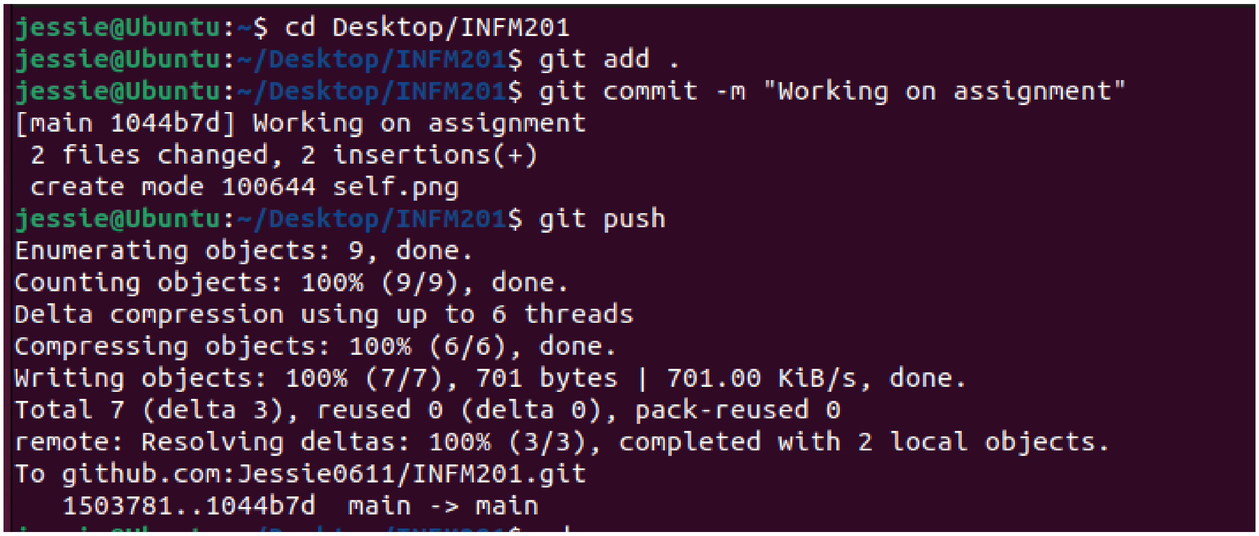
GitHub is a website designed for programmers. It provides a centralized location for programmers to store and share their code, collaborate on projects, track changes made, and manage project issues and bugs. Much of GitHub is navigated from your computer through the command line. If you are not familiar with commands this can cause some issues, however, nothing major that a quick google search cannot fix.

On your GitHub account you create a repository, or repo, which is a collection of files, resources, and documentations for specific projects. On the home page there is an easily recognized button to create a repository. A GitHub repo contains things like the source code, documentation, images, and other files inside the folder of the project. Users can create repositories for their own projects or contribute to existing repositories owned by others. When you create content on your repo you must commit the changes from your PC to your GitHub account. To do so you must submit a commit through the command line. When you open your command line you must navigate to the appropriate project folder on your PC. Once you are in the correct folder in the command line you will enter the commands, “git add .”, “git commit -m ‘message of commit here’.”, then “git push”. If you do not submit a commit after you make changes to the project on your PC then it will display an error message.

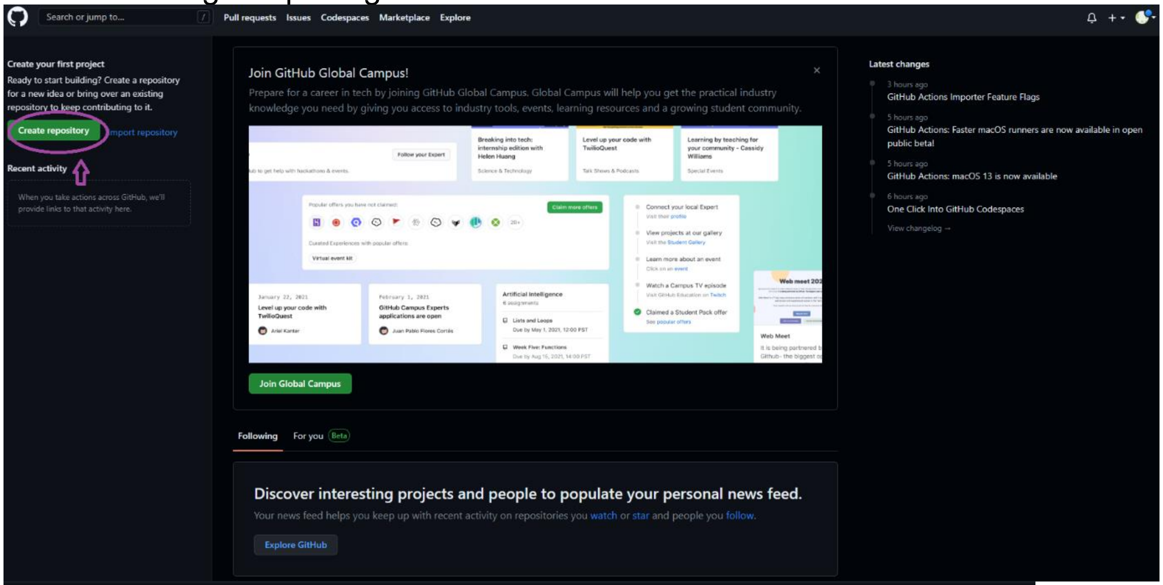
Improving GitHub would involve enhancing aspects of the platform to make it more user-friendly, efficient, and accommodating to those who may not feel so comfortable in the command line. Perhaps adding a page of resources available that touch base on common issues that happen with new GitHub users. Another option would be to implement and easier way to enter the project information directly onto the website itself. Another fun idea is to allow users to create a personalized dashboard with their own projects. To test the updates and new design implemented I would gather volunteers of new programmers in the beginner classes at local colleges to test out the user-experience for new users.

