Introduction to Web Development

A Student-Run Short Course (SRC) conducted by the Student Academic Council in collaboration with the Technical Council.

Instructors:
Reuben Devanesan, Ananthu JP,
Kanishk Singhal, Progyan Das

What is Open Source?

Open Source software is software that can be freely accessed, used, changed, and shared (in modified or unmodified form) by anyone.

Open source software is made by many people, and distributed under licenses that comply with the Open Source Definition.

What is Git?

Git is a distributed version control system. It is an open source project originally developed by Linus Torvalds in 2005.

Advantages:

- The entire codebase and history is available on every developer’s computer, which allows for easy branching and merging.
- Git is not fooled by the names of the files when determining what the storage and version history of the file tree should be, instead, Git focuses on the file content itself.
- The content of the files as well as the true relationships between files and directories, versions, tags and commits, all of these objects in the Git repository are secured with a cryptographically secure hashing algorithm called SHA1.

What is Version Control?

Version control is the practice of tracking and managing changes to software code.

Version control systems (VCS) help software developers manage changes to the source code by keeping track of every modification to the code.

Version control helps teams by tracking every individual change by each contributor and helping prevent concurrent work from conflicting.

Git is one of the most popular VCS tool in use today.

What is Github?

Github is a cloud based service that helps developers manage their code, as well as track and control changes to their code.

Github makes it easier for developers and teams to use Git for version control and collaboration.

Github hosts git repositories. A repository encompasses the entire collection of files and folders associated with a project, along with each file’s revision history.

Create an account of Github: https://github.com/

Git Commands

Git Commands

- **`git init:`** This is one way to start a new project with Git. It transforms the current directory into a Git repository. This command creates a hidden directory called `.git` This hidden `.git` directory is what separates a regular directory from a Git repository.

- **`git remote add origin <URL>:`** This command adds the remote URL to your local git repository. This stores the remote URL under a more human-friendly name, `origin`.

Git Commands

git add: This command adds new or changed files in your working directory to the Git staging area. This is an important command - without it, no git commit would ever do anything.

- **`git add <path>`**: Stage a specific directory or file.
- **`git add .`** : Stage all files (that are not listed in the `.gitignore`) in the entire repository. The `.gitignore` file is a text file that tells Git which files or folders to ignore in a project.

Git Commands

git commit : This command creates a commit, which is like a snapshot of your repository. These commits are snapshots of your entire repository at specific times. You should make new commits often, based around logical units of change.

- **`git commit -m "<descriptive commit message>":`** This starts the commit process, and allows you to include the commit message at the same time.

