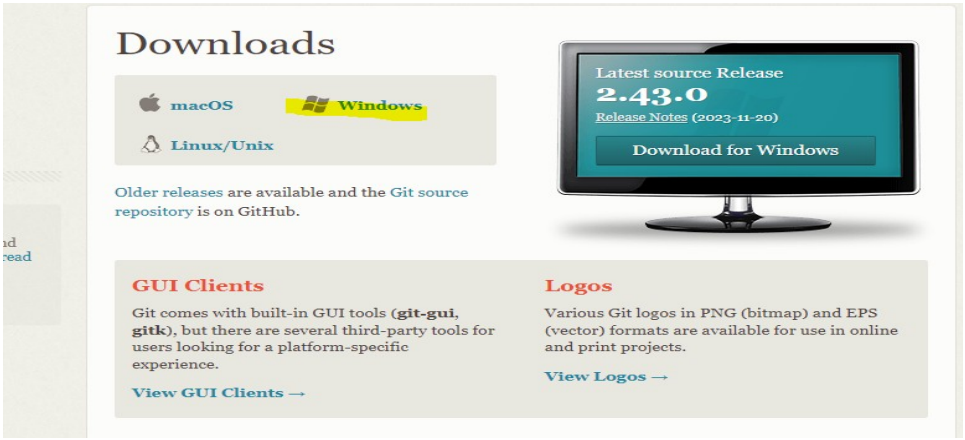
MANITOBA INSTITUTE OF TRADES & TECHNOLOGY

Setting Up Git

1. ***Install Git Bash on your local computer***

- Visit the official Git website: https://git-scm.com/downloads
- Download the latest version of Git Bash for your operating system.
- Follow the installation instructions provided during the installation process.

Note: Do not change or modify any settings during the installation.



Now you should be able to locate Git bash in your start menu. Launch git bash. Set up your git username and email address. This is a very important step. The name and email you set up here will be tagged along with any changes you make to the file afterwards. Essential part of version control.

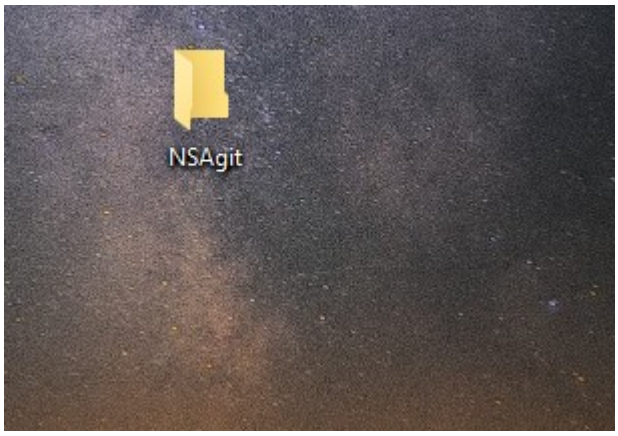
```
DJ's PC@LAPTOP-BA35V1H5 MINGW64 ~/Desktop/NSAgit (master)
$ git config --global user.name JayDixit

DJ's PC@LAPTOP-BA35V1H5 MINGW64 ~/Desktop/NSAgit (master)
$ git config --global user.email "jay.dixit@mitt.ca"

DJ's PC@LAPTOP-BA35V1H5 MINGW64 ~/Desktop/NSAgit (master)
$
```

2. ***Initialize a Repository***

- Create a Folder on your desktop named “NSAGit”



- Use the cd command to navigate to the directory Desktop/NSAGit

```
DJ's PC@LAPTOP-BA35V1H5 MINGW64 ~
$ cd Desktop/NSAgit/

DJ's PC@LAPTOP-BA35V1H5 MINGW64 ~/Desktop/NSAgit
$
```

- Issue “git init” command

```
DJ's PC@LAPTOP-BA35V1H5 MINGW64 ~/Desktop/NSAgit
$ git init
Initialized empty Git repository in C:/Users/DJ's PC/Desktop/NSAgit/.git/

DJ's PC@LAPTOP-BA35V1H5 MINGW64 ~/Desktop/NSAgit (master)
$
```