Ubuntu Command Line:



```
jessie@Ubuntu:~$ cd Desktop/INFM201
jessie@Ubuntu:~/Desktop/INFM201$ git add .
jessie@Ubuntu:~/Desktop/INFM201$ git commit -m "Working on assignment"
[main 1044b7d] Working on assignment
 2 files changed, 2 insertions(+)
 create mode 100644 self.png
jessie@Ubuntu:~/Desktop/INFM201$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 6 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 701 bytes | 701.00 KiB/s, done.
Total 7 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:Jessie0611/INFM201.git
   1503781..1044b7d  main -> main
```

## Creating a repo in git:



**Create your first project**

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

**Recent activity**

When you take actions across GitHub, we'll provide links to that activity here.

Join GitHub Global Campus!

Prepare for a career in tech by joining GitHub Global Campus. Global Campus will help you get the practical industry knowledge you need by giving you access to industry tools, events, learning resources and a growing student community.

Discover interesting projects and people to populate your personal news feed.

Your news feed helps you keep up with recent activity on repositories you watch or star and people you follow.

**Latest changes**

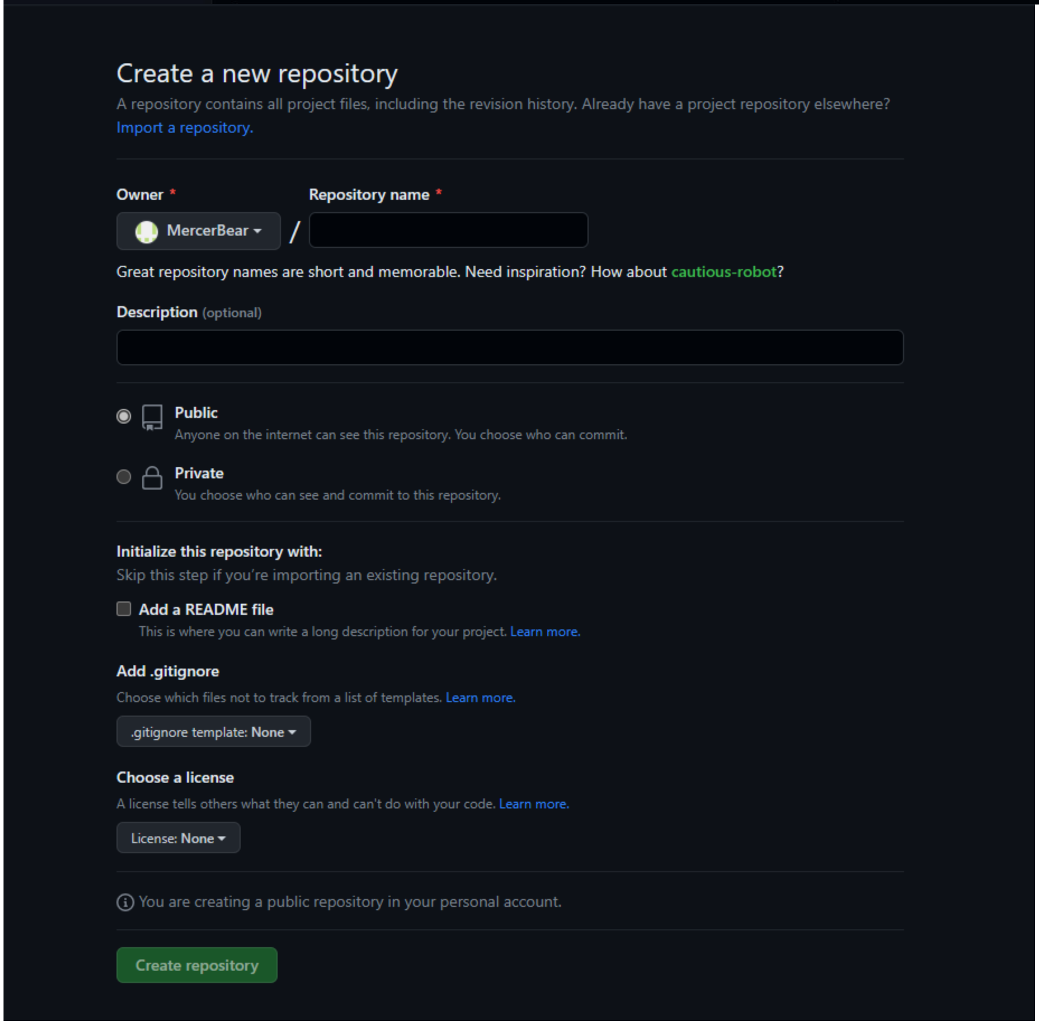
3 hours ago
GitHub Actions Importer Feature Flags

5 hours ago
GitHub Actions: Faster macOS runners are now available in open public beta!

5 hours ago
GitHub Actions: macOS 13 is now available

6 hours ago
One Click Into GitHub Codespaces

View changelog →



### Create a new repository

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

**Owner** * / **Repository name** *

MercerBear ▾ /

Great repository names are short and memorable. Need inspiration? How about cautious-robot?

**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:**
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.

.gitignore template: None ▾

**Choose a license**
A license tells others what they can and can't do with your code. Learn more.

License: None ▾

ⓘ You are creating a public repository in your personal account.