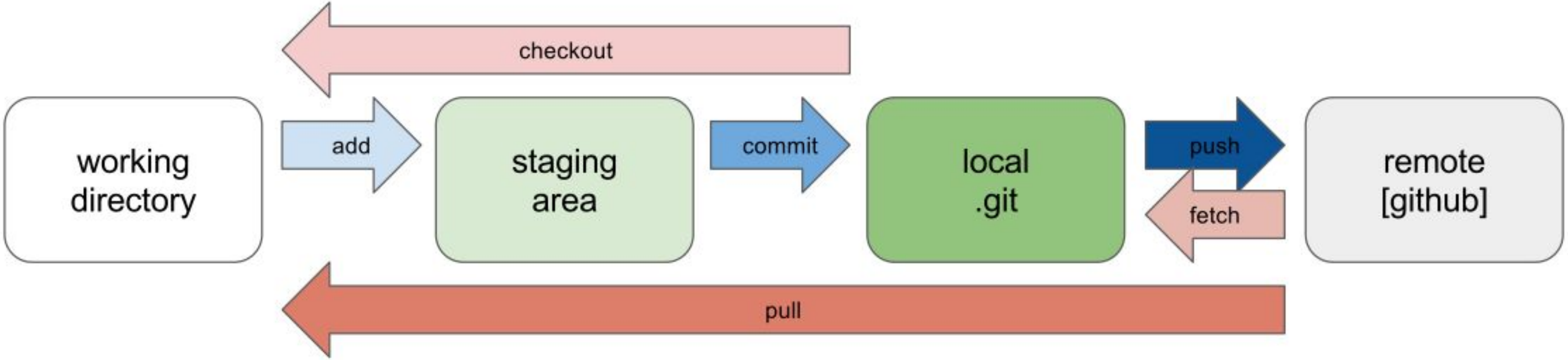
Git Commands

- **`git push:`** This command updates the remote branch with local commits. You can also think of git push as update or publish. Pushing changes to the remote makes your commits accessible to others who you may be collaborating with. As best practice, it's important to run the `git pull` command before you push any new changes to the remote branch.
- **`git push --set-upstream origin <branch name>:`** Sets the default remote branch to <branch name>.
- **`git push --all:`** Push all branches

Git Commands

- **`git pull:`** This command updates your current local working branch, and all of the remote tracking branches. It's a good idea to run `git pull` regularly on the branches you are working on locally. It's important to run this command before you push (`git push)` any new changes to the remote branch.
- **`git pull --all:`** Fetch *all* remotes - this is handy if you are working on a fork or in another use case with multiple remotes.
- **`git revert <UID of commit>:`** This is the safest way to change history with Git. Instead of deleting existing commits, it looks at the changes introduced in a specific commit, then applies the inverse of those changes in a new commit. This is always the recommended way to change history when it's possible.

