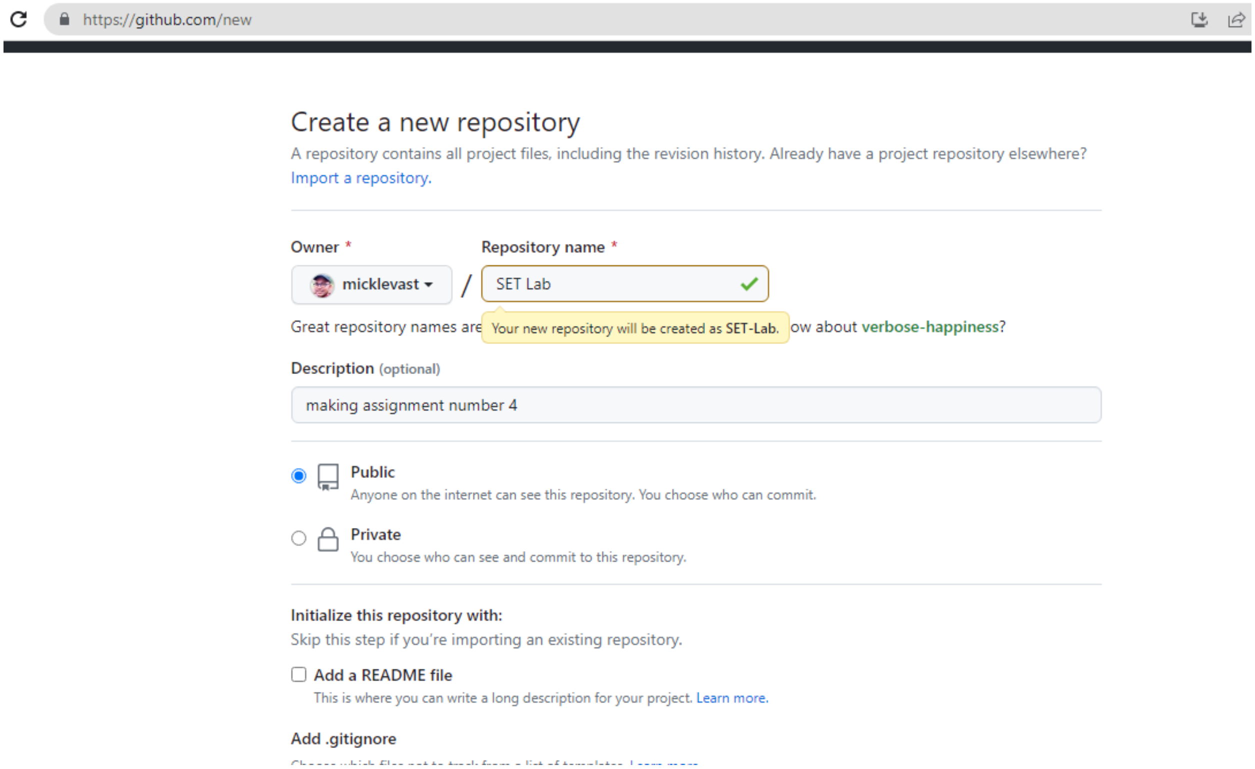
PRN:2020BTECS00090
Name:Vishal Chauhan
Software Engineering Tools Lab
Assignment No-4

Q 1. Create a repository on GitHub named SET Lab and add files into it (you can add implementation files of previous assignment) perform below operations on it. (Add screenshot as an answer to every question)

1. Perform commit on added files
2. Perform update to the existing files (show history)
3. Create another branch
4. Create pull request
5. Perform merging of both branches
6. Perform Fork operation

Step1: Making new repo



https://github.com/new

Create a new repository

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

Owner * micklevast / Repository name * SET Lab ✓

Great repository names are Your new repository will be created as SET-Lab. ow about verbose-happiness?

Description (optional)

making assignment number 4

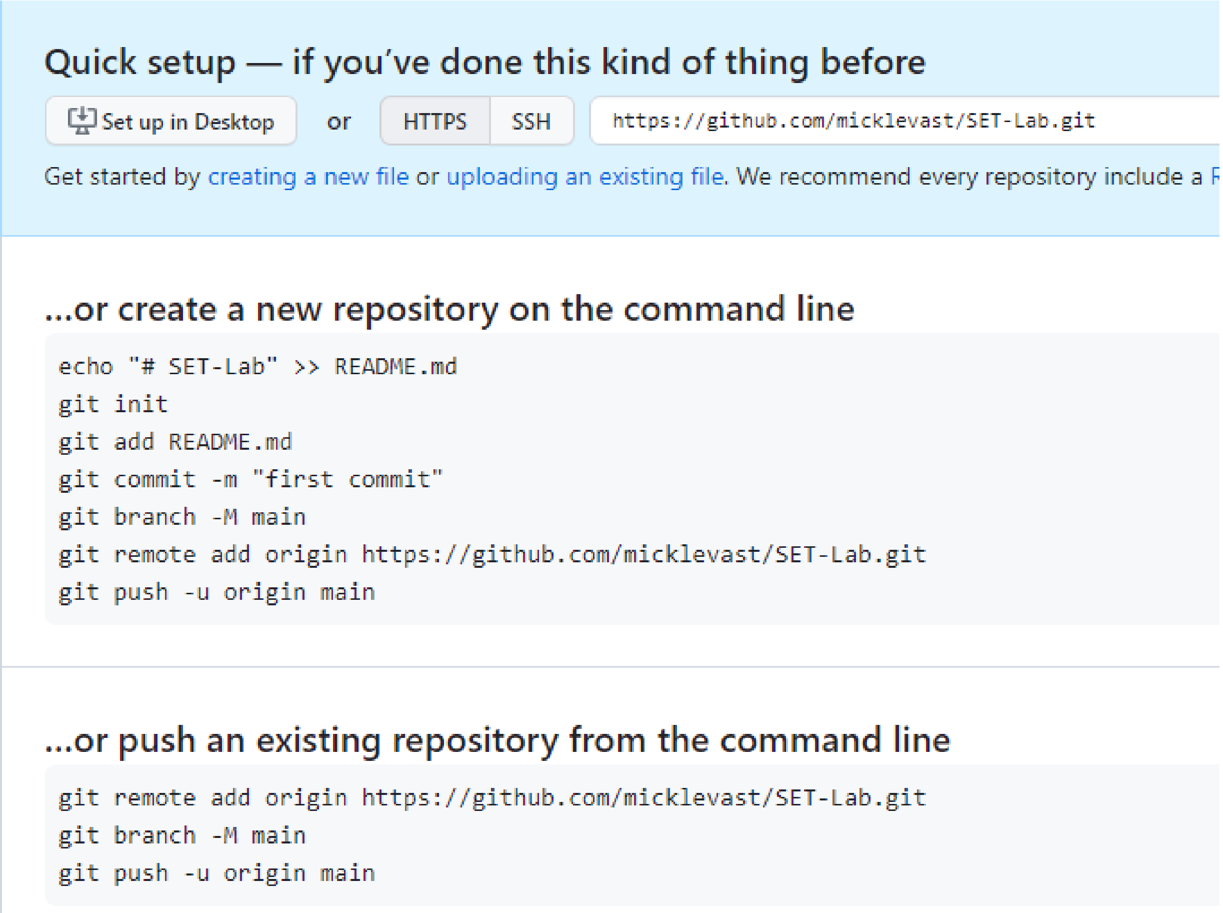
● Public
Anyone on the internet can see this repository. You choose who can commit.

○ 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.



Quick setup — if you've done this kind of thing before

```
https://github.com/micklevast/SET-Lab.git
```

Get started by creating a new file or uploading an existing file. We recommend every repository include a R

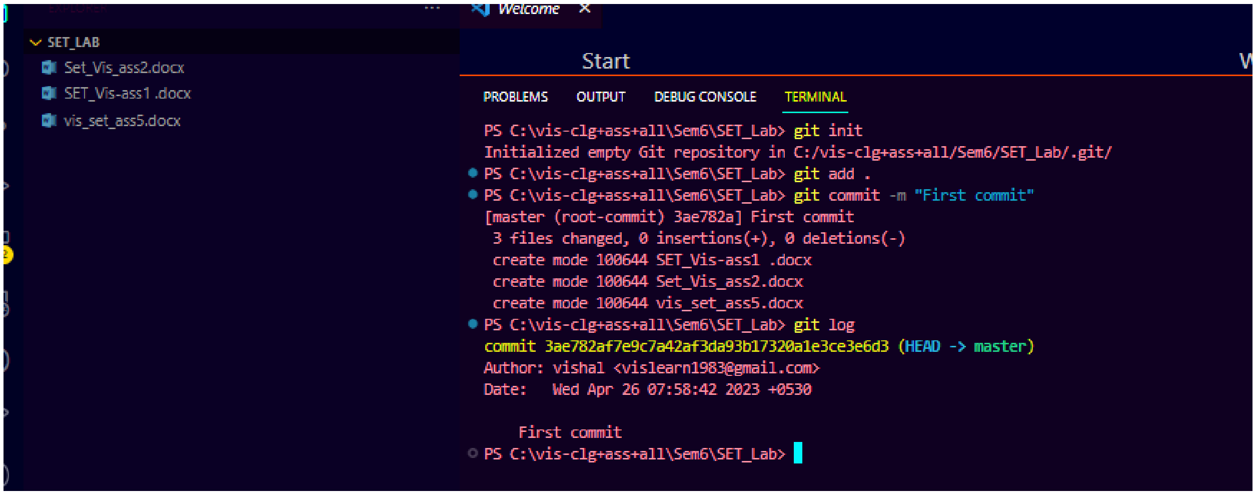
...or create a new repository on the command line

```
echo "# SET-Lab" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/micklevast/SET-Lab.git
git push -u origin main
```

