**Install GIT & make sure it is added into PATH.**

I installed GIT and worked on Commands, Branching, Merging, Rebase, Conflicts while merging and rebase, Pushing code to server and Pulling & Fetching code from server and finally Collaborated with our team members.

**Section 0 -Use GIT as local VCS. Steps to follow:**

1. **Create a directory ‘project\_dir’ & cd to ‘project\_dir’.**

$ mkdir project\_dir # Creating the directory

$ cd project\_dir # Changing into the directory

1. **Initialize git version database. (git init)**

$ git init #Initializing Git VCS to the directory

Initialized empty Git repository in C:/nandu/Git/project\_dir/.git/

1. **Create a new file index.html.**

$ touch index.html #Creating a new HTML file in the directory call index

1. **Check the git status. You should find index.html as untracked file.**

$ git status #getting status of our repository

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

index.html

nothing added to commit but untracked files present (use "git add" to track)

1. **Stage the index.html file.**

$ git add . / $ git index.html # adding the version of the file that you want to

save in next commit

1. **Commit index.html**

$ git commit -m "Added index.html" #saving the file in the repository

[master (root-commit) 640e054] Added index.html

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 index.html

1. **Make few changes in index.html & create a new file info.txt file.**

$ vim index.html 🡪 i 🡪 insert text/code 🡪 Esc 🡪 :wq / :x!

$ touch info.txt # changed some data in index.html and created info.txt

1. **Check git status. You should find index.html & info.txt as untracked files.**

$ git status #checking the status of the repository that shows the tasks we did on the file i.e., files that need to be committed and files that are untracked.

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: index.html

Untracked files:

(use "git add <file>..." to include in what will be committed)

info.txt

no changes added to commit (use "git add" and/or "git commit -a")

1. **Configure GIT to ignore all txt files.**

$ touch .gitignore #created the git ignore file to ignore the files that we want to untracked while tracking the remaining files.

$ vim .gitignore 🡪 i 🡪 info.txt 🡪 Esc 🡪 :wq

$ git add .gitignore #staging and committing the .git ignore file.

$ git commit -m "Added .gitignore"

[master d5bbb49] Added .gitignore

1 file changed, 1 insertion(+)

create mode 100644 .gitignore

1. **Again, check the git status. You should find only index.html as untracked file.**

$ git status #getting the status of our repo that only shows the files that are modified but not the txt files that ignored.

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: index.html

no changes added to commit (use "git add" and/or "git commit -a")

1. **State & commit index.html**

$ git add index.html #staging and committing the index file to the repository.

$ git commit -m "Changes made in index.html"

[master f5f1ef4] Changes made in index.html

1 file changed, 2 insertions(+)

1. **Log all your comments so far.**

$ git log #This log command shows all the commits we have done until now. It shows all the commits on index.html and .gitignore file.

commit f5f1ef4d83d054f08cc9b2935109cd5da94f4e2c (HEAD -> master)

Author: nandini <jampalanandini034@gmail.com>

Date: Fri Jan 7 01:19:05 2022 +0530

Changes made in index.html

commit d5bbb49ae383c59c6c2b08c56b631a424bf2feed

Author: nandini <jampalanandini034@gmail.com>

Date: Fri Jan 7 01:12:21 2022 +0530

Added .gitignore

commit 640e05456d1e45b22c71a383f5605ea0228591b7

Author: nandini <jampalanandini034@gmail.com>

Date: Fri Jan 7 00:35:53 2022 +0530

Added index.html

1. **Make some changes in index.html.**

$ vim index.html 🡪 i 🡪 make changes 🡪 :wq

$ git add index.html #Again we made some changes in the index file, Staged them and committed them.

$ git commit -m "Changes in index again"

[master efccf51] Changes in index again

1 file changed, 2 insertions(+)

1. **Revert the change made in the previous step using git command.**

$ git revert HEAD # We used git revert command to undo the last commit we did on index file.

[master 4594f26] Revert "Changes in index again"

1 file changed, 2 deletions(-)

1. **Again change index.html.**

$ vim index.html 🡪 i 🡪 make changes 🡪 :wq

# Again we changed some changes in the index.html

1. **Stage index.html**

$ git add index.html #Again we added the version of the file in that you want save in next commit.

1. **Revert the last stage.**

$ git reset #unstaging the last stage we made by using revert command in git.

Unstaged changes after reset:

M index.html

1. **Rename ‘add’ command to ‘my-add’.**

$ git config --global alias.my-add add

#By using git config command we renamed the add command into my-add using alias option.

