

# VERSIONING CONTROL WITH GITHUB

GitHub Desktop

Presented By : Shahir



# TOPICS



What is Versioning Control?



Why use Versioning Control?



List of versioning control tool



GitHub



GitHub Flow



Branches



Commit



Pull Request



Git Bash



Git Configuration



Demo



Practice

# OBJECTIVES



- By the end of this session, you should be able to
  - Understand the concepts of versioning or source control
  - Understand the basics of GitHub
  - Explain about GitHub flow
  - Practice GitHub usage in a project with GitHub Desktop

# PRE-REQUISITES



- Participants should have
  - Registered for a GitHub account (it's free)
  - Downloaded and installed
    - GitHub Desktop - <https://desktop.github.com/>
    - Git - <https://git-scm.com/downloads>



# VERSIONING CONTROL

Brief overview of Versioning Control



# WHAT IS VERSIONING CONTROL?



A component of Software Configuration Management



Also known as Revision Control, Source Control



Manage changes of

Documents

Computer programs/software

etc

# WHY USE VERSION CONTROL?



Easier to track changes

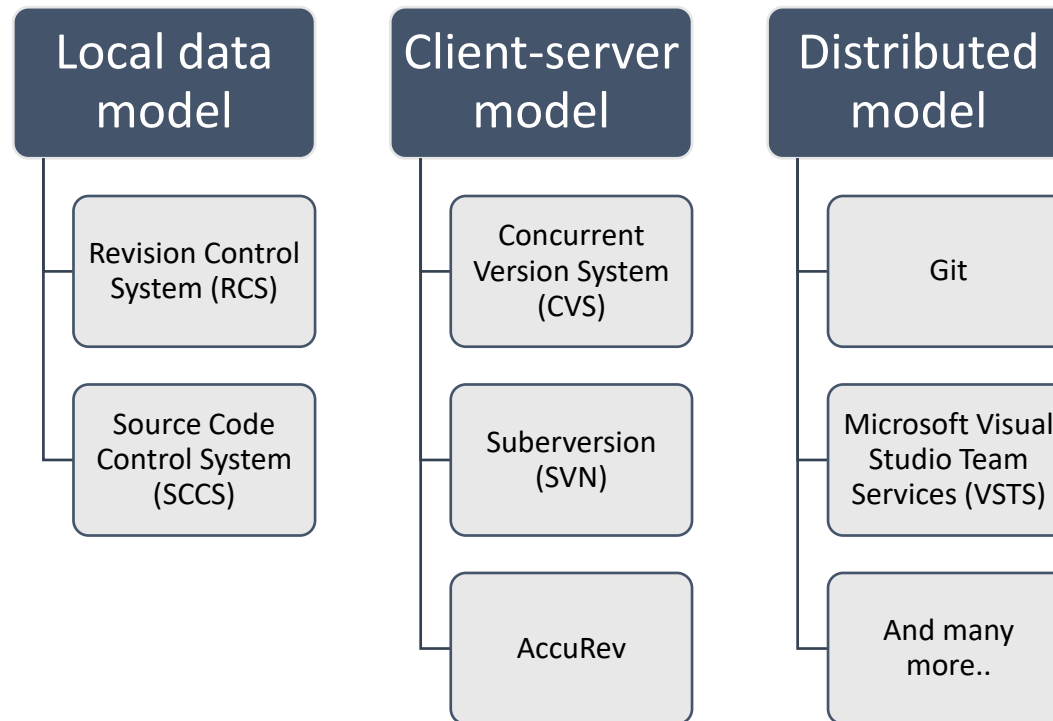
Error can be fixed more effectively as there are historical data



Team members can collaborate effectively

# EXAMPLE OF VERSIONING CONTROL TOOL

13



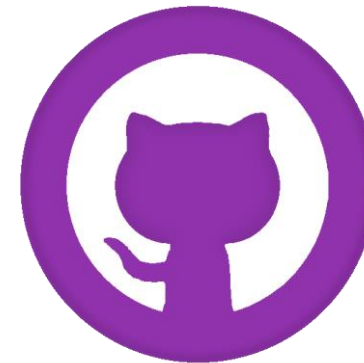


# GITHUB

Introduction to GitHub

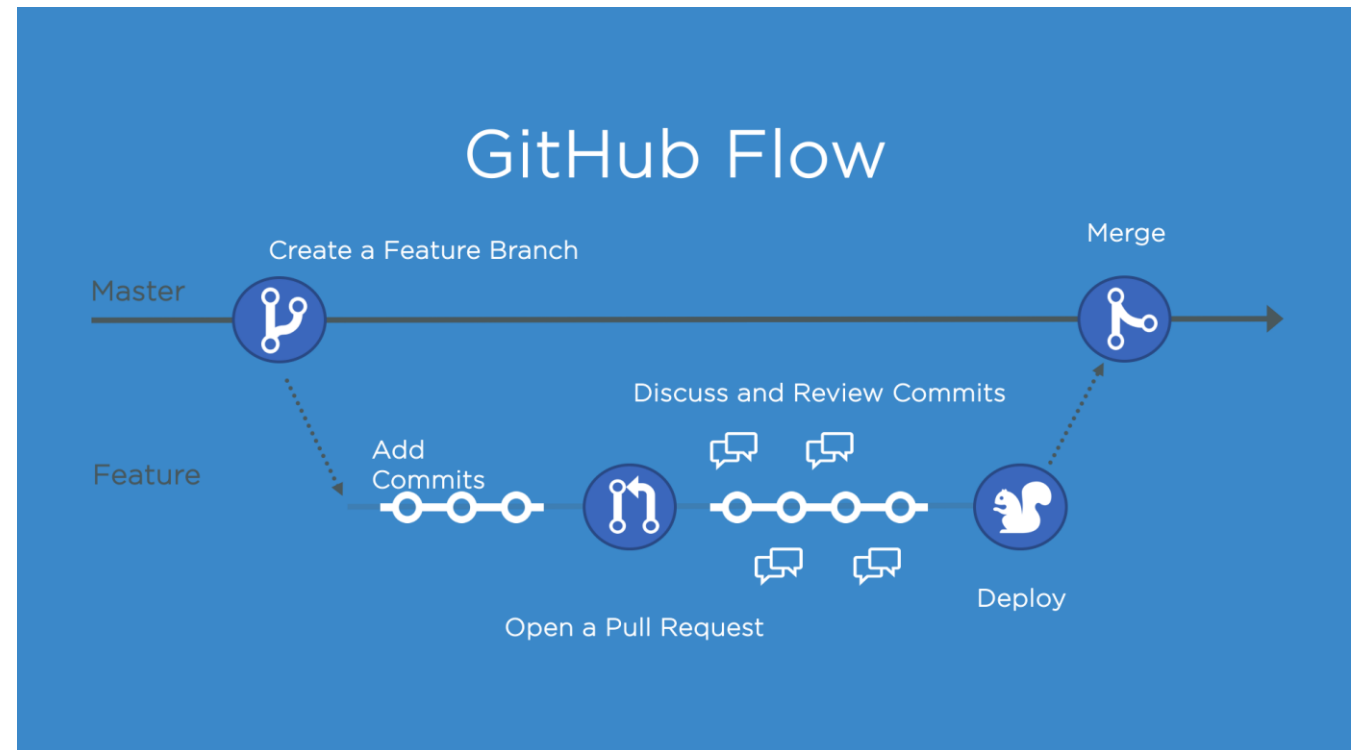


- GitHub is a Git repository hosting service
- It adds a lot of its own features
- Provides a GUI over traditional CLI
- Provides access control
- Other collaboration features:
  - Wikis
  - Basic task management tools

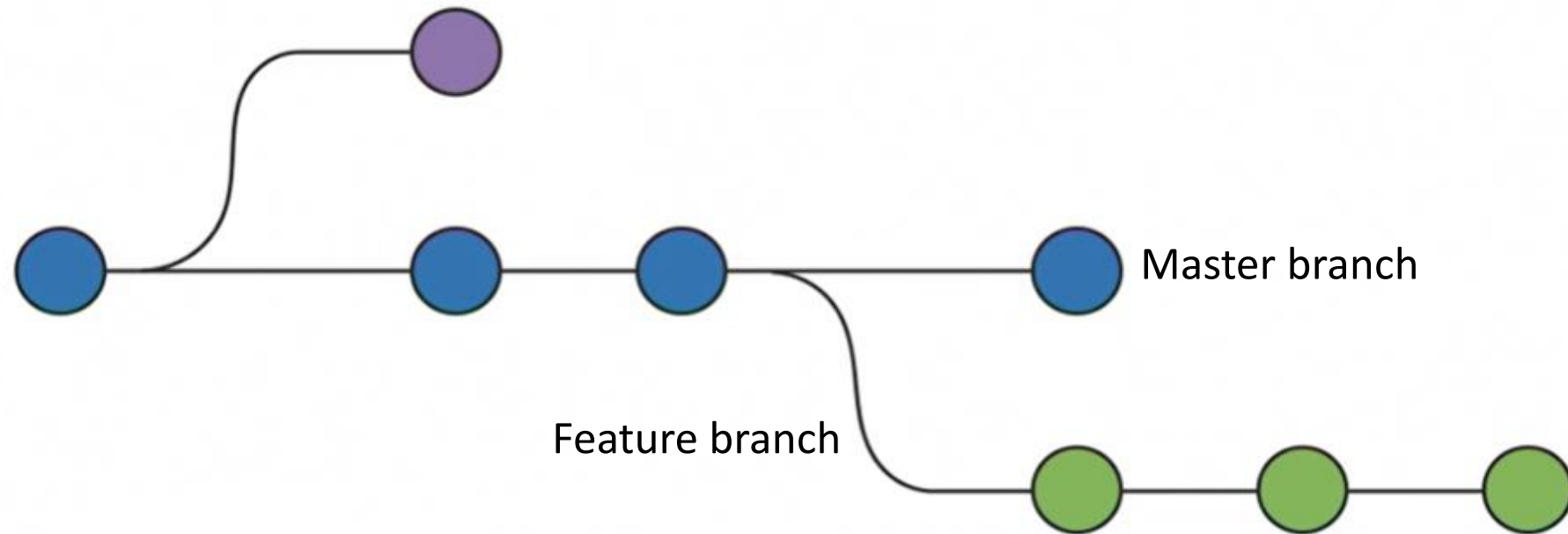


# GITHUB FLOW

- A lightweight, branch-based workflow.



