

Lesson-End Project

Implementing Advance Operations in Git

Project agenda: To execute advanced Git operations for enhanced version control and collaboration

Description: Imagine you are a software developer who has been asked to work on the Codex repository on GitHub. The goal is to solve issues using Git. First, create the Codex repo. Then, perform key Git tasks like tagging and branching. Understand Git rebase versus Git revert, explore Git log, and use Git rm with Git status. This project aims to improve teamwork and issue tracking for the Codex repository.

Tools required: Git and GitHub

Prerequisites: You must have Git installed in the lab to proceed.

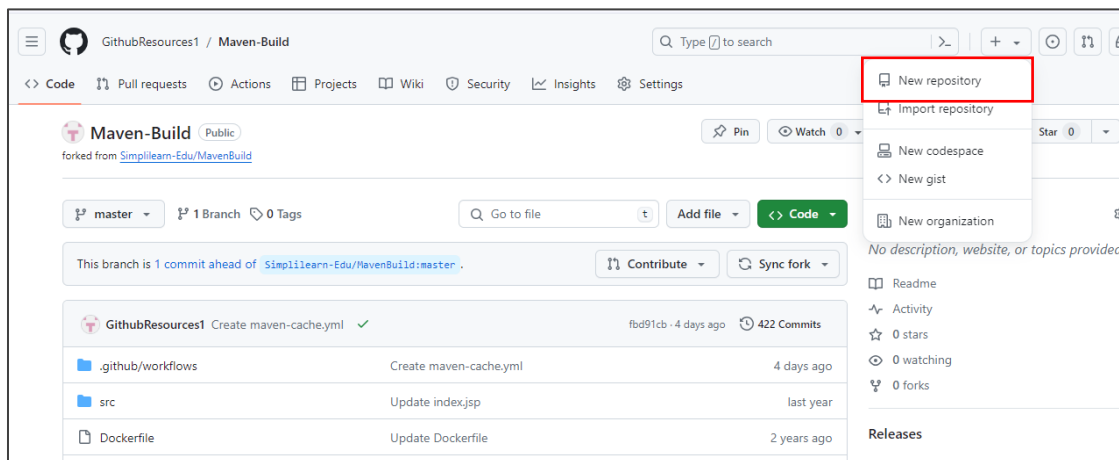
Expected deliverables: Creation of a GitHub repository named Codex with specified operations executed

Steps to be followed:

1. Create a new repository in Git
2. Create a tag in Git
3. Create a new branch in Git
4. Revert to the previous commit
5. Rebase the branch to integrate the changes
6. Remove the files from the Git index

Step 1: Create a new repository in Git

1.1 Click on the + icon in the upper-right corner of the page and select **New repository** from the drop-down menu



1.2 Provide the **Repository name** as **Codex**, select the **Add a README file**, choose **Public** for the repository type, and click on **Create repository**

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Required fields are marked with an asterisk ().*

Owner *

GithubResources1

Repository name *

Codex

Codex is available.

Great repository names are short and memorable. Need inspiration? How about [bookish-octo-broccoli](#) ?

Description (optional)

☒ 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:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore
.gitignore template: **None** ▾
Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license
License: **None** ▾
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository

Step 2: Create a tag in Git

2.1 Open the terminal tab and execute the following command to create and navigate into the **Codex** project directory:

mkdir Codex

cd Codex

```
syedsharozsimpl@ip-172-31-40-171:~$ mkdir Codex
syedsharozsimpl@ip-172-31-40-171:~$ cd Codex
syedsharozsimpl@ip-172-31-40-171:~/Codex$ █
```

2.2 Execute the following command to clone the repository:

git clone "repo path"

cd "repo name"

```

syedsharozsimpl@ip-172-31-40-171:~/Codex$ git clone https://github.com/GithubRes
sources1/Codex.git
Cloning into 'Codex'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
Receiving objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 1 (delta 0), pack reused 0
syedsharozsimpl@ip-172-31-40-171:~/Codex$ cd Codex
syedsharozsimpl@ip-172-31-40-171:~/Codex/Codex$ █

```

2.3 Execute the following command to create a new file:

touch testfile1

```

syedsharozsimpl@ip-172-31-40-171:~/Codex/Codex$ touch testfile1
syedsharozsimpl@ip-172-31-40-171:~/Codex/Codex$ █