1. **Using my\_add command Stage index.html again & commit the changes**.

$ git my-add index.html #By using renamed add command we staged the index.html and commit the changes.

$ git commit

[master 715fad3] Added some changes in index

1 file changed, 3 insertions(+)

1. **Revert the last commit.**

$ git reset --soft # by using the revert command in git we have undo the las commit.

*GIT Branching*

**Objective: Commit HTML, CSS & JavaScript assignments into GIT.**

**SECTION-1 (HTML assignments) - Steps to follow:**

1. **First take a backup of your assignments & projects. This is required because due to incorrect GIT operation you may lose your files.**

$ cd /c/nandu/clone #took the backup of my project by clone it the git hub account to another folder

$ git clone https://github.com/NJnandini/freshersbatch-Dec28.git

Cloning into 'freshersbatch-Dec28'...

remote: Enumerating objects: 60, done.

remote: Counting objects: 100% (60/60), done.

remote: Compressing objects: 100% (33/33), done.

remote: Total 60 (delta 27), reused 56 (delta 25), pack-reused 0

Receiving objects: 100% (60/60), 6.83 KiB | 635.00 KiB/s, done.

Resolving deltas: 100% (27/27), done.

1. **Create an empty directory ‘Assignments’ & cd to ‘Assignments’.**

$ mkdir Assignments #created the assignment directory

$ cd Assignments #changing into the Assignments directory

$ git init #initializing it into git repository

1. **Create a file README.txt inside ‘Assignments’ & write few lines about the contents of ‘Assignments’ folder.**

$ touch README.txt #creating README text file, adding some data into it

$ vim README.txt

$ git add README.txt #adding the version of the file into repository to save for the next commit

1. **Commit README.txt file.**

#Saving the README text file into the repository

$ git commit -m "First Html commit"

[master (root-commit) 3a6f228] First Html commit

1 file changed, 1 insertion(+)

create mode 100644 README.txt

1. **Now create a new branch ‘html-assignments’.**

$ git branch html-assignments #creating a html-assignments branch for the Assignments repository

1. **Switch to ‘html-assignments’ branch.**

$ git checkout html-assignments #switching into the html-assignments branch

Switched to branch 'html-assignments'

1. **Copy all HTML assignments inside ‘Assignments’ folder.**

$ git checkout master

Switched to branch 'master' #to create a .html files into the Assignments folder we are switching into the master repository that is Assignments directory

$ touch login.html && vim login.html #creating .html file in the master branch

$ git add . #adding all the .html files versions into the master

$ git commit -m "Added login form"

[master 8798a44] Added login form

1 file changed, 9 insertions(+) #saving the .html files into the repository

$ git checkout html-assignments

Switched to branch 'html-assignments' #again switching to html-assignments branch

$ git add –-all #copying all the files in the master into the html-assignments branch

1. **Commit HTML assignments into ‘html-assignments’ branch.**

$ git commit -m "copied files from master to html-assignments branch"

[html-assignments c3cda7a] copied files from master to html-assignments branch

2 files changed, 10 insertions(+)

create mode 100644 README.txt

create mode 100644 login.html #commiting the copied files in the html-assignments

$ git status

On branch html-assignments

nothing to commit, working tree clean #getting the status of the html-assignments branch

1. **Make minor changes into few files belonging to ‘html-assignments’ branch**.

$ vim login.html #making some changes in the files of the html-assignments branch that we are copied

$ git status

On branch html-assignments

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: login.html #getting the status of the html-assignments branch

$ git add . #adding the versions of files to the repo

1. **Commit those changed files.**

$ git commit -m "Changes in login are commited"

[html-assignments 9254378] Changes in login are commited

1 file changed, 1 insertion(+) #saving the changes we made in the files

1. **Switch to master branch.**

$ git checkout master

Switched to branch 'master' #switching into the master branch

1. **Make minor changes into README.txt file & commit those changes into master.**

$ vim README.txt #making some changes in the README text file in the master branch and saving it

$ git commit -m "changes in README are commited"

[master a38bdee] changes in README are commited

2 files changed, 1 insertion(+)

create mode 100644 index.html

1. **Again, switch to ‘html-assignments’ branch.**

$ git checkout html-assignments

Switched to branch 'html-assignments' #switching to the html-assignments branch

1. **Make minor changes into few files belonging to ‘html-assignments’ branch.**

$ vim login.html #again making some changes in the files of html-assignments branch and staging it

$ git status

On branch html-assignments

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: login.html

$ git add .