# BRANCHES



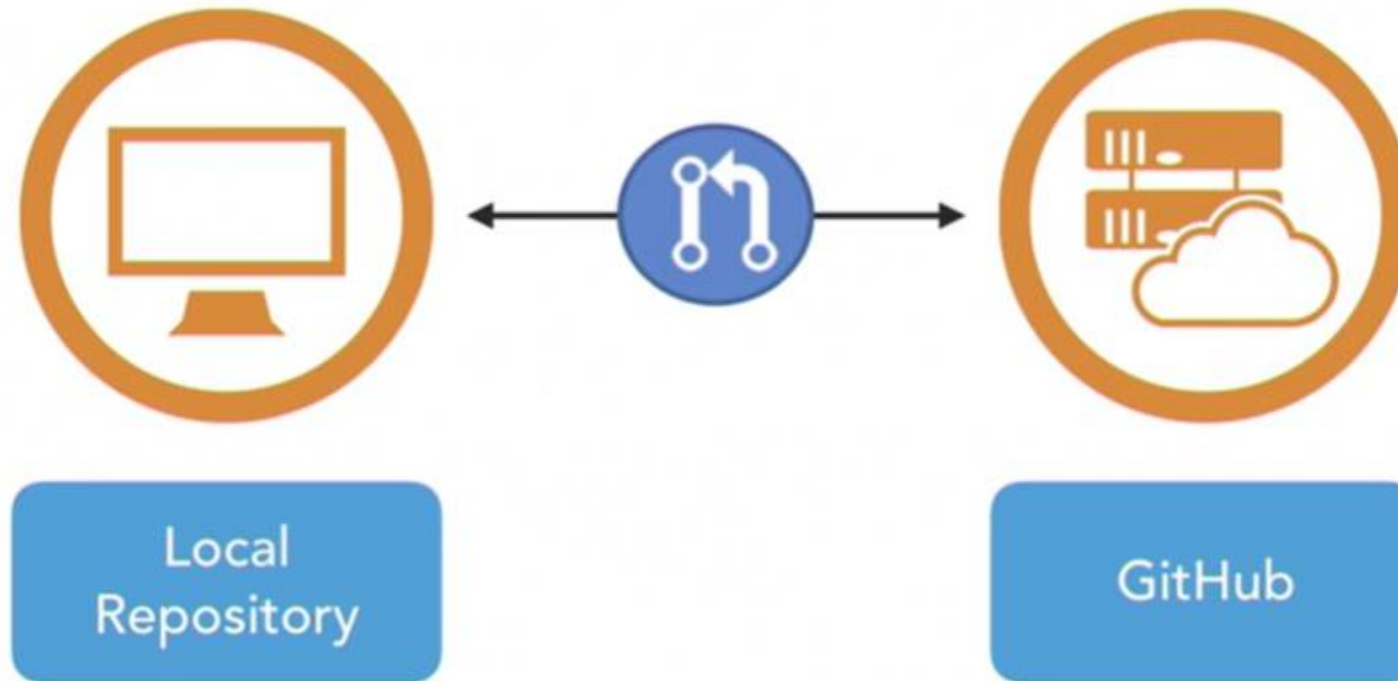
# COMMITS





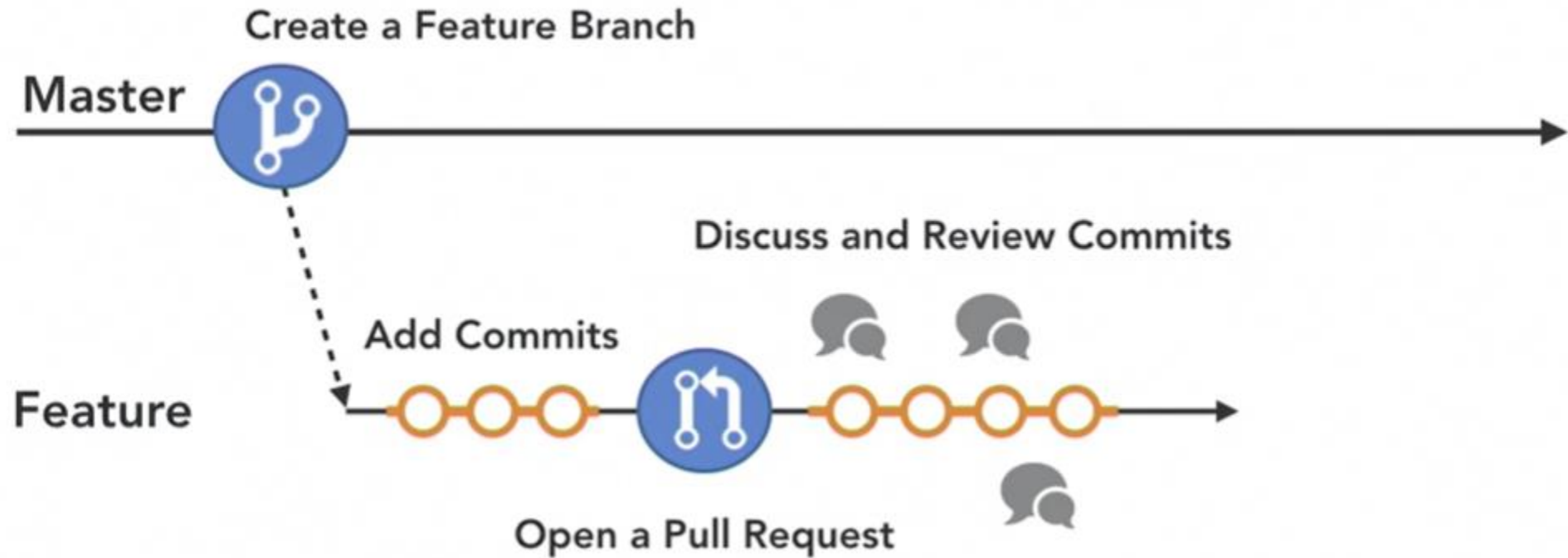
# PULL REQUEST

13



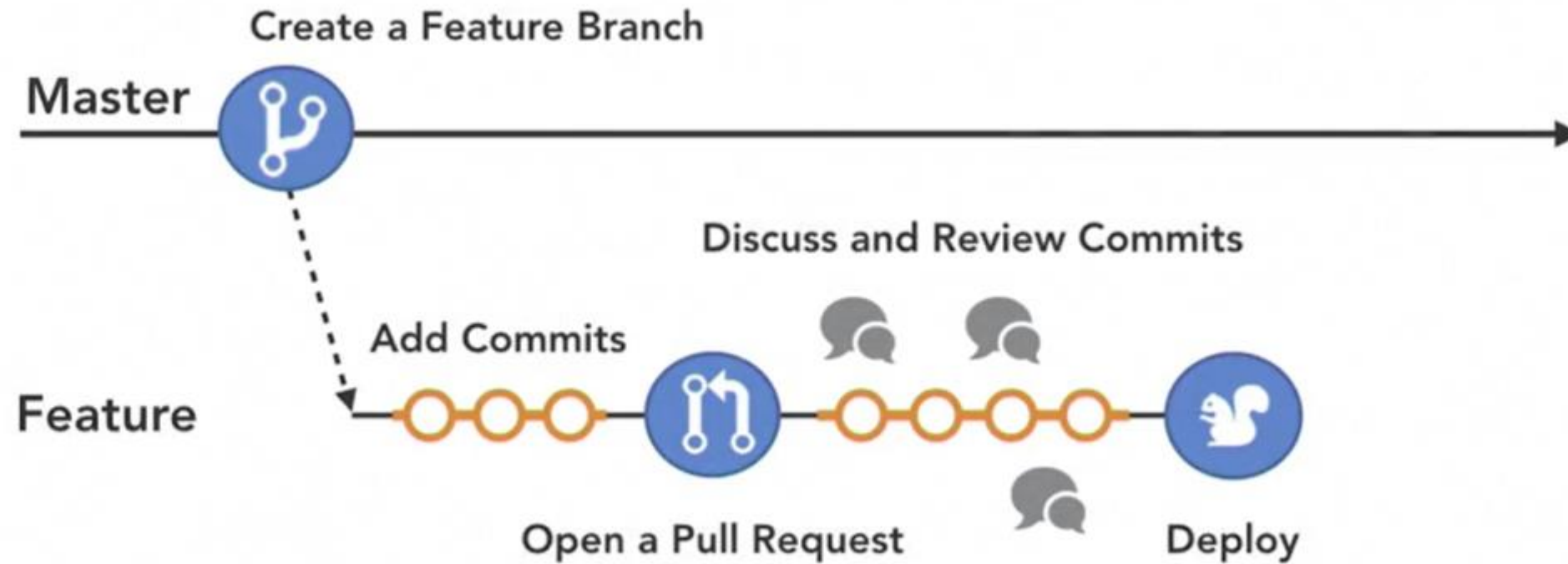
# REVIEW PULL REQUEST

13

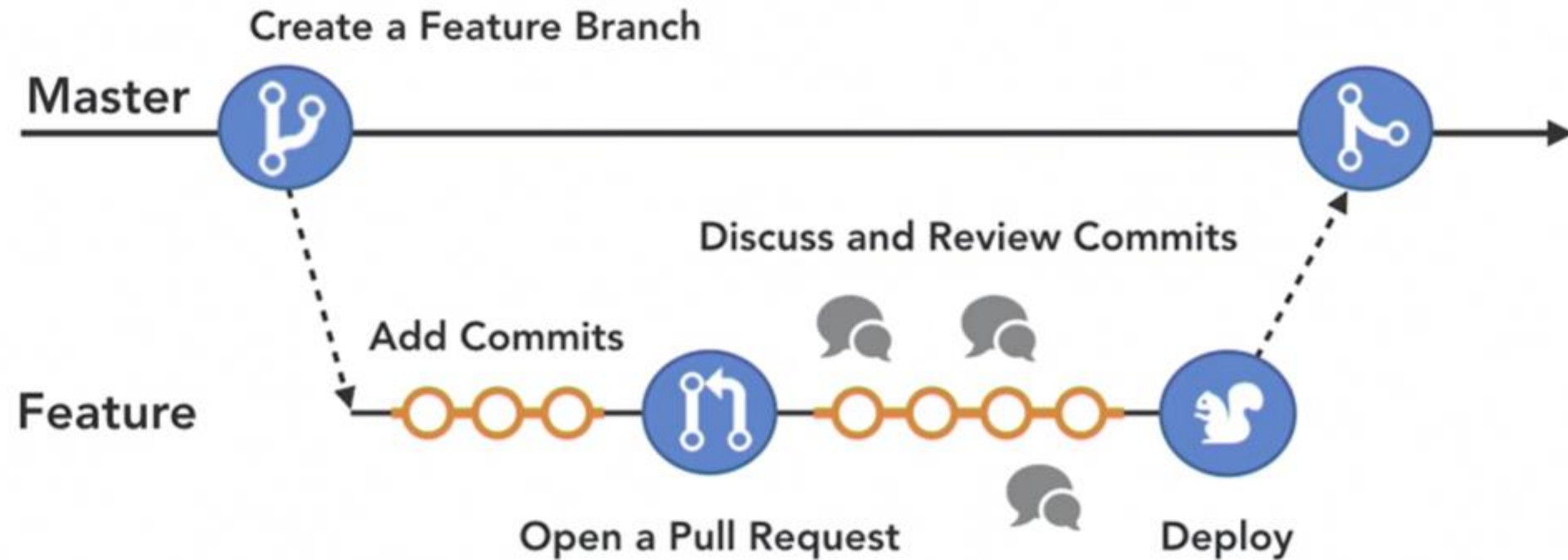


# DEPLOY PULL REQUEST

13



# MERGE PULL REQUEST



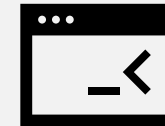
# DIFFERENT WAYS OF USING GIT/GITHUB

13



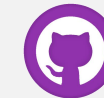
Command line interface

Git Bash



GUI Clients

GitHub Desktop  
SourceTree  
TortoiseGit  
Etc..



IDEs

Visual Studio  
Eclipse  
Etc..



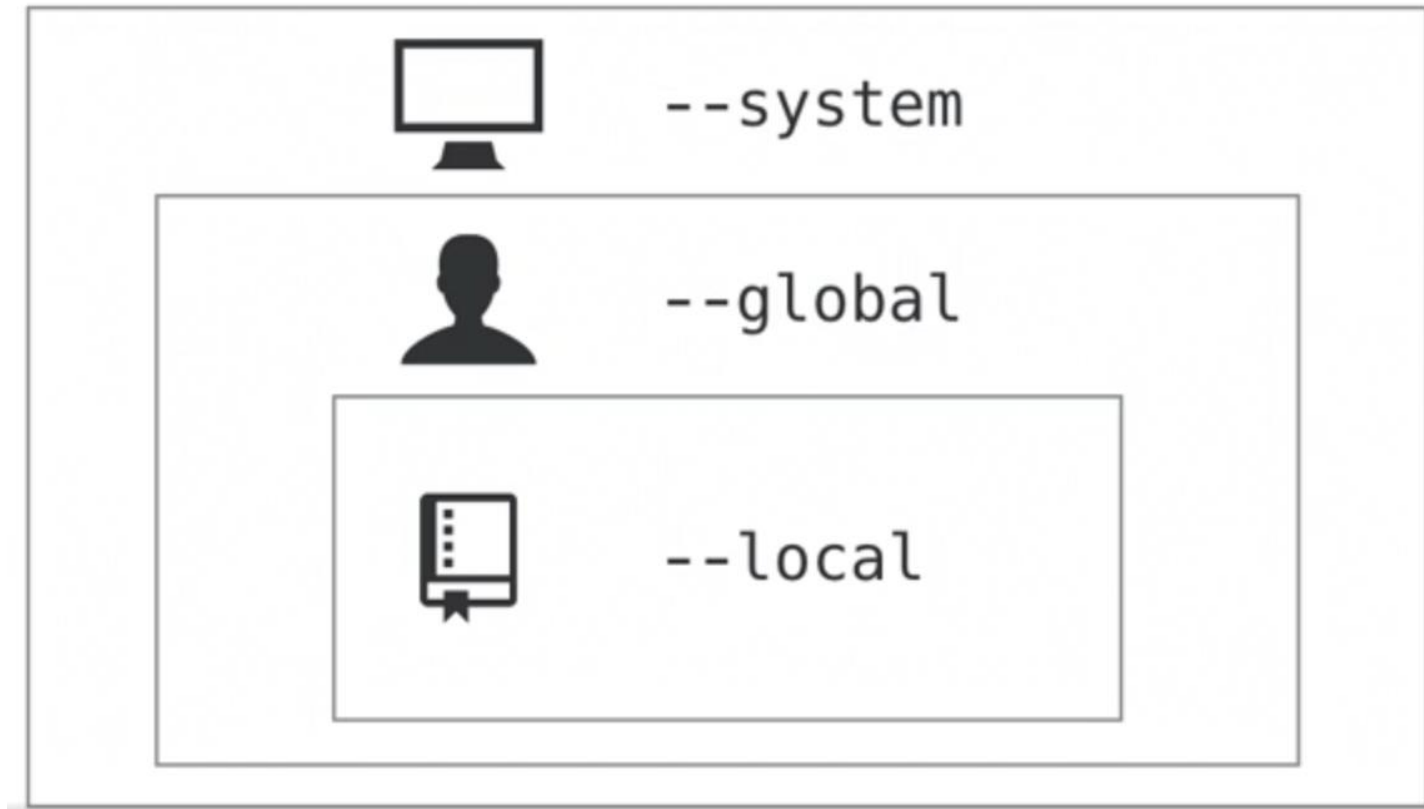


# BASIC LIST OF COMMANDS

<code>pwd</code>	• print working directory
<code>cd</code>	• change directory
<code>dir</code>	• list down content in drectory
<code>copy con</code>	• create an empty file
<code>mkdir</code>	• create an empty folder
<code>clear</code>	• to clear the console screen
<code>git – version</code>	• check if Git installed
<code>git pull</code>	• update the local repository to follow latest version in remote repository (GitHub)
<code>git add</code>	• to stage changes before commit
<code>git commit</code>	• commit changes to local branch
<code>git push</code>	• push changes done on a local branch to remote branch
<code>git checkout -- &lt;file&gt;</code>	• to undo changes on a file