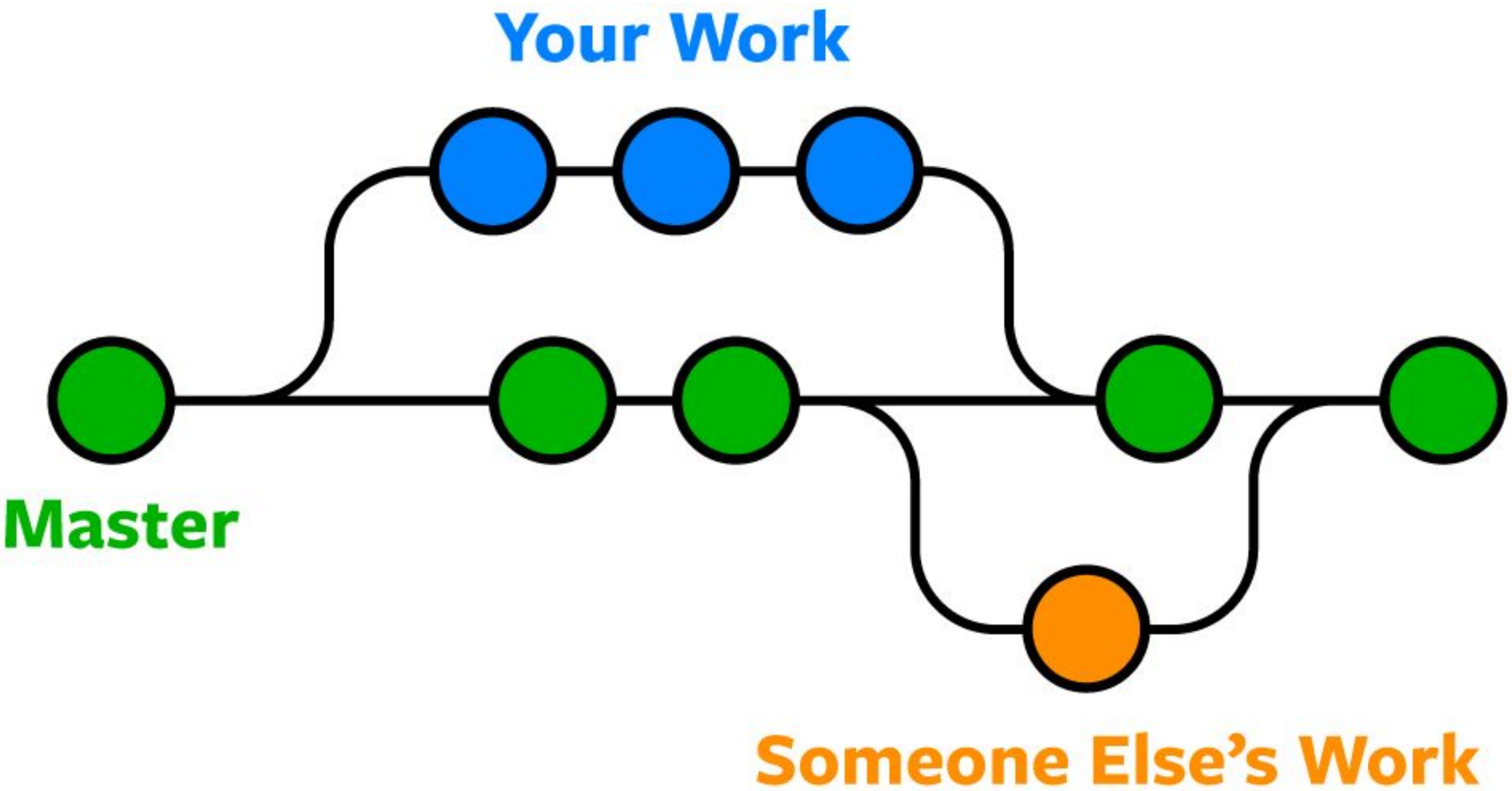
Git Workflow



https://uidaholib.github.io/get-git/images/workflow.png

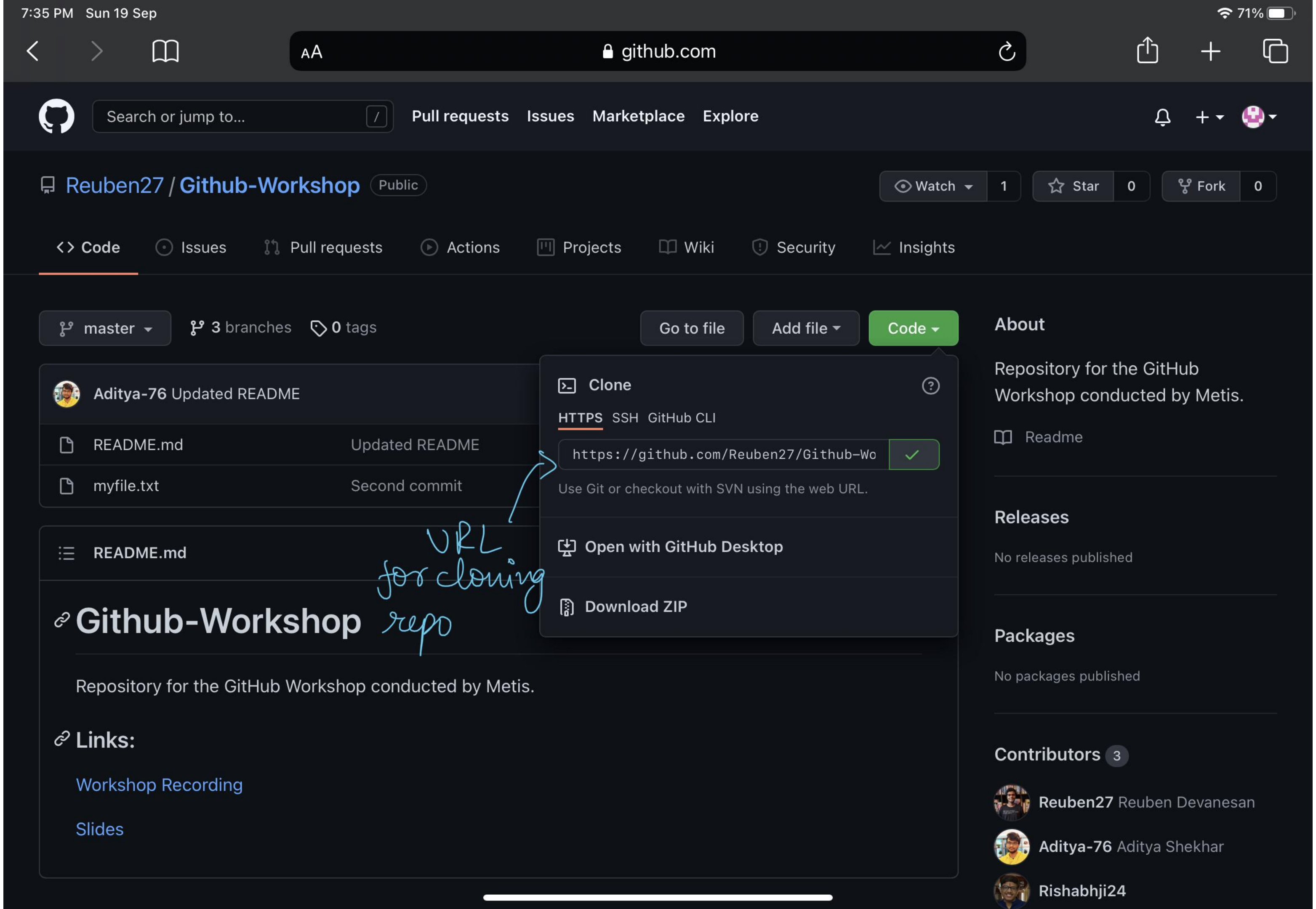
Git Commands

- **`git branch:`** List branches (the asterisk denotes the current branch)
- **`git branch -a:`** List all branches (local and remote)
- **`git checkout <branch name>:`** This command lets you navigate between the branches created by `git branch`.
- **`git checkout -b <new branch name>:`** Creates a new branch.
- **`git checkout -b <new-branch> <existing-branch>:`** Creates a new branch from the existing branch.


Your Work
Master
Someone Else's Work

Git Commands

- **`git clone [url]:`** Clone (download) a repository that already exists on GitHub, including all of the files, branches, and commits.
- **`git clone [url] --branch [branch] --single-branch:`** Clone only a single branch. Cloning only one branch does not add any benefits unless the repository is very large and contains binary files that slow down the performance of the repository. The recommended solution is to optimize the performance of the repository before relying on single branch cloning strategies.



7:35 PM Sun 19 Sep

71%

github.com

Search or jump to... / Pull requests Issues Marketplace Explore

Reuben27 / Github-Workshop Public

Watch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights

master 3 branches 0 tags Go to file Add file Code

Aditya-76 Updated README

README.md	Updated README
myfile.txt	Second commit

Clone

HTTPS SSH GitHub CLI

```
https://github.com/Reuben27/Github-Wo
```

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

URL for cloning repo

README.md

Github-Workshop

Repository for the GitHub Workshop conducted by Metis.

Links:

Workshop Recording

Slides

About

Repository for the GitHub Workshop conducted by Metis.

