**GIT/Github    
Documentation :** [**https://www.atlassian.com/git/tutorials**](https://www.atlassian.com/git/tutorials)

Step 1 : **Install git for mac and windows**

Follow the link below to download git for mac and windows

Mac : <https://git-scm.com/download/mac>

Windows : <https://git-scm.com/download/win>

Once installed please test the following command in the terminal(mac) and git bash(windows)

* git  – version (two dashes - - )

Step 2 : **Create an account on github**

 To create an account on github use your email address and choose a unique username to create an account.

Step 3: **Create Repository on github**

  Create repositories

Graphical user interface, text, application, email

Description automatically generated

At the bottom of the page click on Create repository and it you will see the following page here.

Graphical user interface, text, application, email

Description automatically generated

echo "# pragra" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/Surinderpragra/pragra.git

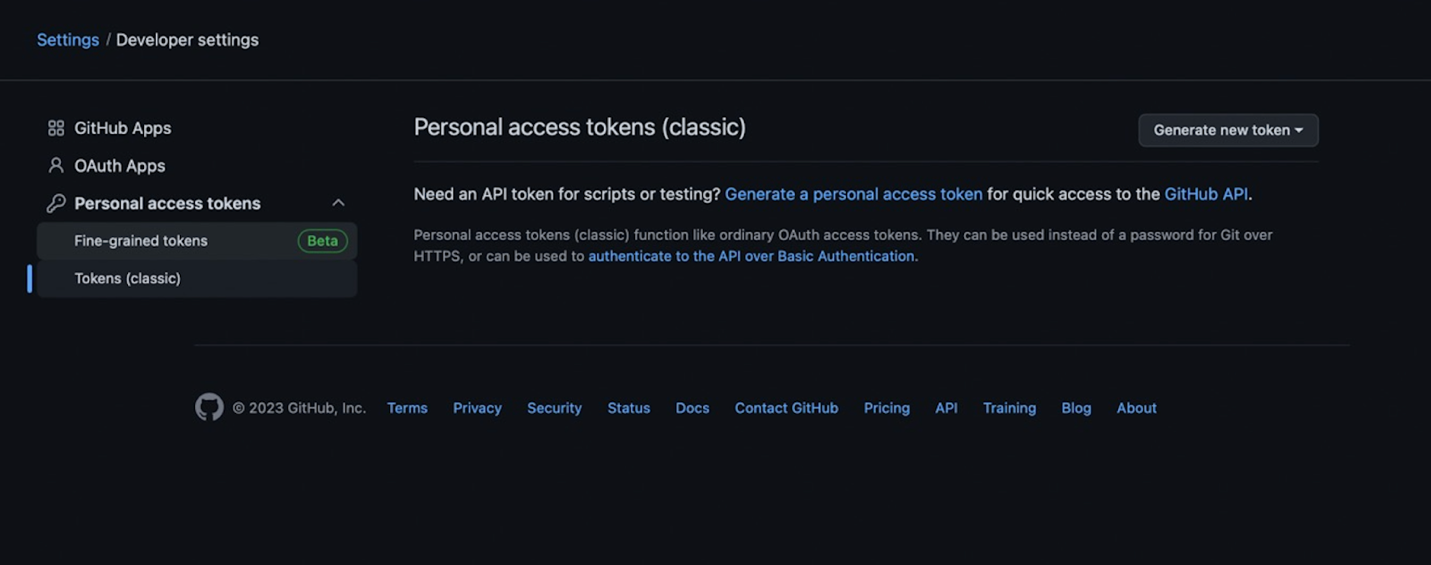
git push -u origin main

Step 4: Once you have create a repository, go to mac terminal or windows git bash or open intellij terminal for connecting local git to github

Commands :

* If you are connecting the github to local git for the first time, add the username and email with the following commands
* **NOTE** : Use your own username and email
* **git config --global user.name "pragra"**
* **git config --global user.email pragra**[**@gmail.com**](mailto:bugs@gmail.com)
* **git init** →  to initialize the git repository on local machine
* **git add .** → to add the git files to the staging area
* **git add file\_name**  → to just add one file
* **git commit -m “add you message inside these quotes”**
* **git remote add origin** [**https://github.com/Surinderpragra/pragra.git**](https://github.com/Surinderpragra/pragra.git)
* git push -u origin main   → this will push your local code/files to the remote(github account)

Now you will get a prompt to either login to github via username or password. If it asks for any login please add.   
Otherwise create a token from the link below

<https://github.com/settings/tokens>  
  
Graphical user interface, application

Description automatically generated

**Git Branching**

Diagram

Description automatically generated

**Git stages**

Diagram

Description automatically generated

========================================================================**Git commands:**

Git –help

Configure git :

Git config  - -global user.name “surinder”

Git config  - -global user.email [surinder032@gmail.com](mailto:surinder032@gmail.com)

Git config - -global color.ui auto

* git config -l   → to check the configuration
* git rm --cached index.html  → to untrack a file from git
* git rm --cached . -r   → for all files
* Git add -A → to add all the files even outside of current directory
* Git commit -m “added javascript code”
* Git log → to check the commit hash
* Git show 43a847a096ccc636a0289bd84d32780821f9aa40
* Git diff → to see the difference
* git commit --amend -m "added console log in index.js" → to make changes to current log message
* Git pull

Branch

* Git branch -r → to check the branch on remote
* Git branch -a → to check all the branches local and remote
* Git branch feature-a
* Git checkout -b feature-b → to create and checkout at the same time

**Merge branches**

If you want to merge branch A into Branch B

* Git checkout branch B
* Git merge A

===============

Branches

main/ master    – restricted (no one can directly push to this branch )

1. Develop
2. Staging

— > feature branch —>

Task 123 :

Create a branch called feature-task123

Feature1

Feature2

Feature3

  → master