# GIT CONFIGURATION LEVEL

13



# CONFIGURE USERNAME & EMAIL

13

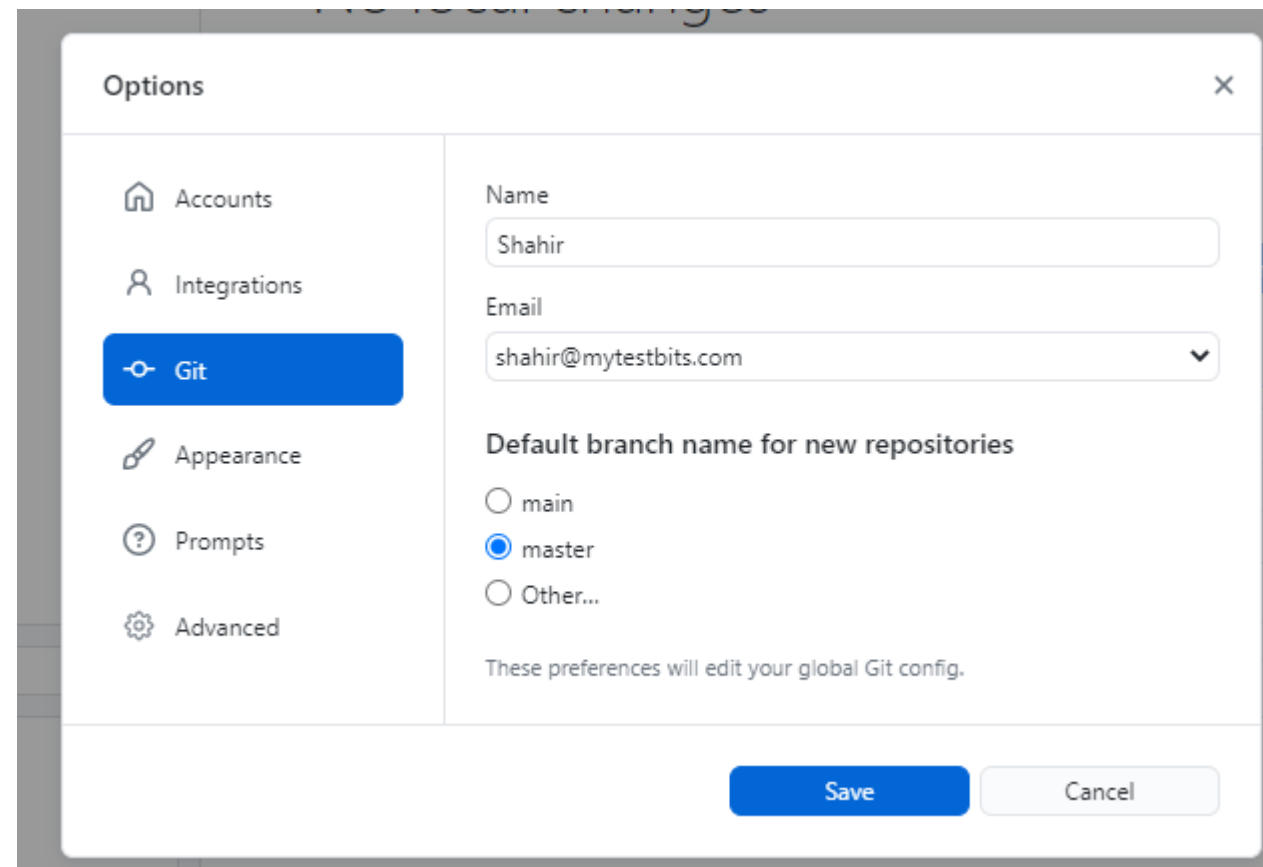
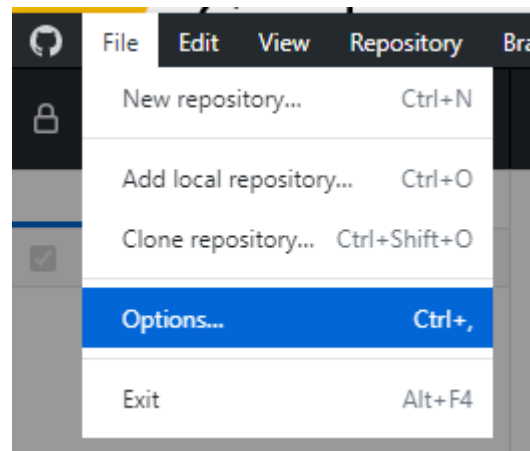
```
git config --global user.name "username"
```

```
git config --global user.email "email@example.com"
```

# CONFIGURE USERNAME & EMAIL – GITHUB DESKTOP

13

- File > Options
- On 'Options' popup, select 'Git'



# CONFIGURE DEFAULT TEXT EDITOR

13

```
git config --global core.editor "'C:\Program  
Files\Notepad++\notepad++.exe'"
```

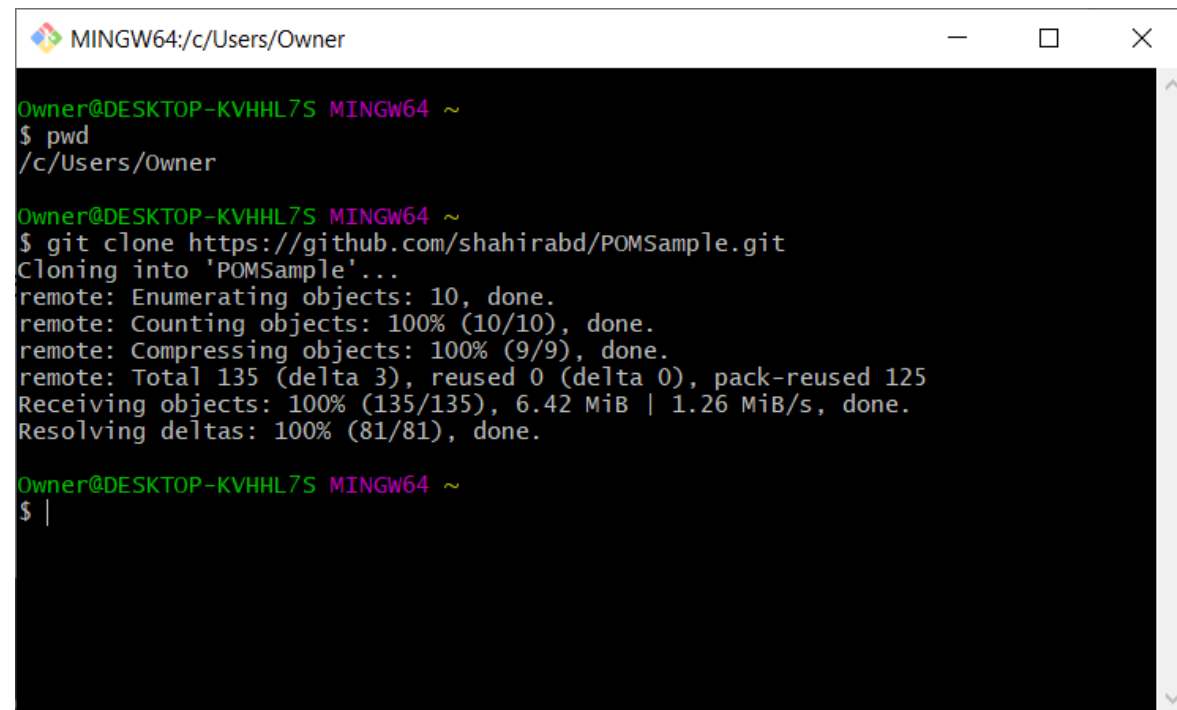


# DEMO

GitHub Desktop

# CLONE REPOSITORY - CLI

`git clone <repository URL>`

A screenshot of a Windows terminal window titled "MINGW64:/c/Users/Owner". The terminal shows the execution of the 'git clone' command. The user first runs 'pwd' and gets the path '/c/Users/Owner'. Then, they run 'git clone https://github.com/shahirabd/POMSample.git'. The terminal output shows the cloning progress: 'Cloning into 'POMSample'...', 'remote: Enumerating objects: 10, done.', 'remote: Counting objects: 100% (10/10), done.', 'remote: Compressing objects: 100% (9/9), done.', 'remote: Total 135 (delta 3), reused 0 (delta 0), pack-reused 125', 'Receiving objects: 100% (135/135), 6.42 MiB | 1.26 MiB/s, done.', and 'Resolving deltas: 100% (81/81), done.'. The prompt returns to '\$ |' at the bottom.

```
MINGW64:/c/Users/Owner

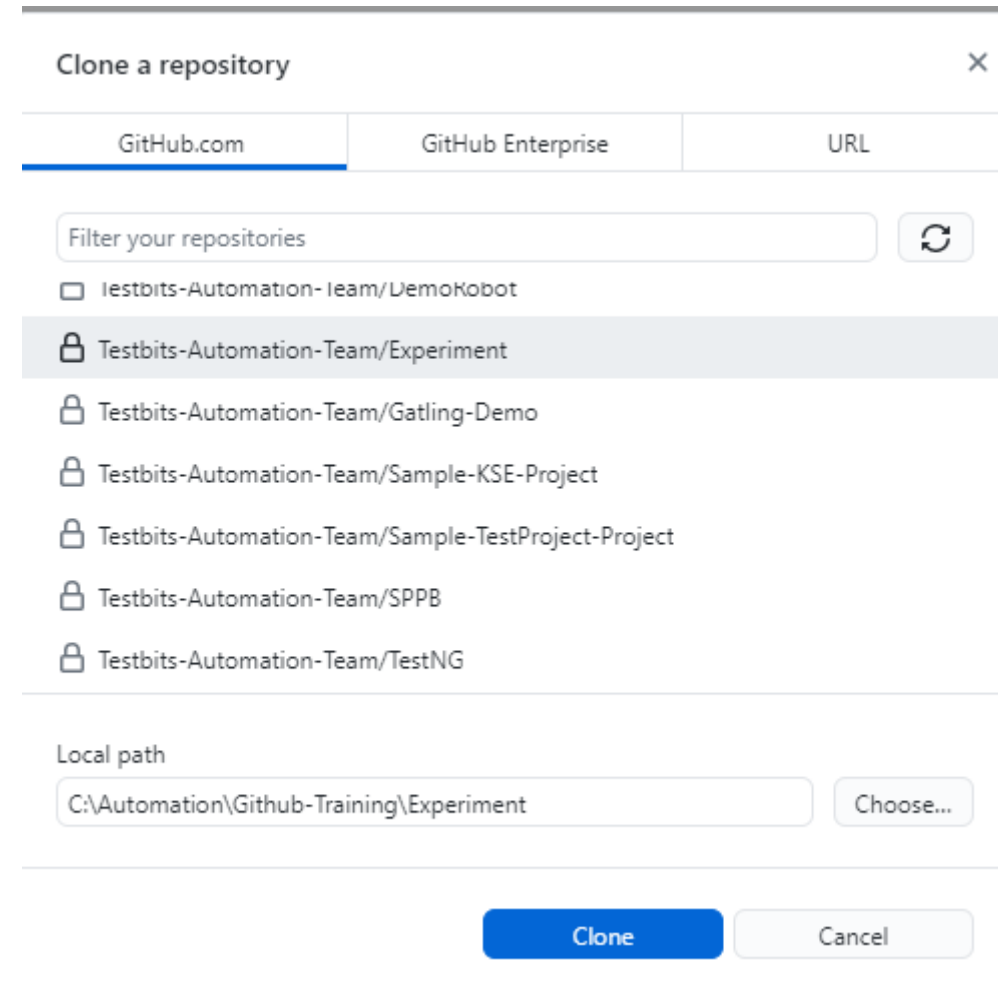
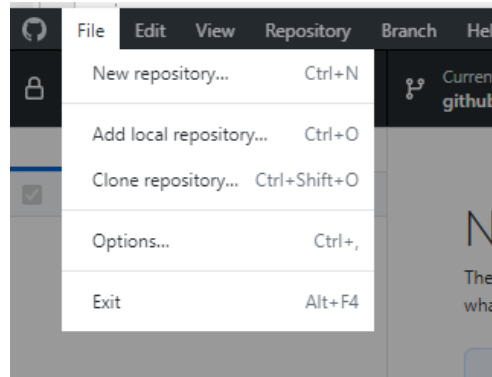
Owner@DESKTOP-KVHHL7S MINGW64 ~
$ pwd
/c/Users/Owner

Owner@DESKTOP-KVHHL7S MINGW64 ~
$ git clone https://github.com/shahirabd/POMSample.git
Cloning into 'POMSample'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 135 (delta 3), reused 0 (delta 0), pack-reused 125
Receiving objects: 100% (135/135), 6.42 MiB | 1.26 MiB/s, done.
Resolving deltas: 100% (81/81), done.

Owner@DESKTOP-KVHHL7S MINGW64 ~
$ |
```

