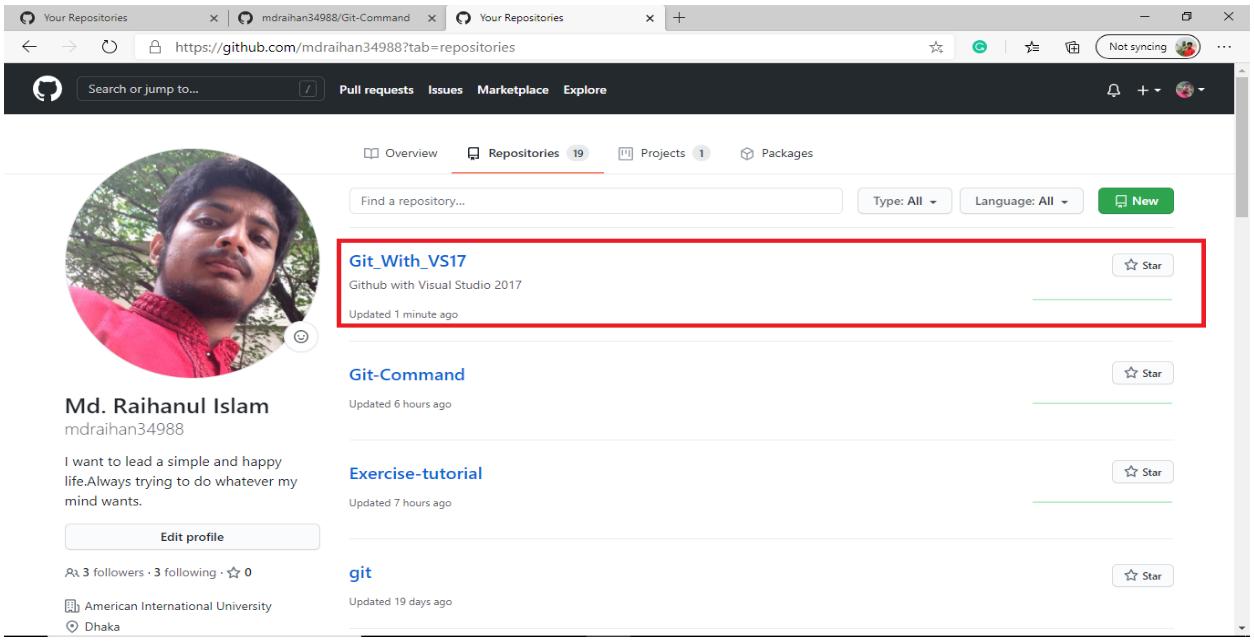


Create a GitHub Repository

GitHub

Name: Git_With_VS17
Description: Github with Visual Studio 2017
Local path: C:\Users\Raihan\Source\Repos Browse
Git ignore: VisualStudio
License: None
mdraihan34988
Private Repository
Create

Team Explorer - Connect
Connect | Offline
Manage Connections
GitHub
Clone | Create | Sign out
Exercise-tutorial C:\Users\Raihan\source\...
Online-Education-System C:\Users\Raiha...
Hosted Service Providers
Azure DevOps
Microsoft Corporation
Services to help you ship high quality software. On time, every time. Focus on your code. We'll simplify the rest.
Connect... Get started for free
Local Git Repositories (5)
New | Add | Clone | View Options
Blood Bank Management System | C:\Users...
console app | C:\Users\Raihan\source\repo...
Desktop | C:\Users\Raihan\Desktop
NewRepo | C:\Users\Raihan\Source\Repos\...
Online-Education-System | C:\Users\Raiha...



https://github.com/mdraihan34988?tab=repositories

Md. Raihanul Islam
mdraihan34988
I want to lead a simple and happy life.Always trying to do whatever my mind wants.
3 followers · 3 following · 0
American International University
Dhaka

Repositories 19

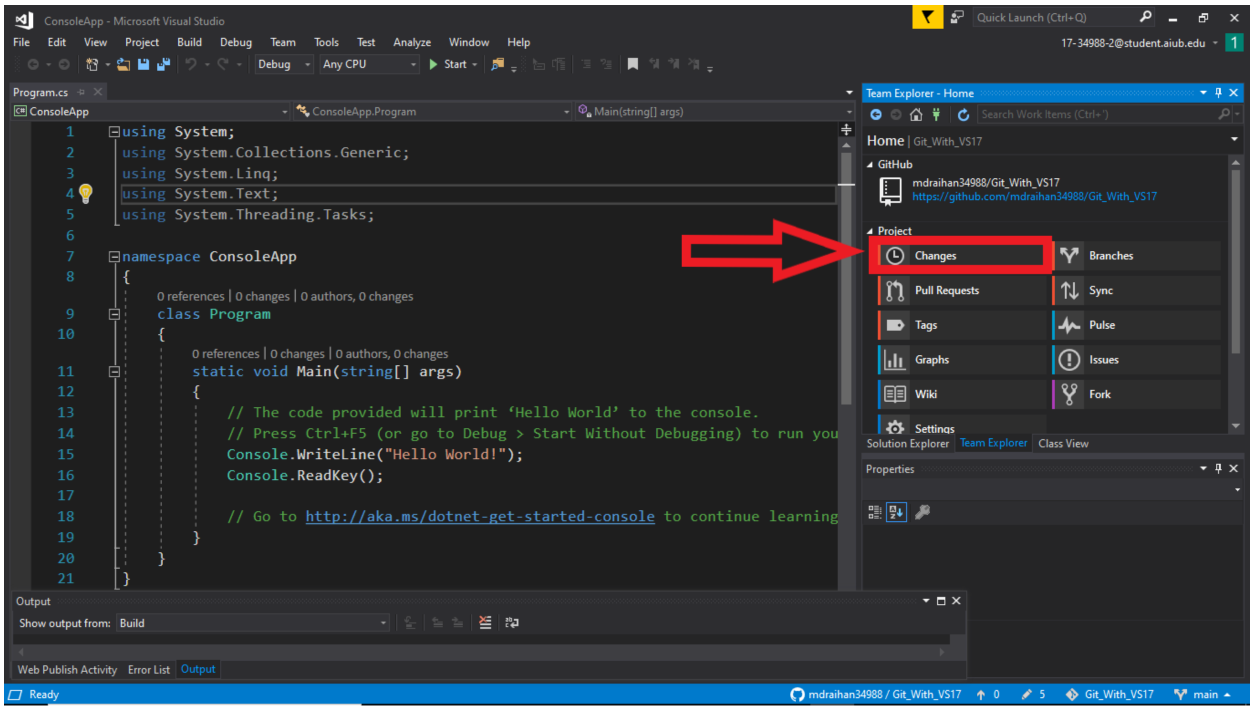
Git_With_VS17
Github with Visual Studio 2017
Updated 1 minute ago

Git-Command
Updated 6 hours ago

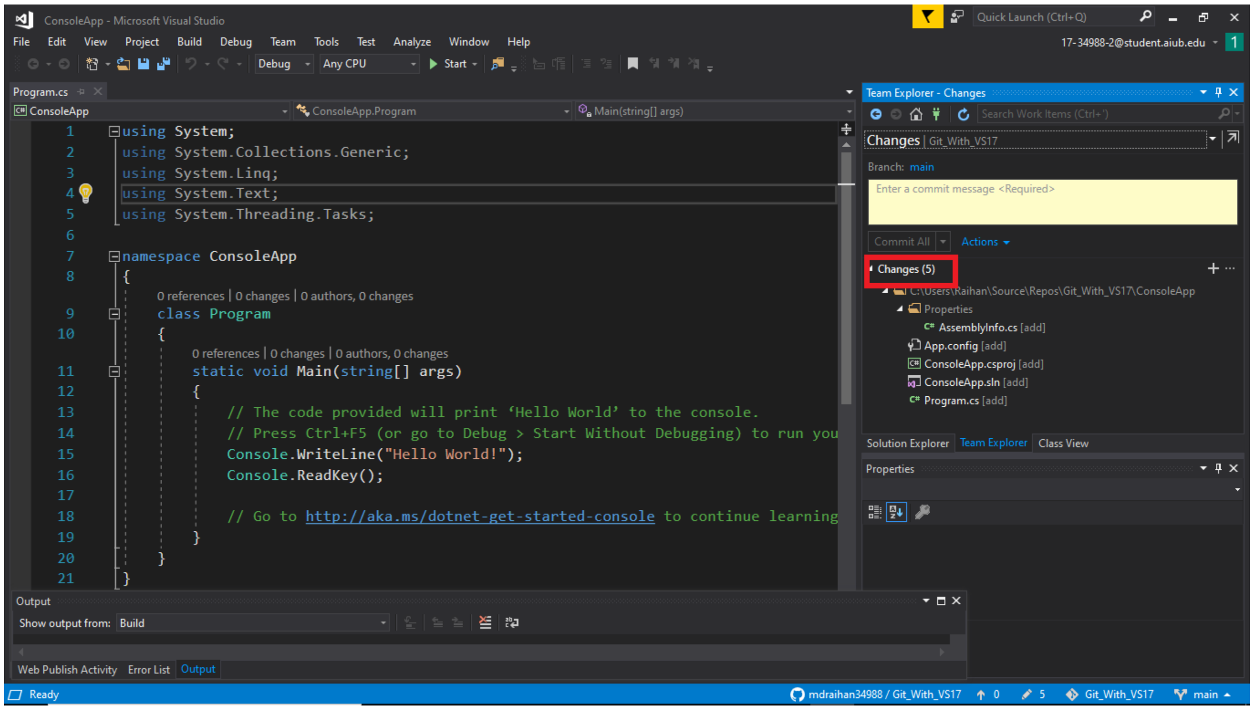
Exercise-tutorial
Updated 7 hours ago

git
Updated 19 days ago

Now create a project into local repository **Git_With_VS17** and see the number of changes into local repositories.



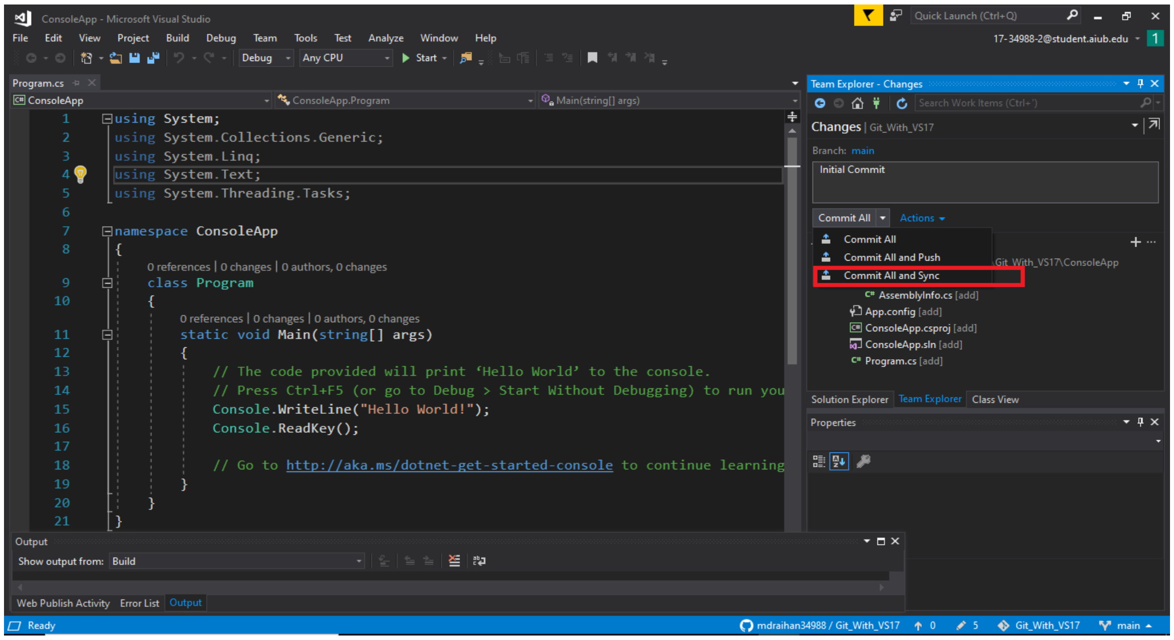
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp
{
    0 references | 0 changes | 0 authors, 0 changes
    class Program
    {
        0 references | 0 changes | 0 authors, 0 changes
        static void Main(string[] args)
        {
            // The code provided will print ‘Hello World’ to the console.
            // Press Ctrl+F5 (or go to Debug > Start Without Debugging) to run you
            Console.WriteLine("Hello World!");
            Console.ReadKey();

            // Go to http://aka.ms/dotnet-get-started-console to continue learning
        }
    }
}
```



```
C:\Users\Raihan\Source\Repos\Git_With_VS17\ConsoleApp
    Properties
        AssemblyInfo.cs [add]
    App.config [add]
    ConsoleApp.csproj [add]
    ConsoleApp.sln [add]
    Program.cs [add]
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp
{
    0 references | 0 changes | 0 authors, 0 changes
    class Program
    {
        0 references | 0 changes | 0 authors, 0 changes
        static void Main(string[] args)
        {
            // The code provided will print ‘Hello World’ to the console.
            // Press Ctrl+F5 (or go to Debug > Start Without Debugging) to run you
            Console.WriteLine("Hello World!");
            Console.ReadKey();

