Stasned Untracked files to→

git stash ...... git stash pop

git stash -a

git --version

git clone <link>

git status

ls

ls -a (Alternate)

git init

git add .

git commit -m "message"

git push origin main

Stash with message

git stash save "message text"

Find changes Done in Speific Stash

git stash show stash@{index}

Apply Speific stash

git stash apply stash@{index}

pop

git init

git remote add origin <link>

git remote -v (to verify remote)

git branch (to check branch)

git branch -M main (to rename branch)

git push origin main

Untracked

new files that git doesn't yet track

modified - Changed

Staged

file is ready to be Committed

Unmodified

Unchanged

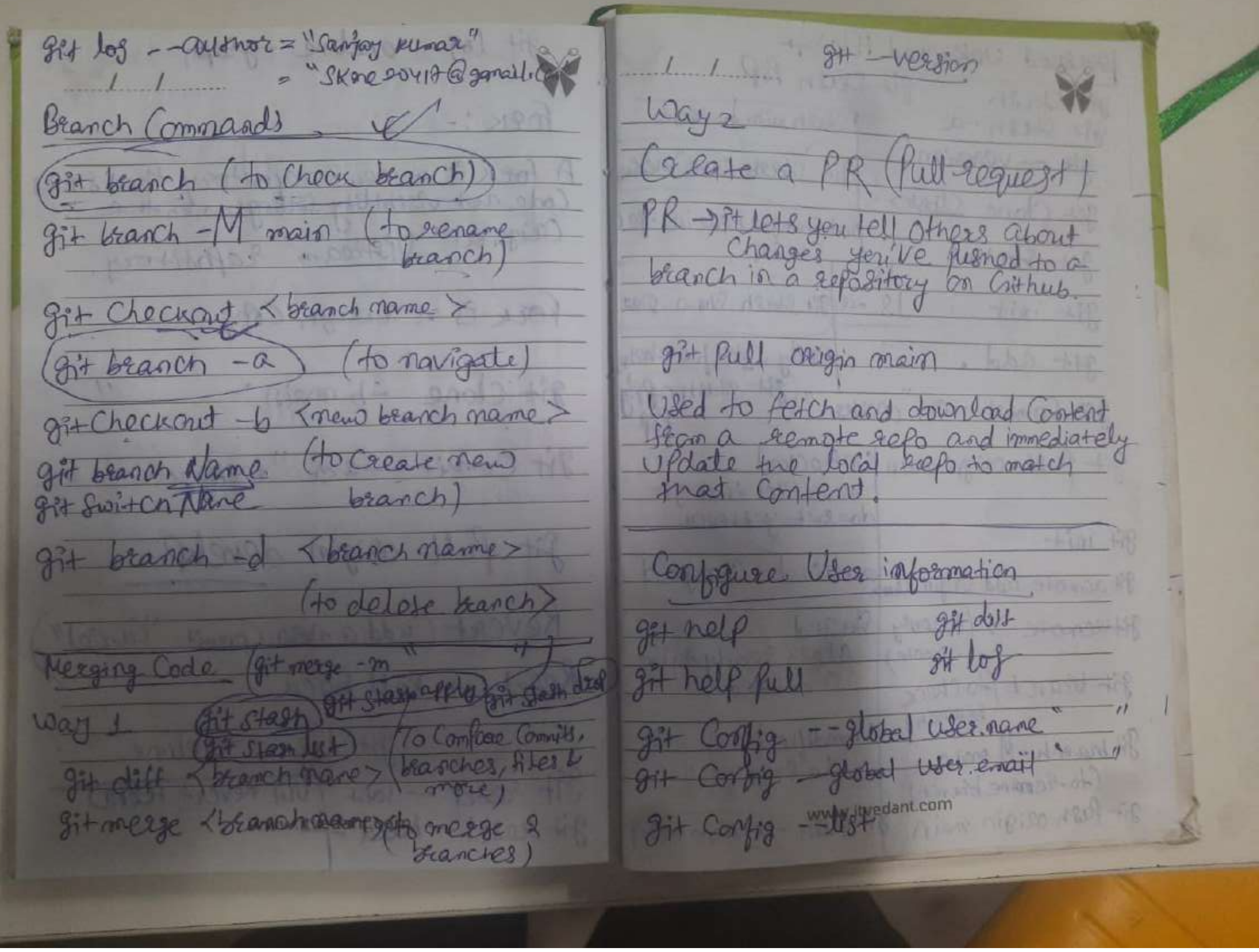
git push -u origin main

To set upstream

CELEBRATING OUR SELF-PLACED NINJAS

When the combined efforts of [illegible]'s training and our student's hard work lands them in lucrative job roles, we celebrate the achievement with a specially curated gift

Our Self-placed Avengers



git log --author = "Sanjay kumar"
" " = "Skne20417@gmail.c

Branch Commands

git branch (to Check branch)

git branch -M main (to rename branch)

git Checkout <branch name >

git branch -a (to navigate)

git Checkout -b <new branch name>

git branch Name (to Create new
git Switch Name branch)

git branch -d <branch name>
(to delete branch>

Merging Code (git merge -m " ")

way 1 git stash git stash apply git stash drop
git stash list
git diff <branch name> (To Compare Commits, branches, files & more)

git merge <branch name> (to merge 2 branches)

git --version

Way 2

Create a PR (Pull request)

PR → it lets you tell others about Changes you've pushed to a branch in a repository on Github.

git pull origin main

Used to fetch and download Content from a remote repo and immediately update the local repo to match that Content.

Configure User information

git help git diff
git help pull git log

git Config --global user.name " "
git Config --global user.email " "

git Config --list

git branch -m <oldName> <newName>

Resolving Merge Conflicts

An event that takes place when git is unable to automatically resolve differences in code between two commits.

Undoing Changes :-

git log
q
for
quit

Case 1 : Staged Changes

```
git reset <filename>
git reset
```

Case 2 : Commited Changes (for one Commit)

```
git reset HEAD~1
```

Case 3 : Commited Changes (for many Commits)

```
git reset <Commit hash>
git reset --hard <Commit hash>
```

```
git log --oneline --graph
```

Fork :-

A fork is a new repository that shares Code and visibility settings with the Original "Upstream" repository.

Fork is a rough Copy

```
git clone -b main "  "
git Commit -am "message"
git pull origin develop
```

Revert (add a new Commit "Cancel")

Reset → Roll back

```
git log --oneline
git reset --soft
git reset --hard
```

| git Revert HEAD