Readme

Releases

No releases published

Packages

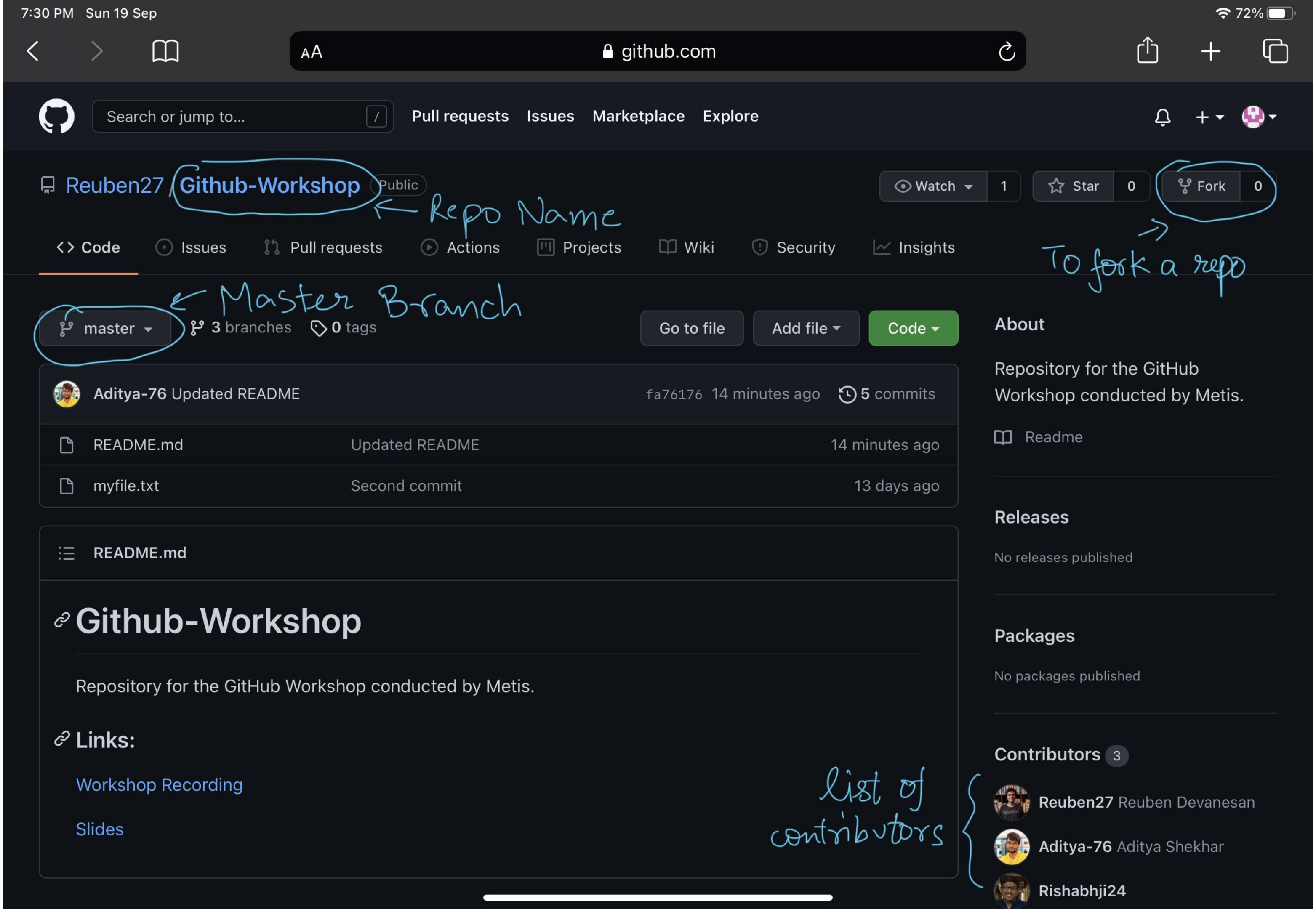
No packages published

Contributors 3

Reuben27 Reuben Devanesan

Aditya-76 Aditya Shekhar

Rishabhji24



Repo Name

To fork a repo

Master Branch

Aditya-76 Updated README	fa76176 14 minutes ago	5 commits
README.md	Updated README	14 minutes ago
myfile.txt	Second commit	13 days ago

README.md

Github-Workshop

Repository for the GitHub Workshop conducted by Metis.

Links:

Workshop Recording

Slides

About

Repository for the GitHub Workshop conducted by Metis.