            // Go to http://aka.ms/dotnet-get-started-console to continue learning
        }
    }
}
```

Enter our commit message “Initial commit”. For commit we have 3 options. One is commit all which is used to commit the changes into our local repositories. Then commit all and push which is used to commit changes into our local repositories as well as push changes into git repositories. Finally, commit all and sync which is used commit changes into local repositories and also into the git repositories and pull down the changes into your github account repositories. To commit our changes we select Commit All and sync.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp
{
    0 references | 0 changes | 0 authors, 0 changes
    class Program
    {
        0 references | 0 changes | 0 authors, 0 changes
        static void Main(string[] args)
        {
            // The code provided will print ‘Hello World’ to the console.
            // Press Ctrl+F5 (or go to Debug > Start Without Debugging) to run you
            Console.WriteLine("Hello World!");
            Console.ReadKey();

            // Go to http://aka.ms/dotnet-get-started-console to continue learning
        }
    }
}
```

Team Explorer - Changes

Changes | Git_With_VS17

Branch: main

Initial Commit

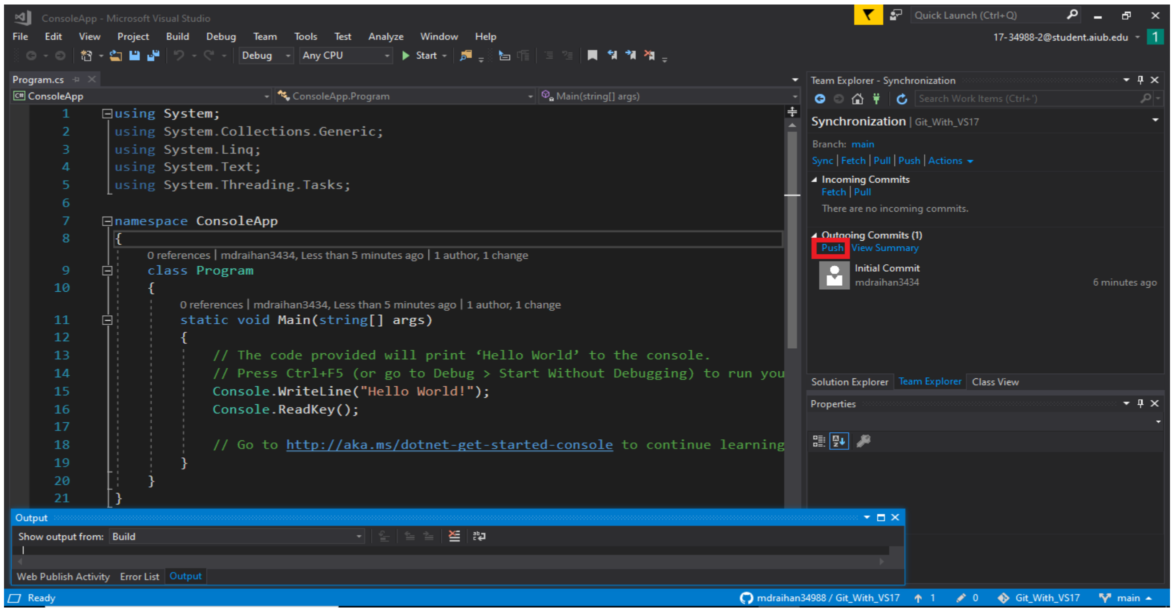
Commit All | Actions

- Commit All
- Commit All and Push
- Commit All and Sync

Git_With_VS17\ConsoleApp

- AssemblyInfo.cs [add]
- App.config [add]
- ConsoleApp.csproj [add]
- ConsoleApp.sln [add]
- Program.cs [add]

Now push changes into git repository.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp
{
    0 references | mdraihan3434, Less than 5 minutes ago | 1 author, 1 change
    class Program
    {
        0 references | mdraihan3434, Less than 5 minutes ago | 1 author, 1 change
        static void Main(string[] args)
        {
            // The code provided will print ‘Hello World’ to the console.
            // Press Ctrl+F5 (or go to Debug > Start Without Debugging) to run you
            Console.WriteLine("Hello World!");
            Console.ReadKey();