# CLONE REPOSITORY – GITHUB DESKTOP

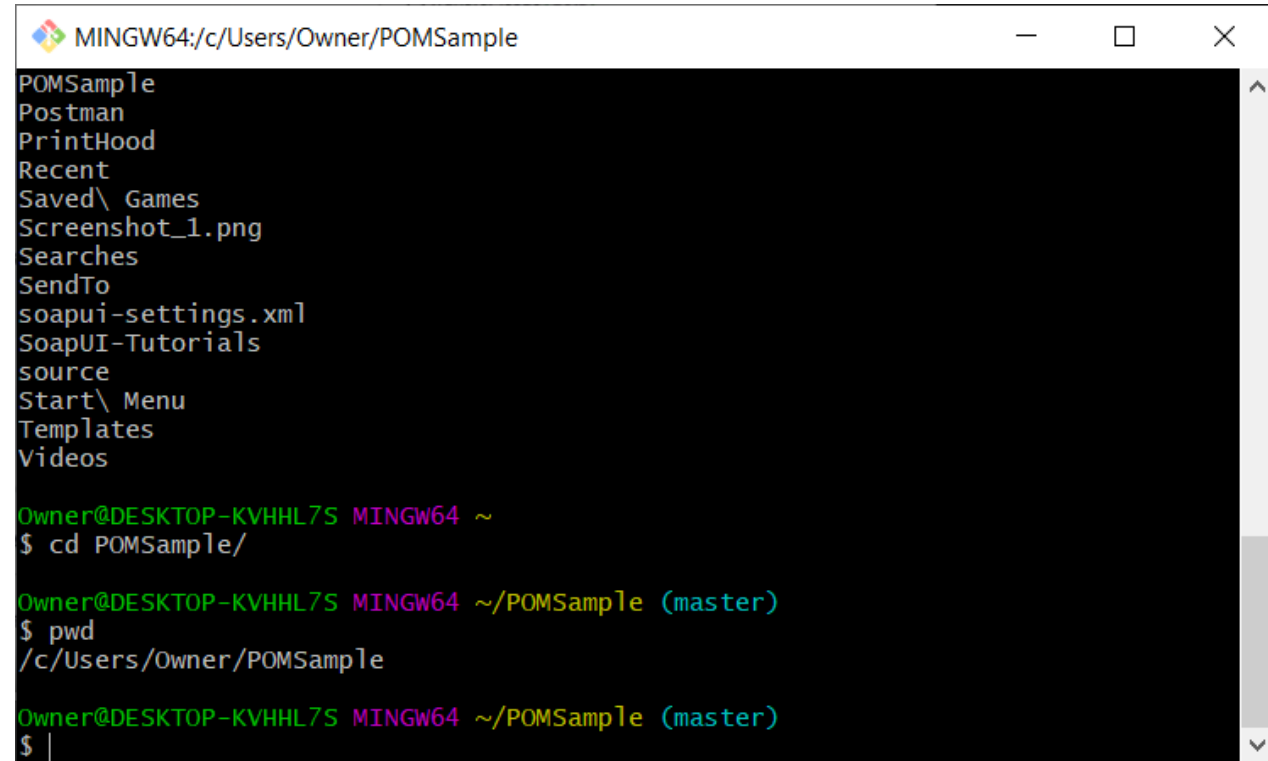
- File > Clone repository
- Select which repository you wish to clone to your local machine
- Select the local path of your repository
- Click the 'Clone' button



# CHANGE DIRECTORY - CLI

13

- `cd <folder_name>`



A screenshot of a Windows command prompt window titled "MINGW64:/c/Users/Owner/POMSample". The window shows a directory listing of the current directory, which includes folders like "POMSample", "Postman", "PrintHood", "Recent", "Saved\ Games", "Screenshots\_1.png", "Searches", "SendTo", "soapui-settings.xml", "SoapUI-Tutorials", "source", "Start\ Menu", "Templates", and "Videos". Below the listing, the user enters the command `cd POMSample/`, and the prompt changes to `Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (master)`. The user then enters `pwd`, and the output is `/c/Users/Owner/POMSample`. The prompt returns to `Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (master)` with a cursor at the end of the line.

```
MINGW64:/c/Users/Owner/POMSample
POMSample
Postman
PrintHood
Recent
Saved\ Games
Screenshots_1.png
Searches
SendTo
soapui-settings.xml
SoapUI-Tutorials
source
Start\ Menu
Templates
Videos

Owner@DESKTOP-KVHHL7S MINGW64 ~
$ cd POMSample/

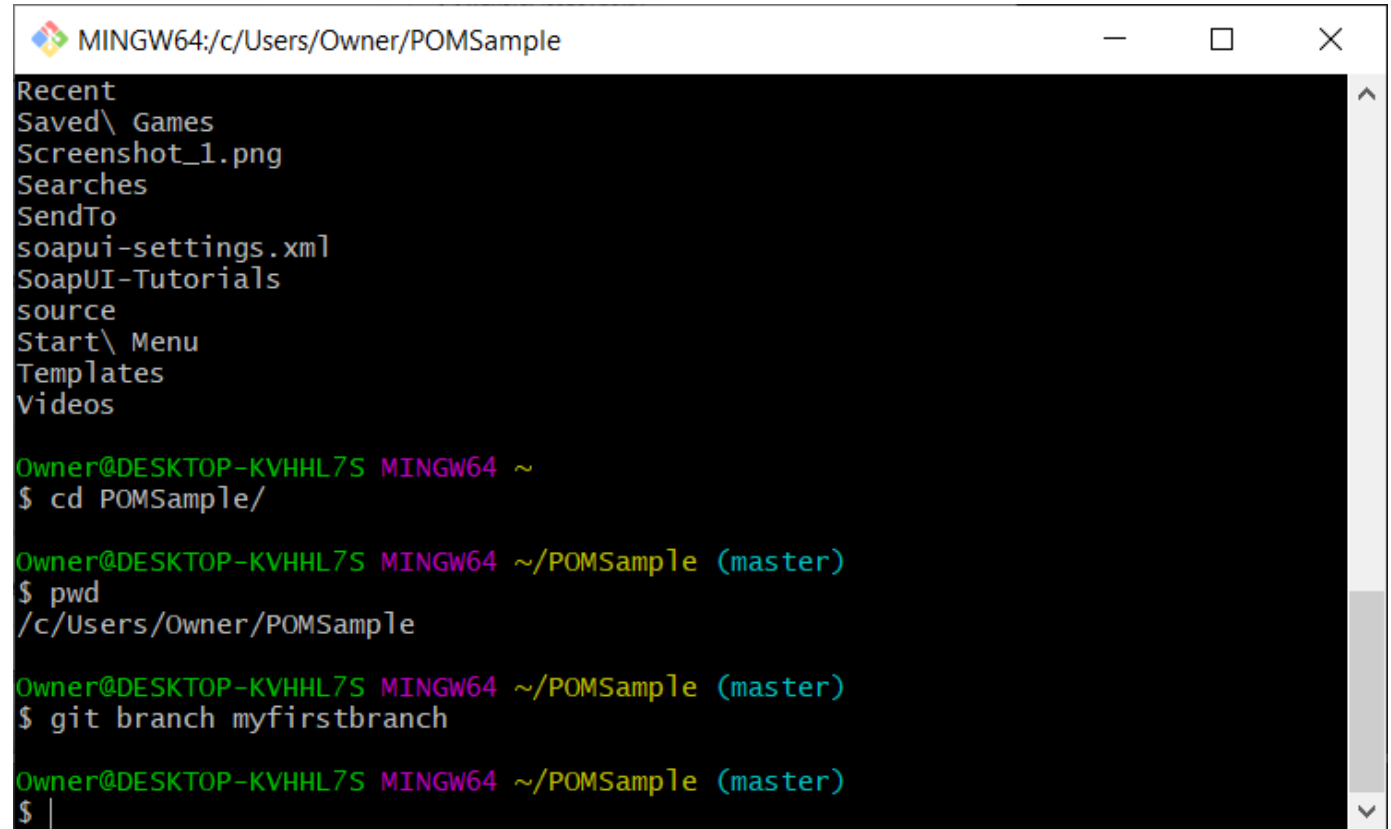
Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (master)
$ pwd
/c/Users/Owner/POMSample

Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (master)
$ |
```

# CREATE BRANCH - CLI

13

- `git branch <branch_name>`
- `git branch --all`



```
MINGW64:/c/Users/Owner/POMSample
Recent
Saved\ Games
Screenshot_1.png
Searches
SendTo
soapui-settings.xml
SoapUI-Tutorials
source
Start\ Menu
Templates
Videos

Owner@DESKTOP-KVHHL7S MINGW64 ~
$ cd POMSample/

Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (master)
$ pwd
/c/Users/Owner/POMSample

Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (master)
$ git branch myfirstbranch

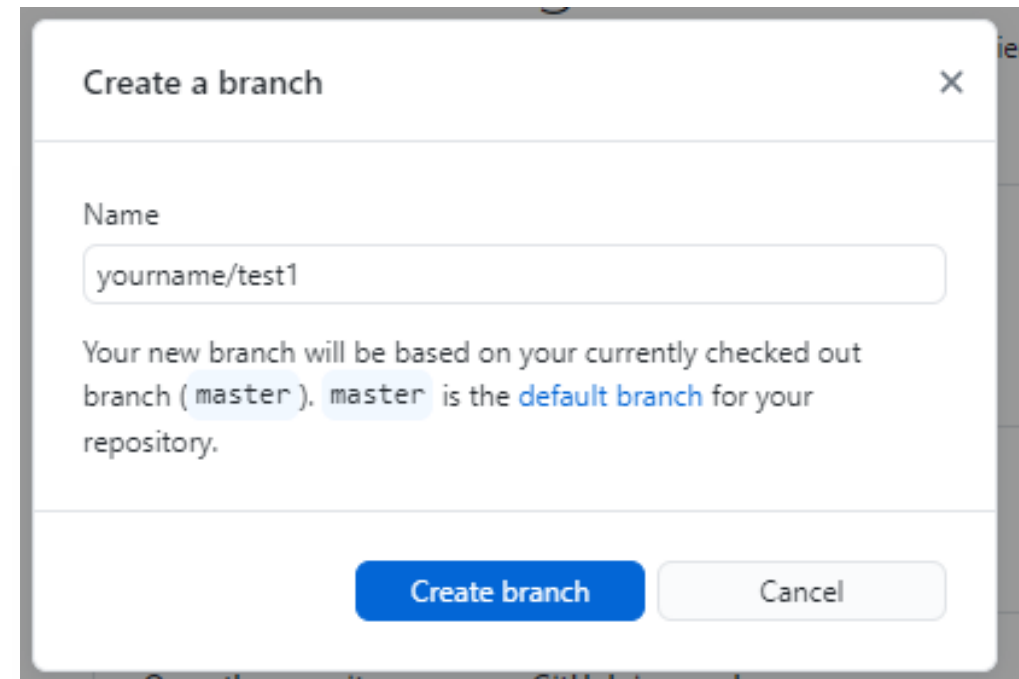
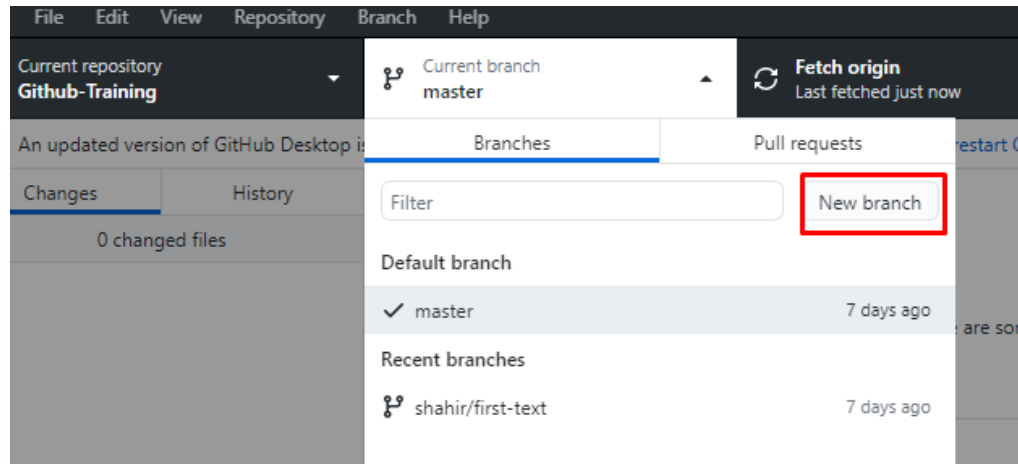
Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (master)
$ |
```