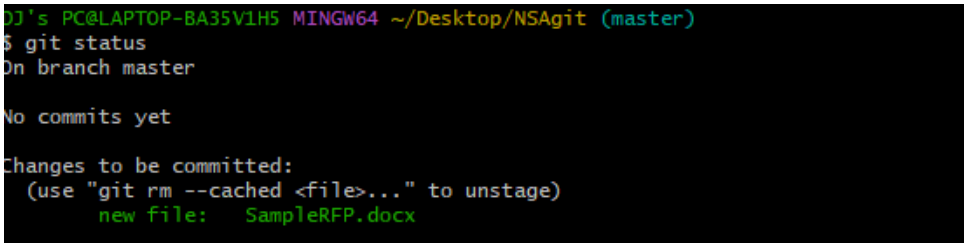
3. ***Add files to Repository***

- Move the sample RFP “Request for Proposal Vulnerability Scanning Tool Deployment” to the NSAgit folder

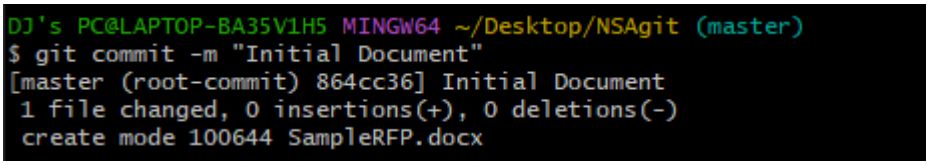
- Use “git add” command to add the initial, original version of the RFP file to staging area.

```
DJ's PC@LAPTOP-BA35V1H5 MINGW64 ~/Desktop/NSAgit (master)
$ git add SampleRFP.docx
```

- Use “git status” command to verify the current state of the file. You can see it has been added to the staging area but has not been committed yet.

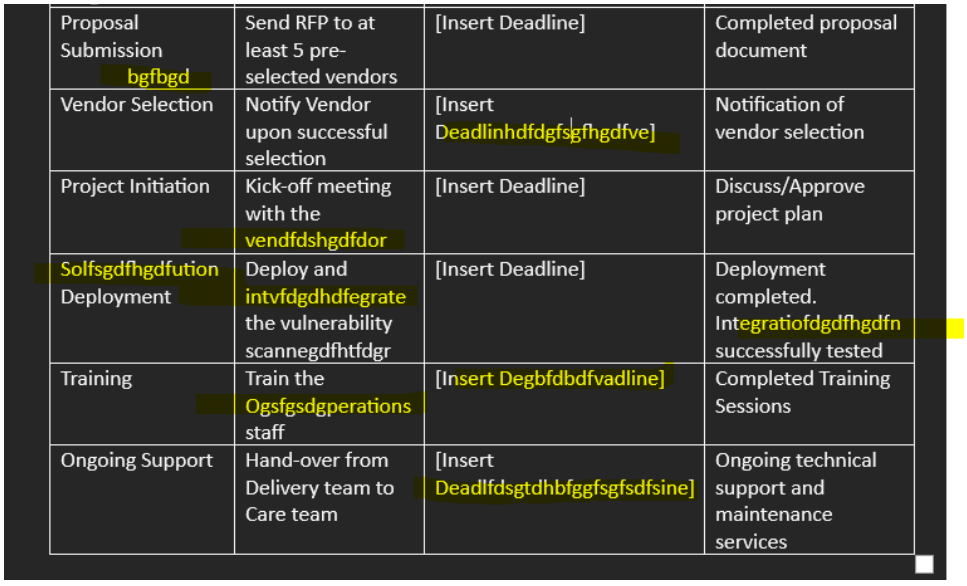


- Issue “git commit –m “command to add a snapshot of the initial document to the Repository database.



4. ***Make Changes to a document in Working Directory***

- Now Open the RFP by double clicking on it and make some changes. For example, I have added some random characters to the table displayed in the screenshot below. Save and close the file.



