**Start for Github**

1. Install a github in pc and make account on web.
2. Initial commit(ctr+Enter)
3. View->commandpallete->**Add remote** and paste repository link created in github ie on first line(if file is not opened then it has to be used)
4. If not creating repository from cmd cli chose for push option github codes in github 2nd part
5. Register for username and email if asked for 1st time
6. Vscode **gitenabled** should be checked

**Commit Project to VScode**

1. Initial commit or init or first commit
2. View->commandpallete->**Add remote** and paste repository link created in github ie on first line(if file is not opened then it has to be used)
3. Copy three line codes to cmd in project vscode
4. Project repository to be created in github

**Cloning**

1. Vscode **gitenabled** should be checked
2. New window ->**git clone** in cmd pallete
3. View->cmd\_pallete->paste **clone** url link from github
4. After search in cmd\_pallete folder has to be generated for **select repository folder** anywhere and open it below, It will open automatically in vscode
5. After change initial commit and then push … project to repository, changes can be seen in github
6. For other clone projects from other repository delete **.git** from selected repository folder in pc otherwise it will not be commited after changes made then push it to github(it may be hidden)

**Delete**

1. Simple in repository account

**Fork and Pull**

1. Click fork in repository copy clone url link goto cmd\_pallete->**gitclone** or clone repository from sidebar same as above(2nd)
2. Readme file change in vscode then intial commit and push to github
3. Readme file is only for organization I have made no changes as my account not created for organization
4. Check and compare (**forked from)** link
5. Profile main page repository **New pull request** -> named the changes made and **Pull** new forked request(check pull request)
6. This is only to remind and request for the change in project if changes made or not it will be messaged in github account.

**Merge, Branch and Pull**

1. Cmd\_pallete->**Git checkout to**..-->create New Branch->type branch name(Eg:content) see left side corner down(content)
2. Push to github initial commit not to be used
3. Whatever changes made to content push to github it will show in only content branch not in main(master) with naming its folder
4. Cmd\_pallete->**Git checkout to**..-->create New Branch->content/main to check the changes
5. Cmd\_pallete->**Git merge branch**(local way) or goto github compare **pull request** and it will merge from main to content or vice-versa
6. If changes on comparison in both main and branch file conflict may occur changes made by collegues in main branch then it necessary to find out on merging so **Netlify** software it test the changes let us know sametime
7. Cmd\_pallete->**Git checkout to**..-->create New Branch->content/main
8. Now changes in content branch appear in main after merging.
9. For check delete content goto **Code->overview->branch**

**Github**

**Getting & Creating Projects**

| **Command** | **Description** |
| --- | --- |
| git init | Initialize a local Git repository |
| git clone ssh://git@github.com/[username]/[repository-name].git | Create a local copy of a remote repository |

**Basic Snapshotting**

| **Command** | **Description** |
| --- | --- |
| git status | Check status |
| git add [file-name.txt] **Example**: $git add index.html | Add a file to the staging area |
| git add -A | Add all new and changed files to the staging area |
| git commit -m "[commit message]" | Commit changes |
| git rm -r [file-name.txt] | Remove a file (or folder) |

**Branching & Merging**

| **Command** | **Description** |
| --- | --- |
| git branch | List branches (the asterisk denotes the current branch) |
| git branch -a | List all branches (local and remote) |
| git branch [branch name] | Create a new branch |
| git branch -d [branch name] | Delete a branch |
| git push origin --delete [branch name] **Example:** $git push origin master | Delete a remote branch |
| git checkout -b [branch name] **Example**: $git checkout development\_branch | Create a new branch and switch to it |
| git checkout -b [branch name] origin/[branch name] | Clone a remote branch and switch to it |
| git branch -m [old branch name] [new branch name] | Rename a local branch |
| git checkout [branch name] | Switch to a branch |
| git checkout - | Switch to the branch last checked out |
| git checkout -- [file-name.txt] | Discard changes to a file |
| git merge [branch name]  **Example :** git checkout -b dev  # Edit some files git add <file> git commit -m "commit changes"  # Merge in the dev branch git checkout master git merge dev git branch -d dev | Merge a branch into the active branch |
| git merge [source branch] [target branch] | Merge a branch into a target branch |
| git reset --merge | Cancel merging |
| git stash | Stash changes in a dirty working directory |
| git stash clear | Remove all stashed entries |

**Sharing & Updating Projects**

| **Command** | **Description** |
| --- | --- |
| git push origin [branch name] | Push a branch to your remote repository |
| git push -u origin [branch name] | Push changes to remote repository (and remember the branch) |
| git push | Push changes to remote repository (remembered branch) |
| git push origin --delete [branch name] | Delete a remote branch |
| git pull | Update local repository to the newest commit |
| git pull origin [branch name] **Example**: $git pull origin master | Pull changes from remote repository |
| git remote add origin ssh://git@github.com/[username]/[repository-name].git | Add a remote repository |
| git remote set-url origin ssh://git@github.com/[username]/[repository-name].git | Set a repository's origin branch to SSH |