```

2.4 Execute the following command to commit the changes and create a tag in Git:

git commit -m "soruce ver 1.0"

git tag -a 1.0 -m"version 1.0"

```

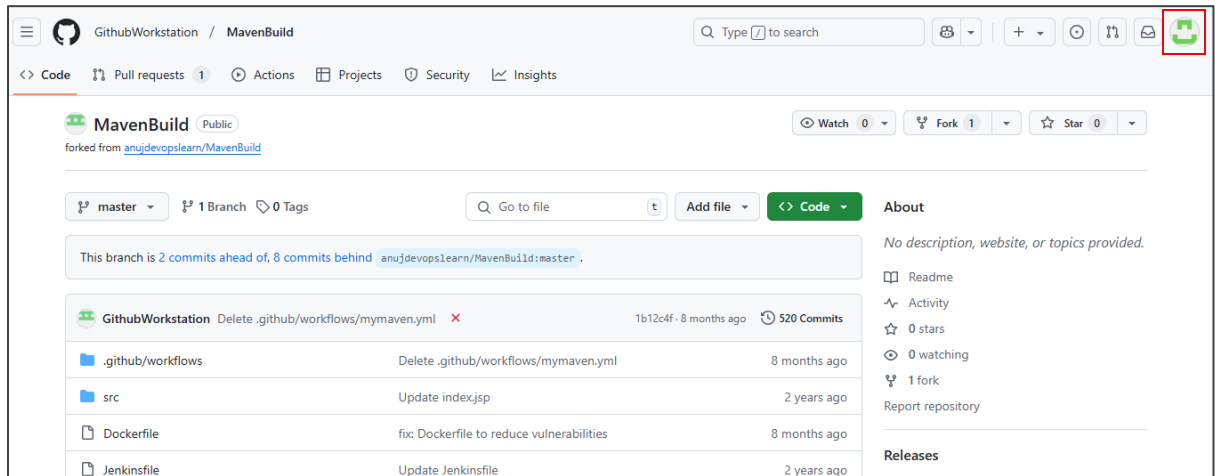
syedsharozsimpl@ip-172-31-40-171:~/Codex/Codex$ git commit -m "soruce ver 1.0"
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    testfile1

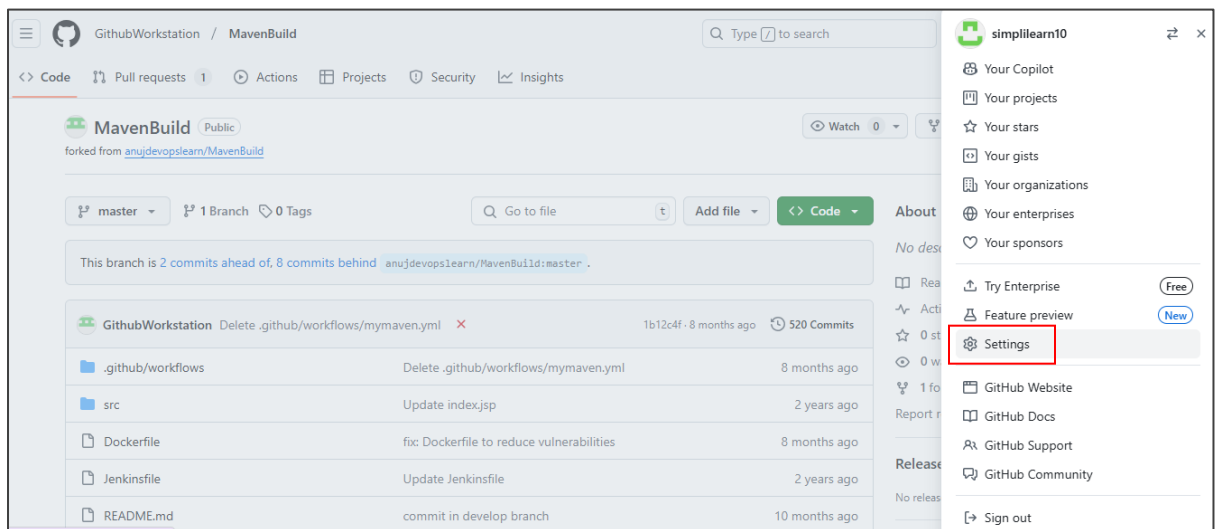
nothing added to commit but untracked files present (use "git add" to track)
syedsharozsimpl@ip-172-31-40-171:~/Codex/Codex$ git tag -a 1.0 -m"version 1.0"
syedsharozsimpl@ip-172-31-40-171:~/Codex/Codex$ █