...or push an existing repository from the command line

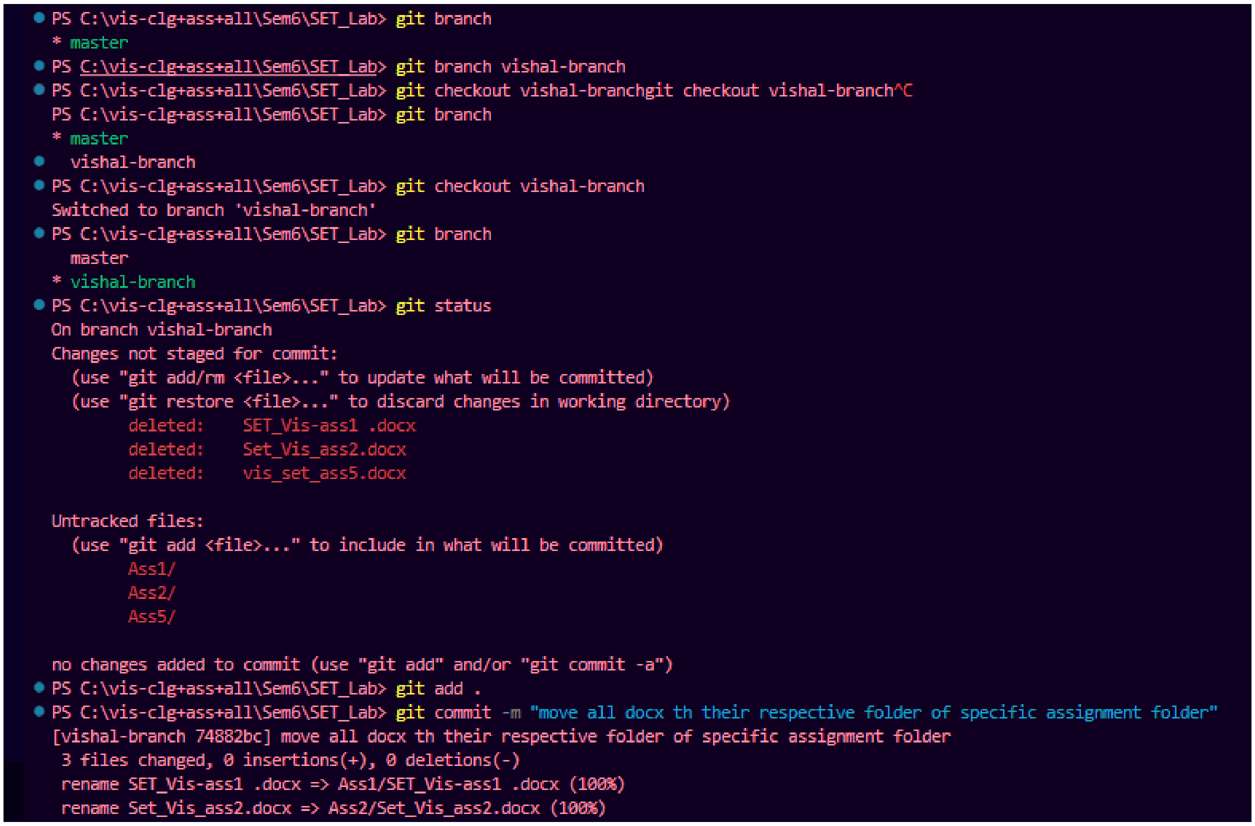
```
git remote add origin https://github.com/micklevast/SET-Lab.git
git branch -M main
git push -u origin main
```

Step2 : Initializing new git repo and adding file and making first commit and showing history



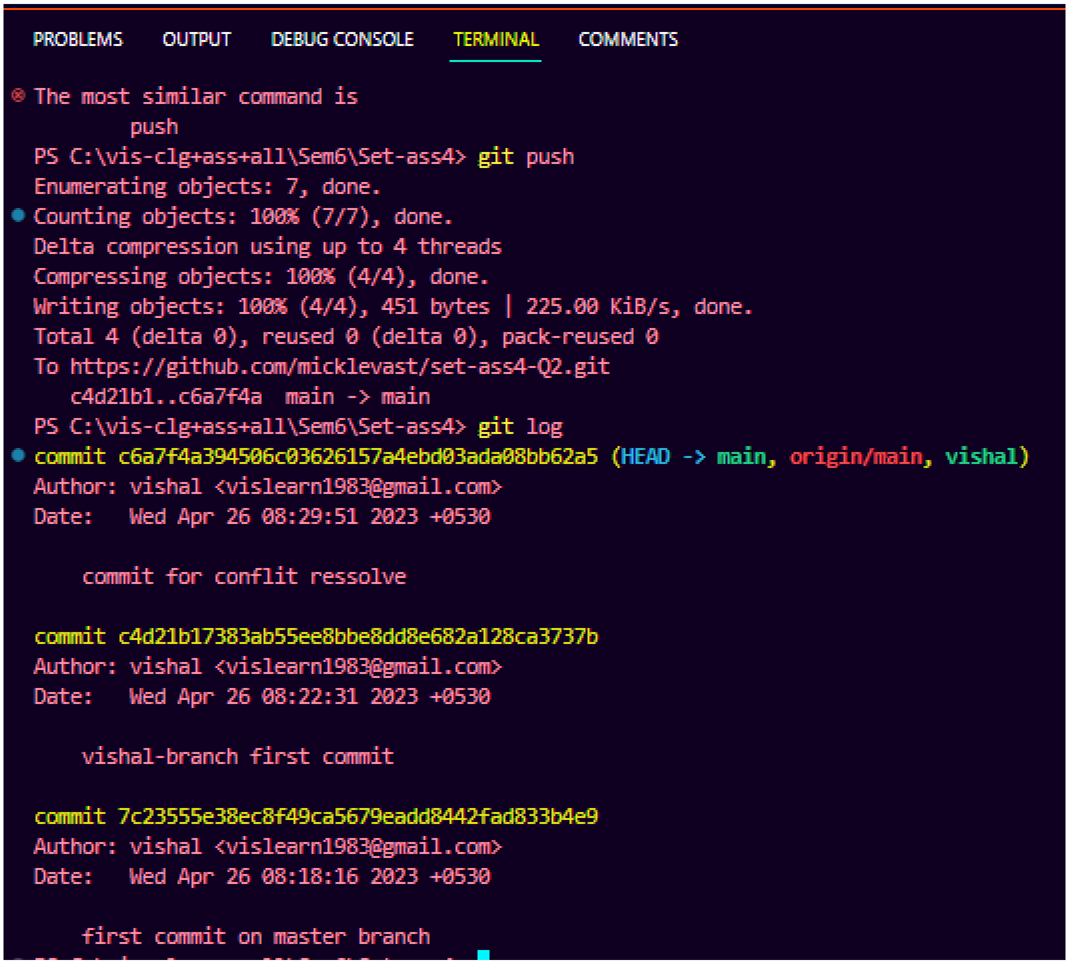
```
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git init
Initialized empty Git repository in C:/vis-clg+ass+all/Sem6/SET_Lab/.git/
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git add .
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git commit -m "First commit"
[master (root-commit) 3ae782a] First commit
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 SET_Vis-ass1 .docx
 create mode 100644 Set_Vis_ass2.docx
 create mode 100644 vis_set_ass5.docx
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git log
commit 3ae782af7e9c7a42af3da93b17320a1e3ce3e6d3 (HEAD -> master)
Author: vishal <vislearn1983@gmail.com>
Date:   Wed Apr 26 07:58:42 2023 +0530

    First commit
PS C:\vis-clg+ass+all\Sem6\SET_Lab>
```