            // Go to http://aka.ms/dotnet-get-started-console to continue learning
        }
    }
}
```

Team Explorer - Synchronization

Synchronization | Git_With_VS17

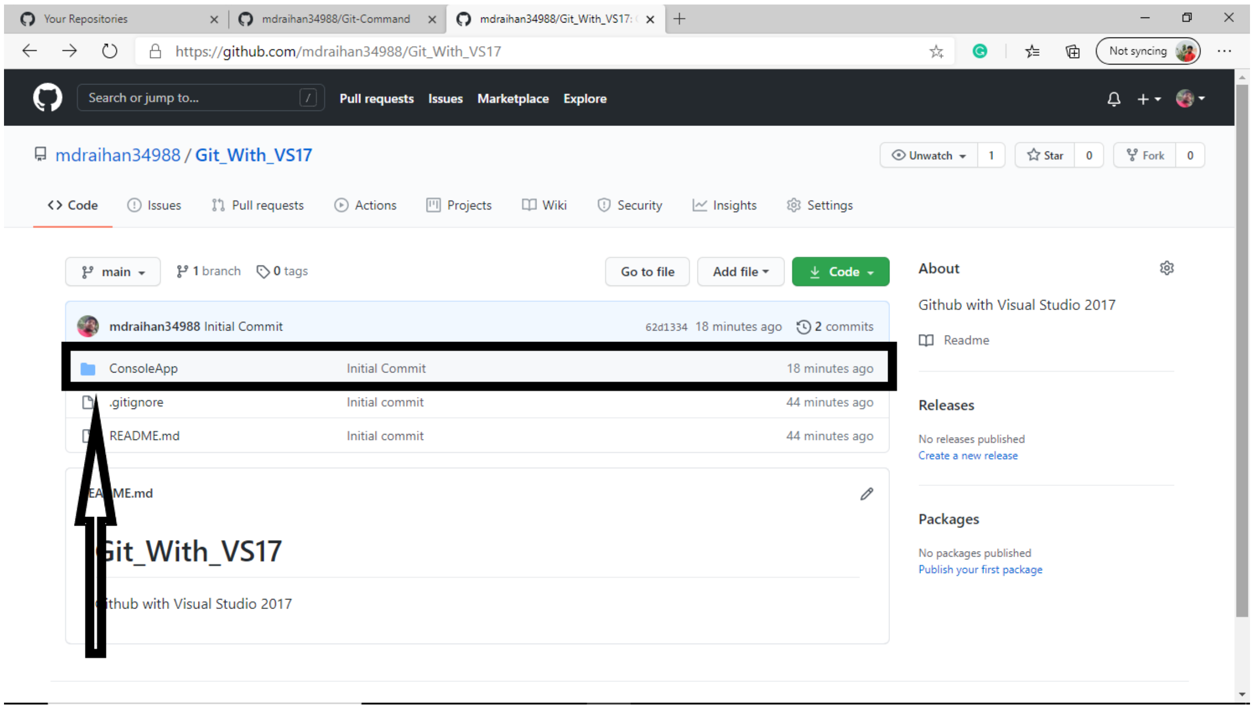
Branch: main

Sync | Fetch | Pull | Push | Actions

Incoming Commits
Fetch | Pull
There are no incoming commits.

Outgoing Commits (1)
Push | View Summary
Initial Commit
mdraihan3434 6 minutes ago

In this way, we successfully upload our project using visual studio.


https://github.com/mdraihan34988/Git_With_VS17

mdraihan34988 / Git_With_VS17

main · 1 branch · 0 tags

mdraihan34988 Initial Commit · 62d1334 18 minutes ago · 2 commits

ConsoleApp	Initial Commit	18 minutes ago
.gitignore	Initial commit	44 minutes ago
README.md	Initial commit	44 minutes ago

README.md

Git_With_VS17

Github with Visual Studio 2017

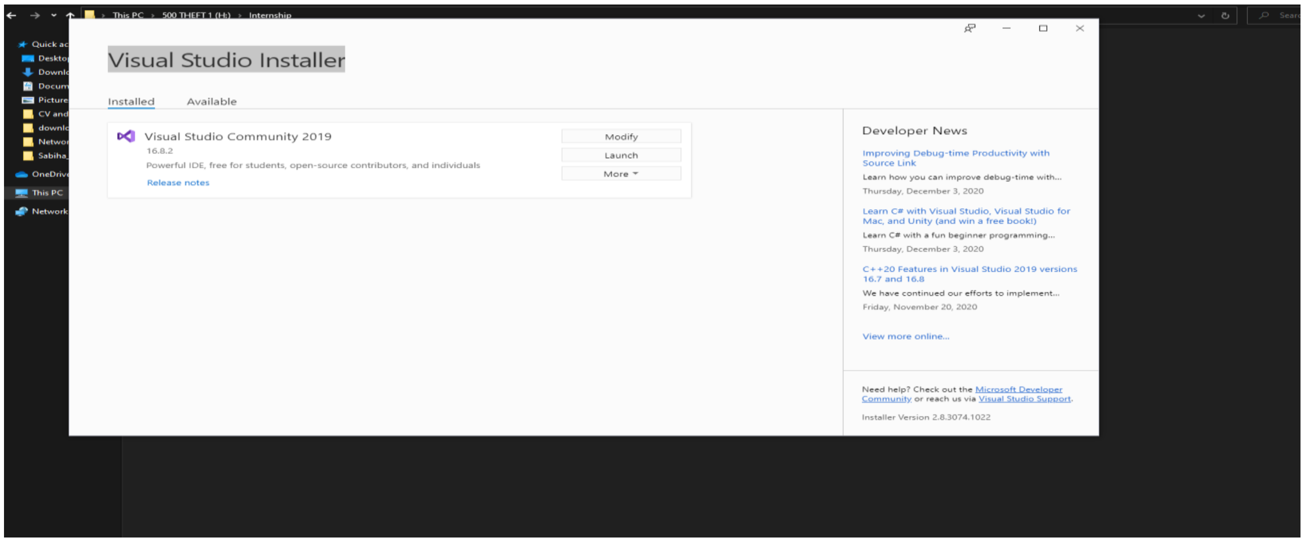
About: Github with Visual Studio 2017

Releases: No releases published

Packages: No packages published

Use Git repository with Visual Studio 2019

To add GitHub in VS 2019 you need to go to the **search bar** at task bar and type **Visual Studio Installer**. Once you do it a window will pop up. Click on the **Modify**


Visual Studio Installer

Installed · Available

Visual Studio Community 2019
16.8.2
Powerful IDE, free for students, open-source contributors, and individuals
Release notes

Modify · Launch · More

Developer News

Improving Debug-time Productivity with Source Link
Learn how you can improve debug-time with...
Thursday, December 3, 2020

Learn C# with Visual Studio, Visual Studio for Mac, and Unity (and win a free book!)
Learn C# with a fun beginner programming...
Thursday, December 3, 2020

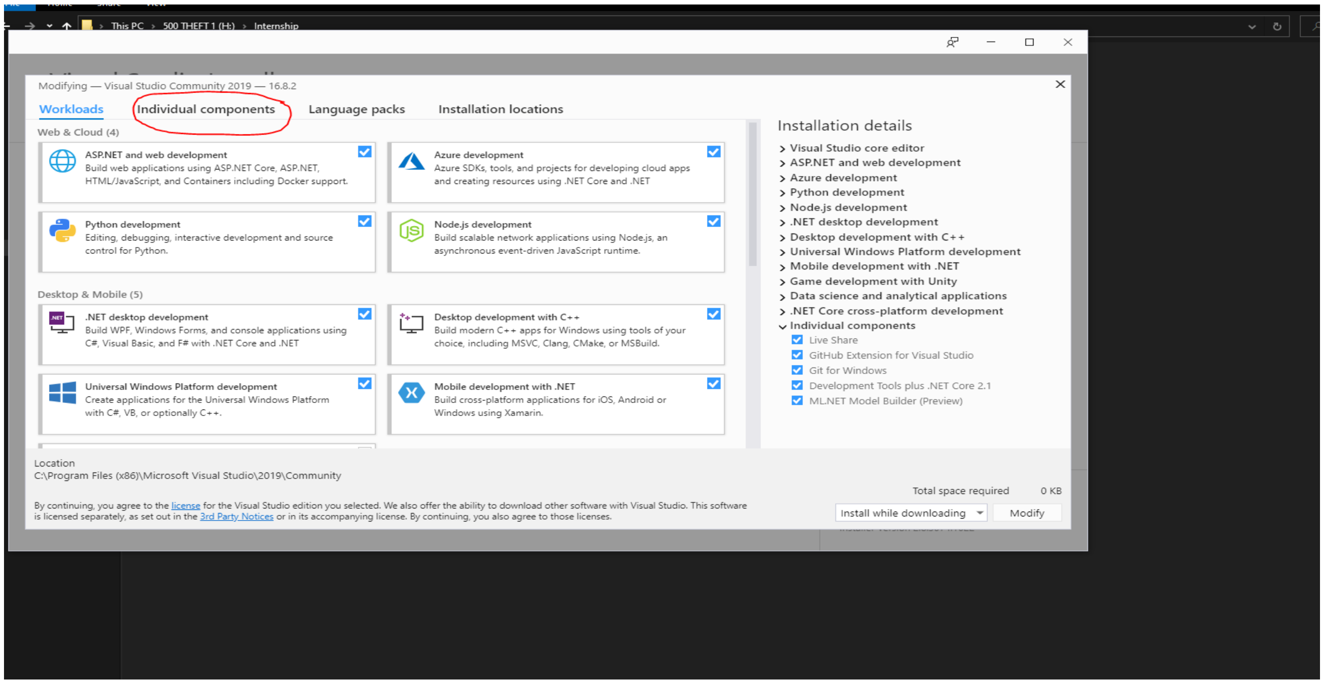
C++20 Features in Visual Studio 2019 versions 16.7 and 16.8
We have continued our efforts to implement...
Friday, November 20, 2020

View more online...

Need help? Check out the Microsoft Developer Community or reach us via Visual Studio Support.

Installer Version 2.8.3074.1022

Button. Click on the **Individual Components** to bring about a screen where the Visual Studio installer will show all extensions and components that can be added to VS 19.


This PC › 500 THEFT 1 (H:) › Internship
Modifying — Visual Studio Community 2019 — 16.8.2
Workloads
Individual components
Language packs
Installation locations

Web & Cloud (4)

- ☑ ASP.NET and web development — Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker support.
- ☑ Azure development — Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET Core and .NET
- ☑ Python development — Editing, debugging, interactive development and source control for Python.
- ☑ Node.js development — Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Desktop & Mobile (5)

- ☑ .NET desktop development — Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET Core and .NET
- ☑ Desktop development with C++ — Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.
- ☑ Universal Windows Platform development — Create applications for the Universal Windows Platform with C#, VB, or optionally C++.
- ☑ Mobile development with .NET — Build cross-platform applications for iOS, Android or Windows using Xamarin.

Installation details

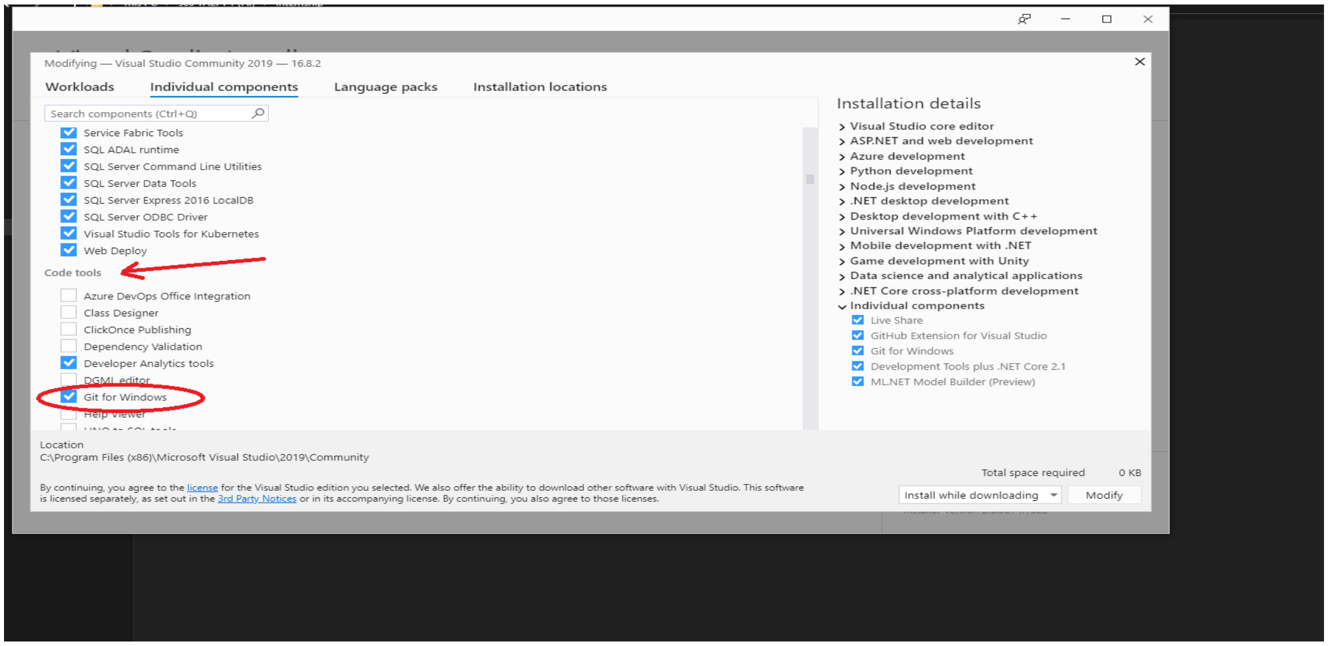
- › Visual Studio core editor
- › ASP.NET and web development
- › Azure development
- › Python development
- › Node.js development
- › .NET desktop development
- › Desktop development with C++
- › Universal Windows Platform development
- › Mobile development with .NET
- › Game development with Unity
- › Data science and analytical applications
- › .NET Core cross-platform development
- ⌄ Individual components
  - ☑ Live Share
  - ☑ GitHub Extension for Visual Studio
  - ☑ Git for Windows
  - ☑ Development Tools plus .NET Core 2.1
  - ☑ ML.NET Model Builder (Preview)

Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 0 KB
Install while downloading | Modify

From there go down to **Code Tools** section and select the **Git for Windows** and go down the **Uncategorized** and select **GitHub Extension for Visual Studio.**


Modifying — Visual Studio Community 2019 — 16.8.2
Workloads
Individual components
Language packs
Installation locations
Search components (Ctrl+Q)

- ☑ Service Fabric Tools
- ☑ SQL ADAL runtime
- ☑ SQL Server Command Line Utilities
- ☑ SQL Server Data Tools
- ☑ SQL Server Express 2016 LocalDB
- ☑ SQL Server ODBC Driver
- ☑ Visual Studio Tools for Kubernetes
- ☑ Web Deploy

Code tools

- ☐ Azure DevOps Office Integration
- ☐ Class Designer
- ☐ ClickOnce Publishing
- ☐ Dependency Validation
- ☑ Developer Analytics tools
- ☐ DGML editor
- ☑ Git for Windows
- ☐ Help Viewer

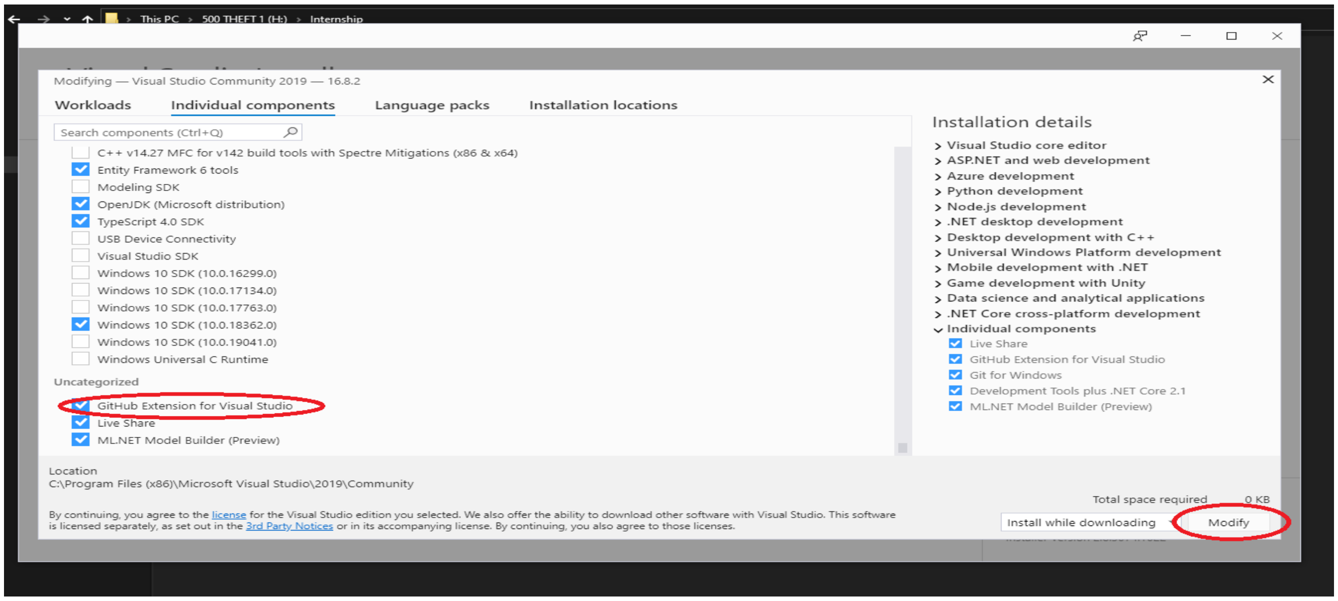
Installation details

- › Visual Studio core editor
- › ASP.NET and web development
- › Azure development
- › Python development
- › Node.js development
- › .NET desktop development
- › Desktop development with C++
- › Universal Windows Platform development
- › Mobile development with .NET
- › Game development with Unity
- › Data science and analytical applications
- › .NET Core cross-platform development
- ⌄ Individual components
  - ☑ Live Share
  - ☑ GitHub Extension for Visual Studio
  - ☑ Git for Windows
  - ☑ Development Tools plus .NET Core 2.1
  - ☑ ML.NET Model Builder (Preview)

Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 0 KB
Install while downloading | Modify



Modifying — Visual Studio Community 2019 — 16.8.2

Workloads | Individual components | Language packs | Installation locations

- [ ] C++ v14.27 MFC for v142 build tools with Spectre Mitigations (x86 & x64)
- [x] Entity Framework 6 tools
- [ ] Modeling SDK
- [x] OpenJDK (Microsoft distribution)
- [x] TypeScript 4.0 SDK
- [ ] USB Device Connectivity
- [ ] Visual Studio SDK
- [ ] Windows 10 SDK (10.0.16299.0)
- [ ] Windows 10 SDK (10.0.17134.0)
- [ ] Windows 10 SDK (10.0.17763.0)
- [x] Windows 10 SDK (10.0.18362.0)
- [ ] Windows 10 SDK (10.0.19041.0)
- [ ] Windows Universal C Runtime

Uncategorized

- [x] GitHub Extension for Visual Studio
- [x] Live Share
- [x] ML.NET Model Builder (Preview)

Installation details
- Visual Studio core editor
- ASP.NET and web development
- Azure development
- Python development
- Node.js development
- .NET desktop development
- Desktop development with C++
- Universal Windows Platform development
- Mobile development with .NET
- Game development with Unity
- Data science and analytical applications
- .NET Core cross-platform development
- Individual components
  - [x] Live Share
  - [x] GitHub Extension for Visual Studio
  - [x] Git for Windows
  - [x] Development Tools plus .NET Core 2.1
  - [x] ML.NET Model Builder (Preview)

Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community

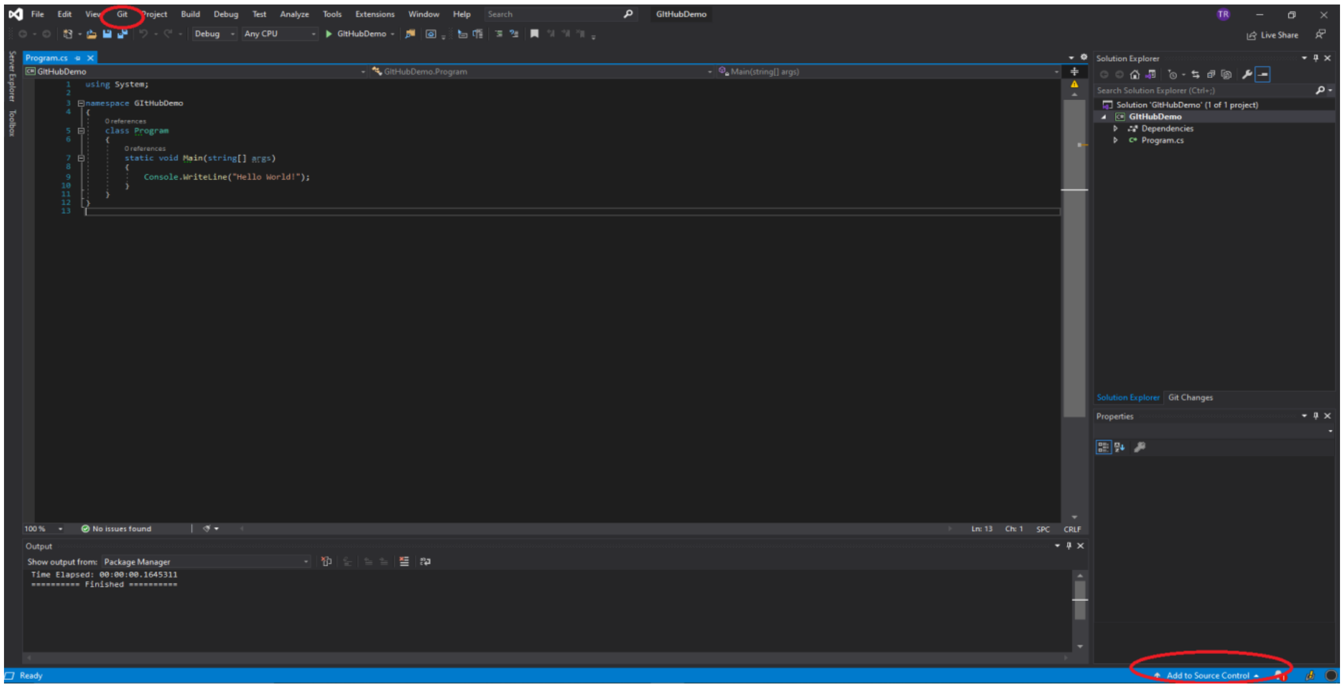
By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 0 KB

Install while downloading | Modify

After selecting click on the **Modify** button to add the changes to Visual Studio.

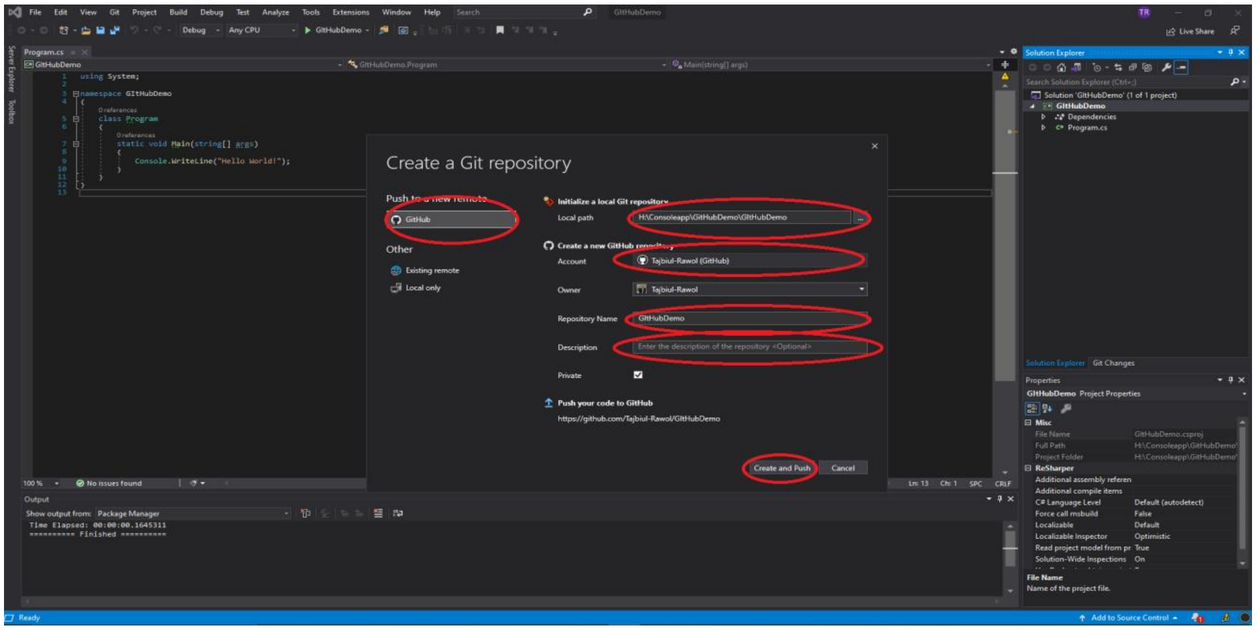
Now to create a repository, you can either click at the righthand bottom corner of the IDE saying **Add to Source Control** or you can create a new repository by clicking on the **Git > Create Git Repository** menu that is on the menu bar above.


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
GitHubDemo
Solution Explorer
Solution 'GitHubDemo' (1 of 1 project)
Dependencies
Program.cs
Git Changes
Properties
Ready
Add to Source Control