Proposal Submission bgfbgd	Send RFP to at least 5 pre-selected vendors	[Insert Deadline]	Completed proposal document
Vendor Selection	Notify Vendor upon successful selection	[Insert Deadlinhdfdgfsgfhgdfve]	Notification of vendor selection
Project Initiation	Kick-off meeting with the vendfdshgdfdor	[Insert Deadline]	Discuss/Approve project plan
Solfsgdfhgdfution Deployment	Deploy and intvfdgdhdfegrate the vulnerability scannegdfhtfdgr	[Insert Deadline]	Deployment completed. Integratiofdgdfhgdfn successfully tested
Training	Train the Ogsfgsdgperations staff	[Insert Degbfdbdfvadline]	Completed Training Sessions
Ongoing Support	Hand-over from Delivery team to Care team	[Insert Deadlfdsgtdhbfggfsgfsdfsine]	Ongoing technical support and maintenance services

5. **Stage your Changes**

- Issue “git status” command to check the current state. You will observe that a File has been modified but has not been staged or committed.

```
DJ's PC@LAPTOP-BA35V1H5 MINGW64 ~/Desktop/NSAgit (master)
$ git status
On branch master
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:   SampleRFP.docx

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

- Stage the changes you made in the previous step

```
DJ's PC@LAPTOP-BA35V1H5 MINGW64 ~/Desktop/NSAgit (master)
$ git add SampleRFP.docx
```

6. Commit the changes

- "-m" option allows you to add comments (messages) along with the changes made. Very important Version Control functionality.

```
DJ's PC@LAPTOP-BA35V1H5 MINGW64 ~/Desktop/NSAgit (master)
$ git commit -m "Changes made to RFP"
[master f841cc7] Changes made to RFP
 1 file changed, 0 insertions(+), 0 deletions(-)
```

7. Rollback to one of the previous versions

- So far you have two versions (snapshots) of the same file. The initial version and the one after you made changes. Evey time you issue "commit" a snapshot of the file is taken. You can rollback to any of those snapshots at any time in future.
- If you want to list the snapshots available, in other words if you want to see the history of changes made to the document, you can use "git log" command.

```
DJ's PC@LAPTOP-BA35V1H5 MINGW64 ~/Desktop/NSAgit (master)
$ git log
commit f841cc7725f86843504f07543379b25c99361c32 (HEAD -> master)
Author: JayDixit <jay.dixit@mitt.ca>
Date:   Sun Dec 17 21:47:54 2023 -0500

    Changes made to RFP

commit 864cc360527856607a25e89016eb6c90d934fe84
Author: JayDixit <jay.dixit@mitt.ca>
Date:   Sun Dec 17 21:38:07 2023 -0500

    Initial Document
```

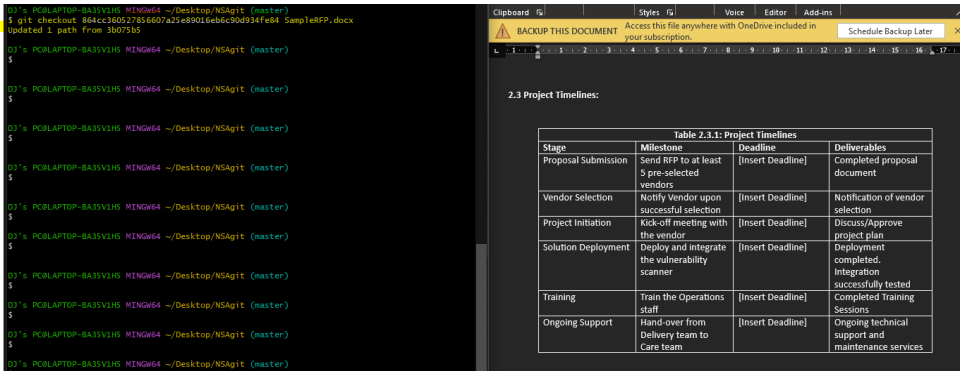
You can see the list of changes made in reverse chronological order

- *The first field is the "commit hash." It is like the identity of a snapshot. You will use this commit hash to rollback.*
- *You can also see the Author who made the changes along their username and email address.*
- *Next is the date and time.*
- *Finally, you see the comments on what changes were made by the author.*

- To rollback to a previous version of the document, you can use “git checkout” command.
- If I wanted to remove all the unwanted changes from my RFP document, I should rollback to the initial version of the document.