Step3:Creating another branch and making commit over after changes

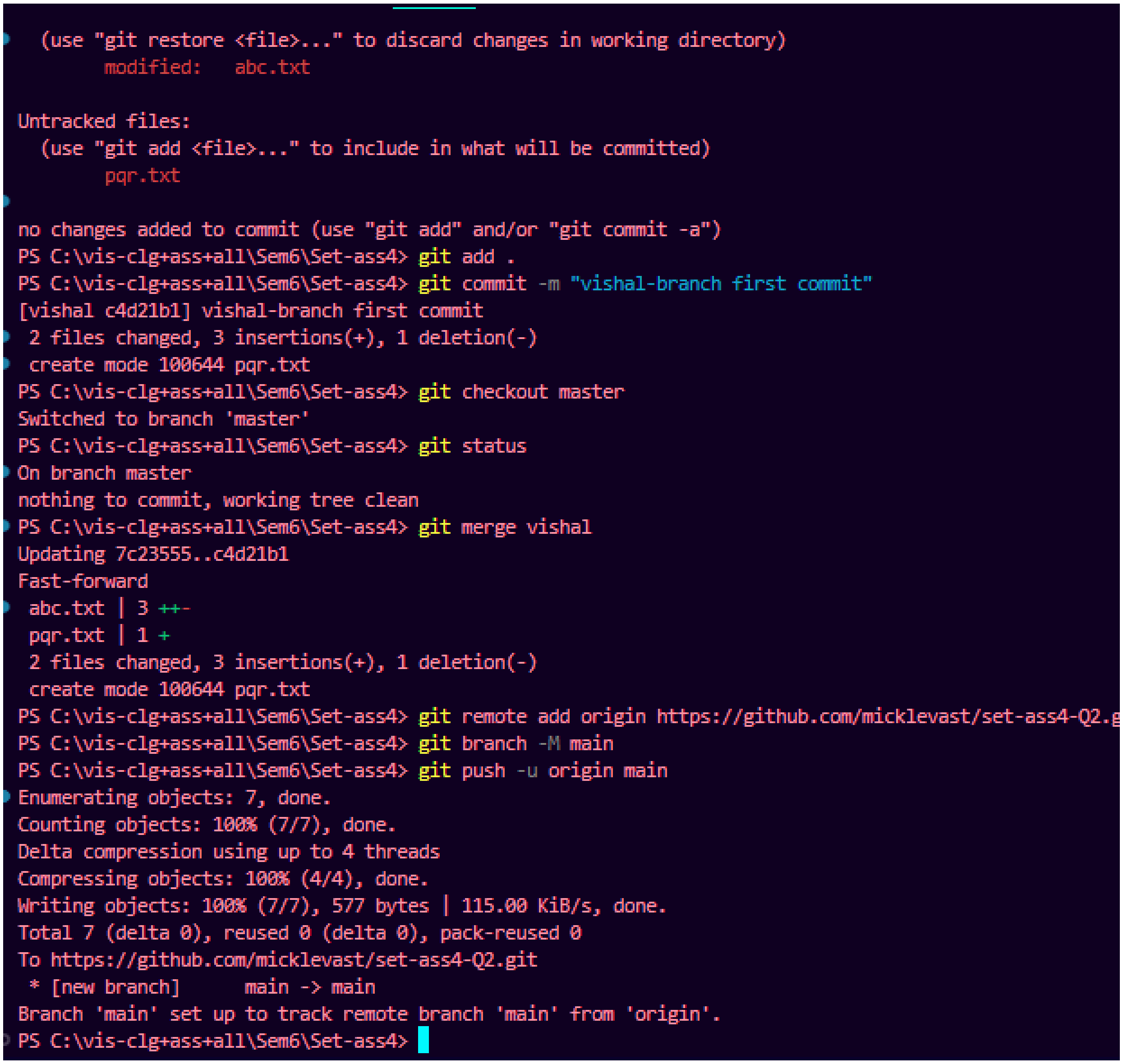


```
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git branch
* master
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git branch vishal-branch
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git checkout vishal-branchgit checkout vishal-branch^C
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git branch
* master
  vishal-branch
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git checkout vishal-branch
Switched to branch 'vishal-branch'
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git branch
  master
* vishal-branch
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git status
On branch vishal-branch
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    SET_Vis-ass1 .docx
        deleted:    Set_Vis_ass2.docx
        deleted:    vis_set_ass5.docx

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Ass1/
        Ass2/
        Ass5/

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git add .
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git commit -m "move all docx th their respective folder of specific assignment folder"
[vishal-branch 74882bc] move all docx th their respective folder of specific assignment folder
 3 files changed, 0 insertions(+), 0 deletions(-)
 rename SET_Vis-ass1 .docx => Ass1/SET_Vis-ass1 .docx (100%)
 rename Set_Vis_ass2.docx => Ass2/Set_Vis_ass2.docx (100%)
```

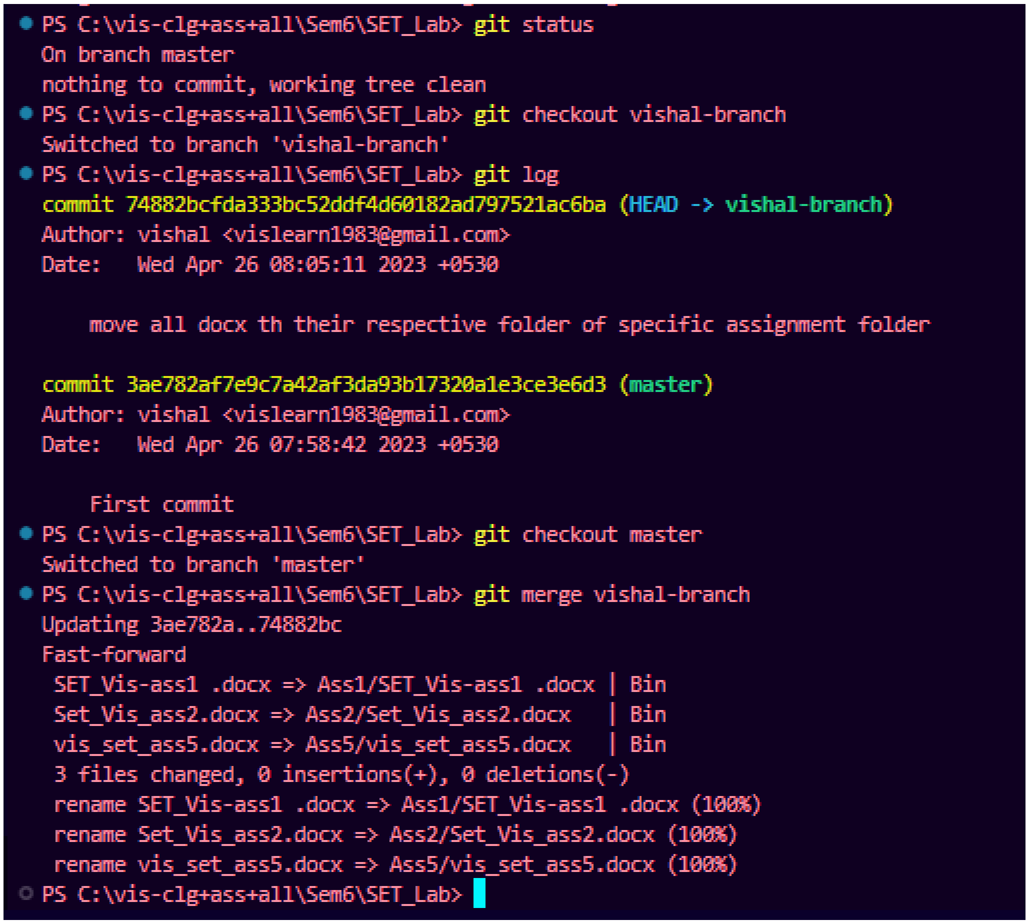
Step4:Merging two branch and pushing it into github remote repo



```
The most similar command is
        push
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 451 bytes | 225.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/micklevast/set-ass4-Q2.git
   c4d21b1..c6a7f4a  main -> main
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git log
commit c6a7f4a394506c03626157a4ebd03ada08bb62a5 (HEAD -> main, origin/main, vishal)
Author: vishal <vislearn1983@gmail.com>
Date:   Wed Apr 26 08:29:51 2023 +0530

    commit for conflit ressolve

commit c4d21b17383ab55ee8bbe8dd8e682a128ca3737b
Author: vishal <vislearn1983@gmail.com>
Date:   Wed Apr 26 08:22:31 2023 +0530

    vishal-branch first commit

commit 7c23555e38ec8f49ca5679eadd8442fad833b4e9
Author: vishal <vislearn1983@gmail.com>
Date:   Wed Apr 26 08:18:16 2023 +0530

    first commit on master branch
```

