

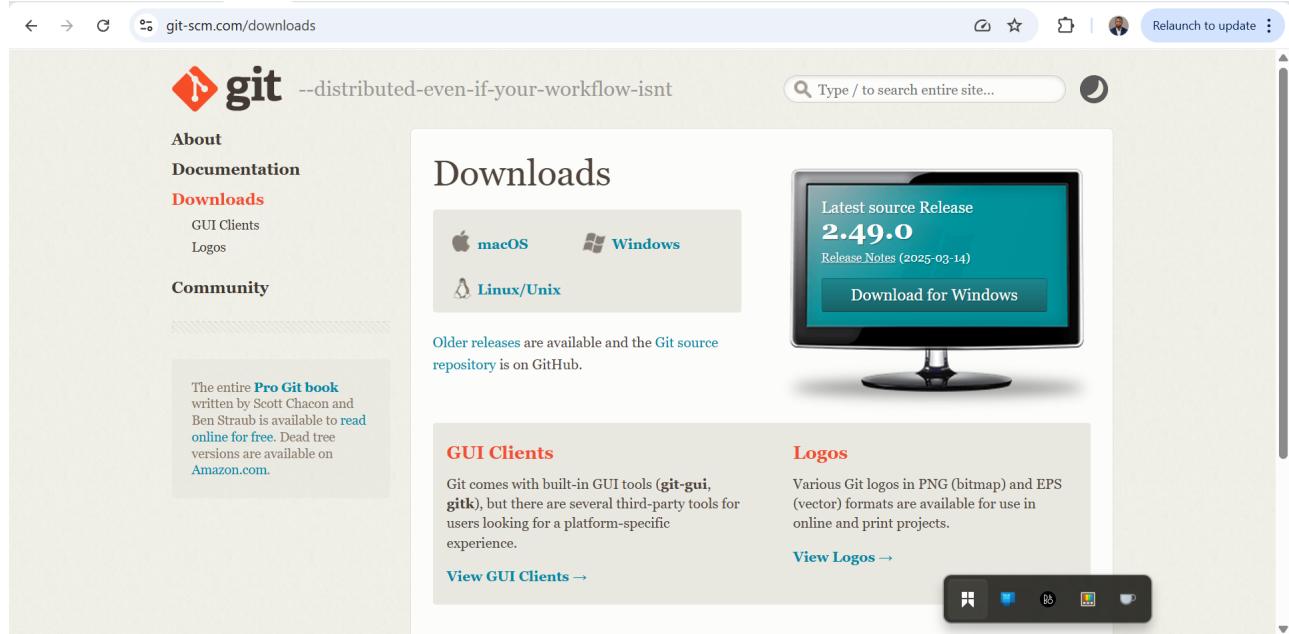
Hands-On Git Project: Collaborative Website Development with Git and GitHub

In this mini project, we simulate a collaborative Git and GitHub workflow using two contributors — Tom and Jerry. The project covers Git installation, setting up a GitHub repository, cloning, branching, modifying content, committing changes, and pushing to the main repository.

Part 1: Setup and Initial Configuration

1. Install Git

- Visit the [official Git website](#) and [download](#) the version of Git for your operating system.
- Follow the installation instructions.



2. Create a GitHub Repository

- Sign up or log in to [GitHub](#).
- Click the "+" icon in the top-right corner and select "**New repository**".

The screenshot shows a GitHub user profile for 'Oluwaseunoa'. The profile picture is a circular portrait of a man with short hair and a beard, wearing a dark shirt. The user's name is 'oluwaseunoa' and they are listed as 'Oluwaseunoa · he/him'. Below the profile, it says 'Cybersecurity || DevOps'. A sidebar on the right contains a list of repository options: 'New repository' (highlighted with a red box), 'Import repository', 'New codespace', 'New gist', 'New organization', and 'New project'. Popular repositories listed include 'currency-converter' (Public, 1 star, 1 commit), 'drumkit' (Public, 1 star, 1 commit), 'Nested-For-Loop' (Public, 1 star, 1 commit), 'Oluwaseun-Bank-Presentation' (Public, 1 star, 1 commit), and 'Portfolio' (Public, 1 star, 1 commit).

- Name your repository (e.g., [03. Mini Project - Basic Git Commands](#)) and initialize it with a **README** file.
- Click "**Create repository**".

The screenshot shows the 'Create repository' form on GitHub. It includes fields for 'Description (optional)' containing 'This is a repository as a DevOps Engineer', a 'Public' radio button selected (highlighted with a red box), a 'Private' radio button, 'Initialize this repository with:' (checkbox checked for 'Add a README file'), and a 'Create repository' button at the bottom right (highlighted with a red box). Other sections include 'Add .gitignore' (template: None), 'Choose a license' (None selected), and a note about the default branch being 'main'.