```
DJ's PC@LAPTOP-BA35V1H5 MINGW64 ~/Desktop/NSAgit (master)
$ git checkout 864cc360527856607a25e89016eb6c90d934fe84 SampleRFP.docx
Updated 1 path from 3b075b5
```

- Now open the document by double clicking it and verify that all the unwanted changes are gone.



Setting up GitHub

The steps above show you how you can efficiently deploy a DVCS using Git.

One thing you must have noticed that Git Bash is installed and deployed on your local computer. Git is a distributed version control system (DVCS) that runs locally on your computer. It tracks changes in source code/document and manages version history within your project directory.

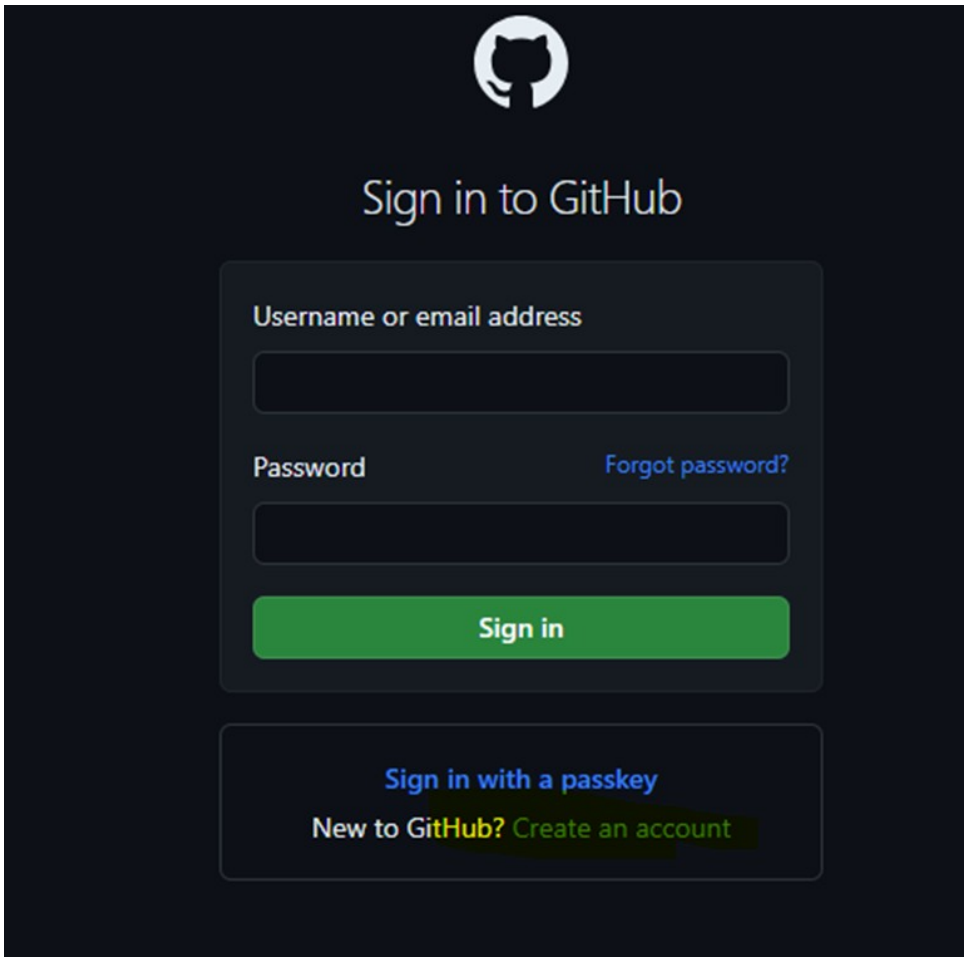
What if you wanted others on your team to have access to my Git repository and make changes to it? This is where GitHub comes into picture. GitHub is a web-based platform that extends Git's capabilities. It hosts Git repositories on the web, making it a central hub for collaborative version control.

