Git Workflow

Prepare Your Git Account:

* Create a GitHub Account --
* A repository - - "Create a repo."
* Create a Branch in your repository

- "Creating and deleting branches within your repository."

**General Workflow:**

1. Create a local repository.

**There are several ways to do this. Below are some of them.**

* Clone an existing the repository into a new directory AND switch to the new branch.

git clone <ssh key> <name of new branch>

* + - Switch to an existing branch:

Git switch <name of branch>

* + - Create a new branch and switch to it:

Git switch -c <new branch name>

* + - * -c – if a branch of the same name exists it will switch to it otherwise it creates a new branch with the new branch name.
      * -C – forces a fresh branch with new branch name. Resets the start point.

1. Make your changes.
2. Stage changes in the staging area.

git add . – for all

* + - ‘.’ – add all to staging area.
    - <file name> add specified files to staging area.
    - Note: you can use git reset <file name> to remove a file from the staging area.

1. Commit staged files to the branch.

git commit -m “<comment>”

* + - ‘-m’ flag to add comment to commit. Otherwise you have to do it in the editor.

1. Push changes to remote branch repository.

git push --set-upstream origin <branch to push to>

e.g. git push --set-upstream origin GraceShopper/CNC/returnitems\_with\_order

1. Create pull request. This is done in github.com
   * 1. Click “Compare & pull request” button in GitHub.
     2. Add comment describing changes.
     3. Click “Create pull request” button.
     4. Request review.

- Click gear by “Reviewers” on right side.

1. Review your change.

a. Click “Add Review”.

* + - Can click on ‘+’ by line of code. This allows you to add comments and suggesBons. - Either add a single comment by clicking “Add single comment” buXon. OR
    - Click start review, to package all comments in one.
    - Check “Viewed” check box to indicate you have reviewed the file. b. Click “Finish your review” buXon.
    - Leave comment.
    - Select “Comment”, “Approve”, or “Request change.”

1. Merge the commit
   * 1. Click “Merge Pull Request”
     2. Click “Confirm merge”

1. Delete the “Remote Branch” if you’re done with it:

Click the Delete Branch BuXon in GitHub

OR in the Terminal git push <remote> --delete <branch>

(e.g. git push origin -- delete FT\_TR/CNC/HomePage)

1. Delete the “Local Branch” if you’re done with it:
   * 1. Switch to main or another branch: git switch main
     2. Delete local branch: git branch -d git branch -d <branch name>
        + You cannot be on the branch you want to delete.
        + -d opton deletes the only if it has been pushed and merged.
        + - D option forces the delete regardless of push merge status.

**Resolving Merge Conflicts**

This section is used when feature branch is behind your main branch

Note: these instructons use the terminal to resolve the conflict. Git has other instructions where you can resolve the conflicts online, but they cause conflicts between the local and remote repositories. This is OK if you’re going to delete the local branch.

1. Switch to the “Main” branch.

1. Clone the repository or update your local repository with the latest changes.

1. Switch to the head branch of the pull request (feature branch).

git checkout

<

head

branch >

1. Merge the base branch into the head branch.

git merge main

5.

Fix the conflicts and commit the result.

a.

Run git stat

us to get a list of

files with confl

icts

.

g

it

statu

s

b.

O

pen

the conflict files in vs code

c. Follow steps 5 and 6 in the github’s *Resolving a merge conflict using the command line* page (link).

f. Push you changes.

d.

S

tage your changes

g

it

add

.

e.

Commit you

r

changes

.

g

it

commit

-

m

“

R

esolving

merge conflict

s

”

g

it

push

OR

git push

--

set

-

upstream origin

<

branch to push to>