```
using System;

namespace GItHubDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

```
Show output from: Package Manager
Time Elapsed: 00:00:00.1645311
========== Finished ==========
```

Once you click on it a window will appear where you will be prompted to enter the description of the project. **Note:** Before creating a repo through VS 19 make sure you have an account in GitHub otherwise Visual Studio will not auto detect your account and cannot create a repository. You can choose to make your project **public** by unchecking the **private** checkbox so that everyone can see your project. The **Local Path** shows where your project is locally in your machine. Once done click on the **Create and Push** button and VS 19 will automatically create and push your project in GitHub.



```
using System;

namespace GItHubDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

Create a Git repository

Push to a new remote: GitHub

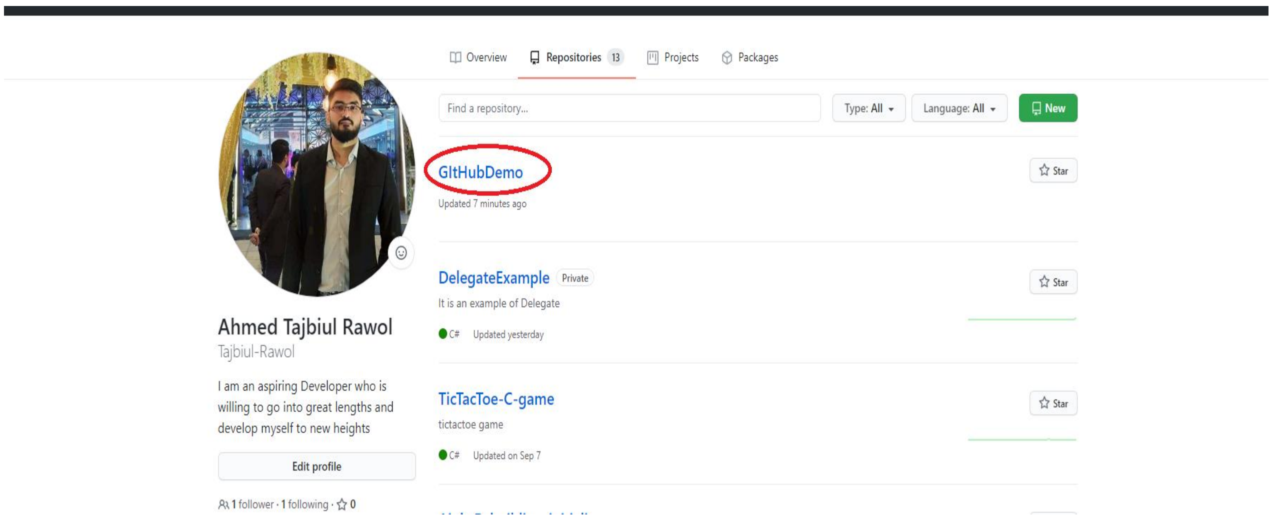
Other: Existing remote, Local only

Initialize a local Git repository
Local path: H:\Consoleapp\GitHubDemo\GItHubDemo

Create a new GitHub repository
Account: Tajbiul-Rawol (GitHub)
Owner: Tajbiul-Rawol
Repository Name: GItHubDemo
Description: Enter the description of the repository <Optional>
Private: ✓

Push your code to GitHub
https://github.com/Tajbiul-Rawol/GItHubDemo

Create and Push | Cancel



Overview | Repositories 13 | Projects | Packages

Ahmed Tajbiul Rawol
Tajbiul-Rawol

I am an aspiring Developer who is willing to go into great lengths and develop myself to new heights

1 follower · 1 following · 0

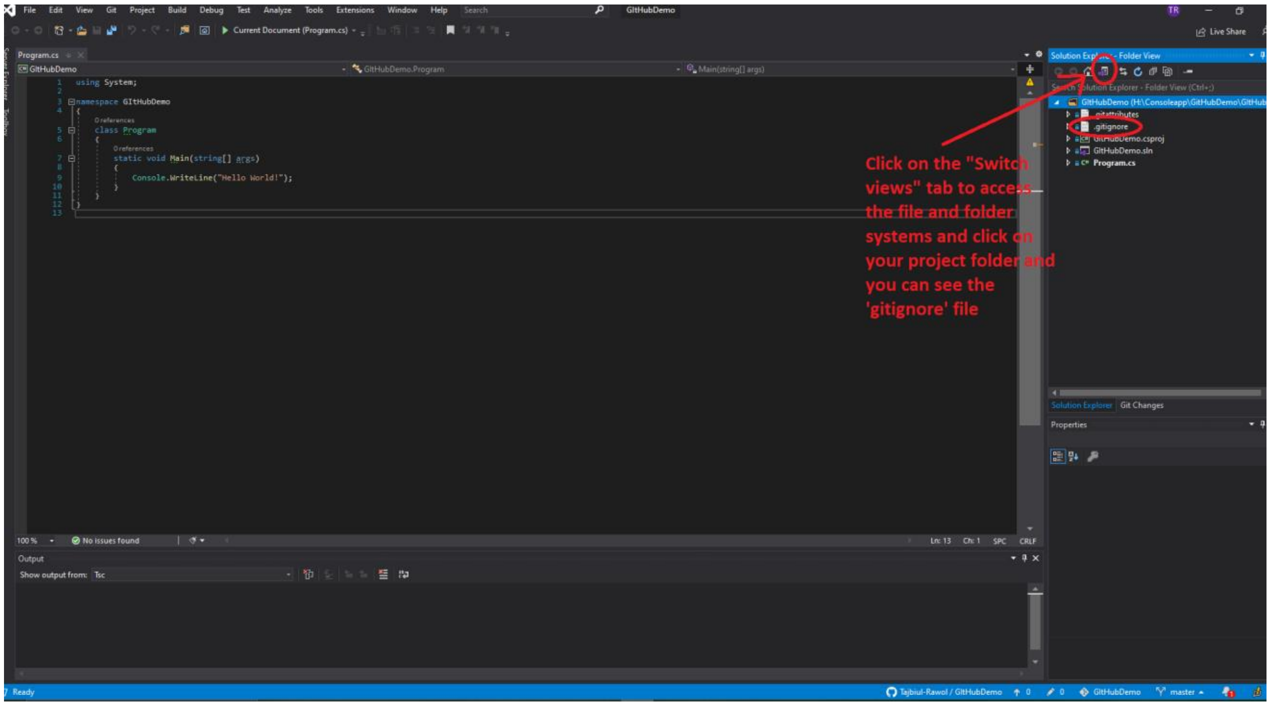
GItHubDemo
Updated 7 minutes ago

DelegateExample Private
It is an example of Delegate
C# Updated yesterday

TicTacToe-C-game
tictactoe game
C# Updated on Sep 7

Once you create and push your project to GitHub you can see that add the bottom right it states that you are on the **master** branch and if you click on it you can see that there are more options such as **New Branch, Manage Branches, Create a new Pull, Fetch, Pull, Push** and **View History**.

By default, you may see the **gitignore** file at the solution as VS 19 has worked around a new way to see and access the file system instead of just the project.

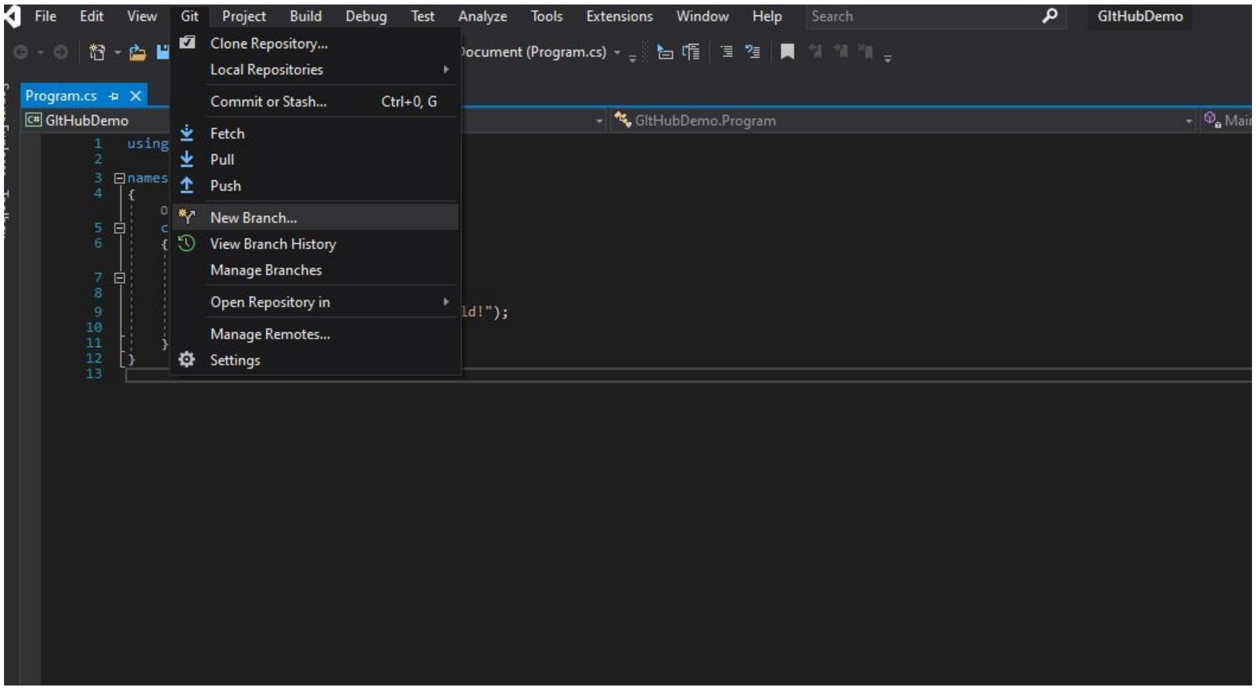

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Current Document (Program.cs)
Program.cs
GitHubDemo
Solution Explorer - Folder View
GitHubDemo (H:\Consoleapp\GitHubDemo\GitHub
.gitattributes
.gitignore
GitHubDemo.csproj
GitHubDemo.sln
Program.cs
Solution Explorer
Git Changes
Properties
No issues found
Output
Ready
Tajbiul-Rawol / GitHubDemo
master