```

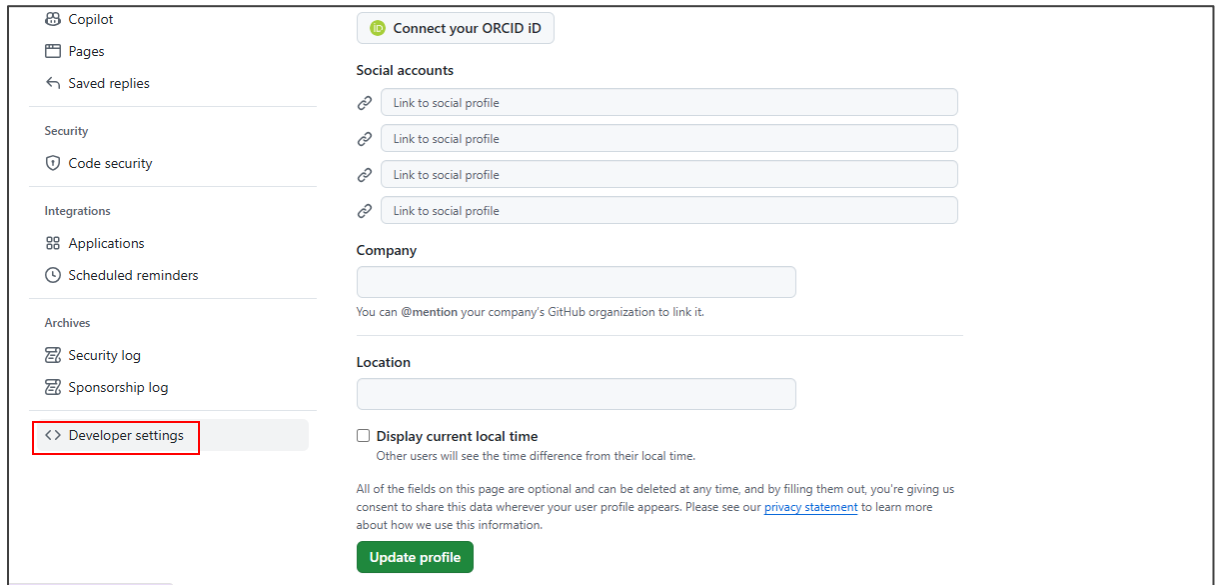
2.5 Navigate back to your github account and click on your profile in the top right corner



2.6 Click **Setting**



2.7 Click **Developer settings**



The screenshot shows the GitHub Developer Settings page. On the left is a sidebar with navigation links: Copilot, Pages, Saved replies, Security, Code security, Integrations, Applications, Scheduled reminders, Archives, Security log, and Sponsorship log. The 'Developer settings' link is highlighted with a red box. The main content area includes a 'Connect your ORCID iD' button, a 'Social accounts' section with four 'Link to social profile' buttons, a 'Company' section with a text input field and a note about mentioning the company's GitHub organization, a 'Location' section with a text input field, and a 'Display current local time' checkbox. At the bottom, there is a paragraph of disclaimer text and an 'Update profile' button.

Copilot

Pages

Saved replies

Security

Code security

Integrations

Applications

Scheduled reminders

Archives

Security log

Sponsorship log

<> Developer settings

Connect your ORCID iD

Social accounts

Link to social profile

Link to social profile

Link to social profile

Link to social profile

Company

You can @mention your company's GitHub organization to link it.

Location

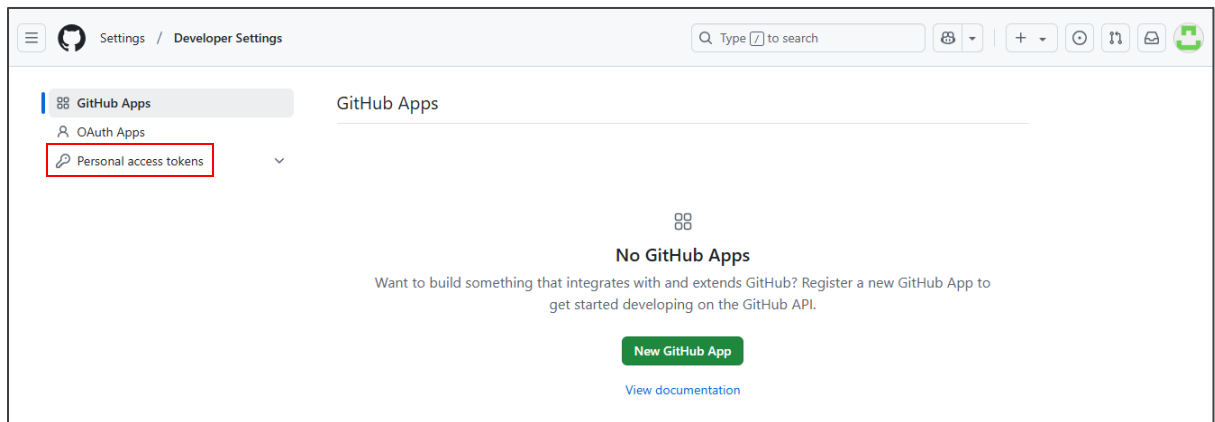
☐ Display current local time

Other users will see the time difference from their local time.