$ git status

On branch html-assignments

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: login.html

1. **Commit those changes.**

$ git commit -m "changes made in branch login are commited"

[html-assignments 886db5d] changes made in branch login are commited

1 file changed, 1 insertion(+)

#saving those changes in the repository

1. **Switch to master.**

$ git checkout master

Switched to branch 'master' #switching to the master branch

1. **Merge ‘html-assignments’ branch into master. Confirm all html assignments are shown in master.**

$ git branch #checking how many branches in the repository

html-assignments

\* master

$ git diff master..html-assignments #checking the differences in the master branch and html-assignments branch

**diff --git a/README.txt b/README.txt**

**index 0f80c3b..cd70653 100644**

**--- a/README.txt**

**+++ b/README.txt**

@@ -1,2 +1 @@

This folder has the assignments of the HTML files

-This folder also has the index of html topic

**diff --git a/login.html b/login.html**

**new file mode 100644**

**index 0000000..69b3564**

**--- /dev/null**

**+++ b/login.html**

@@ -0,0 +1,11 @@

+<html>

+ <head>

+ <h1> LOGIN FORM </h1>

+ <h2> WELCOME </h2>

+ </head>

+ <body>

+ <b> <br>This is the login form of html</b>

+ <i> <br> Hello!!! Welcome to login form...</i>

+ </body>

+</html>

$ git merge html-assignments #merging both master and html-assignments branches

Merge made by the 'ort' strategy.

login.html | 11 +++++++++++

1 file changed, 11 insertions(+)

create mode 100644 login.html

1. **Finally delete the ‘html-assignments’ branch.**

$ git branch --merged #checking which branches are merged

html-assignments

\* master

$ git branch -d html-assignments #finally deleting the html-assignments branch

Deleted branch html-assignments (was a4f8a1f).

$ git branch --merged #checking which branches are there

\* master

**SECTION-2 - (CSS assignments) Steps to follow:**

1. **Create a new branch ‘css-assignments’.**

$ git branch css-assignments #created a new branch called css-assignments

$ git branch #checking which branches are in the repo

css-assignments

\* master

1. **Switch to ‘css-assignments’ branch.**

$ git checkout css-assignments #switching to the css-assignments branch

Switched to branch 'css-assignments'

1. **Copy all CSS assignments inside ‘Assignments’ folder.**

$ git checkout master

Switched to branch 'master' #to create some files in the master branch we switched to it

$ touch index.css #created a css file in the master

$ vim index.css #added some data into the css file

$ git add . #staged the css file

$ git commit -m "Added CSS file to the repo" #saved that css file into the repo

[master 30c3a9f] Added CSS file to the repo

1 file changed, 40 insertions(+)

create mode 100644 index.css

$ git checkout css-assignments #switching to css-assignments branch

Switched to branch 'css-assignments'

$ git checkout master "index.css" #copying the files in the master branch into css-assignments branch

Updated 1 path from 5f427d6

$ git status #getting the status of the css-assignnments branch it showed the files we copied

On branch css-assignments

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: index.css

1. **Commit CSS assignments into ‘css-assignments’ branch.**

$ git commit -m "copied files from master to css-assignments"

[css-assignments bb3de22] copied files from master to css-assignments

1 files changed, 58 insertions(+)

create mode 100644 index.css #saving those files into css-assignments branch

$ git status

On branch css-assignments

nothing to commit, working tree clean

1. **Make minor changes into README.txt file on line 1 belonging to ‘css-assignments’ branch.**

$ git checkout master "README.txt" #copying README text file into the css-assignments branch

Updated 0 paths from 6253bb7

$ vim "README.txt" #editing the data in README file of css-assignments branch

$ git status #getting status of the CSS-assignments branch

On branch css-assignments

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: README.txt

$ git add . #staging those changes in the file

1. **Commit those changed files.**

$ git commit -m "Changed some data in README.txt"

[css-assignments bcefd43] Changed some data in README.txt

1 file changed, 1 insertion(+) #saving the changes in the file

1. **Switch to master branch.**

$ git checkout master

Switched to branch 'master' #switching to master branch

1. **Make minor changes into README.txt file on line 3 & commit those changes into master.**

$ vim README.txt #editing the data in the README file of the master branch

$ git commit -a #saving those changes

[master 22604fb] changes made in README has commited

1 file changed, 1 insertion(+)