# CREATE BRANCH – GITHUB DESKTOP

13

- Click on 'Current branch' > 'New branch'



# PUBLISH THE BRANCH – CLI

13

- `git push --set-upstream origin <branch_name>`
- When prompted, enter your GitHub username and password

```
MINGW64:/c/Users/Owner/POMSample

Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (master)
$ pwd
/c/Users/Owner/POMSample

Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (master)
$ git branch myfirstbranch

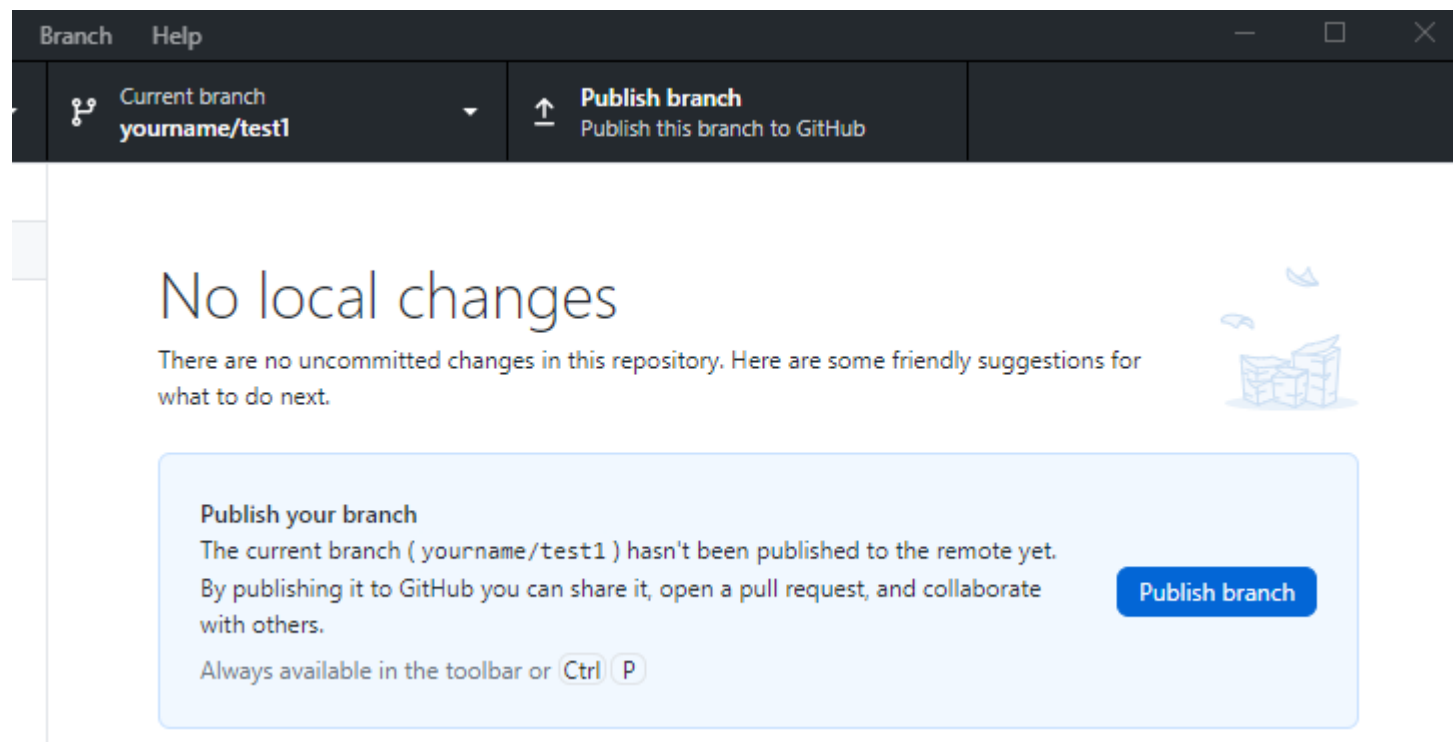
Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (master)
$ git push --set-upstream origin myfirstbranch
Logon failed, use ctrl+c to cancel basic credential prompt.
Username for 'https://github.com': shahirabd
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'myfirstbranch' on GitHub by visiting:
remote:      https://github.com/shahirabd/POMSample/pull/new/myfirstbranch
remote:
To https://github.com/shahirabd/POMSample.git
 * [new branch]      myfirstbranch -> myfirstbranch
Branch 'myfirstbranch' set up to track remote branch 'myfirstbranch' from 'origin'.

Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (master)
$
```

# PUBLISH THE BRANCH – GITHUB DESKTOP

13

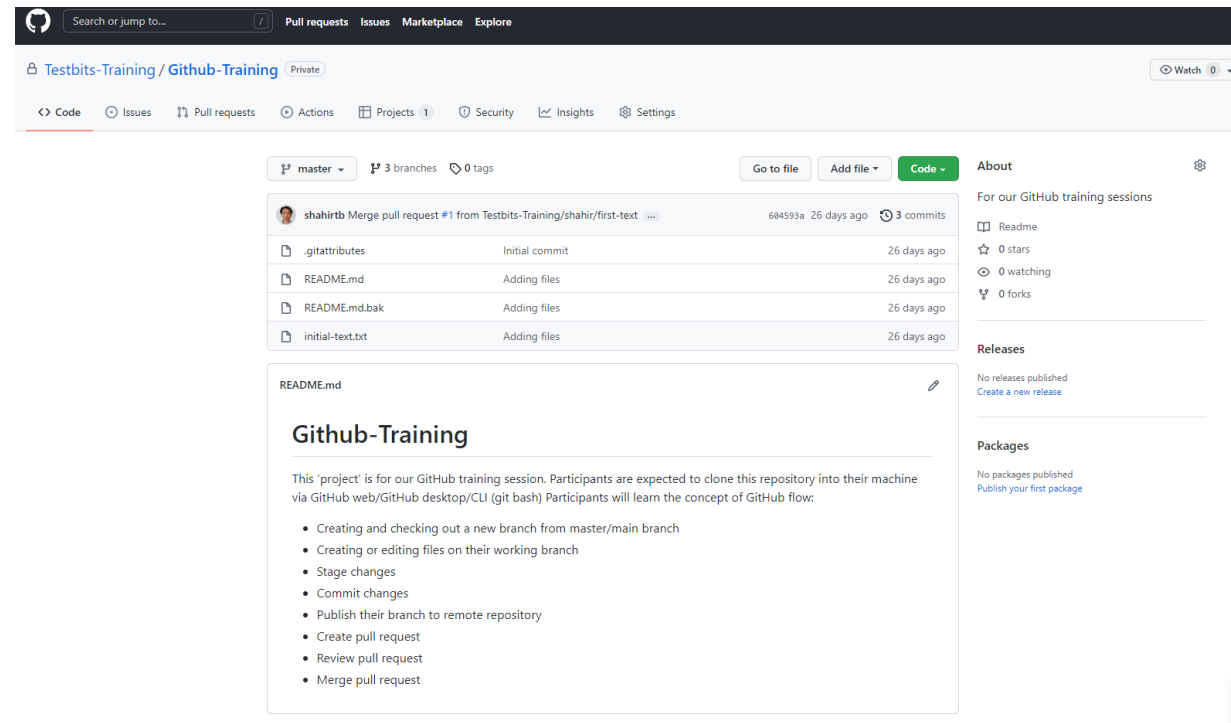
- Click on 'Publish branch' button



# BROWSE TO GITHUB PORTAL

13

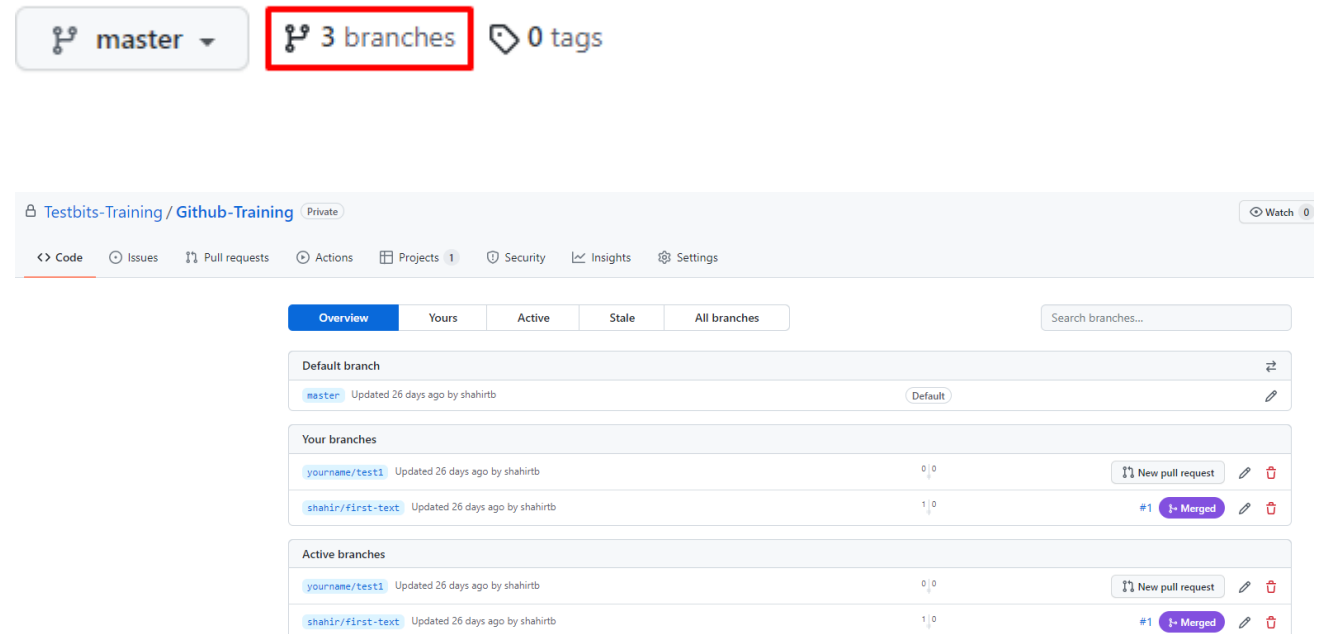
- Since we are not making any changes to the project, there are no changes displayed



# BROWSE TO GITHUB PORTAL

13

- View your branch and other branches in the repository
- Click on the 'branches' link



The screenshot shows the GitHub repository page for `Testbits-Training / Github-Training`. The repository is private. The navigation bar includes links for Code, Issues, Pull requests, Actions, Projects (1), Security, Insights, and Settings. The 'branches' tab is selected, and the '3 branches' link is highlighted with a red box. Below the tabs, there are sections for 'Default branch' (master), 'Your branches' (yourname/test1, shahir/first-text), and 'Active branches' (yourname/test1, shahir/first-text). Each branch entry shows the branch name, the user who updated it, the time ago, and the commit count. The 'shahir/first-text' branch is marked as merged.

master 3 branches 0 tags

Testbits-Training / Github-Training Private Watch 0

<> Code Issues Pull requests Actions Projects 1 Security Insights Settings

Overview Yours Active Stale All branches Search branches...