3. Clone the Repository

- On your repository page on GitHub, click the "**Code**" button and copy the **HTTPS URL**.

You now have push access to the Oluwaseunoa/DevOps-Projects repository.

DevOps-Projects Public

main · 1 Branch · 0 Tags

Oluwaseunoa Added initial commit of the AI Website

- 01. Tech Environment Setup Initial commit
- 02. Git-Version-Control-Basics Added initial commit
- README.md Initial commit

README

About

No description, website, or topics provided.

Activity

0 stars

1 watching

0 forks

Report repository

Releases

No releases published

Create a new release

Packages

No packages published

[Publish your first package](#)

- Open your terminal or command prompt.
- Create a folder named **DevOps-Projects** in your desired directory (e.g., **Documents/Workspace/DevOps-Projects**) and navigate to it:

```
mkdir ~/Documents/Workspace/DevOps-Projects
cd ~/Documents/Workspace/DevOps-Projects
```

- Clone (download) the repository using:

```
git clone https://github.com/Oluwaseunoa/DevOps-Projects.git
```

```
MINGW64:/c/Users/HP/Documents/Workspace/Workspace
$ cd workspace/
$ git clone https://github.com/Oluwaseunoa/DevOps-Projects.git
Cloning into 'DevOps-Projects'...
remote: Enumerating objects: 81, done.
remote: Counting objects: 100% (81/81), done.
remote: Compressing objects: 100% (69/69), done.
```

- Your default branch will be **main**.
- Navigate into the cloned repository:

```
cd /DevOps-Projects/03. Mini Project - Basic Git Commands/
```

- Create a file named `index.html` and add the content below:

```
touch index.html
echo "This is the Admin creating an index.html file for Tom and Jerry." >
index.html
```

- Check that changes haven't been staged:

```
git status
```

- Stage the changes:

```
git add index.html
```

- Confirm staging:

```
git status
```

The file name will appear in green, indicating it is staged.

- Commit the changes:

```
git commit -m "This is my first commit"
```

- Push the `main` branch to GitHub:

```
git push origin main
```

This sends commits from your main branch on your laptop to GitHub (Remote Repository).

深厚的 Part 2: Simulating Tom and Jerry's Work

We'll simulate both Tom and Jerry working on the same machine by using branches.

深厚的 Tom's Work

- Navigate to the project directory:

```
cd DevOps-Projects\03. Mini Project - Basic Git Commands
```

- Check the current branch:

```
git branch
```

- Create a new branch for Tom's changes:

```
git checkout -b update-navigation
```

A screenshot of a terminal window titled 'MINGW64:c/Users/HP/Documents/Workspace/Workspace/DevOps-Projects'. The window shows the command history:

```
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/Workspace  
$ cd DevOps-Projects/  
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/Workspace/DevOps-Projects (main)  
$ git checkout -b update-navigation  
Switched to a new branch 'update-navigation'  
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/Workspace/DevOps-Projects (update-navigation)  
$
```

- Check the active branch again:

```
git branch
```

A screenshot of a terminal window showing the current branch is 'update-navigation'. The command history is as follows:

```
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/Workspace/DevOps-Projects (update-navigation)  
$ git branch  
  main  
* update-navigation  
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/Workspace/DevOps-Projects (update-navigation)  
$
```

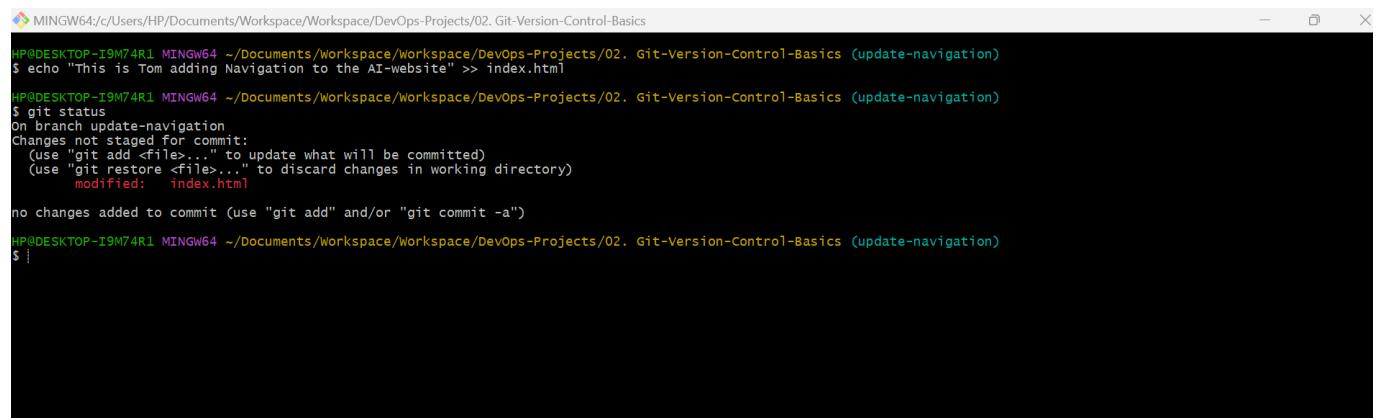
- Modify `index.html` and add:

This is Tom adding Navigation to the AI-website.

