GIT refresher

- **Commands**
  - git clone - Used to create a local copy of a remote repository
  - git pull - Used to download and merge the current state of the remote repository into the local repository
  - git add <files to add, or just "." for all>- Used to stage files for uploading. When pushing or committing, only files that have been added via this command. Usage usually goes (git add .) - This will stage all the files.
  - git commit -m "" - The command for a local upload and update of the repository. What this means, the repository will be updated to a new state, but only locally. For remote repository update, use the git push command. The "" denote a space where a commit message should be left. Commit messages should be meaningful. E.g., git commit –m "Fixed main navbar's dropdown not closing on outside click".
  - git push – The command to update the state of the remote repository. Git push will not work properly unless files have been staged and committed. In case of a non-default branch, the CLI will throw an error, saying an upstream branch has to be set. This can be mitigated in one of two ways: either push to the branch name with the keyword "origin" (git push origin branchName) or by setting the upstream branch ONCE (git push –u origin branchName) after which, plain old git push should work.
  - git checkout <branch-name> – The command to switch to an EXISTING branch (git checkout existingBranch). If used on a non-existant branch, it will throw an error. In order to create a branch the –b flag needs to be used (git checkout –b newBranchName)
  - git merge <branch-name> - Merges the target branch into the current branch. Essential in coordinating branches, and the only tool for resolving merge conflicts.
  - git reset – Used to unstage a specific commit. This is usually done with the flags –soft HEAD for unstaging a local update (commit) and git reset –hard HEAD for a remote update reset. This will revert the last commit.

1. **Sample Flow**
   a. Pull the latest changes from the master branch (main on GitHub) using (git pull)
   b. Checkout to a new branch (git checkout –b new-feature)
   c. Do your work and then either add only the changed files or all of them using (git add .)
   d. Local upload using (git commit –m "the work that was done")
   e. Remote upload using (git push origin new-feature)
   f. Open a merge request using the VCS' UI (simplest option)
   g. Merge changes in master branch