Readme

Releases

No releases published

Packages

No packages published

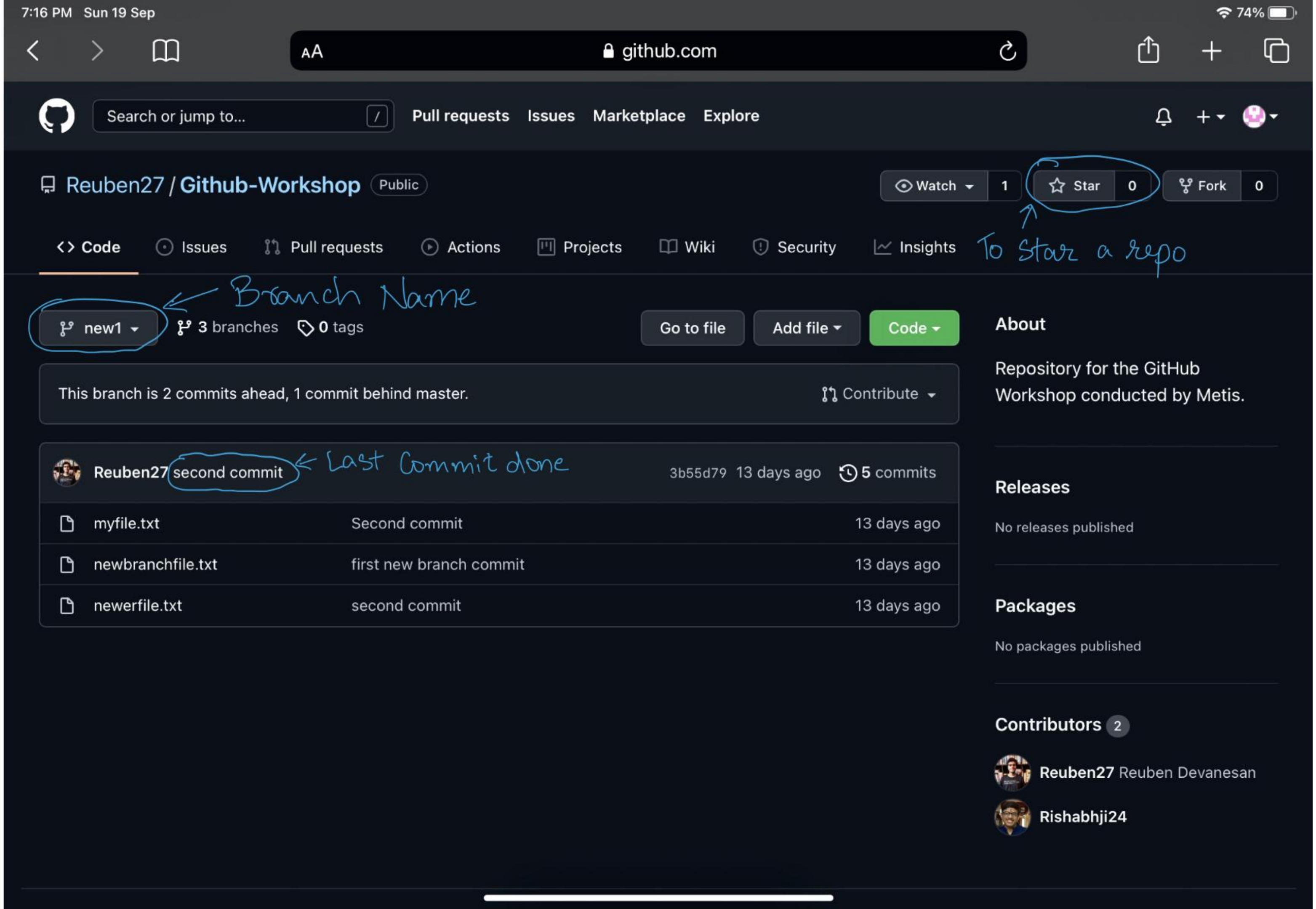
Contributors 3

Reuben27 Reuben Devanesan

Aditya-76 Aditya Shekhar

Rishabhji24

list of contributors



Reuben27 / Github-Workshop Public

To Star a repo

Branch Name

This branch is 2 commits ahead, 1 commit behind master.

Reuben27 second commit 3b55d79 13 days ago 5 commits

Last Commit done

myfile.txt	Second commit	13 days ago
newbranchfile.txt	first new branch commit	13 days ago
newerfile.txt	second commit	13 days ago

About

Repository for the GitHub Workshop conducted by Metis.