Default branch

master Updated 26 days ago by shahirb Default

Your branches

yourname/test1	Updated 26 days ago by shahirb	0   0	New pull request		
shahir/first-text	Updated 26 days ago by shahirb	1   0	#1 Merged		

Active branches

yourname/test1	Updated 26 days ago by shahirb	0   0	New pull request		
shahir/first-text	Updated 26 days ago by shahirb	1   0	#1 Merged		



# CHECKOUT BRANCH - CLI

13

- `git checkout <your_branch_name>`

```
MINGW64:/c/Users/Owner/POMSample
remote:
remote: Create a pull request for 'myfirstbranch' on GitHub by visiting:
remote:   https://github.com/shahirabd/POMSample/pull/new/myfirstbranch
remote:
To https://github.com/shahirabd/POMSample.git
* [new branch]      myfirstbranch -> myfirstbranch
Branch 'myfirstbranch' set up to track remote branch 'myfirstbranch' from 'origin'.

Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (master)
$ git branch --all
* master
  myfirstbranch
remotes/origin/HEAD -> origin/master
remotes/origin/master
remotes/origin/myfirstbranch

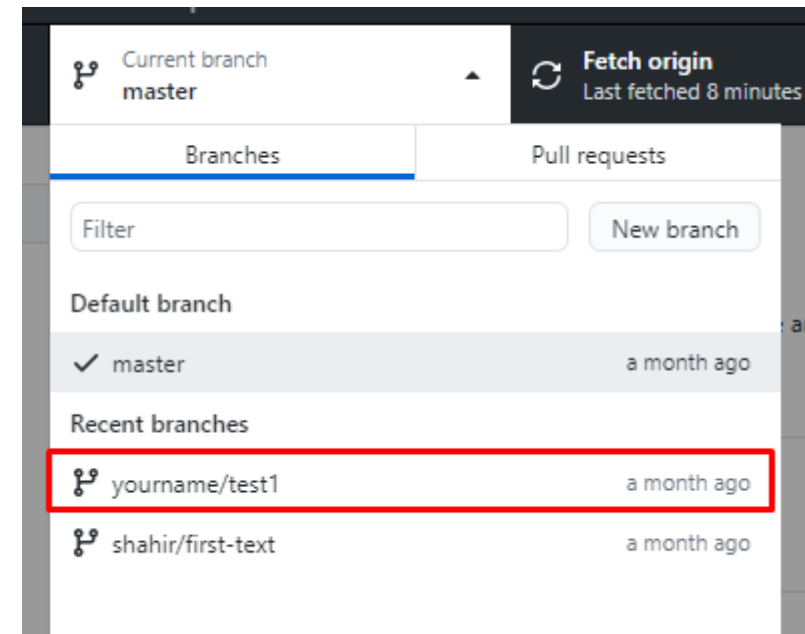
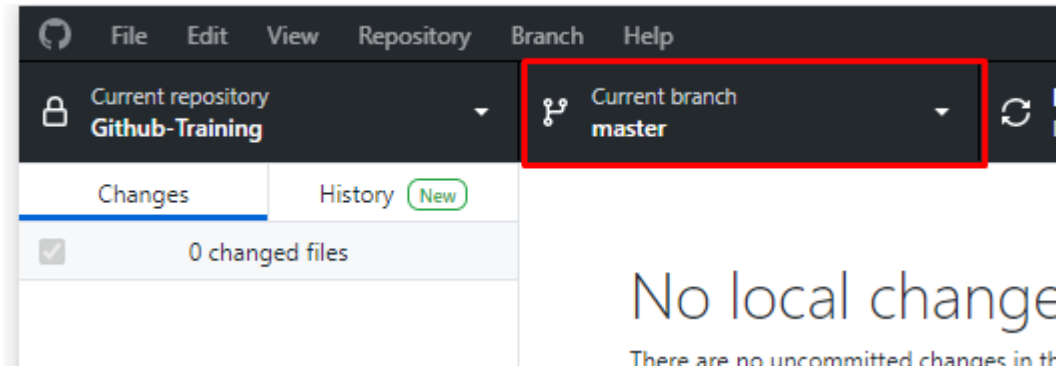
Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (master)
$ git checkout myfirstbranch
Switched to branch 'myfirstbranch'
Your branch is up to date with 'origin/myfirstbranch'.

Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (myfirstbranch)
$ |
```

# CHECKOUT BRANCH – GITHUB DESKTOP

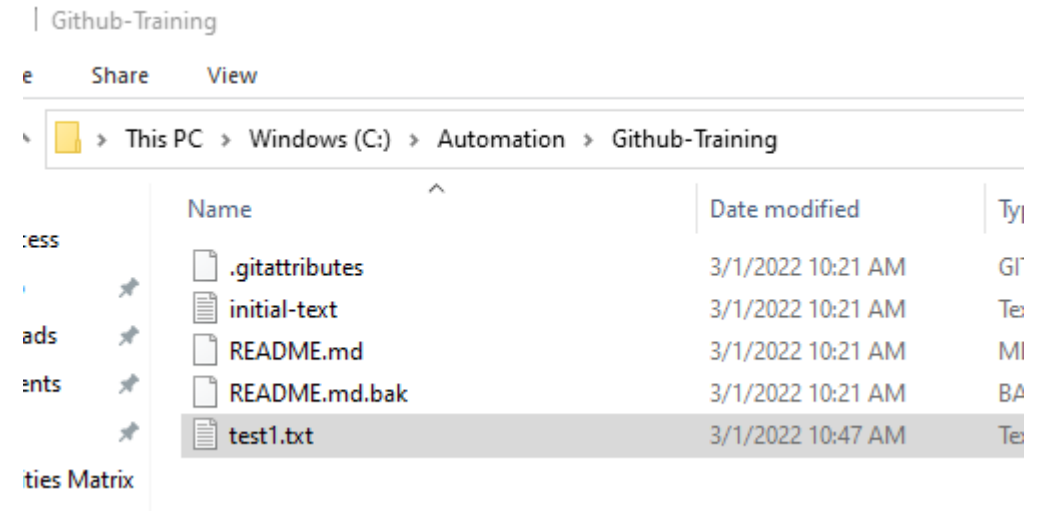
13

- Click on 'Current branch' > 'yourname/test1' branch or your branch name



# ADD A NEW FILE TO THE PROJECT FOLDER

- Open the project folder in Windows Explorer
- Add a text file under 'Github-Training' folder
- Open the new text file and add some text in there



# STAGE THE FILE - CLI

13

- Go back to Git Bash console
- To view the status
  - git status
- To stage the file
  - git add <filename>

```
MINGW64:/c/Users/Owner/POMSample
(use "git checkout -- <file>..." to discard changes in working directory)

        modified:   bin/Debug/netcoreapp3.1/nunit_random_seed.tmp

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .vs/
        Tests/UnitTest2.cs
        obj/Debug/netcoreapp3.1/POMSample.assets.cache
        obj/Debug/netcoreapp3.1/POMSample.csprojAssemblyReference.cache
        obj/POMSample.csproj.nuget.dgspec.json
        obj/POMSample.csproj.nuget.g.props
        obj/POMSample.csproj.nuget.g.targets
        obj/project.assets.json
        obj/project.nuget.cache

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

Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (myfirstbranch)
$ git add Tests/UnitTest2.cs

Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (myfirstbranch)
$ |
```

# CHECK STATUS - CLI

13

- To view the status
  - git status
- Notice that the file we staged is in green font

```
MINGW64:/c/Users/Owner/POMSample
$ git add Tests/UnitTest2.cs

Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (myfirstbranch)
$ git status
On branch myfirstbranch
Your branch is up to date with 'origin/myfirstbranch'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   Tests/UnitTest2.cs

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   bin/Debug/netcoreapp3.1/nunit_random_seed.tmp

Untracked files:
  (use "git add <file>..." to include in what will be committed)

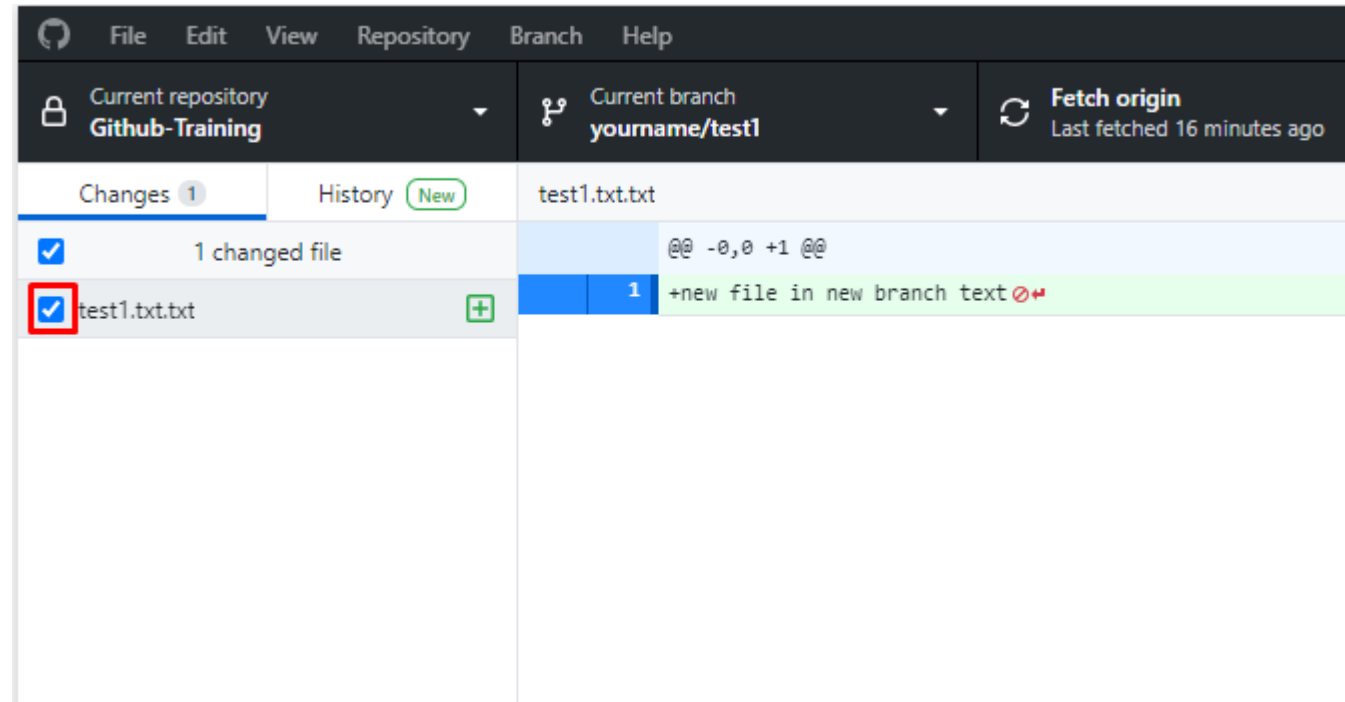
    .vs/
    obj/Debug/netcoreapp3.1/POMSample.assets.cache
    obj/Debug/netcoreapp3.1/POMSample.csprojAssemblyReference.cache
```



# STAGE THE FILE – GITHUB DESKTOP

13

- Tick the checkbox on the left to stage the file



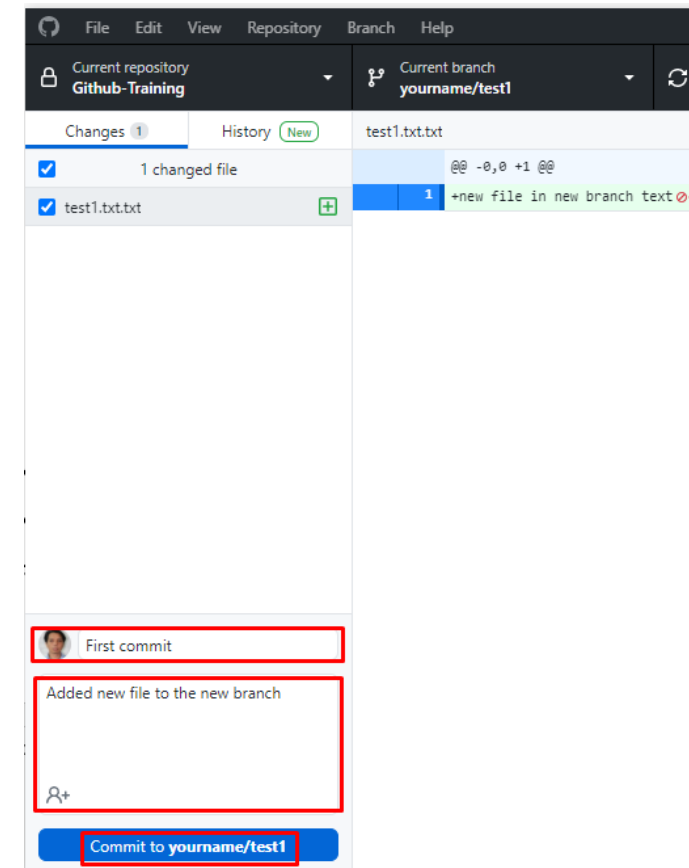
# COMMIT – CLI

- `git commit -m "file_name"`

```
MINGW64:/c/Users/Owner/POMSample
Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (myfirstbranch)
$ git commit -m "Add new test class"
[myfirstbranch 9d56dca] Add new test class
1 file changed, 10 insertions(+)
create mode 100644 Tests/UnitTest2.cs
Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (myfirstbranch)
$
```