**Inspection & Comparison**

| **Command** | **Description** |
| --- | --- |
| git log | View changes |
| git log –p -1 |  |
| git log –p -2 |  |
| git log --summary | View changes (detailed) |
| git log --oneline | View changes (briefly) |
| git diff [source branch] [target branch] | Preview changes before merging |

1. **git config --global user.name Abhishek**
2. **git config --global user.email** [**abhi144k@gmail.com**](mailto:abhi144k@gmail.com)
3. **git config –global --edit(**check name n password**)**
4. **git --version**
5. **git init**
6. **ls –lart**(hidden files)
7. **ls -a**
8. **code .**
9. **cmd+k** will clear all command
10. **git add** filename
11. **git commit**
12. **git commit –m “initial commit”** -m is used for message with commit and
13. **git status**(keep running)
14. **touch** contact.html(touch command creates blank page)
15. **git add –A**(add all file at once in staged)
16. **git commit –m “Added more htmls”**
17. **git checkout** hello.blade.php(some one changes our file check for last changed file by mistakes)
18. **git checkout –f**(someone changes our file check for last changed file by mistakes restores it back to original)
19. **git log**(list all history commit)
20. **git log –p -1**(list last only 1 commit)
21. **git log –p -2**(list last only 2 commit)
22. **git diff**(gittree will be matched with staging area) and after it add **git add –A** or **git add .** then no difference will show after change in file after change
23. **git diff –stagged**(check last commit with staging area)
24. **git commit –a –m”Skipped staging area and fixed”**(eg only certain number of file we r working on currently) and **git commit –a –m^C(cancel git commit –a –m command)**
25. **ls (**list all file**)**
26. create **touch** waste.html
27. **git commit –a –m “Adding Waste”**
28. **git rm waste.html** (deletes file) and **git rm --cached waste.html**(remove file from stagged area only and put in untracked file)
29. **git status –s**(summarise short status)

**note:**1st one is for staged area and 2nd is for working area M means modified in working ans stage area

1. **touch .gitignore**
2. **cd.. (for back)**
3. **for AWS git push --set-upstream origin yellow**  for git checkout –b yellow main

**---------------Branch----------------**

1. **git log**  is to check all operations in branch always run it
2. **git log** and then copy **git checkout <**b6733e3cacd09727fc7cea2afb22a7f0ca8c9e5a**>(**to enter in that author log to see changes**)**

**error:** Another git process seems to be running in this repository, e.g.

**sol:** **rm -f .git/index.lock**

if want to go back to previous one/same/back **git checkout master/main**

1. **git branch dev(**create branch dev**)**
2. **git branch(** list all branch**)**
3. **git checkout dev(**go to **dev** that branch**)**

**git checkout –b abhishek/multi(**under parent branch **dev** create and checkout directly in one command **multi** branch**)**

create/add any folder eg:multiply.html to project and then add and commit

but this multiply.html folder is in **multi** branch not in parent branch **dev**

so goto **dev** branch and merge eg: **git checkout dev** and **git merge abhishek/multi**

and to cancelmerging use **git reset –merge**

---------------**gitignore**---------------

1. Some files to be ignored in github not to be uploaded

**cd$ touch .gitignore**

1. Mention file which not to be uploaded and name mention in .gitignore

----------------**push**--------------------

1. Create repository in github
2. Login for 1st time
3. **git remote add origin** [**https://github.com/AKS144/gittest.git** (origin](https://github.com/AKS144/gittest.git%20(origin) can be named by other)
4. **git remote –v** (origin will appear for push)
5. **git branch -M main**
6. **git push -u origin main**
7. now push to branch **dev**

**git push origin**

**git checkout dev**

**git push –u origin dev**

**---------Multiple Collaboration------------**

1. Other to work on project github->settings->invite collaborator(email invitation will be accepted)

--------------------**Pull,clone**------------------

1. Make repo and **fork** any project
2. Goto others repo eg:bitfumes and **fork** it will create all code in our repo same to same
3. Now code-> <https://github.com/AKS144/phonebook.git> and **git clone https://github.com/AKS144/phonebook.git** in cmd
4. **Eg:** gitbash in www and **git clone** [**https://github.com/AKS144/phonebook.git**](https://github.com/AKS144/phonebook.git)
5. **git config –global --edit(**create name n password for other user,type it in again for organization contributor**)**
6. **To** exit **vi** editor without save a.**Escape** b.press ‘**:’** c.’**q**’and a.**Escape** b.press ‘**:’** c.**’wq’**
7. **git add** and **git commit –m “hello” and git push**
8. now we can see in commit in github code tag
9. Now **code**->**contribute**->**pull request**
10. **Check for video from** 30-34 minutes for pull request

Fork from [abhi144k@gmail.com](mailto:abhi144k@gmail.com)(1st) in [aks144k@gmail.com](mailto:aks144k@gmail.com)(2nd) then make changes and also using for **git config –global –edit user and email of 2nd github** then make changes create pull request to merge and now in 1st github save request check in **file change and review changes click** approve checkbox with comment n submit. Then **Merge** request