All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us consent to share this data wherever your user profile appears. Please see our [privacy statement](#) to learn more about how we use this information.

Update profile

2.8 Click **Personal access tokens**



The screenshot shows the GitHub Developer Settings page for 'GitHub Apps'. The left sidebar has 'GitHub Apps' selected, with 'Personal access tokens' highlighted by a red box. The main content area is titled 'GitHub Apps' and displays 'No GitHub Apps' with a message encouraging the user to register a new GitHub App to integrate with the GitHub API. A 'New GitHub App' button and a 'View documentation' link are provided at the bottom.

Settings / Developer Settings

GitHub Apps

OAuth Apps

Personal access tokens

GitHub Apps

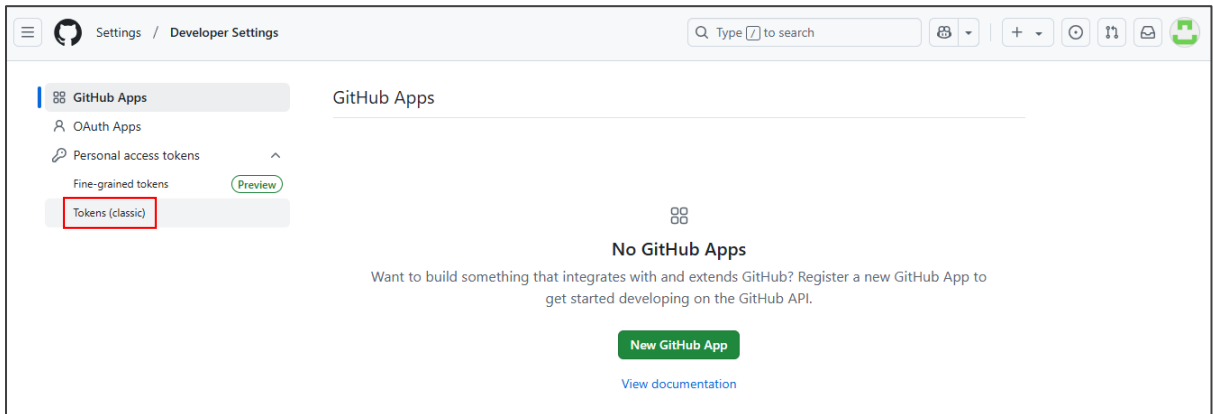
No GitHub Apps

Want to build something that integrates with and extends GitHub? Register a new GitHub App to get started developing on the GitHub API.

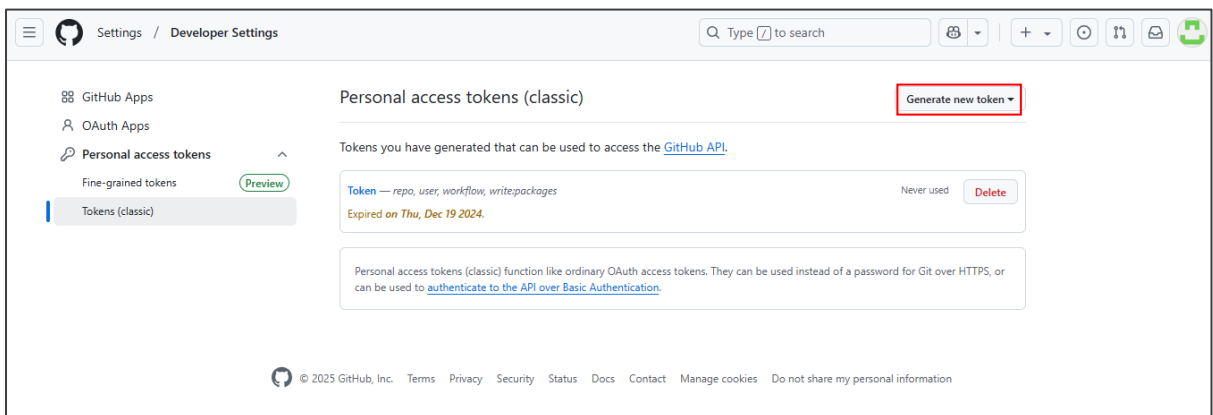
New GitHub App

[View documentation](#)