```
using System;

namespace GItHubDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

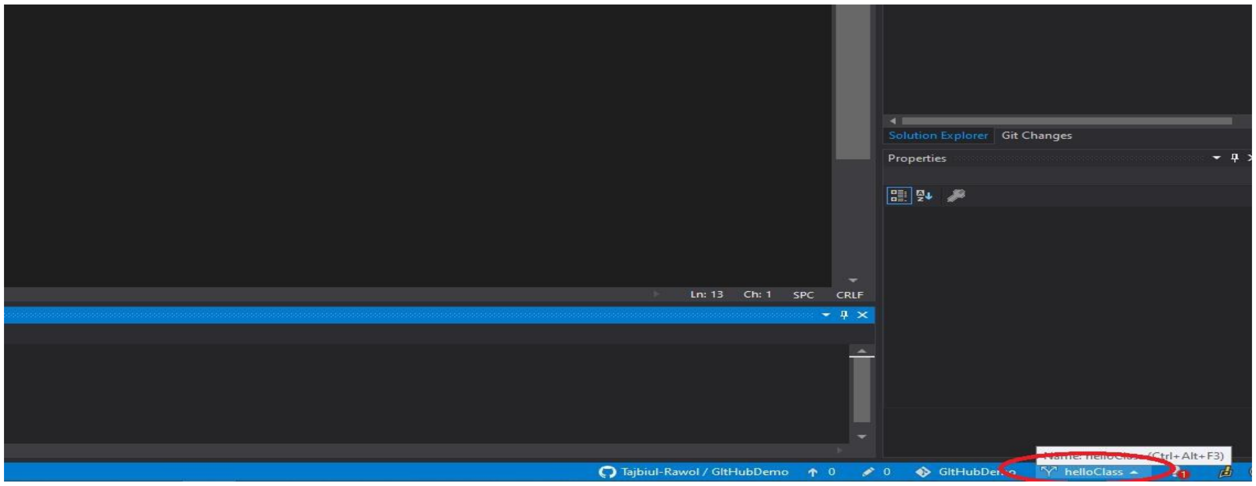
Click on the "Switch views" tab to access the file and folder systems and click on your project folder and you can see the 'gitignore' file

Click on the **switch views** tab on solution explorer and select your folder and you can see it. **Gitignore** is a file that logs all the files that you don't want to commit or add change to in a project. It lists down all the files and ignores any changes when a change is made to a project. You can also see that there are 'blue locks' beside your classes or files. This indicates that the files are added committed and pushed into the Git repository all in one go. If you make any changes to the code, you will see a change in icon which we will see later.

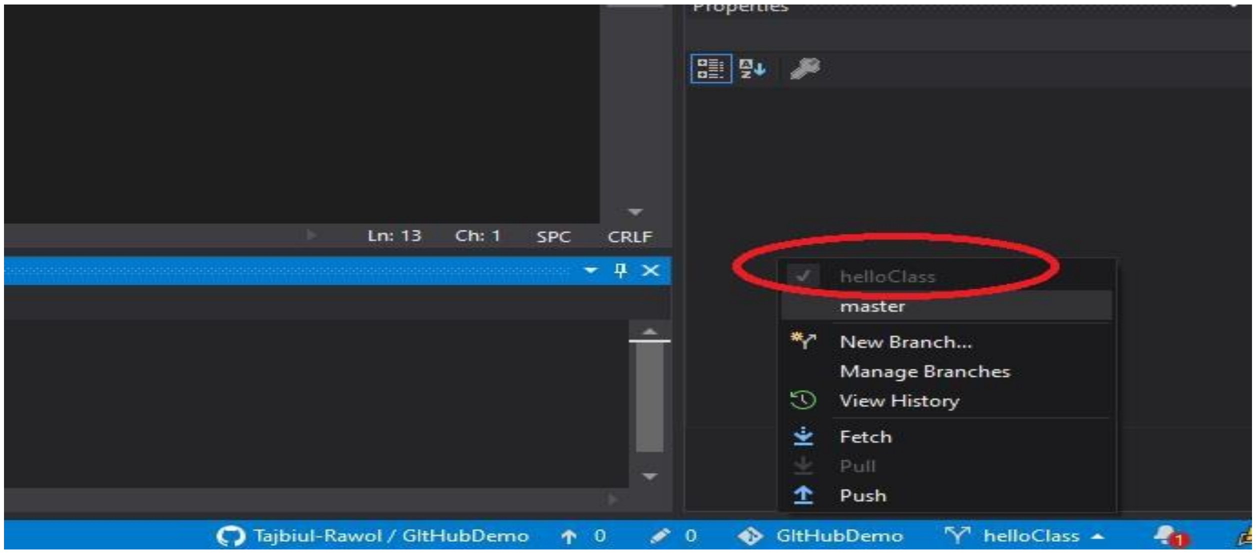
By default, VS 19 will keep you at the **master** branch which is the main branch the project is under development.



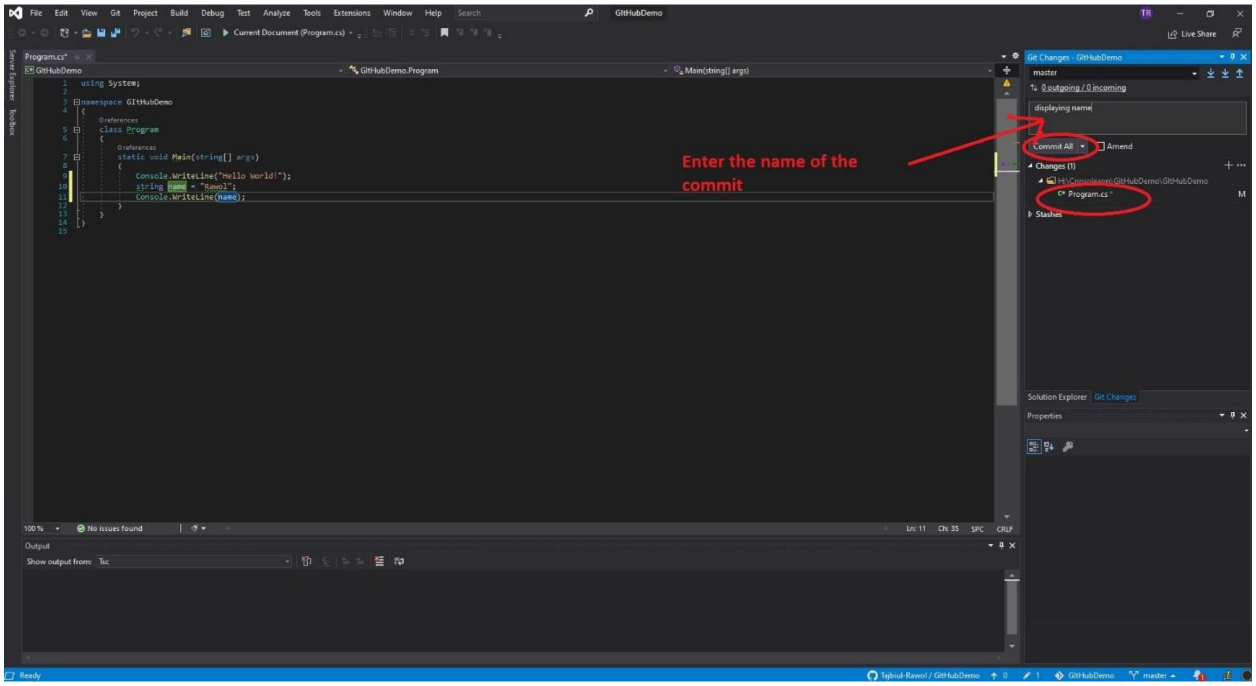
To create a new branch, you can click on the **Git** menu on the menu bar or the **master** menu on the right bottom corner which already shows you are on the master branch. Once you click on the **New Branch** you will be prompted to create a branch with a branch name and a check box labeled **checkout branch** which means once you create a new branch from the master you will be taken to the new branch and you can see below if you are on the master branch or the new branch.



Here as you can see you are already on the new branch as there is a tick mark beside it.



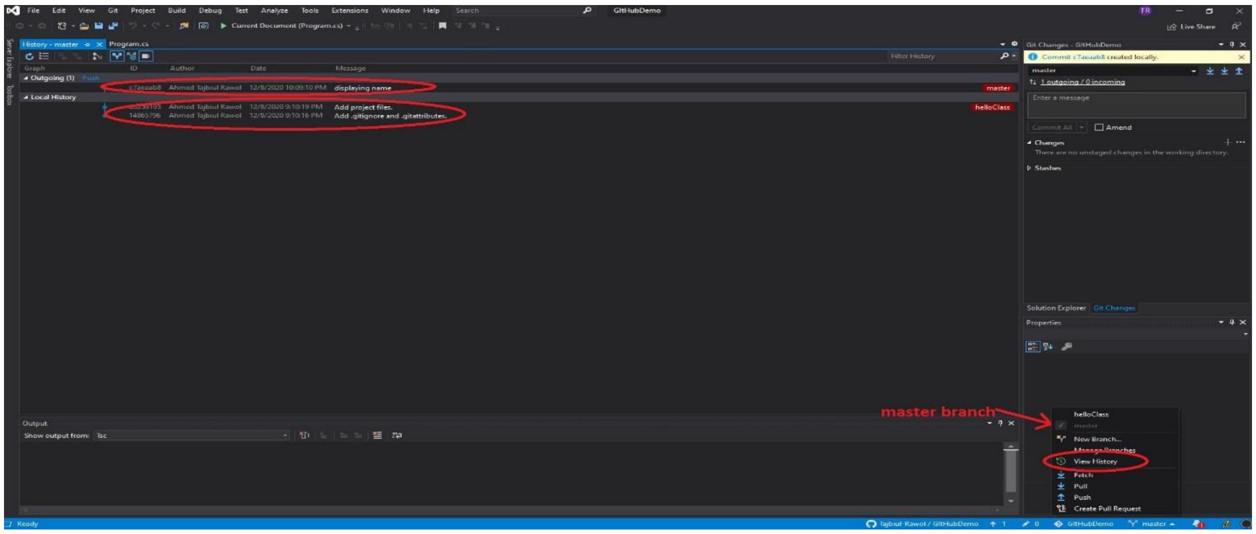
Now Suppose you changed or added some lines of code in your project in the **master** branch and you would want to commit the changes just click **Git > Commit or Stash** and you will see that you will be prompted to enter a name of your commit at the right hand side of your IDE.



```
using System;

namespace GItHubDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
            string name = "Rawol";
            Console.WriteLine(name);
        }
    }
}
```

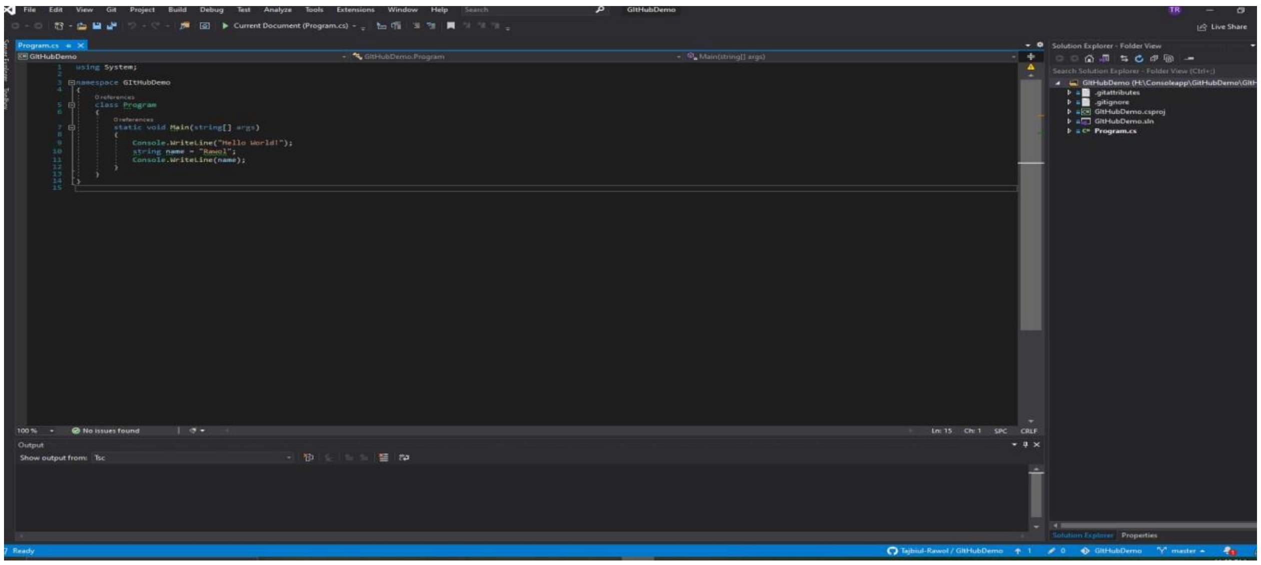
displaying name

You can also see on which part or class of the program changes were made and click **Commit All** to commit all changes to the master branch. To see the change history, click on **master > View History** on the bottom right menu and you will see changes made on each branch. As you can see there were changes made to the master branch with the commit names beside and you can also see the and the commit ids and name of the author who made the changes. Notice that the changes were made on the master branch and the changes were not affecting any other branches.