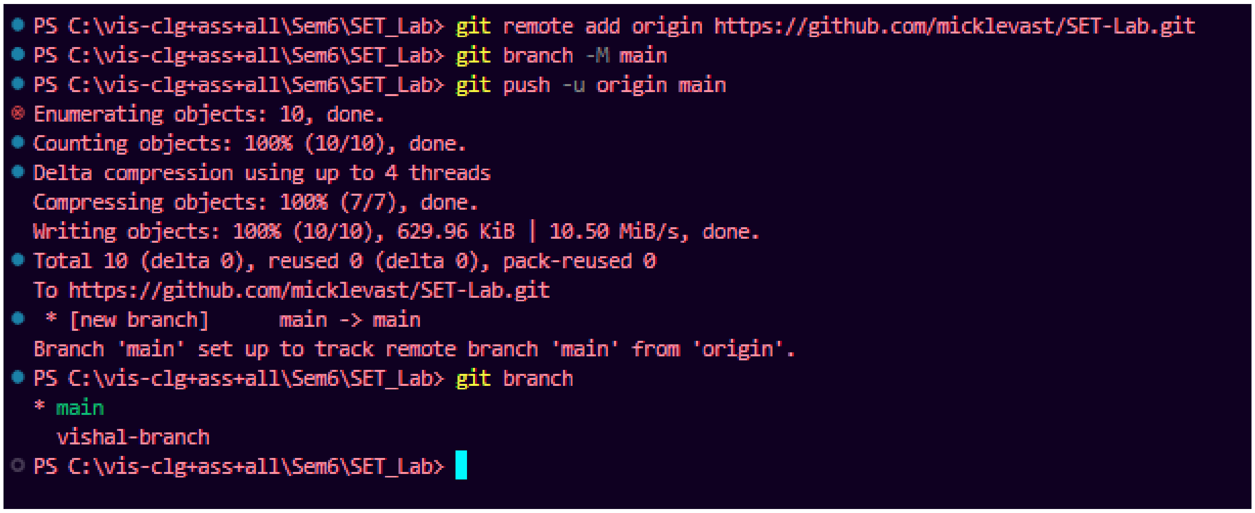


```
  (use "git restore <file>..." to discard changes in working directory)
        modified:   abc.txt

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

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git add .
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git commit -m "vishal-branch first commit"
[vishal c4d21b1] vishal-branch first commit
 2 files changed, 3 insertions(+), 1 deletion(-)
 create mode 100644 pqr.txt
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git checkout master
Switched to branch 'master'
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git status
On branch master
nothing to commit, working tree clean
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git merge vishal
Updating 7c23555..c4d21b1
Fast-forward
 abc.txt | 3 ++-
 pqr.txt | 1 +
 2 files changed, 3 insertions(+), 1 deletion(-)
 create mode 100644 pqr.txt
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git remote add origin https://github.com/micklevast/set-ass4-Q2.g
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git branch -M main
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (7/7), 577 bytes | 115.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/micklevast/set-ass4-Q2.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
PS C:\vis-clg+ass+all\Sem6\Set-ass4>
```



```
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git status
On branch master
nothing to commit, working tree clean
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git checkout vishal-branch
Switched to branch 'vishal-branch'
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git log
commit 74882bcfda333bc52ddf4d60182ad797521ac6ba (HEAD -> vishal-branch)
Author: vishal <vislearn1983@gmail.com>
Date:   Wed Apr 26 08:05:11 2023 +0530

    move all docx th their respective folder of specific assignment folder

commit 3ae782af7e9c7a42af3da93b17320a1e3ce3e6d3 (master)
Author: vishal <vislearn1983@gmail.com>
Date:   Wed Apr 26 07:58:42 2023 +0530

    First commit
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git checkout master
Switched to branch 'master'
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git merge vishal-branch
Updating 3ae782a..74882bc
Fast-forward
 SET_Vis-ass1 .docx => Ass1/SET_Vis-ass1 .docx | Bin
 Set_Vis_ass2.docx => Ass2/Set_Vis_ass2.docx   | Bin
 vis_set_ass5.docx => Ass5/vis_set_ass5.docx   | Bin
 3 files changed, 0 insertions(+), 0 deletions(-)
 rename SET_Vis-ass1 .docx => Ass1/SET_Vis-ass1 .docx (100%)
 rename Set_Vis_ass2.docx => Ass2/Set_Vis_ass2.docx (100%)
 rename vis_set_ass5.docx => Ass5/vis_set_ass5.docx (100%)
PS C:\vis-clg+ass+all\Sem6\SET_Lab>
```

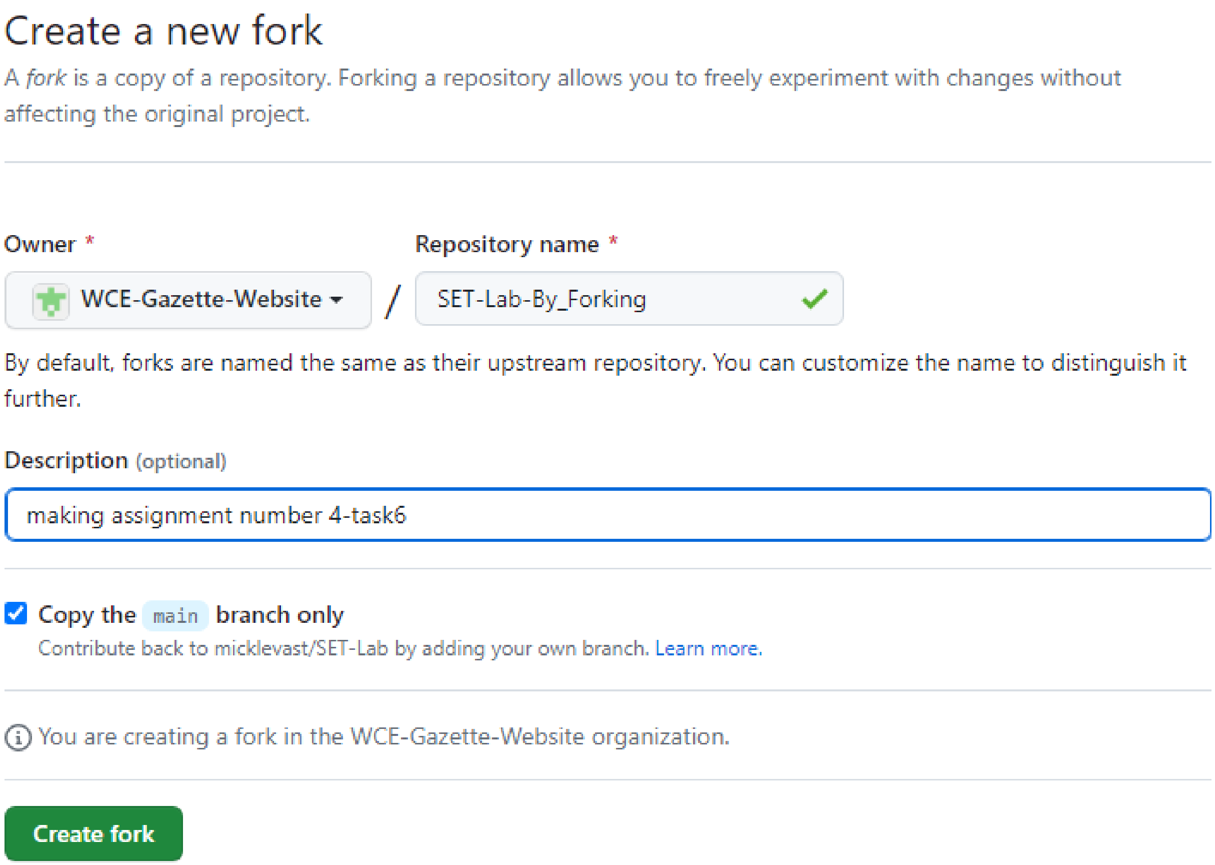
Step5:Pushing local repo over github i.e. on remote repo



```
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git remote add origin https://github.com/micklevast/SET-Lab.git
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git branch -M main
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git push -u origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (10/10), 629.96 KiB | 10.50 MiB/s, done.
Total 10 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/micklevast/SET-Lab.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
PS C:\vis-clg+ass+all\Sem6\SET_Lab> git branch
* main
  vishal-branch
PS C:\vis-clg+ass+all\Sem6\SET_Lab>
```

Step6:Performing Fork

git fork: This command creates a copy of a remote repository on the user's GitHub account.



Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Owner * WCE-Gazette-Website / Repository name * SET-Lab-By_Forking ✓