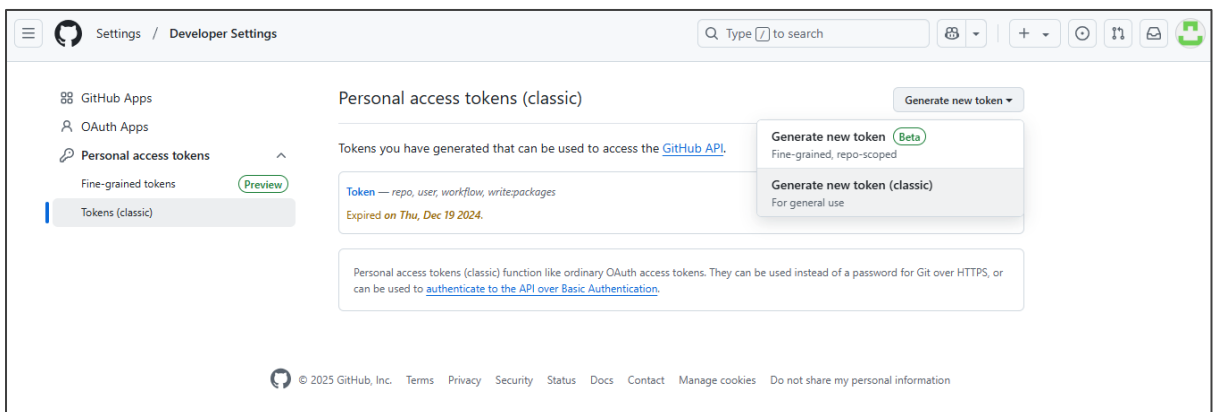
2.9 Click **Tokens (classic)**



2.10 Click **Generate new token**



2.11 Click **Generate new token (classic)**



2.12 Provide a name to the token

The screenshot shows the GitHub interface for creating a new personal access token. The left sidebar contains navigation links: GitHub Apps, OAuth Apps, Personal access tokens (selected), Fine-grained tokens (with a 'Preview' button), and Tokens (classic). The main content area is titled 'New personal access token (classic)'. It includes a 'Note' section, a 'Token' input field (highlighted with a red box), an 'Expiration' dropdown set to '30 days', and a 'Select scopes' section with a list of checkboxes and their descriptions.

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

Token

What's this token for?

Expiration *

30 days ▾ The token will expire on Sun, Mar 2 2025

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events

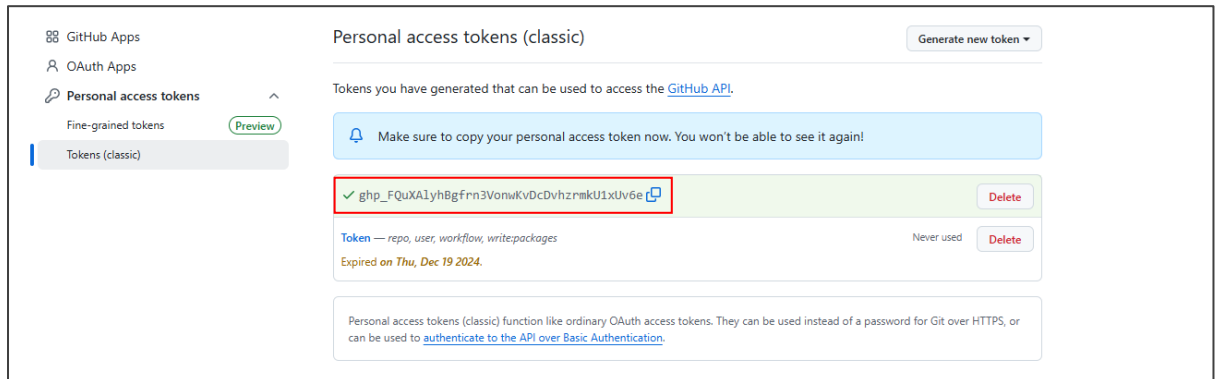
2.13 Scroll down and click **Generate token**

The screenshot shows the bottom portion of the GitHub 'Generate token' page. It features a list of checkboxes and their descriptions, followed by a 'Generate token' button (highlighted with a red box) and a 'Cancel' link.

<input type="checkbox"/> codespace	Full control of codespaces
<input type="checkbox"/> codespace:secrets	Ability to create, read, update, and delete codespace secrets
<input type="checkbox"/> copilot	Full control of GitHub Copilot settings and seat assignments
<input type="checkbox"/> manage_billing:copilot	View and edit Copilot Business seat assignments
<input type="checkbox"/> project	Full control of projects
<input type="checkbox"/> read:project	Read access of projects
<input type="checkbox"/> admin:gpg_key	Full control of public user GPG keys
<input type="checkbox"/> write:gpg_key	Write public user GPG keys
<input type="checkbox"/> read:gpg_key	Read public user GPG keys
<input type="checkbox"/> admin:ssh_signing_key	Full control of public user SSH signing keys
<input type="checkbox"/> write:ssh_signing_key	Write public user SSH signing keys
<input type="checkbox"/> read:ssh_signing_key	Read public user SSH signing keys