# COMMIT – GITHUB DESKTOP

- Enter the commit title and summary, then click on the 'Commit to..' button



# PUSH – CLI

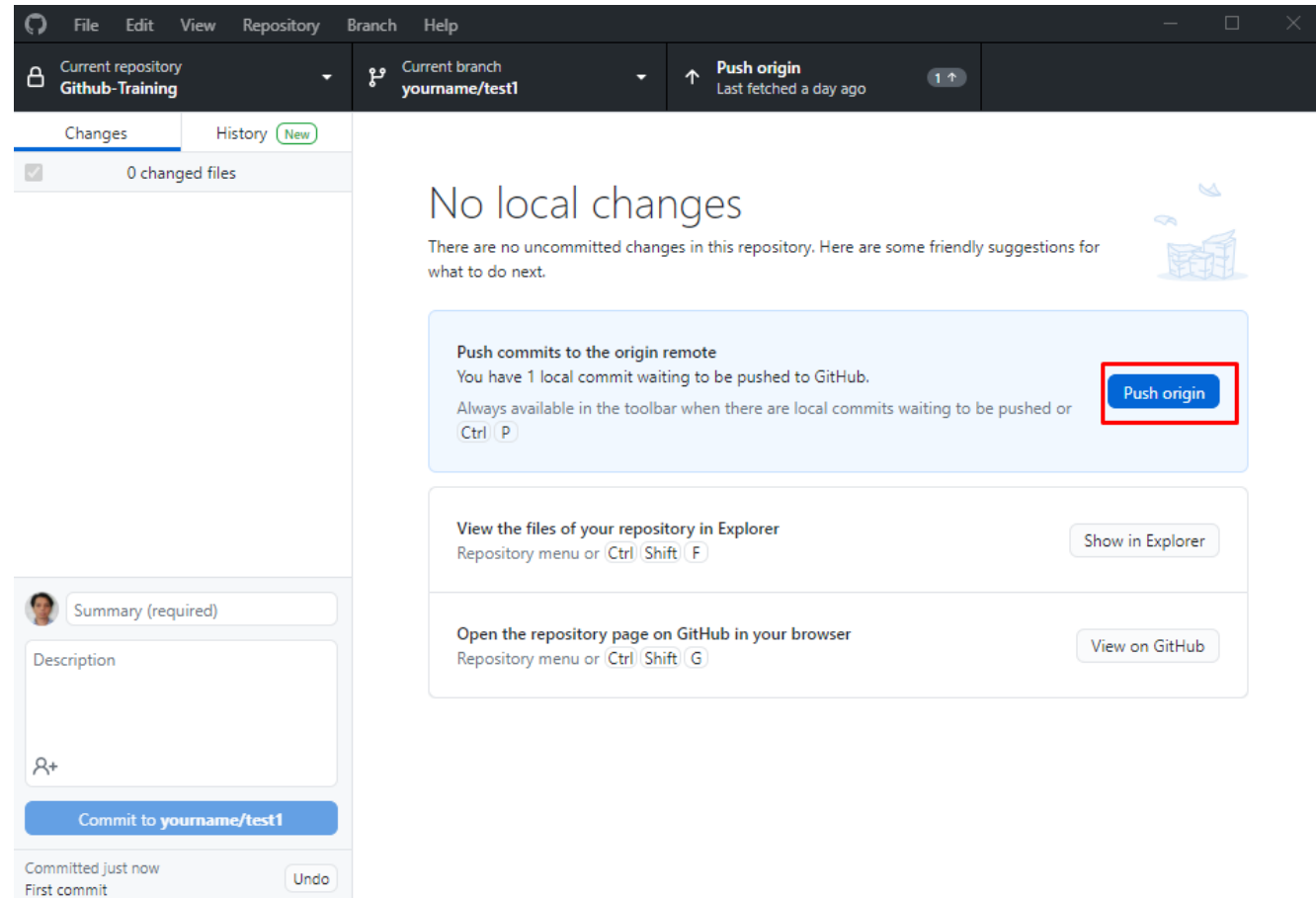
- git push

```
MINGW64; c:/Users/Owner/POMSample
Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (myfirstbranch)
$ git push
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% (4/4), 434 bytes | 434.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/shahirabd/POMSample.git
   f515e2a..9d56dca  myfirstbranch -> myfirstbranch

Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (myfirstbranch)
$ |
```

# PUSH – GITHUB DESKTOP

- Click on the 'Push origin' button







# DEMO RECAP

GitHub Desktop

# STATES OF A FILE

13



Working



Staging



History

# WORKING DIRECTORY

13



Working

New



Modified



Staging



History

# STAGING AREA

13



Working

New

Modified



Staging



History

# COMMITTED HISTORY

13



Working

New

Modified



Staging



History







# DEMO

Pull Requests

# CREATE PULL REQUEST

- Go to your repository in GitHub
- You can see the new branch and the green button 'Compare and pull request', click on it

The screenshot shows the GitHub interface for the repository 'shahirabd / POMSample'. The repository has 4 commits, 2 branches, 0 packages, 0 releases, and 1 contributor. A new branch 'myfirstbranch' was pushed 43 minutes ago. A green button labeled 'Compare & pull request' is visible next to the branch name. Below the repository information, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. The file list shows 'Constants' and 'PageObjects' with their respective upload times.

File	Upload Time
Constants	23 hours ago
PageObjects	22 hours ago

# CREATE PULL REQUEST

- Once clicked, this page will be opened
- We can add comment and perform other actions like adding reviewers etc
- Make sure the base branch is 'master'

## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: master

compare: myfirstbranch

✓ Able to merge. These branches can be automatically merged.

Add new test class

Write Preview

AA B i “ < > @

Test comment

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Linked issues

Use [Closing keywords](#) in the description to automatically close issues

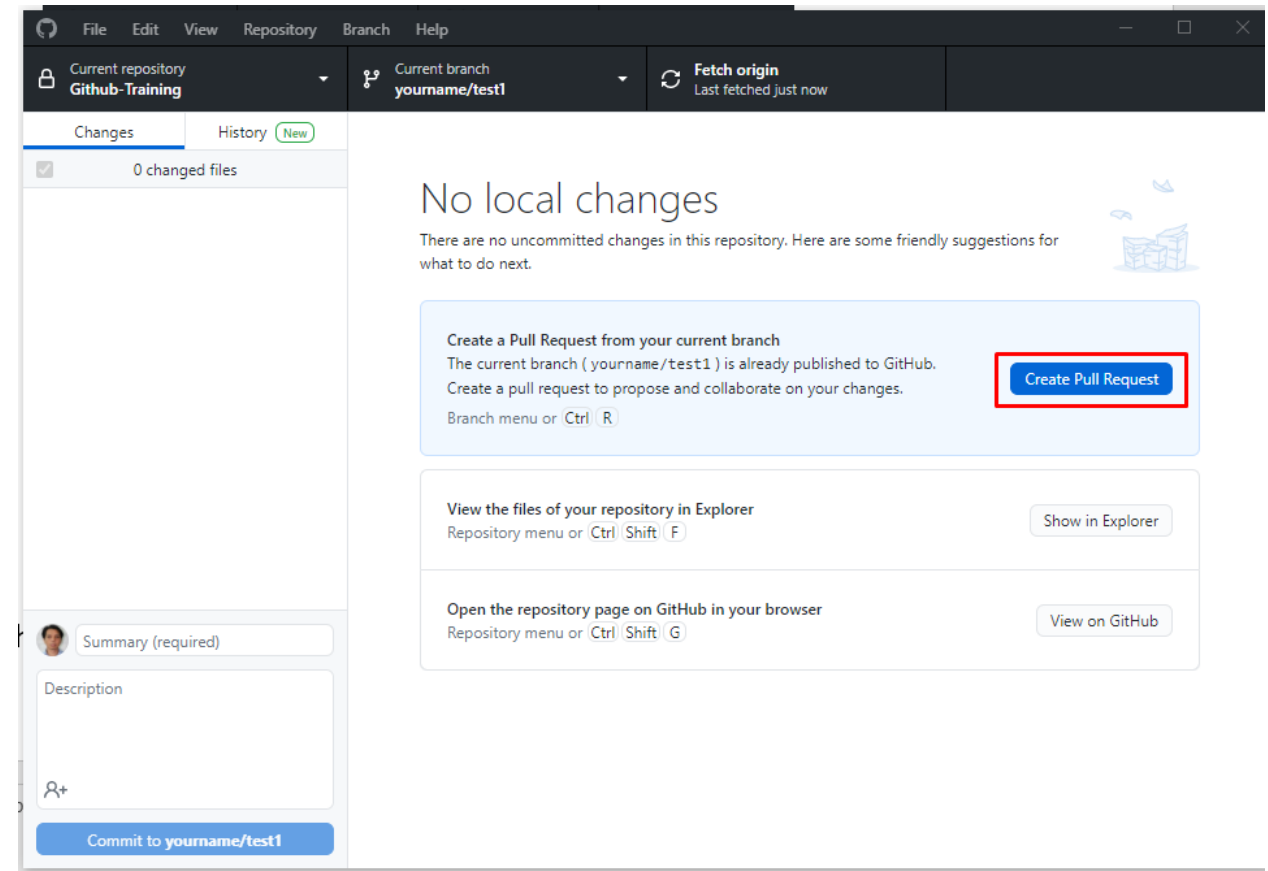
Helpful resources

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

# CREATE PULL REQUEST – GITHUB DESKTOP

13

- Click on the 'Push origin' button'





# CREATE PULL REQUEST

- Pull request created

## Add new test class #1

Edit

 Open shahirabd wants to merge 1 commit into `master` from `myfirstbranch`

 Conversation 0  Commits 1  Checks 0  Files changed 1



+10 -0



shahirabd commented now

Owner 😊 ...

Test comment

  Add new test class

9d56dca

Add more commits by pushing to the `myfirstbranch` branch on shahirabd/POMSample.



Continuous integration has not been set up

[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.



This branch has no conflicts with the base branch

Merging can be performed automatically.

Reviewers



No reviews

Still in progress? Convert to draft

Assignees



No one—assign yourself

Labels



None yet

Projects



None yet

Milestone



No milestone



# REVIEW PULL REQUEST

- Reviewer can approve or request change on the pull request
- Who should review?
  - Other team members
  - Leads
  - Managers
  - ...who are working on the project

## Add new test class #1

The screenshot shows a GitHub pull request titled "Add new test class #1" by user "shahirabd". The pull request is from the "myfirstbranch" to the "master" branch. It shows 1 commit, 0 checks, and 1 file changed. The file "Tests/UnitTest2.cs" is highlighted, showing a diff with 10 lines of code. The code includes using statements for System, System.Collections.Generic, and System.Text, and defines a new class "UnitTest2". The right sidebar shows the "Review changes" section with a "Write" tab selected. It includes a "Leave a comment" text area, a file attachment section, and three review options: "Comment" (selected), "Approve", and "Request changes". A "Submit review" button is at the bottom.

Open shahirabd wants to merge 1 commit into master from myfirstbranch

Conversation 0 Commits 1 Checks 0 Files changed 1 +10 -0

Changes from all commits File filter... Jump to... 0 / 1 files viewed Review changes

Tests/UnitTest2.cs @@ -0,0 +1,10 @@

```
1 + using System;
2 + using System.Collections.Generic;
3 + using System.Text;
4 +
5 + namespace POMSample.Tests
6 + {
7 +     class UnitTest2
8 +     {
9 +     }
10 + }
```

Write Preview AA B i " <> ↺ ⋮ ≡ ≡ @ 📌 ↶

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

☒ Comment Submit general feedback without explicit approval.

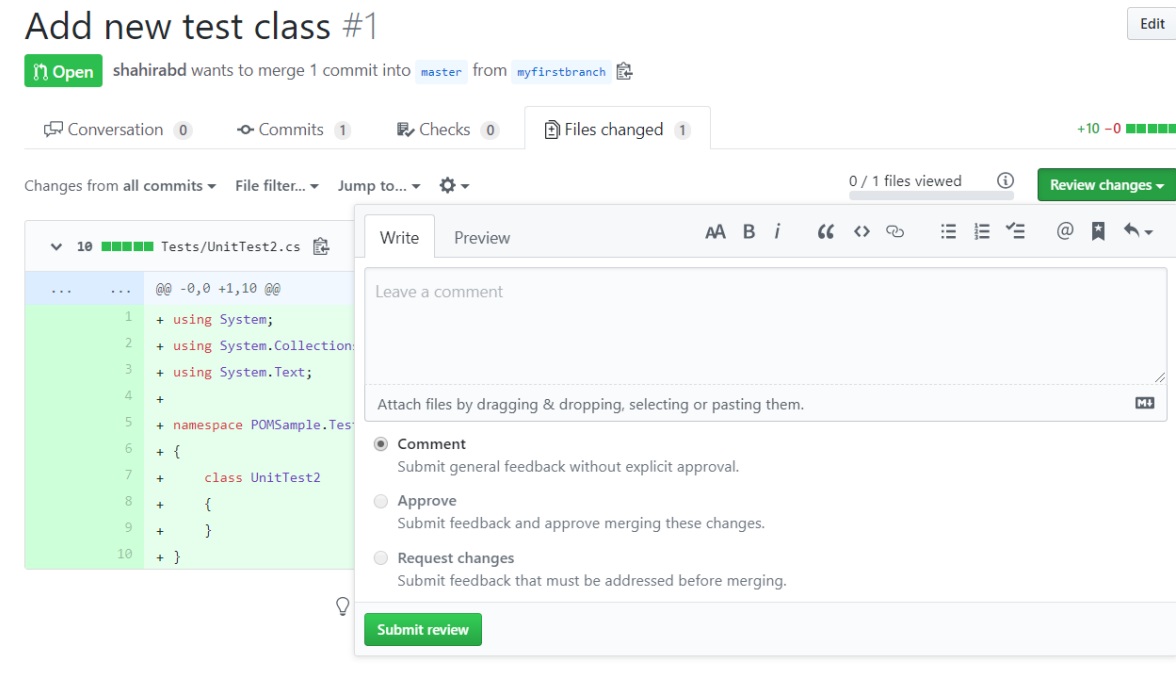
☐ Approve Submit feedback and approve merging these changes.