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

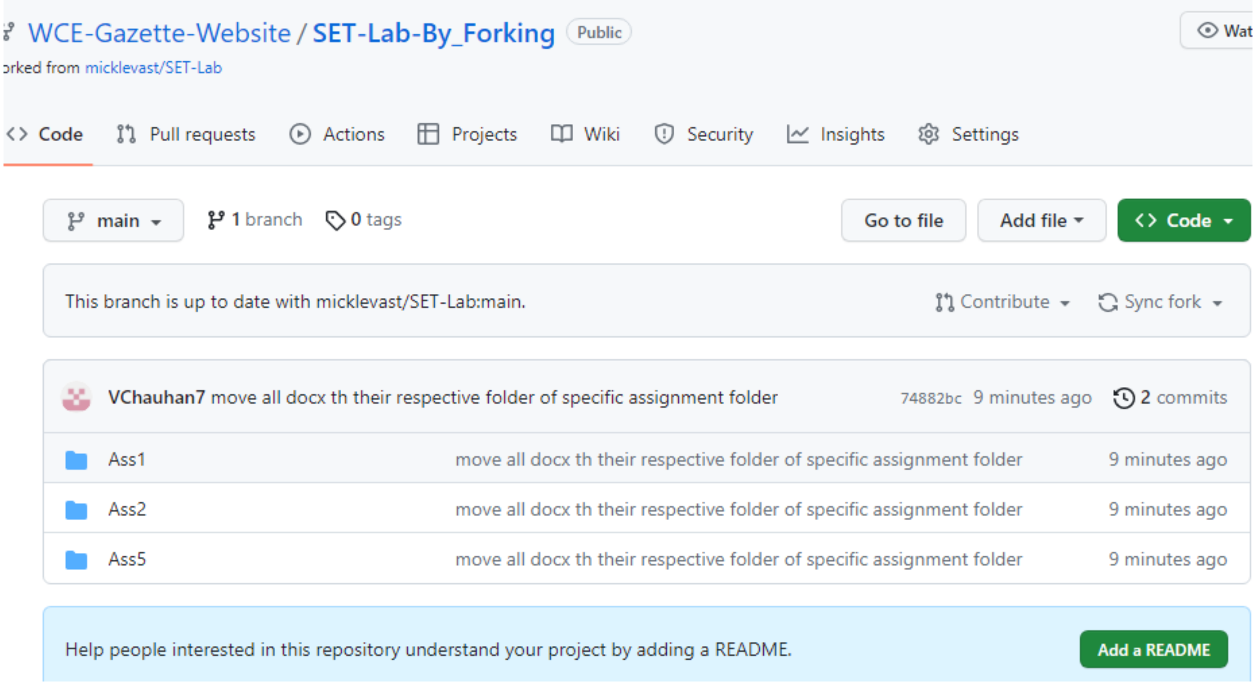
making assignment number 4-task6

☑ Copy the `main` branch only

Contribute back to micklevast/SET-Lab by adding your own branch. Learn more.

ⓘ You are creating a fork in the WCE-Gazette-Website organization.

Fork result:



WCE-Gazette-Website / SET-Lab-By_Forking Public

orked from micklevast/SET-Lab

1 branch 0 tags

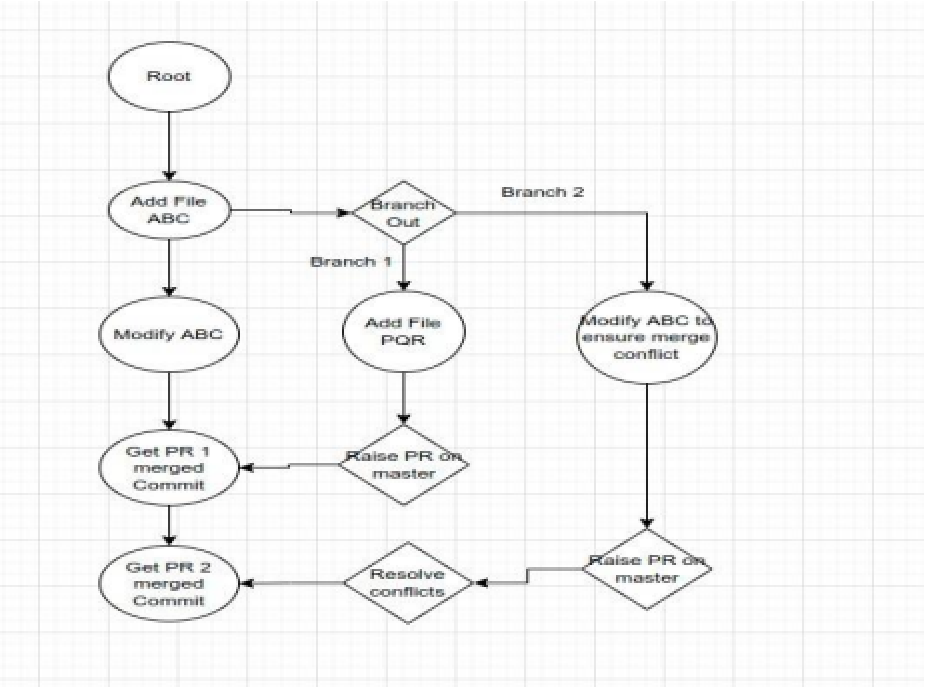
This branch is up to date with micklevast/SET-Lab:main.

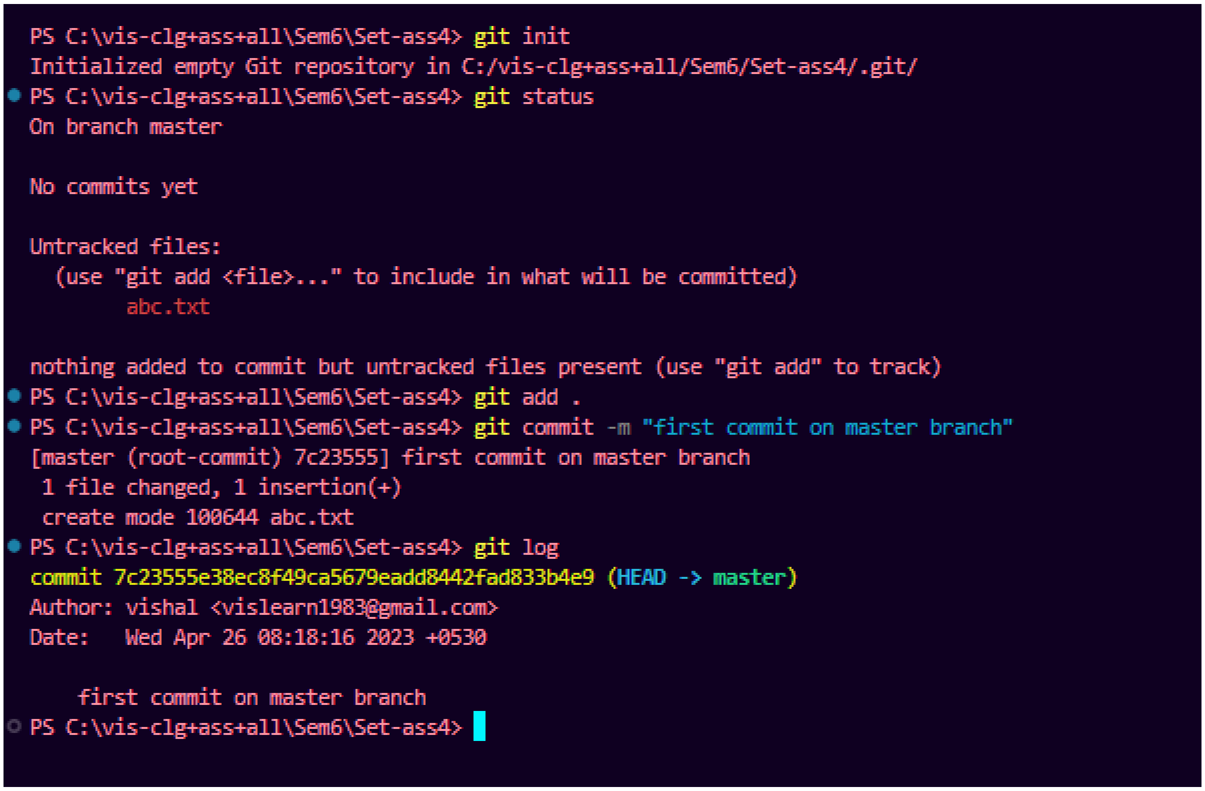
VChauhan7 move all docx th their respective folder of specific assignment folder 74882bc 9 minutes ago 2 commits

Ass1	move all docx th their respective folder of specific assignment folder	9 minutes ago
Ass2	move all docx th their respective folder of specific assignment folder	9 minutes ago
Ass5	move all docx th their respective folder of specific assignment folder	9 minutes ago