1. **Again, switch to ‘css-assignments’ branch.**

$ git checkout css-assignments

Switched to branch 'css-assignments' #switching to the css-assignments branch

1. **Make minor changes into few files belonging to ‘css-assignments’ branch.**

$ vim index.css #editing some code in the css file of css-assignments branch

$ git add . #staging those changes

$ git status #getting the status of the branch

On branch css-assignments

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: index.css

1. **Commit those changes.**

$ git commit -a #saving those changes in the css file

[css-assignments 37d32d9] "Changes in index are commited"

1 file changed, 1 insertion(+), 1 deletion(-)

$ git status #getting the status of the branch

On branch css-assignments

nothing to commit, working tree clean

1. **Switch to master.**

$ git checkout master #switching to the master branch

Switched to branch 'master'

$ git branch #checking which branches are there in the repo

css-assignments

\* master

1. **Merge ‘css-assignments’ branch into master. Confirm all css assignments are shown in master.**

$ git merge css-assignments #merging the both css-assignments branch into the master but got conflicts while merging

Auto-merging README.txt

CONFLICT (content): Merge conflict in README.txt

Auto-merging index.css

CONFLICT (add/add): Merge conflict in index.css

Automatic merge failed; fix conflicts and then commit the result.

$ git status #getting the status of the master that showed that css file and README file are not commited

On branch master

You have unmerged paths.

(fix conflicts and run "git commit")

(use "git merge --abort" to abort the merge)

Unmerged paths:

(use "git add <file>..." to mark resolution)

both modified: README.txt

both added: index.css

$ git add . #staged those files

$ git commit -a #saved those files that automatically merged both branches

$ git merge css-assignments #this showed that our merge is up to date

Already up to date.

$ git branch –merged #checking which branches are merged

css-assignments

\* master

merged

1. **Finally delete the ‘css-assignments’ branch.**

$ git branch -d css-assignments #deleting the css-assignments branch after merging

Deleted branch css-assignments (was 37d32d9).

$ git branch –merged #checking which branches are merged

\* master

merged

**SECTION-3 - (JavaScript assignments) Steps to follow:**

1. **Create a new branch ‘js-assignments’.**

$ git branch js-assignments #creating the new branch called js-assignments in the Assignments/master branch

1. **Switch to ‘js-assignments’ branch.**

$ git checkout js-assignments #switching to the js-assignments branch

Switched to branch 'js-assignments'

1. **Copy all JavaScript assignments inside ‘Assignments’ folder.**

$ git checkout master #to create some files in the master branch we switched to it

Switched to branch 'master'

$ touch sample.js && vim sample.js #created new js file

$ git checkout js-assignments #switching to the js-assignments branch

Switched to branch 'js-assignments'

$ git checkout master . #copying the file in the master to js-assignments

Updated 0 paths from 29592fe

$ git add –all #adding all files into the js-assignmnets

1. **Commit JavaScript assignments into ‘js-assignments’ branch.**

$ git commit -a #saving those files

[js-assignments ac42c81] "Copied files from master to js-assignments"

1 file changed, 11 insertions(+)

create mode 100644 sample.js

1. **Make minor changes into README.txt file on line 1 belonging to ‘js-assignments’ branch.**

$ vim README.txt #editing the data in the README file of the js-assignments

$ git status #getting the status of the js-assignmnets

On branch js-assignments

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: README.txt

no changes added to commit (use "git add" and/or "git commit -a")

1. **Commit those changed files.**

$ git commit -a #saving those changes

[js-assignments 679669f] "Changes made in README.txt are commited"

1 file changed, 1 insertion(+), 4 deletions(-)

1. **Switch to master branch.**

$ git checkout master #switching to the master branch

Switched to branch 'master'

1. **Make minor changes into README.txt file on line 1 & commit those changes into master.**

$ vim README.txt #editing the data in the README file of master branch

$ git commit -a #saving those changes

[master 8d22b6f] "changes made in txt files are commited"

1 file changed, 1 insertion(+), 4 deletions(-)

1. **Again, switch to ‘js-assignments’ branch.**

$ git checkout js-assignments

Switched to branch 'js-assignments' #switching to the js-assignments branch

1. **Make minor changes into few files belonging to ‘js-assignments’ branch.**

$ vim sample.js #editing some code in the js file of js-assignments branch

$ git status #getiing the status of the js-assignments shows the files we modified

On branch js-assignments

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: sample.js

no changes added to commit (use "git add" and/or "git commit -a")

$ git add . #staging those files

1. **Commit those changes.**

$ git commit -a #saving those changes made in the js files

[js-assignments c6f34a3] created head tag in sample.js

1 file changed, 7 insertions(+)

1. **Switch to master.**

$ git checkout master #switching to the master branch

Switched to branch 'master'

1. **Merge ‘js-assignments’ branch into master. Confirm all JavaScript assignments are shown in master.**

$ git merge js-assignments #merging the js-assignments branch into master created a conflict

Auto-merging README.txt

CONFLICT (content): Merge conflict in README.txt

Automatic merge failed; fix conflicts and then commit the result.