Generate token Cancel

2.14 Copy the token



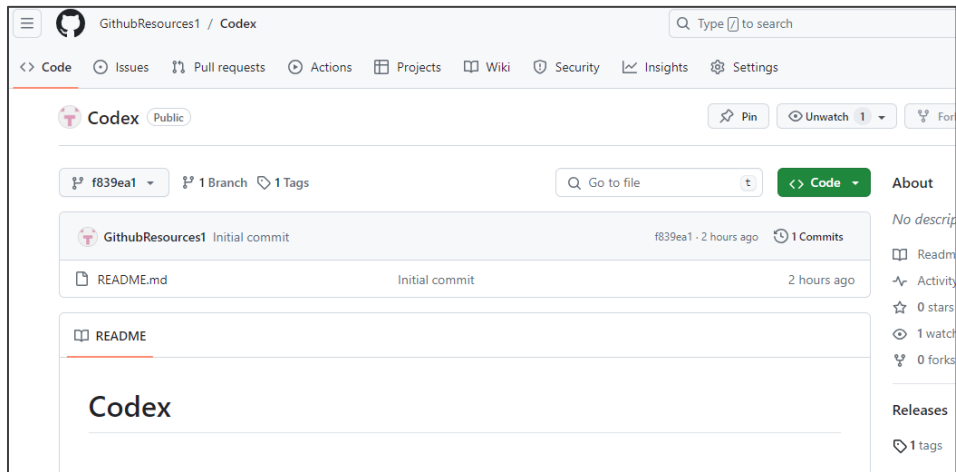
2.15 Navigate back to your terminal and execute the following command to push a tag:

git push origin "tag name"

```
syedsharozsimpl@ip-172-31-40-171:~/Codex/Codex$ git push origin 1.0
Username for 'https://github.com': GithubResources1
Password for 'https://GithubResources1@github.com':
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 159 bytes | 159.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/GithubResources1/Codex.git
 * [new tag]          1.0 -> 1.0
syedsharozsimpl@ip-172-31-40-171:~/Codex/Codex$
```

Note: Enter your GitHub account name as the username, and paste the token copied in Step 2.14 as the password.

2.16 Navigate back to your GitHub repository to check the tag



You can see the tag has been pushed.

Step 3: Create a new branch in Git

3.1 Open the terminal tab and execute the following command to create and switch to a new branch:

git branch "branch name"

git checkout "branch name"

```
syedsharozsimpl@ip-172-31-40-171:~/Codex/Codex$ git branch engineering
syedsharozsimpl@ip-172-31-40-171:~/Codex/Codex$ git checkout engineering
Switched to branch 'engineering'
syedsharozsimpl@ip-172-31-40-171:~/Codex/Codex$
```

3.2 Execute the following commands to create and add a file in the **engineering** branch:

touch file1

git add .

```
syedsharozsimpl@ip-172-31-40-171:~/Codex/Codex$ touch file1
syedsharozsimpl@ip-172-31-40-171:~/Codex/Codex$ git add .
syedsharozsimpl@ip-172-31-40-171:~/Codex/Codex$
```

3.3 Execute the following command to commit the changes:

git commit -m "added files"

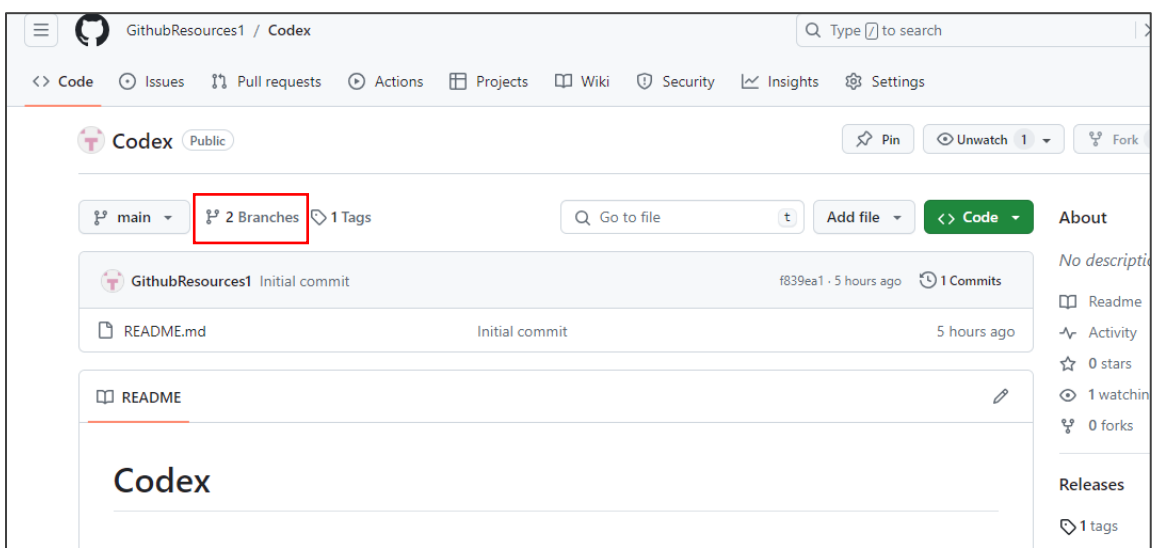
```
syedsharozsimpl@ip-172-31-40-171:~/Codex/Codex$ git commit -m "added files"
[engineering 6758793] added files
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file1
create mode 100644 testfile1
```