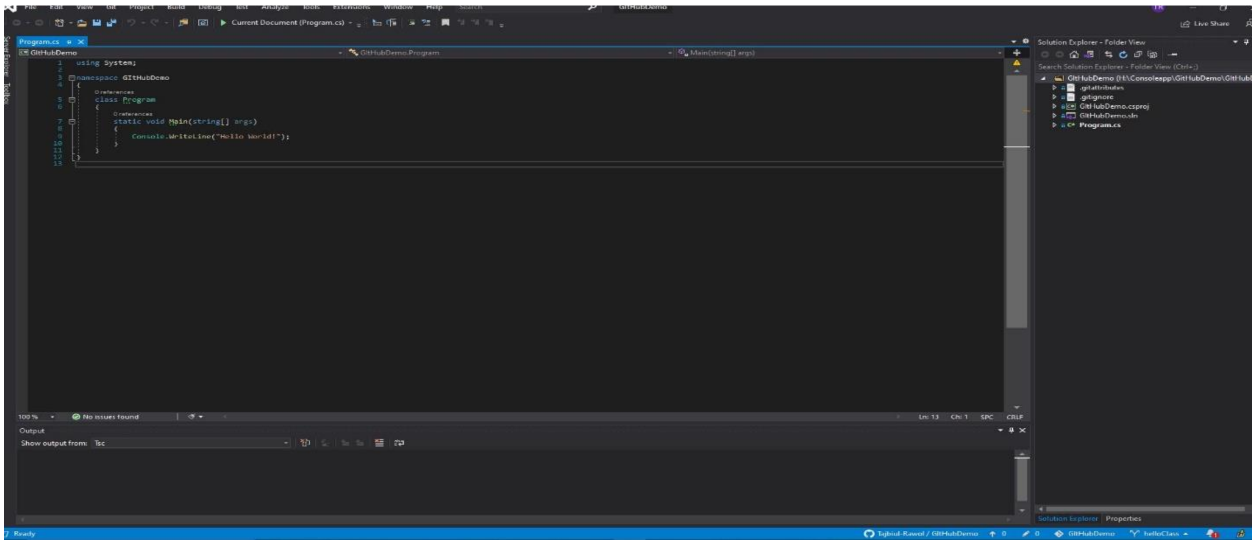


Graph	ID	Author	Date	Message
Outgoing (1)				
	c7aeaab8	Ahmed Tajbiul Rawol	12/8/2020 10:09:10 PM	displaying name
Local History				
	85239195	Ahmed Tajbiul Rawol	12/8/2020 9:10:19 PM	Add project files.
	14865796	Ahmed Tajbiul Rawol	12/8/2020 9:10:16 PM	Add .gitignore and .gitattributes.

Commit c7aeaab8 created locally.



```
using System;

namespace GItHubDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
            string name = "Rawol";
            Console.WriteLine(name);
        }
    }
}
```



```
using System;

namespace GItHubDemo
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

Notice when the branch was changed to the **hello Class** branch you see that the code was reverted, and any changes made on the master branch did not affect the changes of any other branches.

Standard terminal workflow

1. When working together in a group at first open gitbash by right clicking the mouse on the folder that you have cloned your repository or the folder you have your project. Then write the following command, **git pull (retrieve new updates from the repository if any of your members have updated the project files)**.

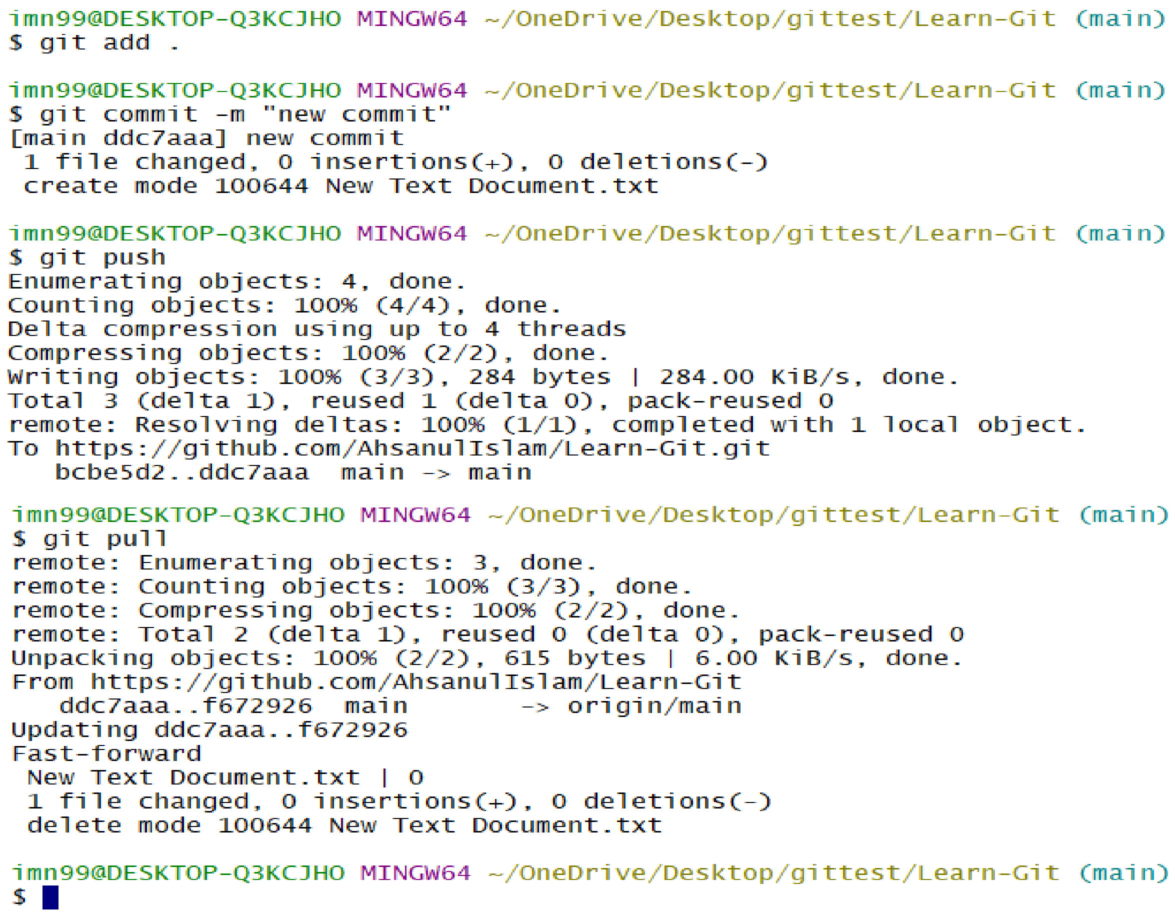
2. If you want to add your files to the repository use the command **git add . (the . here represent all files that you have in your folder and will add all of the files ready to be uploaded).**

3. Then, if you want to know which files you have added or modified use the command **git status (shows which files are changed or modified and also newly added and ready to be committed).**

4. **Next,** add a message to the file that you have committed to make your teammates understand what features have you added in the new file or mention any bugs that you have removed using the command **git commit -m "insert your message here".**

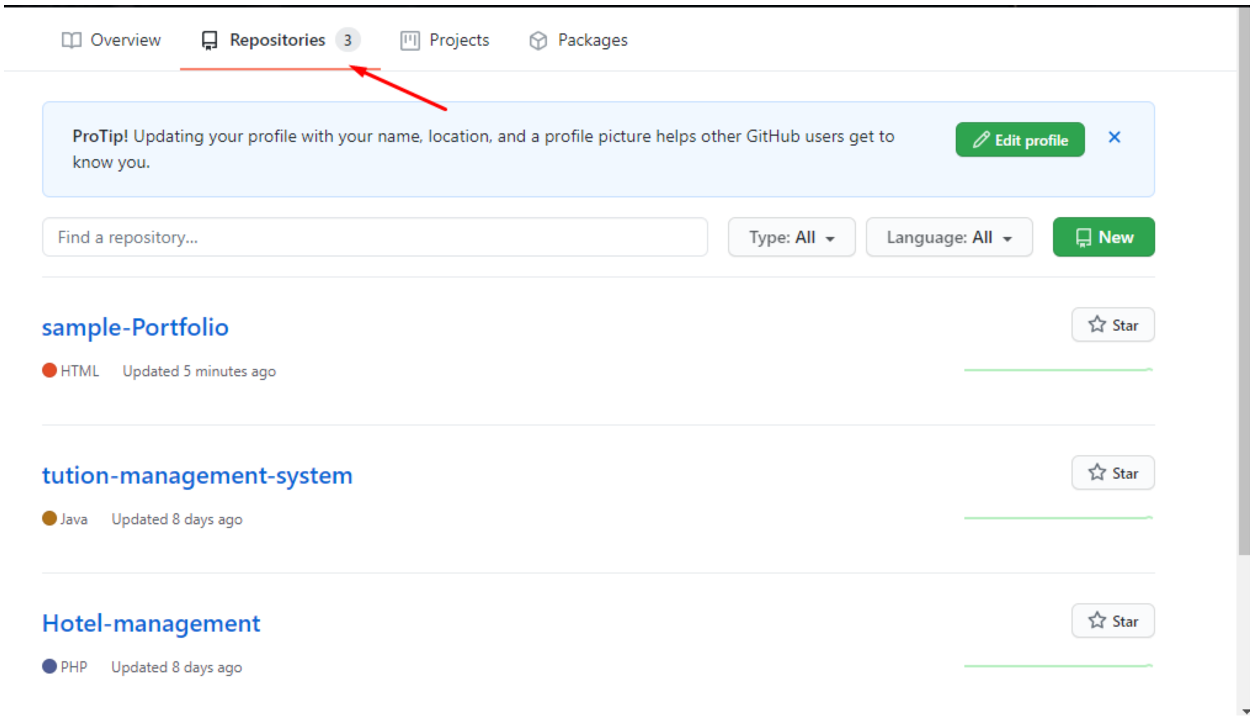
5. After all the previous steps use the command **git push (to commit and upload all of the changes you have made on to your project to own made repository).**

6. Lastly remember to wait until 100% upload is completed **before exiting gitBash** and also refresh your github repository page to make sure that your files have been uploaded with the **correct commit message** on github.



```
imn99@DESKTOP-Q3KCJHO MINGW64 ~/OneDrive/Desktop/gittest/Learn-Git (main)
$ git add .

imn99@DESKTOP-Q3KCJHO MINGW64 ~/OneDrive/Desktop/gittest/Learn-Git (main)
$ git commit -m "new commit"
[main ddc7aaa] new commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 New Text Document.txt

imn99@DESKTOP-Q3KCJHO MINGW64 ~/OneDrive/Desktop/gittest/Learn-Git (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 284 bytes | 284.00 KiB/s, done.
Total 3 (delta 1), reused 1 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/AhsanulIslam/Learn-Git.git
   bcbe5d2..ddc7aaa  main -> main

imn99@DESKTOP-Q3KCJHO MINGW64 ~/OneDrive/Desktop/gittest/Learn-Git (main)
$ git pull
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (2/2), 615 bytes | 6.00 KiB/s, done.
From https://github.com/AhsanulIslam/Learn-Git
   ddc7aaa..f672926  main       -> origin/main
Updating ddc7aaa..f672926
Fast-forward
 New Text Document.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 New Text Document.txt