Releases

No releases published

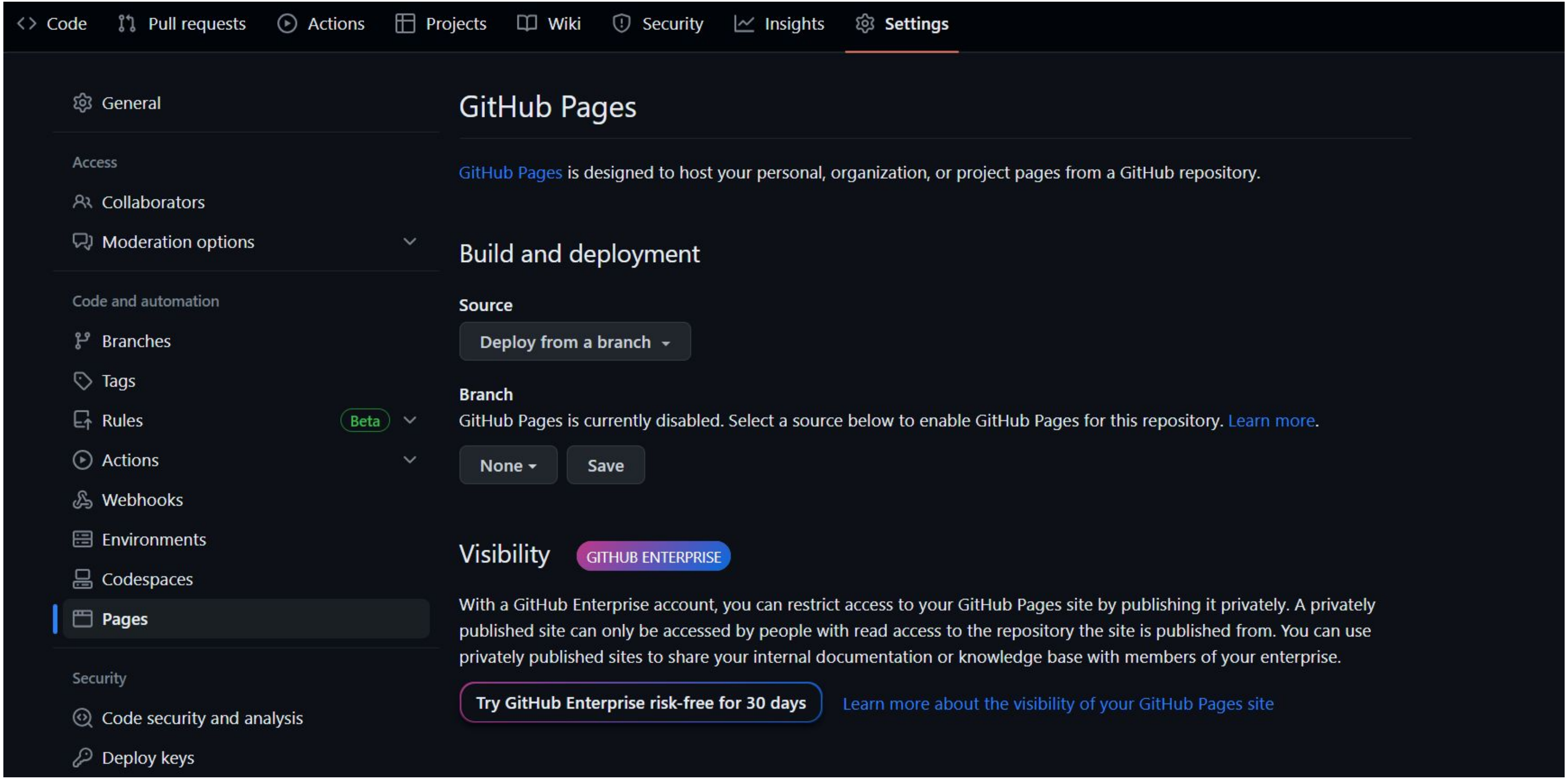
Packages

No packages published

Contributors 2

Reuben27 Reuben Devanesan

Rishabhji24



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Build and deployment

Source

Deploy from a branch

Branch

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. Learn more.

None | Save

Visibility

GITHUB ENTERPRISE

With a GitHub Enterprise account, you can restrict access to your GitHub Pages site by publishing it privately. A privately published site can only be accessed by people with read access to the repository the site is published from. You can use privately published sites to share your internal documentation or knowledge base with members of your enterprise.

Try GitHub Enterprise risk-free for 30 days

Learn more about the visibility of your GitHub Pages site

Host on GitHub pages

- You can host your static sites on GitHub.
- Just choose Branch and folder to be hosted and refresh a few times.

Thank you!