Developers/Engineers typically use Git on their local machines to manage and track changes in their projects. GitHub acts as a remote repository, allowing developers to push their local Git repositories to the cloud, share code with others, and collaborate in a centralized environment. The combination of Git and GitHub facilitates both local and web-based collaboration, making it a powerful and widely adopted solution in the IT community.

Creating your Account

1. **Visit the GitHub Website:**

- Open your web browser and go to https://github.com/.
- Sign Up: On the GitHub homepage, you'll find the "Sign up" or “Create an Account” button. Click on it.

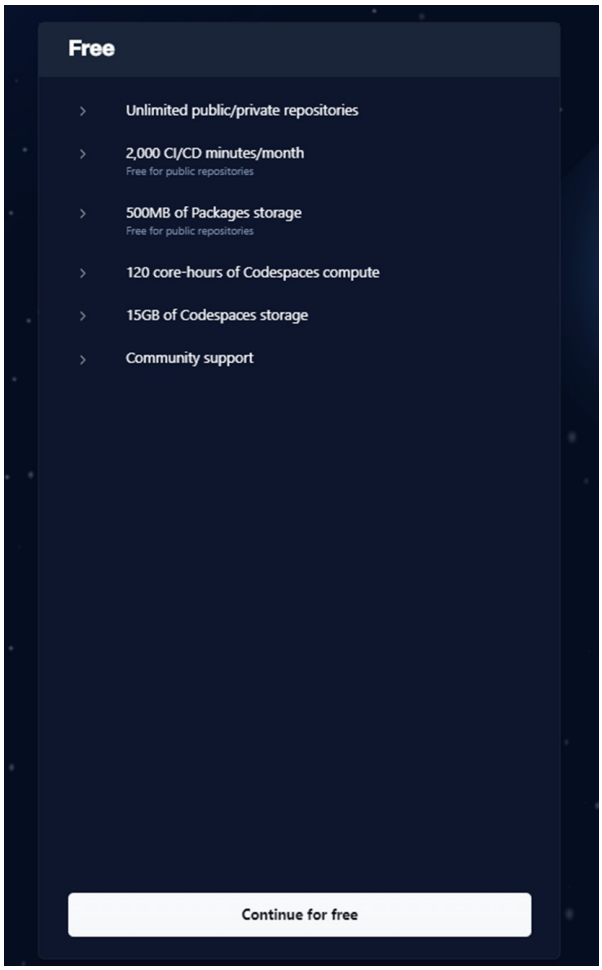


2. **Provide Account Information:**

- Fill in the required information in the provided fields. This typically includes: Your desired username, Your email address, A secure password
- Complete any CAPTCHA or verification challenges to prove that you are a human user.

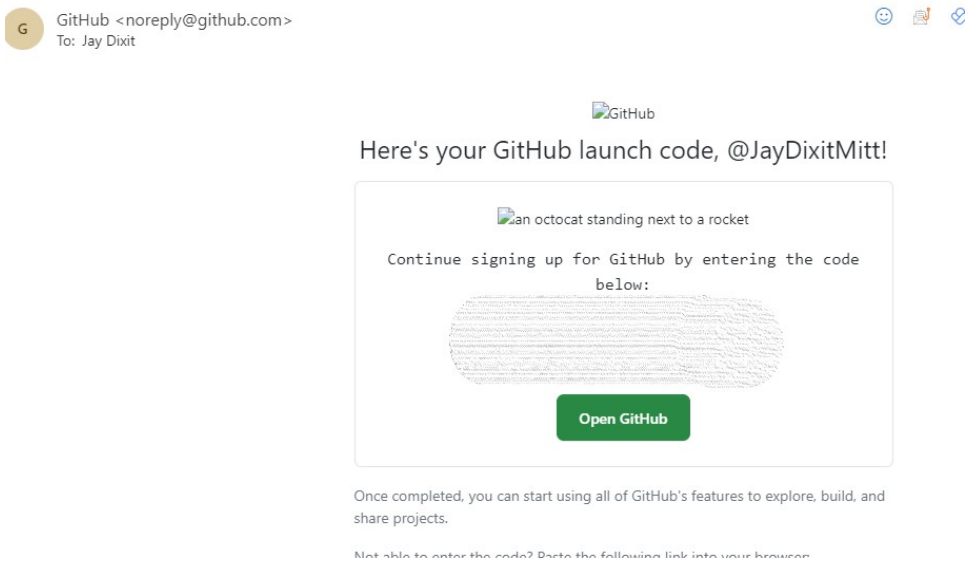