$ git status #getting the status of the master showed the changes to be committed

On branch master

You have unmerged paths.

(fix conflicts and run "git commit")

(use "git merge --abort" to abort the merge)

Changes to be committed:

new file: sample.js

Unmerged paths:

(use "git add <file>..." to mark resolution)

both modified: README.txt

$ git add . #staging those changes

$ git commit -a #saving those changes

[master 724d7ff] Merge branch 'js-assignments'

$ git branch #checking which branches are merged

js-assignments

\* master

merged

1. **Finally delete the ‘js-assignments’ branch.**

$ git branch -d js-assignments #deleting the js-assignments branch

Deleted branch js-assignments (was c6f34a3).

$ git branch #checking the branches in the repo

\* master

merged

*GIT Remoting*

**Objective: Pushing source code into GITHUB & collaborate team members.**

**SECTION-3 (Pushing assignments to remote repository) - Steps to follow:**

1. **Create a GitHub account if you do not have already.**

Done

1. **Login on into GitHub account.**

Done

1. **Create new public repository ‘freshersbatch-oct16’.**

I created a new repository called “Freshersbatch-Dec28’ and pushed the Assignments directory into the remote repository. After pushing I made some changes in the README.txt file.

1. **Commit & push any sample file to this repository under ‘Assignments’ directory.**

$ touch sample.txt && vim sample.txt #Created a sample file

$ git add sample.txt && git commit -a #stage and committed the file

warning: LF will be replaced by CRLF in sample.txt.

The file will have its original line endings in your working directory

[master 29c85b1] SAmple file to push in git

1 file changed, 1 insertion(+)

create mode 100644 sample.txt

$ git remote -v #checking the origin

origin https://github.com/NJnandini/freshersbatch-Dec28.git (fetch)

origin https://github.com/NJnandini/freshersbatch-Dec28.git (push)

$ git push origin master #pushing the sample file into GitHub but occurred with some error cause of the changes we made in README file

To https://github.com/NJnandini/freshersbatch-Dec28.git

! [rejected] master -> master (fetch first)

error: failed to push some refs to 'https://github.com/NJnandini/freshersbtch-Dec28.git'

hint: Updates were rejected because the remote contains work that you do

hint: not have locally. This is usually caused by another repository pushig

hint: to the same ref. You may want to first integrate the remote changes

hint: (e.g., 'git pull ...') before pushing again.

hint: See the 'Note about fast-forwards' in 'git push --help' for details.

$ git pull origin master #pulling the code from GitHub account into the local repo

From https://github.com/NJnandini/freshersbatch-Dec28

\* branch master -> FETCH\_HEAD

Merge made by the 'ort' strategy.

README.txt | 3 ---

1. file changed, 3 deletions(-)

$ git commit -a #commiting those changes

On branch master

Your branch is ahead of 'origin/master' by 2 commits.

(use "git push" to publish your local commits)

nothing to commit, working tree clean

$ git push origin master #succesfully pushed the code into the server that is GitHub repo

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (5/5), 609 bytes | 609.00 KiB/s, done.

Total 5 (delta 2), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

To https://github.com/NJnandini/freshersbatch-Dec28.git

7e09727..763dd7e master -> master

**SECTION-4 (Pushing source code to remote repository using Eclipse GIT plugin) - Steps to follow:**

1. **One developer from project team will create eclipse projects ‘SampleProj’ & add sample source code files. Then commit all files through eclipse GIT plugin.**

One of our team members downloaded eclipse and created a project ‘SAmpleProj’ and added source code files named as sp1.html, sp2.css and sp3.js.She committed the code

In the eclipse and pushed it in her GitHub account.

1. **Collaborate other team members with your GitHub account so that they can also modify the committed files.**

She collaborated with us by adding our GitHub accounts in the project. I was able to make changes in the files >> Adding and Deleting lines >> Committing them.

1. **Other developers from same team will checkout all files from remote repository. This might get conflicts since certain files fail to merge. In such case, merge it manually.**
2. **Commit & push the ‘SampleProj’ project.**

After committing I was able to push the changes that I made by clicking on the “commit” button. Other collaborators were able to see the changes that I made and also I was able to see the changes that they made after committing and pushing the files.