imn99@DESKTOP-Q3KCJHO MINGW64 ~/OneDrive/Desktop/gittest/Learn-Git (main)
$
```

View your file on GitHub



ProTip! Updating your profile with your name, location, and a profile picture helps other GitHub users get to know you.

sample-Portfolio
HTML Updated 5 minutes ago

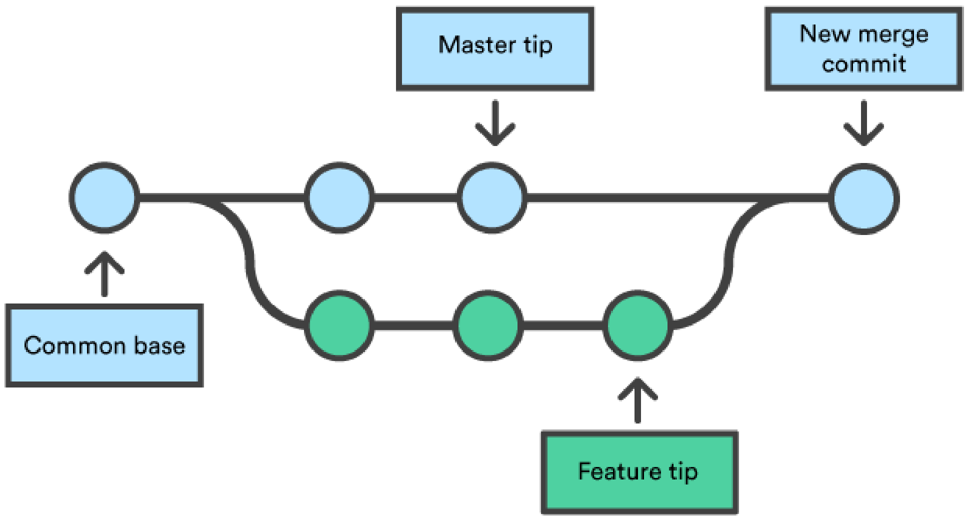
tution-management-system
Java Updated 8 days ago

Hotel-management
PHP Updated 8 days ago

For advanced stuff

Branching

A branch is like a parallel copy of the master branch. It allows changes to occur in parallel. The master branch does not get modified until the branch is merged into the master branch. In a group work Branching is very helpful. In group project member can work parallel. Master branch is the final version of project.



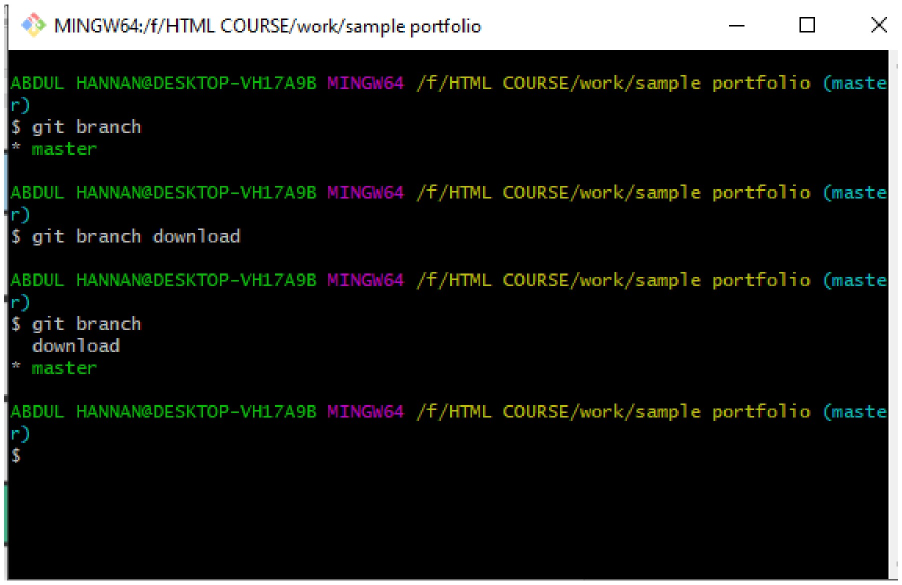
Use the command **git checkout -b first-branch (creates a branch named "first-branch" and move to that branch for commit work).** Branching is used where lots of people are working on a project to make sure that the main file(on the main branch) does not get corrupted with any development bugs and if a feature is complete and after notifying all developers that there are no bugs, the features of a particular branch is pushed in to the main branch and is considered as a new version release.

To get back to the master/main branch use the command **git checkout main (use the command "git checkout branch_name" to traverse through different branches).**

Remember, to be sure use the command **git branch (it shows how many branches are in your repository and the * branch_name, denotes the branch you are in).**

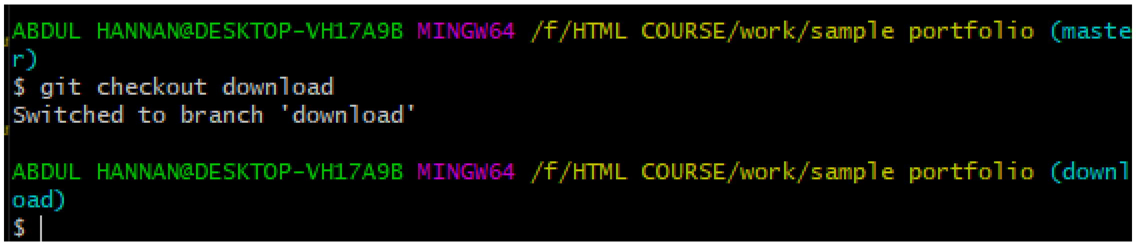
How to create branch

At first go to your project location then open git bash software, then type, $ git branch " branch name"


MINGW64:/f/HTML COURSE/work/sample portfolio

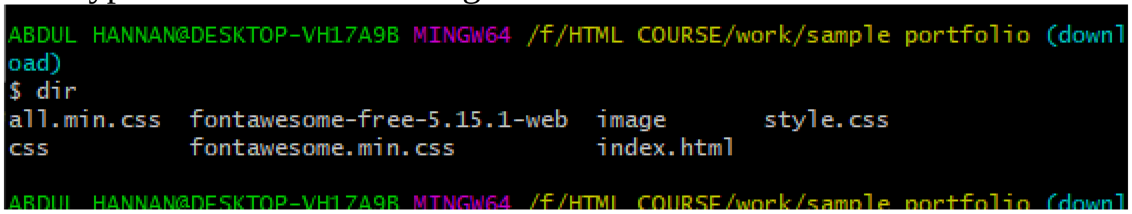
```
ABDUL HANNAN@DESKTOP-VH17A9B MINGW64 /f/HTML COURSE/work/sample portfolio (maste
r)
$ git branch
* master

ABDUL HANNAN@DESKTOP-VH17A9B MINGW64 /f/HTML COURSE/work/sample portfolio (maste
r)
$ git branch download

ABDUL HANNAN@DESKTOP-VH17A9B MINGW64 /f/HTML COURSE/work/sample portfolio (maste
r)
$ git branch
  download
* master

ABDUL HANNAN@DESKTOP-VH17A9B MINGW64 /f/HTML COURSE/work/sample portfolio (maste
r)
$
```

After creating local branch go to master to local branch, type $ git checkout 'local branch name'



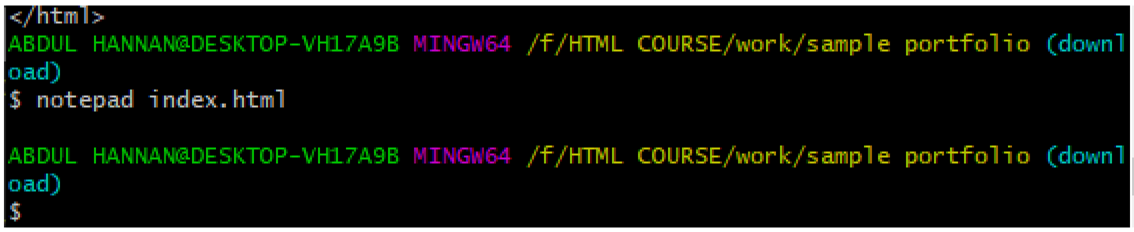
```
ABDUL HANNAN@DESKTOP-VH17A9B MINGW64 /f/HTML COURSE/work/sample portfolio (maste
r)
$ git checkout download
Switched to branch 'download'

ABDUL HANNAN@DESKTOP-VH17A9B MINGW64 /f/HTML COURSE/work/sample portfolio (downl
oad)
$ |
```

And type $ dir to see folder in github .



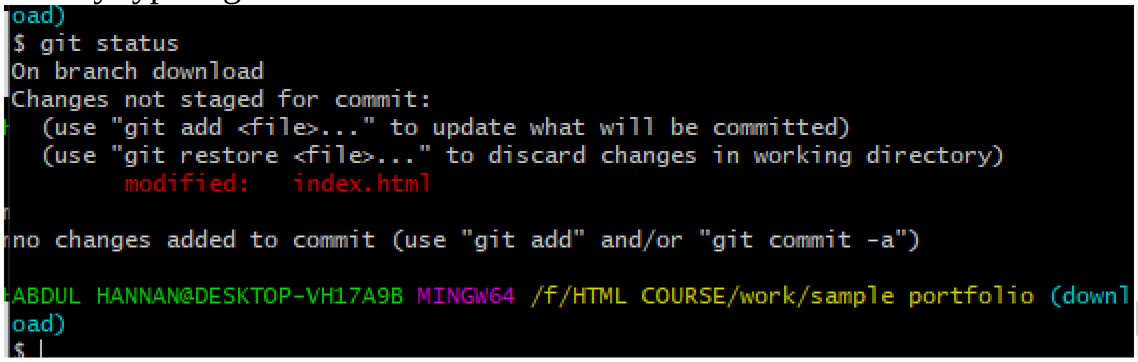
```
ABDUL HANNAN@DESKTOP-VH17A9B MINGW64 /f/HTML COURSE/work/sample portfolio (downl
oad)
$ dir
all.min.css  fontawesome-free-5.15.1-web  image       style.css
css          fontawesome.min.css          index.html

ABDUL HANNAN@DESKTOP-VH17A9B MINGW64 /f/HTML COURSE/work/sample portfolio (downl
```

To modify file by local type $ notepad 'file name'



```
</html>
ABDUL HANNAN@DESKTOP-VH17A9B MINGW64 /f/HTML COURSE/work/sample portfolio (downl
oad)
$ notepad index.html

ABDUL HANNAN@DESKTOP-VH17A9B MINGW64 /f/HTML COURSE/work/sample portfolio (downl
oad)
$
```

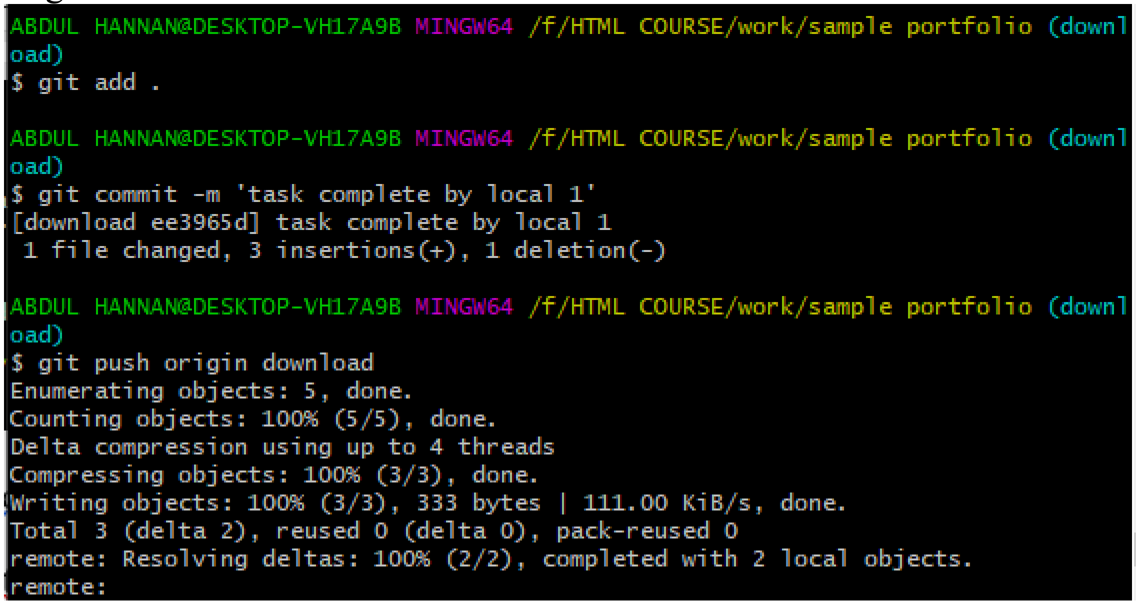
Then bash open a notepad for you, modify and save file. and to check which file modify type $ git status.



```
oad)
$ git status
On branch download
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")

ABDUL HANNAN@DESKTOP-VH17A9B MINGW64 /f/HTML COURSE/work/sample portfolio (downl
oad)
$ |
```

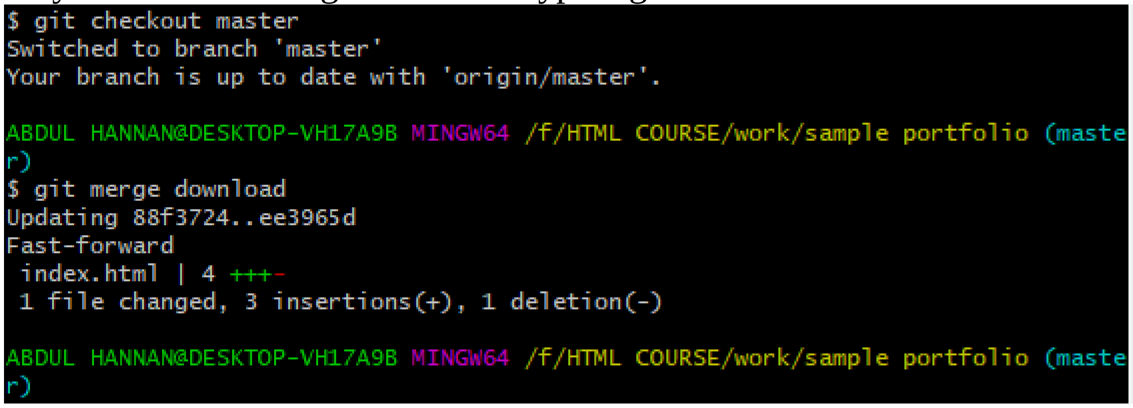
For adding modified file type $ git add . (here . means all)

Then type $git commit -m ‘message’.then push your file to github type $ git push origin ‘local user name’



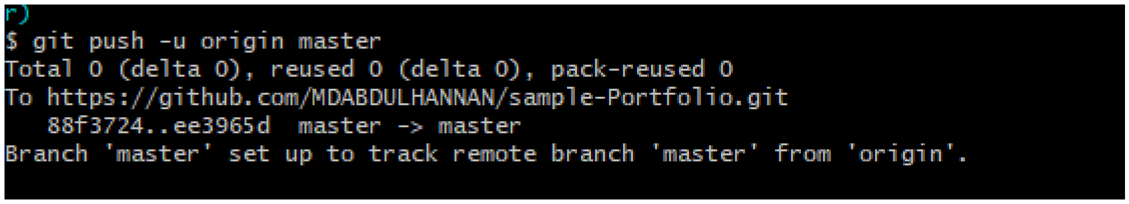
```
ABDUL HANNAN@DESKTOP-VH17A9B MINGW64 /f/HTML COURSE/work/sample portfolio (downl
oad)
$ git add .