3. **Choose a Plan (Free Recommended)**

- GitHub offers different plans, including a free plan for individual developers. Select the plan that suits your needs.



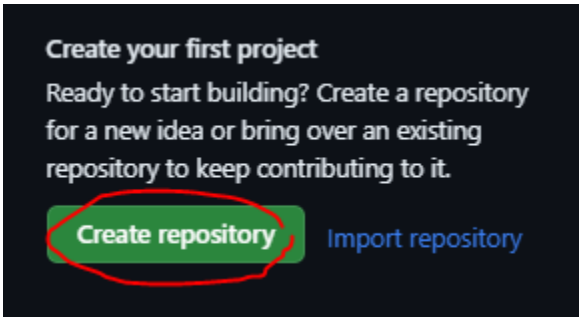
4. **Verify Your Email Address:**

- After completing the sign-up process, GitHub will send you a verification email to the email address you provided. Open the email and click on the verification link to confirm your account.

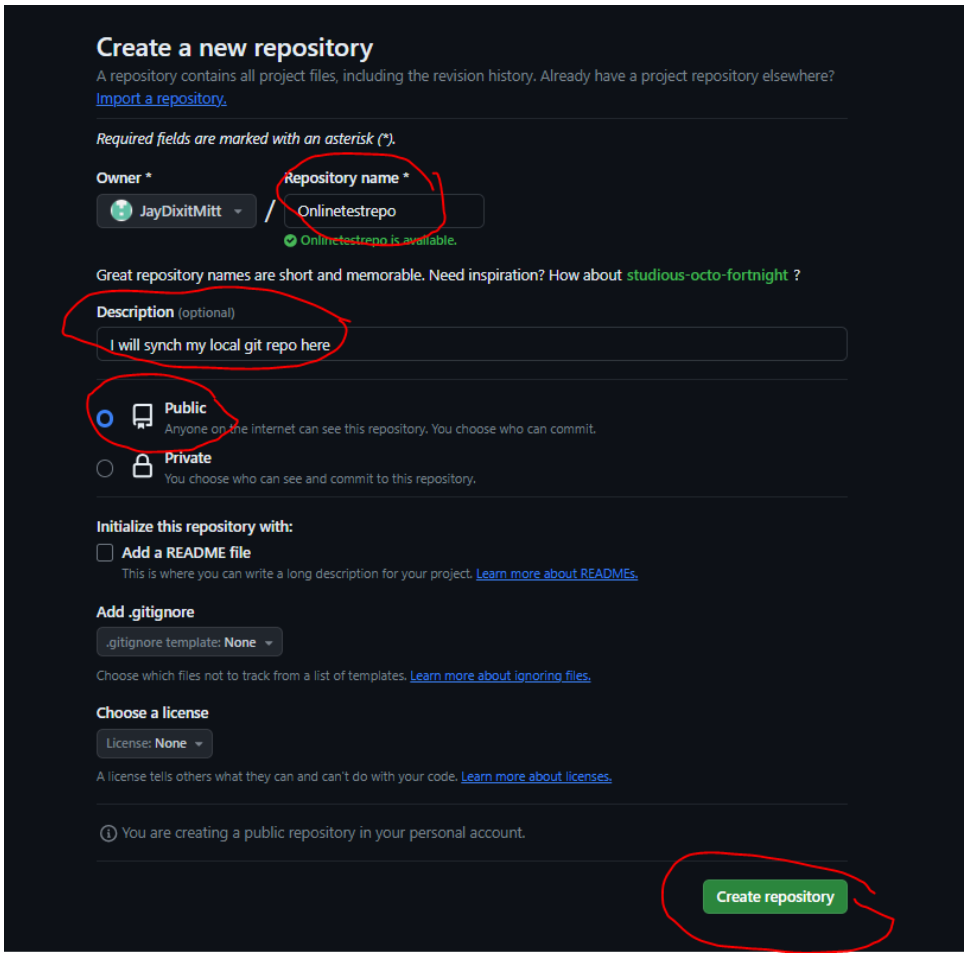


Create a new Repository on GitHub and synch it with your local Git Repository

1. Login to your GitHub account and create a new repository.

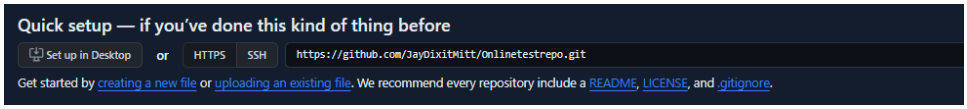


2. Provide a name, description and create this repo as a Public repo so your team members can access this repo using a Web URL. All other options can be left as is.



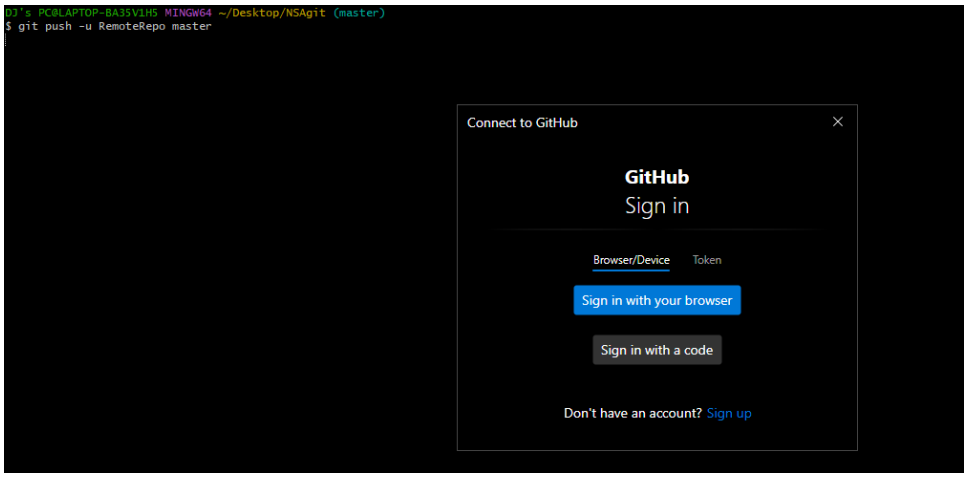
3. Go to Git Bash on your local computer and create a Remote Repo. Name does not have match the GitHub repo. The name is used to identify a remote repo on your local Git database.

Note: URL to use with this command can be found under “Quick Setup” on GitHub.




```
DJ's PC@LAPTOP-BA35V1H5 MINGW64 ~/Desktop/NSAgit (master)
$ git remote add "RemoteRepo" https://github.com/JayDixitMitt/Onlinetestrepo.git
```

4. Now synchronize your Master Repository with GitHub using “git push” command. (login to github when prompted)




```
DJ's PC@LAPTOP-BA35V1H5 MINGW64 ~/Desktop/NSAgit (master)
$ git push -u RemoteRepo master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 55.83 KiB | 18.61 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/JayDixitMitt/Onlinetestrepo.git
 * [new branch]      master -> master
branch 'master' set up to track 'RemoteRepo/master'.
```

5. Now refresh your GitHub page and you will notice that the contents of your Git Repo is available on a Web-based Github repository. This repository can be accessed by your team members using a Web-URL which makes team collaboration and document development with version control easy.