This simulates Tom's contribution to the project. This text represents the work he's doing on the navigation bar. In the real world, this will be an actual software code.

- Check for unstaged changes:

```
git status
```



A screenshot of a terminal window titled "MINGW64" showing the output of a "git status" command. The terminal shows the user is on branch "update-navigation". It displays a single unstaged change: "modified: index.html". The terminal window has standard operating system window controls at the top right.

```
MINGW64:c/Users/HP/Documents/Workspace/Workspace/DevOps-Projects/02. Git-Version-Control-Basics
$ echo "This is Tom adding Navigation to the AI-website" >> index.html
$ git status
On branch update-navigation
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
      modified:   index.html

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

► The file name will appear in red, indicating changes are unstaged.

- Stage the changes:

```
git add index.html
```

- Confirm staging:

```
git status
```

```
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/Workspace/DevOps-Projects/02. Git-Version-Control-Basics (update-navigation)
$ echo "This is Tom adding Navigation to the AI-website" >> index.html
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/Workspace/DevOps-Projects/02. Git-Version-Control-Basics (update-navigation)
$ git status
On branch update-navigation
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
      modified:   index.html

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

HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/Workspace/DevOps-Projects/02. Git-Version-Control-Basics (update-navigation)
$ git add index.html
warning: in the working copy of '02. Git-Version-Control-Basics/index.html', LF will be replaced by CRLF the next time Git touches it

HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/Workspace/DevOps-Projects/02. Git-Version-Control-Basics (update-navigation)
$ git status
On branch update-navigation
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   index.html

HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/Workspace/DevOps-Projects/02. Git-Version-Control-Basics (update-navigation)
$
```



- Commit Tom's changes:

```
git commit -m "Update navigation bar"
```

- Push Tom's branch:

```
git push origin update-navigation
```

```
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/Workspace/DevOps-Projects/02. Git-Version-Control-Basics (update-navigation)
$ git commit -m "Update navigation bar"
[update-navigation 5c82073] Update navigation bar
 1 file changed, 1 insertion(+), 1 deletion(-)

HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/Workspace/DevOps-Projects/02. Git-Version-Control-Basics (update-navigation)
$ git push origin update-navigation
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 399 bytes | 133.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/Oluwaseunoa/DevOps-Projects.git
  02ac05f..5c82073 update-navigation > update-navigation

HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/Workspace/DevOps-Projects/02. Git-Version-Control-Basics (update-navigation)
$
```

👤 Jerry's Work

- Switch back to the **main** branch:

```
git checkout main
```

- Pull latest changes (e.g., Tom's update):

```
git pull origin update-navigation
```

```
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/Workspace/DevOps-Projects/02. Git-Version-Control-Basics (main)
$ git pull origin update-navigation
From https://github.com/Oluwaseunoa/DevOps-Projects
 * branch            update-navigation -> FETCH_HEAD
Updating 02ac05f..5c82073
Fast-forward
 02. Git-Version-Control-Basics/index.html | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/Workspace/DevOps-Projects/02. Git-Version-Control-Basics (main)
$
```

- Create a new branch for Jerry's work:

```
git checkout -b add-contact-info
```

- Edit `index.html` and add the following content:

```
Contact us at: contact@ai-startup.com
```

- Stage Jerry's changes:

```
git add index.html
```

- Commit Jerry's changes:

```
git commit -m "Add contact information"
```

- Push Jerry's branch:

```
git push origin add-contact-info
```

```
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/Workspace/DevOps-Projects/02. Git-Version-Control-Basics (edit-contact-info)
$ git push origin edit-contact-info
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 392 bytes | 65.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
remote:
remote: Create a pull request for 'edit-contact-info' on GitHub by visiting:
remote:     https://github.com/Oluwaseunoa/DevOps-Projects/pull/new/edit-contact-info
remote:
To https://github.com/Oluwaseunoa/DevOps-Projects.git
 * [new branch]      edit-contact-info -> edit-contact-info

HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/Workspace/DevOps-Projects/02. Git-Version-Control-Basics (edit-contact-info)
$
```

Summary

At this point:

- Tom created a branch `update-navigation` and added navigation details.
- Jerry created a branch `add-contact-info` and added contact information.
- Both contributors successfully pushed their branches to GitHub.

🌐 GitHub Repository

 [View on GitHub](#)