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

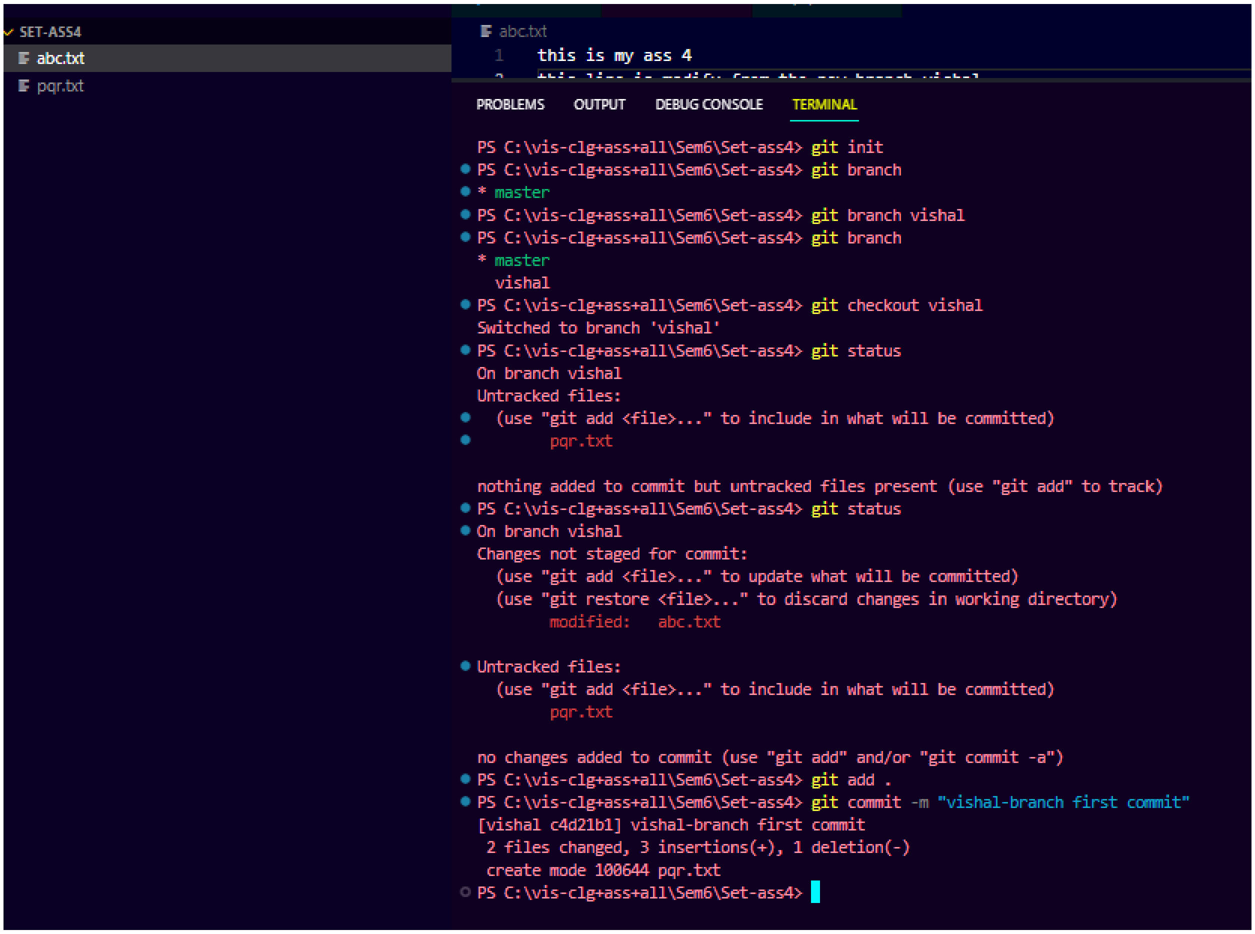
Q 2. For the diagram given below create a GitHub repository and perform operations given in the diagram. (Perform commit operations as given)(Add screenshots as an answer to this question)





```
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git init
Initialized empty Git repository in C:/vis-clg+ass+all/Sem6/Set-ass4/.git/
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git add .
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git commit -m "first commit on master branch"
[master (root-commit) 7c23555] first commit on master branch
 1 file changed, 1 insertion(+)
 create mode 100644 abc.txt
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git log
commit 7c23555e38ec8f49ca5679eadd8442fad833b4e9 (HEAD -> master)
Author: vishal <vislearn1983@gmail.com>
Date:   Wed Apr 26 08:18:16 2023 +0530

    first commit on master branch
PS C:\vis-clg+ass+all\Sem6\Set-ass4> 
```

Step:making new branch and adding new file pqr.txt and adding and committing in it



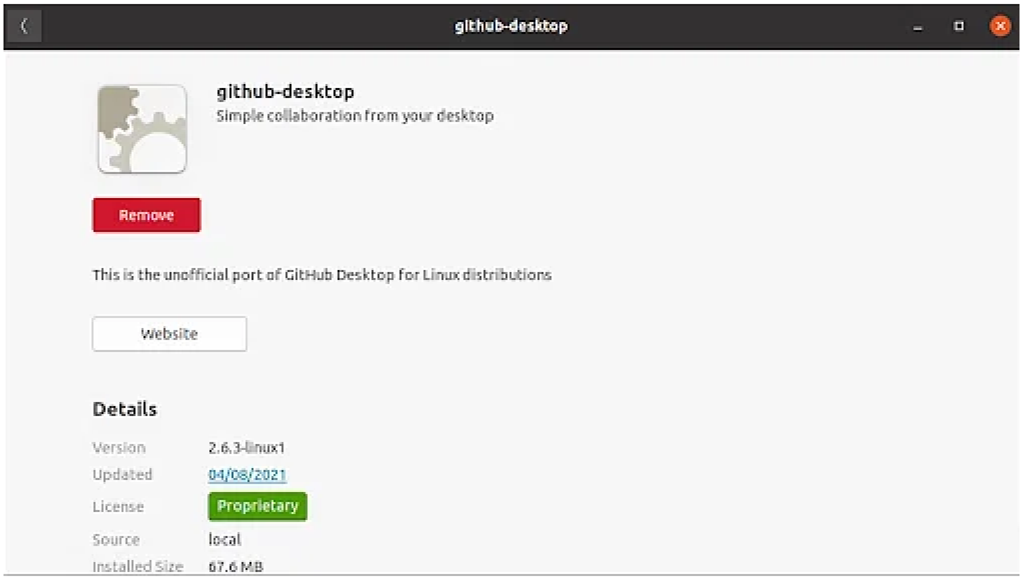
```
this is my ass 4
```

```
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git init
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git branch
* master
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git branch vishal
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git branch
* master
  vishal
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git checkout vishal
Switched to branch 'vishal'
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git status
On branch vishal
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        pqr.txt

nothing added to commit but untracked files present (use "git add" to track)
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git status
On branch vishal
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:   abc.txt

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

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git add .
PS C:\vis-clg+ass+all\Sem6\Set-ass4> git commit -m "vishal-branch first commit"
[vishal c4d21b1] vishal-branch first commit
 2 files changed, 3 insertions(+), 1 deletion(-)
 create mode 100644 pqr.txt
PS C:\vis-clg+ass+all\Sem6\Set-ass4>
```

Q 3. What is GitHub desktop? How to install GitHub on local machine? Install GitHub on your local machine and access repository created in question no 1 (add screenshots).

GitHub Desktop is a graphical user interface (GUI) for GitHub, a popular web-based Git repository hosting service. It allows users to manage their Git repositories, perform version control tasks, and collaborate with others using a simple and intuitive interface.
To install GitHub Desktop on your local machine, you can follow these steps:

1. Go to the GitHub Desktop website at https://desktop.github.com/ and click on the "Download for" button for your operating system.
2. Once the installer is downloaded, run it to start the installation process. Follow the prompts in the installer to complete the installation.
3. After the installation is complete, launch GitHub Desktop and sign in to your GitHub account if prompted.
4. Click on the "Clone repository" button on the GitHub Desktop home screen.
5. Select the "URL" tab and enter the URL for the repository you want to clone. For example, if the repository URL is https://github.com/username/repository-name.git, enter this URL into the "URL" field.



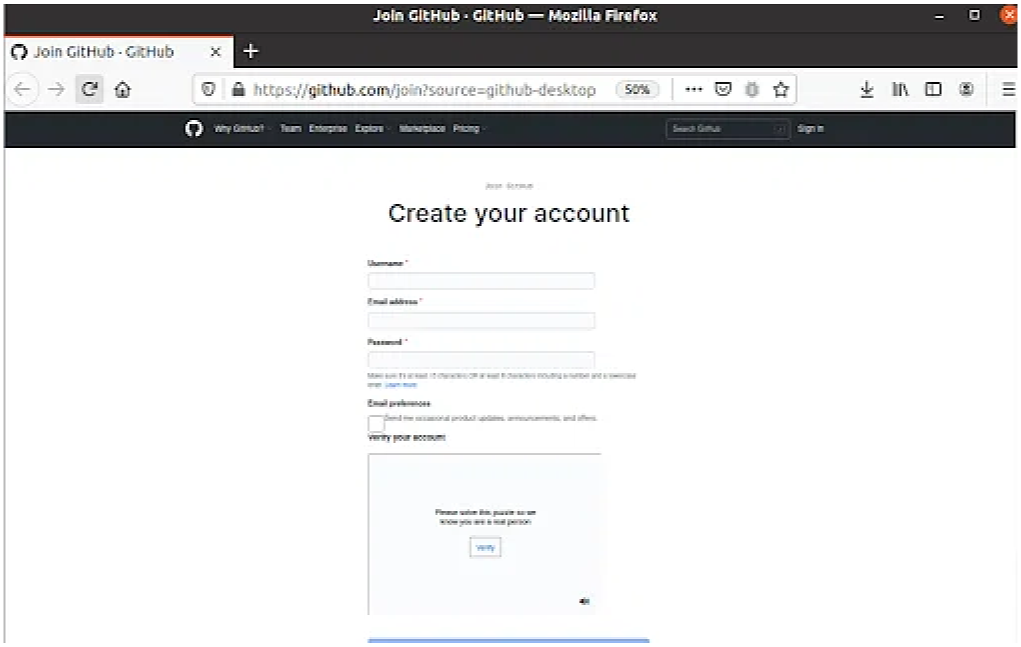
github-desktop

Simple collaboration from your desktop

This is the unofficial port of GitHub Desktop for Linux distributions

Details

Version	2.6.3-linux1
Updated	04/08/2021
License	Proprietary
Source	local
Installed Size	67.6 MB