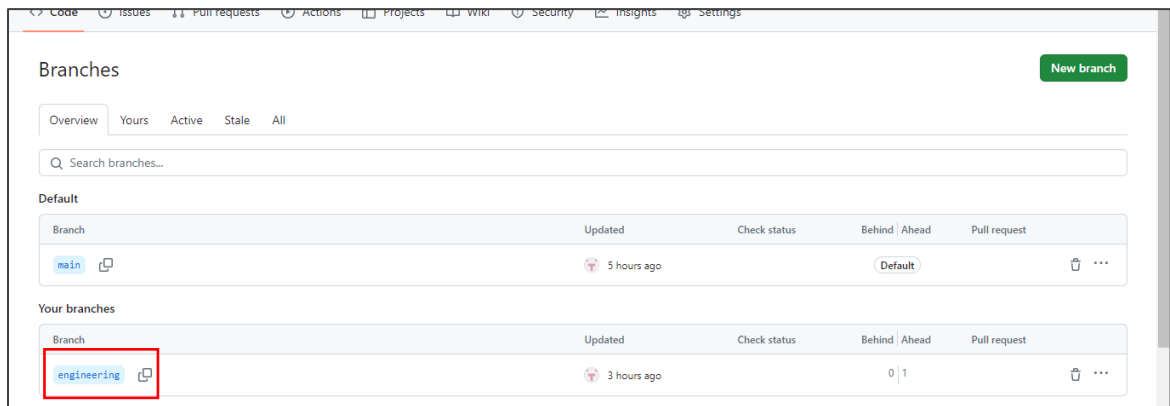
3.4 Execute the following command to push the changes to the remote repository:

git push origin "new branch name"

```
syedsharozsimpl@ip-172-31-40-171:~/Codex/Codex$ git push origin engineering
Username for 'https://github.com': GithubResources1
Password for 'https://GithubResources1@github.com':
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), 279 bytes | 279.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'engineering' on GitHub by visiting:
remote:   https://github.com/GithubResources1/Codex/pull/new/engineering
remote:
To https://github.com/GithubResources1/Codex.git
 * [new branch]      engineering -> engineering
syedsharozsimpl@ip-172-31-40-171:~/Codex/Codex$
```

3.5 Navigate to the **Codex** repository in GitHub, and click on **Branches** to verify whether the branch has been pushed or not





You can see that a new branch is created.

Step 4: Revert to the previous commit

4.1 Use the command given below to create and switch to a new branch:

git checkout -b test-branch

```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git checkout -b test-branch
Switched to a new branch 'test-branch'
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$
```

4.2 Execute the following commands to create a new file and modify it:

touch example.txt

echo "Hello, this is an example file." > example.txt

```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ touch example.txt
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ echo "Hello, this is an example file." > example.txt
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$
```

4.3 Run the following commands to add and commit the changes:

git add example.txt

git commit -m "Add example file"

```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git add example.txt
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git commit -m "Add example file"
[test-branch 0691c04] Add example file
1 file changed, 1 insertion(+)
create mode 100644 example.txt
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ █
```

4.4 Run the following command to push the changes to the **test-branch** of the remote repository:

git push origin test-branch

```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git push origin test-branch
Username for 'https://github.com': GithubResources1
Password for 'https://GithubResources1@github.com':
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), 353 bytes | 353.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'test-branch' on GitHub by visiting:
remote:   https://github.com/GithubResources1/Codex/pull/new/test-branch
remote:
To https://github.com/GithubResources1/Codex.git
 * [new branch]      test-branch -> test-branch
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$
```

4.5 Run the following commands to modify the **example.txt** file further:

touch example.txt

echo "Hello world" > example.txt

```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ touch example.txt
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ echo "Hello world" > example.txt
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ █
```