☐ Request changes Submit feedback that must be addressed before merging.

Submit review

# RESPOND TO A REVIEW

- Once review is done, we need to make changes accordingly (if any).
  - Git pull
  - Git checkout <branch-name>
  - Repeat the whole process (change the file, stage, commit, push)
  - Once pushed, the pull request updated automatically, we don't have to create a new pull request from the GitHub website.
- Finally, we can merge the working branch to master branch



# MERGE PULL REQUEST

- Click on the 'merge pull request' button

The screenshot shows a GitHub Pull Request (PR) titled "Add new test class #1" by user shahirabd. The PR is targeting the master branch from the myfirstbranch. The interface includes a green "Open" button at the top left. Below the title, there's a section for "Add more commits by pushing to the myfirstbranch branch on shahirabd/POMSample." This section contains two status messages: "Continuous integration has not been set up" (with a link to GitHub Actions) and "This branch has no conflicts with the base branch" (indicating merging can be performed automatically). A green "Merge pull request" button is prominently displayed, with a dropdown arrow and a note: "You can also open this in GitHub Desktop or view command line instructions." Below this is a "Write" tab for adding a comment, with a "Preview" tab and a rich text editor toolbar. The comment area includes a "Leave a comment" placeholder, a file attachment area, and buttons for "Close pull request" and "Comment". At the bottom, there's a reminder to follow GitHub Community Guidelines and a "ProTip!" about adding .patch or .diff to URLs. On the right side, there's a sidebar with settings for "Assignees" (No one—assign yourself), "Labels" (None yet), "Projects" (None yet), "Milestone" (No milestone), "Linked issues" (Successfully merging this pull request may close these issues), "Notifications" (Unsubscribe button), "1 participant", and a "Lock conversation" option.

# MERGE PULL REQUEST - CLI

- Or use the command line
  - git checkout master
  - git merge <branch-name>

```
MINGW64:/c/Users/Owner/POMSample
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 434 bytes | 434.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/shahirabd/POMSample.git
   f515e2a..9d56dca  myfirstbranch -> myfirstbranch

Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (myfirstbranch)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (master)
$ git merge myfirstbranch
Updating f515e2a..9d56dca
Fast-forward
 Tests/UnitTest2.cs | 10 ++++++++
 1 file changed, 10 insertions(+)
 create mode 100644 Tests/UnitTest2.cs

Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (master)
$ |
```

# MERGE PULL REQUEST - CLI

- git push
- git branch -d <branch-name>

```
MINGW64:/c/Users/Owner/POMSample
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (master)
$ git merge myfirstbranch
Updating f515e2a..9d56dca
Fast-forward
 Tests/UnitTest2.cs | 10 ++++++++
 1 file changed, 10 insertions(+)
 create mode 100644 Tests/UnitTest2.cs

Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (master)
$ git push
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/shahirabd/POMSample.git
 f515e2a..9d56dca master -> master

Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (master)
$ git branch -d myfirstbranch
Deleted branch myfirstbranch (was 9d56dca).

Owner@DESKTOP-KVHHL7S MINGW64 ~/POMSample (master)
$
```



# MERGE PULL REQUEST

- Check the PR on GitHub website

The screenshot shows a GitHub Pull Request (PR) for the repository 'shahirabd / POMSample'. The PR title is 'Add new test class #1'. It shows that the PR has been merged, with a message: 'shahirabd merged 1 commit into master from myfirstbranch 4 minutes ago'. The PR details include a commit '9d56dca' with the message 'Add new test class'. The PR is closed, and a message states: 'Pull request successfully merged and closed. You're all set—the myfirstbranch branch can be safely deleted.' The right sidebar shows settings for the PR, including Reviewers, Assignees, Labels, Projects, Milestone, and Linked issues.

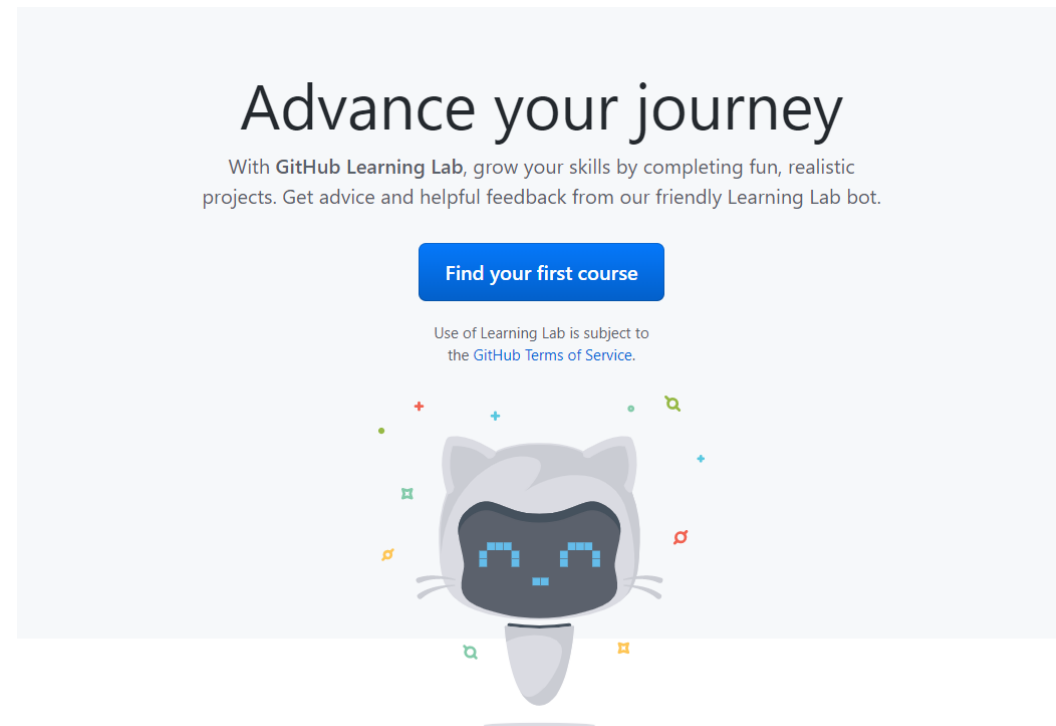


## ADDITIONAL COURSE(S)

For you to explore on your own

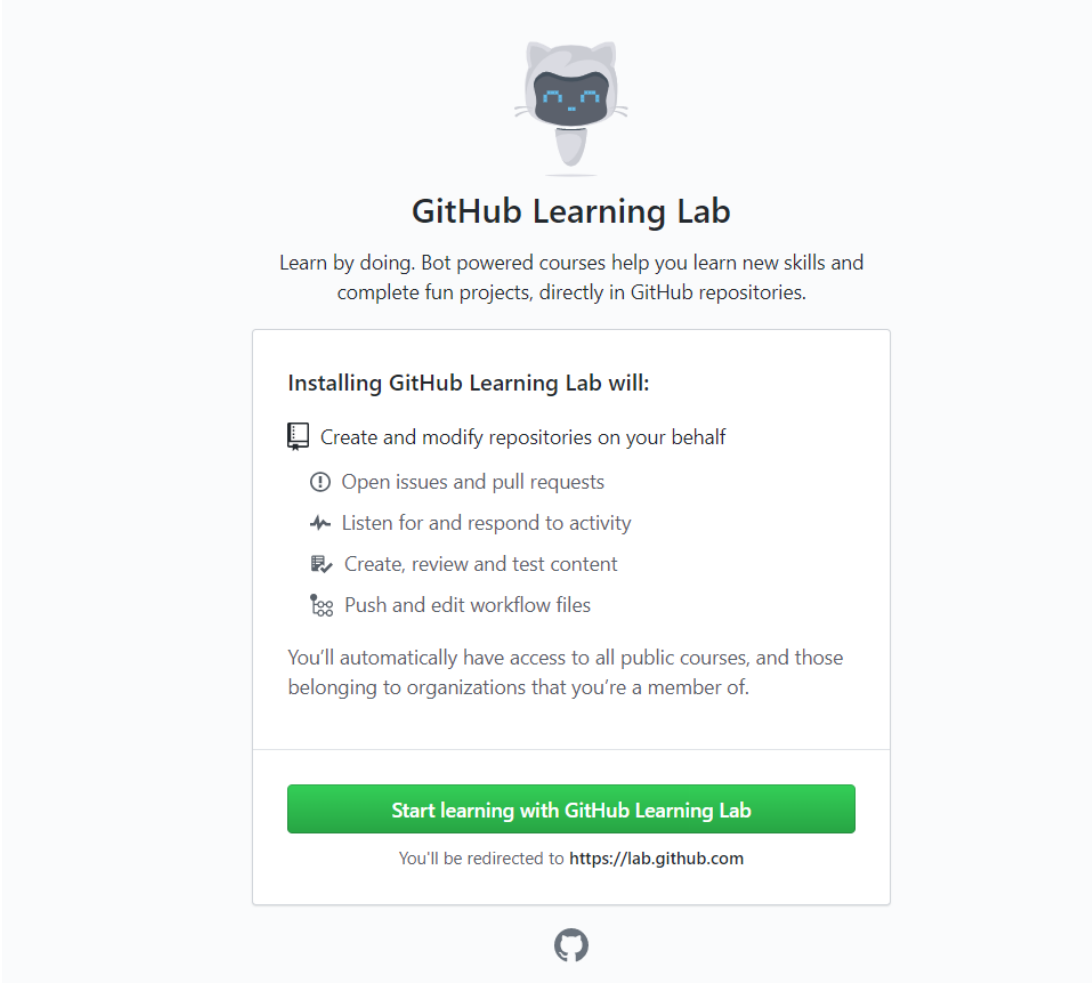
# SELF-LEARNING

- Go to [lab.github.com](https://lab.github.com)
- Login with your GitHub account




# SELF-LEARNING

- Click on start learning button








The image shows a screenshot of the GitHub Learning Lab interface. At the top center is the GitHub Octocat logo. Below it, the text "GitHub Learning Lab" is displayed in a bold, sans-serif font. Underneath this, a paragraph reads: "Learn by doing. Bot powered courses help you learn new skills and complete fun projects, directly in GitHub repositories." Below this text is a white-bordered box containing the following content: The heading "Installing GitHub Learning Lab will:" followed by a list of five items, each with a small icon: a terminal icon for "Create and modify repositories on your behalf", an information icon for "Open issues and pull requests", a lightning bolt icon for "Listen for and respond to activity", a document icon for "Create, review and test content", and a workflow icon for "Push and edit workflow files". Below the list, a paragraph states: "You'll automatically have access to all public courses, and those belonging to organizations that you're a member of." At the bottom of the box is a prominent green button with the text "Start learning with GitHub Learning Lab". Directly below the button, a smaller line of text says: "You'll be redirected to <https://lab.github.com>". At the very bottom center of the interface is the GitHub logo.



## GitHub Learning Lab

Learn by doing. Bot powered courses help you learn new skills and complete fun projects, directly in GitHub repositories.


**Installing GitHub Learning Lab will:**

-  Create and modify repositories on your behalf
-  Open issues and pull requests
-  Listen for and respond to activity
-  Create, review and test content
-  Push and edit workflow files

You'll automatically have access to all public courses, and those belonging to organizations that you're a member of.

[Start learning with GitHub Learning Lab](#)

You'll be redirected to <https://lab.github.com>




# SELF-LEARNING



- Look for this course

## Our most popular courses

### Introduction to GitHub

 The GitHub Training Team

If you are looking for a quick and fun introduction to GitHub, you've found it. This class will get you started using GitHub in less than an hour.

● Git ● GitHub Pages ● Branches  
● Commits ● Pull Requests

[Learn more](#)



# SELF-LEARNING

13

- Click on the start free course button

## Introduction to GitHub

 The GitHub Training Team

If you are looking for a quick and fun introduction to GitHub, you've found it. This class will get you started using GitHub in less than an hour.

[Start free course](#)

*Join 147637 others!*

# SELF-LEARNING

- Click on the start free course button

Begin *Introduction to GitHub*

GitHub Learning Lab will create a repository named **github-slideshow** on your GitHub account.

☒

 **Public** (Default)  
Anyone can see this repository. You choose who can commit.

☐

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

 **Language**  
Choose your preferred language.

English ▾

► Additional Options

Begin *Introduction to GitHub*

# SELF-LEARNING

- Click on the start free course button

Begin *Introduction to GitHub*

×

GitHub Learning Lab will create a repository named **github-slideshow** on your GitHub account.

☒

 **Public** (Default)  
Anyone can see this repository. You choose who can commit.

☐

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

 **Language**  
Choose your preferred language.

English ▾

▶ Additional Options

Begin *Introduction to GitHub*



# ONLINE ASSESSMENT

# ONLINE ASSESSMENT



- <https://extendedforms.io/form/956e038d-3877-4159-81dc-ae3ac7af1418/login>
- Please complete this assessment, once you have studied and practiced
- You can refer to this slide and the official GitHub documentation - docs.github.com.