ABDUL HANNAN@DESKTOP-VH17A9B MINGW64 /f/HTML COURSE/work/sample portfolio (downl
oad)
$ git commit -m 'task complete by local 1'
[download ee3965d] task complete by local 1
 1 file changed, 3 insertions(+), 1 deletion(-)

ABDUL HANNAN@DESKTOP-VH17A9B MINGW64 /f/HTML COURSE/work/sample portfolio (downl
oad)
$ git push origin download
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 333 bytes | 111.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
```

After complete task local branch creat on github but main project do not change, only master can change. After that type $ git checkout master.



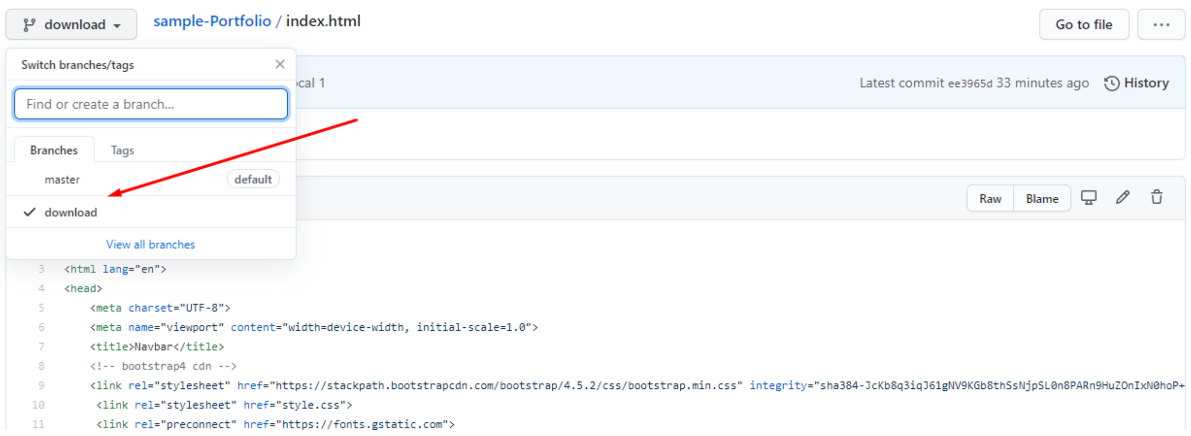
```
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

ABDUL HANNAN@DESKTOP-VH17A9B MINGW64 /f/HTML COURSE/work/sample portfolio (maste
r)
$ git merge download
Updating 88f3724..ee3965d
Fast-forward
 index.html | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)

ABDUL HANNAN@DESKTOP-VH17A9B MINGW64 /f/HTML COURSE/work/sample portfolio (maste
r)
```

Then master, marge file local to master file. After that for remotely marge type git push -u origin master



```
r)
$ git push -u origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MDABDULHANNAN/sample-Portfolio.git
   88f3724..ee3965d  master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

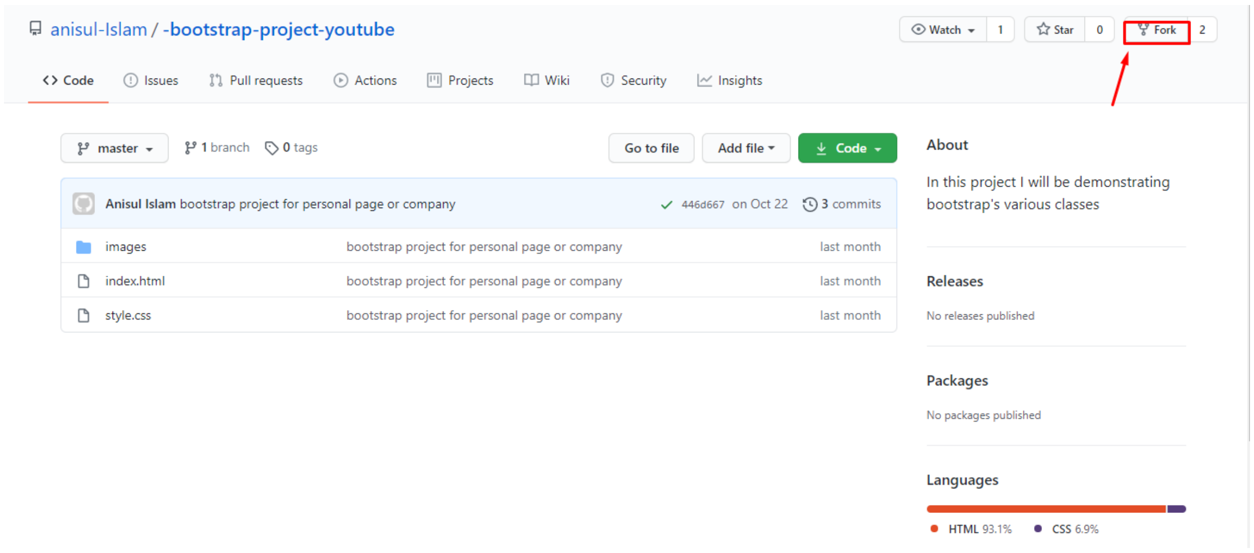
Main file show on GitHub



```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Navbar</title>
    <!-- bootstrap4 cdn -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+
     <link rel="stylesheet" href="style.css">
     <link rel="preconnect" href="https://fonts.gstatic.com">
```

Fork

You can fork any repositorie by clicking the fork button in the upper right hand corner of a r repositories page. Click on the Fork button to fork any repo on github.com. Source: GitHub Guides. When you fork a repo on GitHub, the forked repo is copied to your GitHub account, and you can edit it as the repository's owner.



Anisul Islam bootstrap project for personal page or company ✓ 446d667 on Oct 22 3 commits

images	bootstrap project for personal page or company	last month
index.html	bootstrap project for personal page or company	last month
style.css	bootstrap project for personal page or company	last month

About

In this project I will be demonstrating bootstrap's various classes

Releases

No releases published

Packages

No packages published

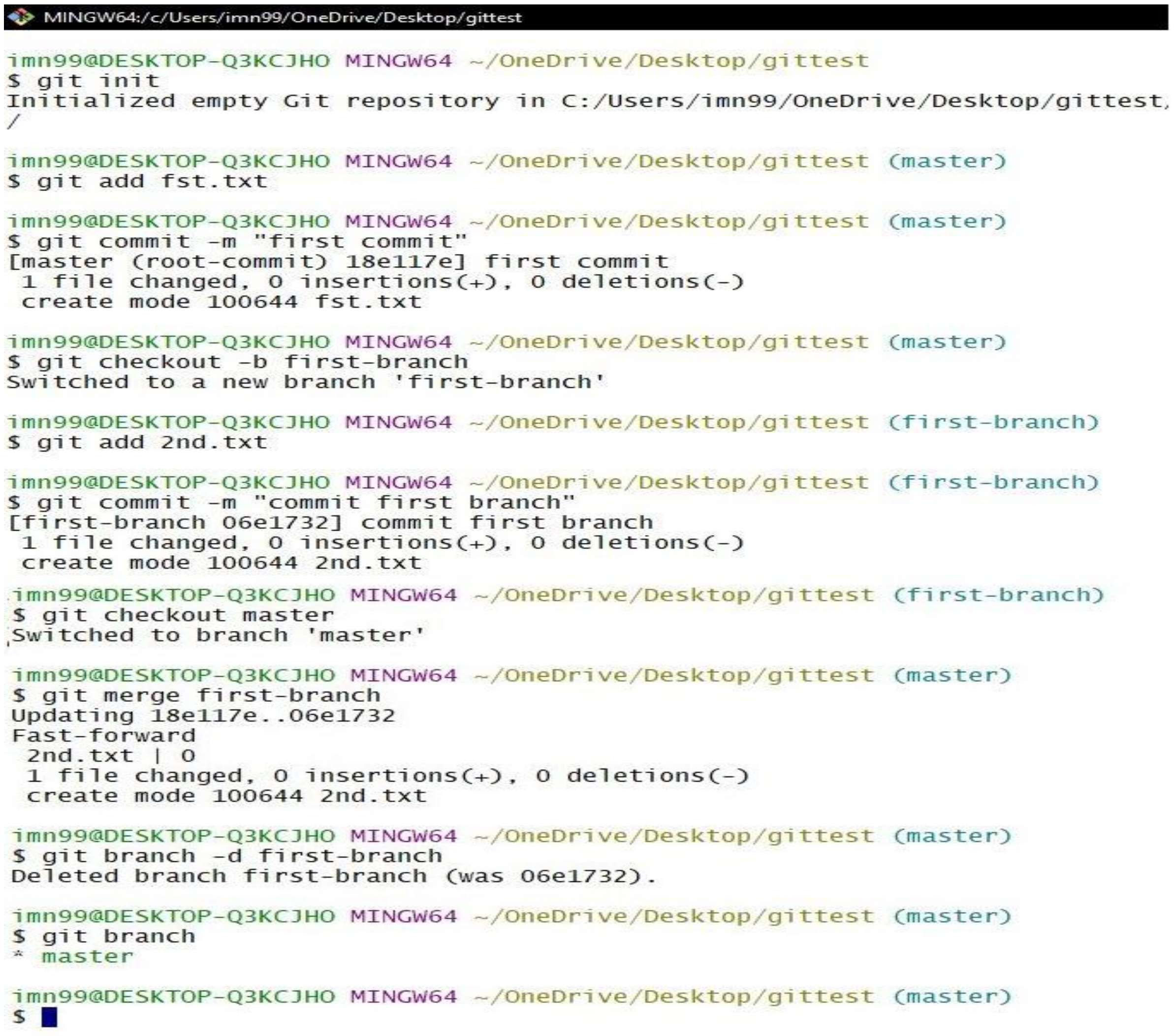
Languages

● HTML 93.1% ● CSS 6.9%

Merging

To merge any other branch on to main branch, at **first go to the main branch** by using the command **git checkout main**. Then type **git merge first-branch (as we have created a branch named first-branch)** it will help to merge and update all the files on the main branch (use merge especially after knowing it will not affect the code on the main branch in a bad way).

Now as all the files have been uploaded on to the main branch including the commit history of the **first-branch.** if you want to delete the other branch use the command **git branch -d first-branch (remember to do it once you are in the main branch and to travel to main branch use the command "git checkout main").**



```
MINGW64:/c/Users/imn99/OneDrive/Desktop/gittest

imn99@DESKTOP-Q3KCJHO MINGW64 ~/OneDrive/Desktop/gittest
$ git init
Initialized empty Git repository in C:/Users/imn99/OneDrive/Desktop/gittest,
/

imn99@DESKTOP-Q3KCJHO MINGW64 ~/OneDrive/Desktop/gittest (master)
$ git add fst.txt

imn99@DESKTOP-Q3KCJHO MINGW64 ~/OneDrive/Desktop/gittest (master)
$ git commit -m "first commit"
[master (root-commit) 18e117e] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 fst.txt

imn99@DESKTOP-Q3KCJHO MINGW64 ~/OneDrive/Desktop/gittest (master)
$ git checkout -b first-branch
Switched to a new branch 'first-branch'

imn99@DESKTOP-Q3KCJHO MINGW64 ~/OneDrive/Desktop/gittest (first-branch)
$ git add 2nd.txt

imn99@DESKTOP-Q3KCJHO MINGW64 ~/OneDrive/Desktop/gittest (first-branch)
$ git commit -m "commit first branch"
[first-branch 06e1732] commit first branch
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 2nd.txt
imn99@DESKTOP-Q3KCJHO MINGW64 ~/OneDrive/Desktop/gittest (first-branch)
$ git checkout master
Switched to branch 'master'

imn99@DESKTOP-Q3KCJHO MINGW64 ~/OneDrive/Desktop/gittest (master)
$ git merge first-branch
Updating 18e117e..06e1732
Fast-forward
 2nd.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 2nd.txt

imn99@DESKTOP-Q3KCJHO MINGW64 ~/OneDrive/Desktop/gittest (master)
$ git branch -d first-branch
Deleted branch first-branch (was 06e1732).

imn99@DESKTOP-Q3KCJHO MINGW64 ~/OneDrive/Desktop/gittest (master)
$ git branch
* master

imn99@DESKTOP-Q3KCJHO MINGW64 ~/OneDrive/Desktop/gittest (master)
$ 
```

Log

Git log --all to check all commit made by the all the users in what time with the commit message. To know further use of log, use this link https://www.thegeekstuff.com/2014/04/git-log/ **as this link provides all info to use git log in various ways**