Join GitHub · GitHub — Mozilla Firefox
Join GitHub · GitHub
50%
Why GitHub? · Team · Enterprise · Explore · Marketplace · Pricing
Search GitHub
Sign in
Verify

https://github.com/join?source=github-desktop

Join GitHub

Create your account

Username *

Email address *

Password *

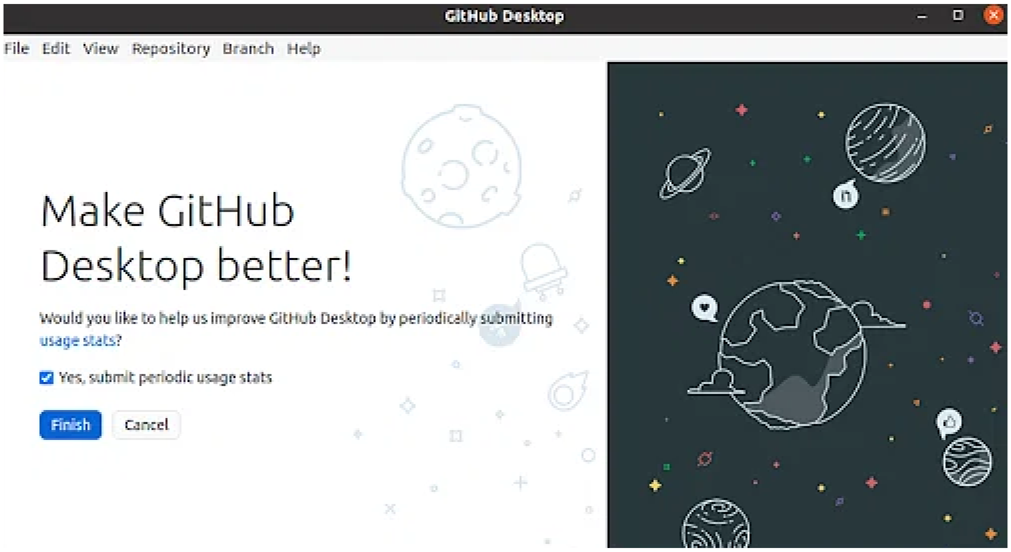
Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.

Email preferences

□ Send me occasional product updates, announcements, and offers.

Verify your account

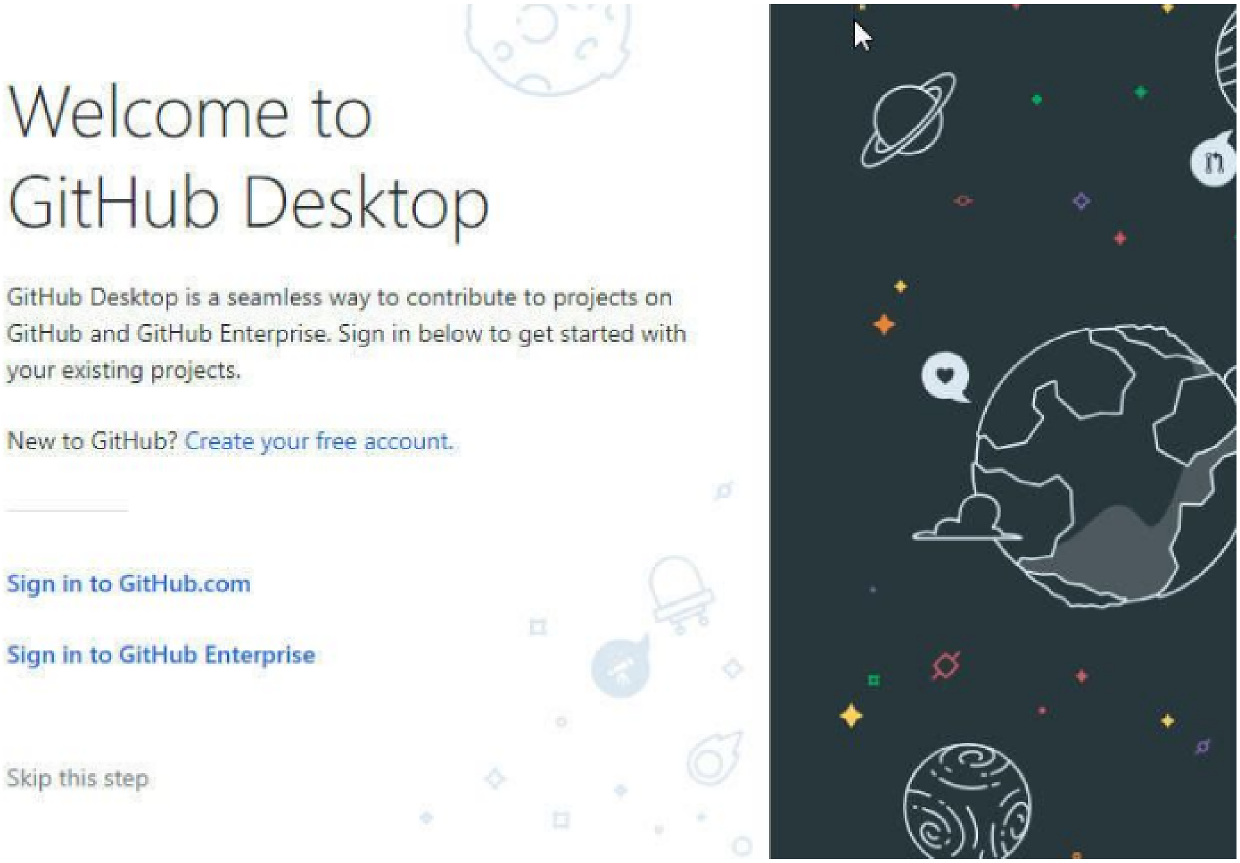
Please solve this puzzle so we know you are a real person



Make GitHub Desktop better!

Would you like to help us improve GitHub Desktop by periodically submitting usage stats?