4.6 Run the following commands to add and commit the changes:

git add example.txt

git commit -m "Modify example file"

```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git add example.txt
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git commit -m "Modify example file"
[test-branch 3982012] Modify example file
1 file changed, 1 insertion(+), 1 deletion(-)
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ █
```

4.7 Execute the following command to inspect the commit history and identify the commit for reversal:

git log --oneline

```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git log --oneline
3982012 (HEAD -> test-branch) Modify example file
0691c04 (origin/test-branch) Add example file
6758793 (origin/engineering, feature-branch, engineering) added files
f839ea1 (tag: 1.0, origin/main, origin/HEAD) Initial commit
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ █
```

4.8 Run the given command to revert to the preceding commit:

git reset HEAD ~1

```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git reset HEAD~1
Unstaged changes after reset:
M   example.txt
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ █
```

4.9 Run the provided command to verify the commit history and confirm the successful reversal:

git log --oneline

```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git log --oneline
0691c04 (HEAD -> test-branch, origin/test-branch) Add example file
6758793 (origin/engineering, feature-branch, engineering) added files
f839ea1 (tag: 1.0, origin/main, origin/HEAD) Initial commit
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ █
```

Step 5: Rebasing the branch to integrate the changes

5.1 Execute the below command to create and switch to a new branch:

git branch testing

git checkout testing

```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git branch testing
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git checkout testing
Switched to branch 'testing'
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ █
```

5.2 Create a file, stage the changes in the current working directory, and commit the staged changes using the following commands:

touch file5

git add .

git commit -m "files added"

```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ touch file5
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git add .
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git commit -m "files added"
[testing 9a65704] files added
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file5
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$
```

5.3 Create another file, stage the changes, and commit the staged changes using the following commands:

touch file6

git add .

git commit -m "file6 added"

```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ touch file6
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git add .
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git commit -m "file6 added"
[testing 9fdf708] file6 added
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file6
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ █
```

5.4 Run the following command to display the commit history of a repository:

git log

```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git log
commit 9fdf708dc0d591447572d20472e5738fe3ba4d1c (HEAD -> testing)
Author: Gitresource <githubresource@gmail.com>
Date: Mon Apr 22 12:32:53 2024 +0000

    file6 added

commit 9a65704307d849899c40a10d5cf070c8f4ccb08c
Author: Gitresource <githubresource@gmail.com>
Date: Mon Apr 22 12:27:41 2024 +0000

    files added
```

5.5 Switch to the main branch using the following command:

git checkout main


```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 3 commits.
(use "git push" to publish your local commits)
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$
```

5.6 List the branch that you are currently working on using the following command:

git branch

```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git branch
* main
  testing
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$
```

5.7 Create a file inside a main branch, stage the changes in the current working directory, and commit the staged changes using the following commands:

touch file7

git add .

git commit -m "file7 added"

```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ touch file7
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git add .
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git commit -m "file7 added"
[main 301f7c1] file7 added
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file7
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$
```

5.8 Run the following command to display the commit history of a repository:

git log

```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git log
commit 301f7c1b182809f60ed971cce73816f76a58fed (HEAD -> main)
Author: Gitresource <githubresource@gmail.com>
Date: Mon Apr 22 12:47:09 2024 +0000

    file7 added
```

5.9 Run the following commands to rebase both the branches and check the commit history of a repository after rebasing:

git rebase testing main

git log

```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git rebase testing main
Successfully rebased and updated refs/heads/main.
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git log
commit 1a6e629df5f9357e87266eacb12d35000d815606 (HEAD -> main)
Author: Gitresource <githubresource@gmail.com>
Date: Mon Apr 22 12:47:09 2024 +0000

    file7 added

commit 9fdf708dc0d591447572d20472e5738fe3ba4d1c (testing)
Author: Gitresource <githubresource@gmail.com>
Date: Mon Apr 22 12:32:53 2024 +0000

    file6 added

commit 9a65704307d849899c40a10d5cf070c8f4ccb08c
Author: Gitresource <githubresource@gmail.com>
Date: Mon Apr 22 12:27:41 2024 +0000

    files added
```

Step 6: Remove the files from the Git index

6.1 Execute the following command to list all the files in the current branch:

ls

```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ ls
file5 file6 file7
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$
```

6.2 Execute the following commands to remove **file5** in the current branch and check the status:

git rm file5

ls

git status

```
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git rm file5
rm 'file5'
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ ls
file6 file7
ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ git status
On branch main
Your branch is ahead of 'origin/main' by 6 commits.
(use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    file5

ravitulsianisim@ip-172-31-31-214:~/Codex/Codex$ █
```

You can see that **file5** has been removed.

By following these steps, you have successfully implemented advanced Git operations to enhance version control and collaboration.