☑ Yes, submit periodic usage stats

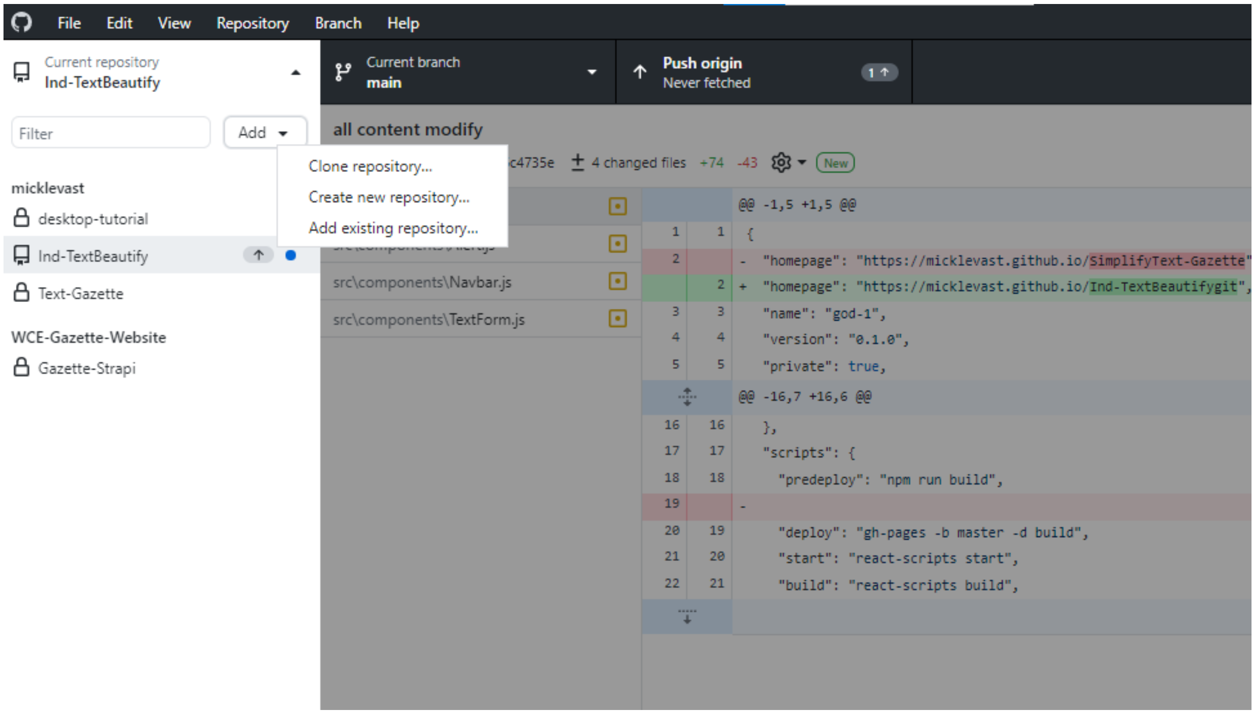


Welcome to GitHub Desktop

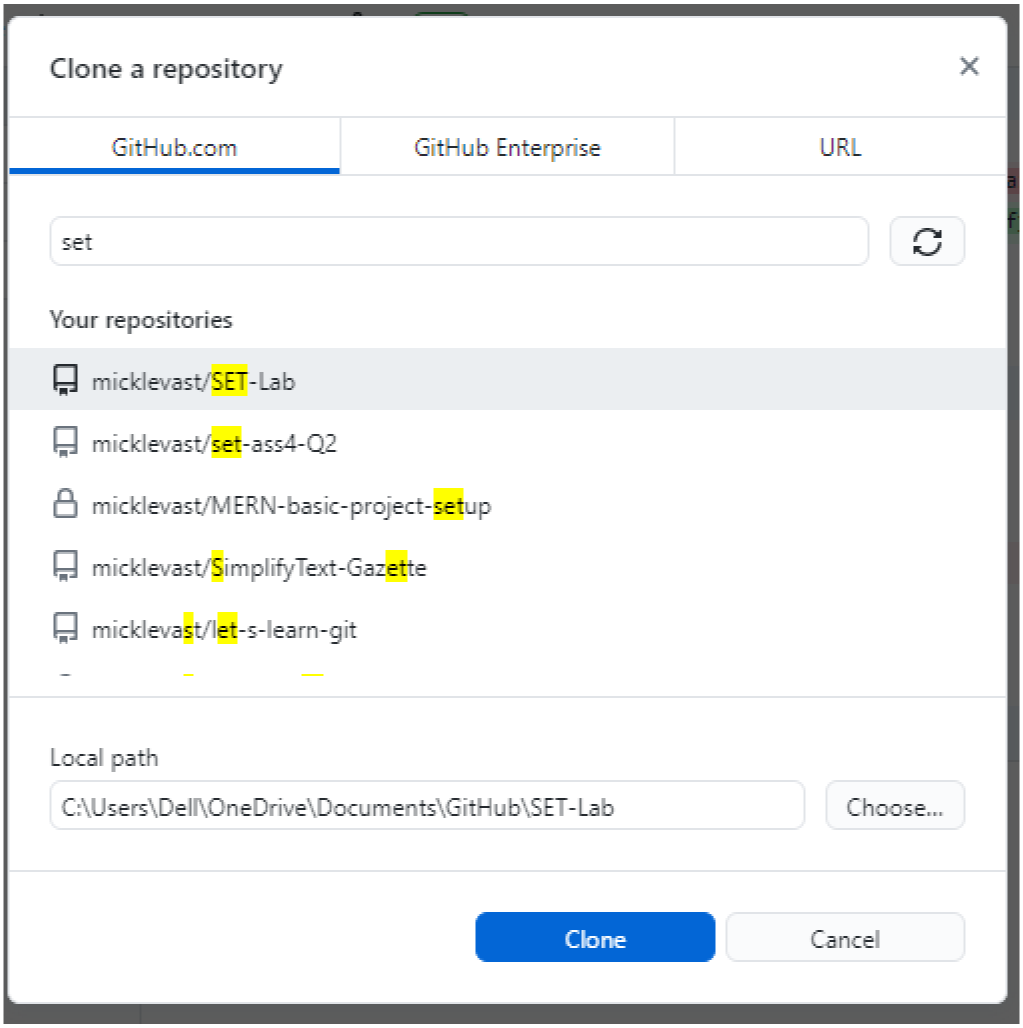
GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise. Sign in below to get started with your existing projects.

New to GitHub? Create your free account.

6. Choose a local path where you want to clone the repository, and click on the "Clone" button.



```
@@ -1,5 +1,5 @@
 {
-  "homepage": "https://micklevast.github.io/SimplifyText-Gazette"
+  "homepage": "https://micklevast.github.io/Ind-TextBeautifygit",
   "name": "god-1",
   "version": "0.1.0",
   "private": true,
@@ -16,7 +16,6 @@
   },
   "scripts": {
     "predeploy": "npm run build",
-
     "deploy": "gh-pages -b master -d build",
     "start": "react-scripts start",
     "build": "react-scripts build",
```



set

Your repositories

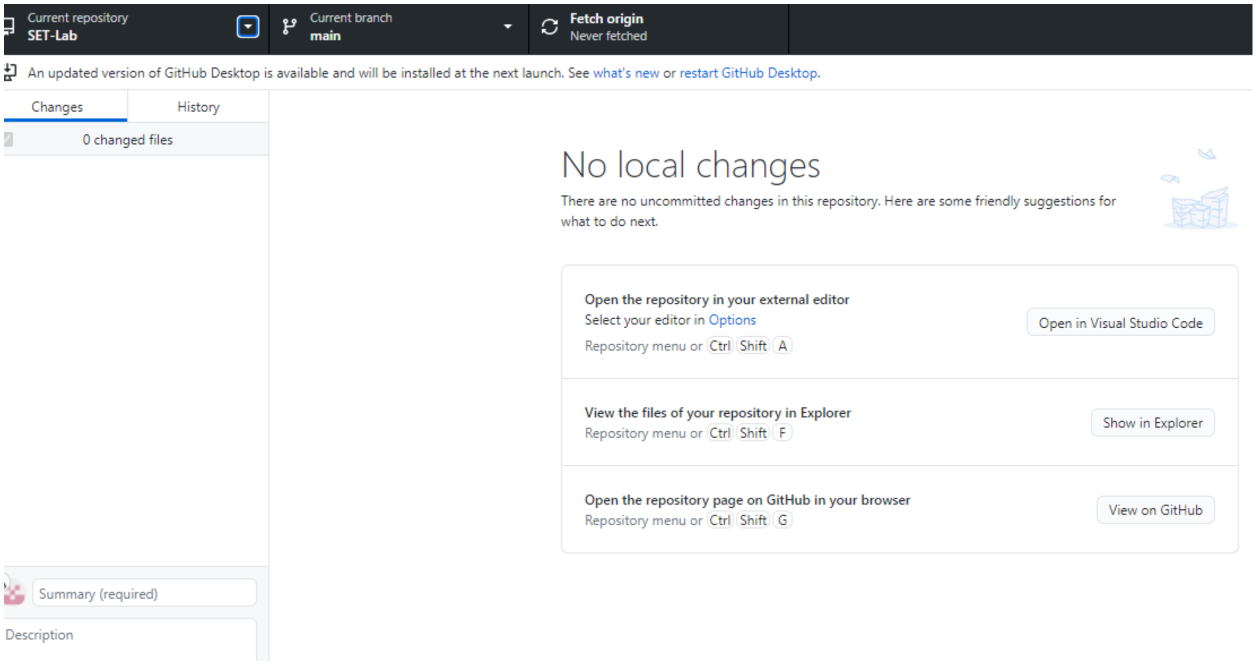
- micklevast/SET-Lab
- micklevast/set-ass4-Q2
- micklevast/MERN-basic-project-setup
- micklevast/SimplifyText-Gazette
- micklevast/let-s-learn-git

Local path

C:\Users\Dell\OneDrive\Documents\GitHub\SET-Lab

7. Once the repository is cloned, you can open it in your preferred code editor or IDE and start making changes.

⇒Accessing Previous repository using github dextop:



An updated version of GitHub Desktop is available and will be installed at the next launch. See what's new or restart GitHub Desktop.

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Open the repository in your external editor
Select your editor in Options
Repository menu or Ctrl Shift A

View the files of your repository in Explorer
Repository menu or Ctrl Shift F

Open the repository page on GitHub in your browser
Repository menu or Ctrl Shift G

Q 4. Differentiate in between GitHub, Git and GitLab.

Git is a distributed version control system used to manage source code and track changes over time. It allows developers to collaborate on a project and keep track of different versions of their code. Git is primarily used through a command-line interface, but there are also various graphical user interfaces available.

GitHub is a web-based platform that provides hosting for Git repositories. It allows developers to collaborate on projects, share code, and track issues and bugs. GitHub provides additional features such as project management tools, pull requests, code reviews, and access control.

GitLab is also a web-based platform that provides hosting for Git repositories. Like GitHub, it offers project management tools, issue tracking, and access control. In addition, GitLab offers continuous integration and deployment features, which enable developers to automatically build, test, and deploy their code changes. GitLab can also be self-hosted, which provides additional control and customization options.

Q5. what is version control?

Version control is the management of changes to documents, code, or other types of files over time. It enables developers to track modifications made to a file or a set of files, revert to earlier versions, and collaborate with others on the same files.
Version control systems, such as Git, provide a way to organize and keep track of changes to code or other files. Instead of just having a single file with the latest version, version control systems maintain a history of changes made to the file, including who made the changes and

when. This history can be used to understand the evolution of the code or file over time and to identify and fix issues or bugs.

Version control also enables collaboration between multiple developers on the same codebase. Developers can work on separate branches of the code, make changes, and merge their changes back into the main codebase. This helps to avoid conflicts and allows for parallel development of different features.

Overall, version control is a critical tool for software development and other types of collaborative projects, as it enables effective management and tracking of changes over time.