Git Assignment

* How do you initialize a new Git repository named "project1" on your local machine?

Consultant@LAPTOP-1L97QQ2-GBOKIRIKA MINGW64 ~

$ cd Documents

Consultant@LAPTOP-1L97QQ2-GBOKIRIKA MINGW64 ~/Documents

$ ls

'GIT AND GITHUB.txt' 'My Music'@ 'git github.txt'

'Hadoop - master slave architecture.txt' 'My Pictures'@ mysql\_prcatice.txt

MYSQL-.txt 'My Videos'@ test\_key.pem

MYSQL.sql desktop.ini

Consultant@LAPTOP-1L97QQ2-GBOKIRIKA MINGW64 ~/Documents

$ git init project1

Initialized empty Git repository in C:/Users/Consultant/Documents/project1/.git/

Consultant@LAPTOP-1L97QQ2-GBOKIRIKA MINGW64 ~/Documents

$ cd project1

* How do you add a new file named "index.html" to the "project1" repository?

Consultant@LAPTOP-1L97QQ2-GBOKIRIKA MINGW64 ~/Documents/project1 (master)

$ git add index.html

* How do you stage the changes made to "index.html" for committing?

Consultant@LAPTOP-1L97QQ2-GBOKIRIKA MINGW64 ~/Documents/project1 (master)

$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: index.html

* How do you commit the changes made to "index.html" with a commit message of "Add index.html"?

Consultant@LAPTOP-1L97QQ2-GBOKIRIKA MINGW64 ~/Documents/project1 (master)

$ git commit -m "Add index.html"

[master (root-commit) ea5309d] Add index.html

1 file changed, 1 insertion(+)

create mode 100644 index.html

* How do you view the commit history of the "project1" repository?

Consultant@LAPTOP-1L97QQ2-GBOKIRIKA MINGW64 ~/Documents/project1 (master)

$ git log

commit ea5309de65705c4584609c26ea0541ce03cdac0f (HEAD -> master)

Author: unknown <ranaabhilash85@gmail.com>

Date: Wed Jul 5 14:21:41 2023 -0400

Add index.html

* How do you create a new branch named "feature-branch" in the "project1" repository?

Consultant@LAPTOP-1L97QQ2-GBOKIRIKA MINGW64 ~/Documents/project1 (master)

$ git branch feature-branch

* How do you switch to the "feature-branch" in the "project1" repository?

Consultant@LAPTOP-1L97QQ2-GBOKIRIKA MINGW64 ~/Documents/project1 (master)

$ git checkout feature-branch

Switched to branch 'feature-branch'

* How do you make changes to "index.html" in the "feature-branch"?

)

$ git commit -m "Modify index.html in feature-branch"

* How do you stage and commit the changes made in the "feature-branch" with a commit message of "Update index.html in feature branch"?

Consultant@LAPTOP-1L97QQ2-GBOKIRIKA MINGW64 ~/Documents/project1 (feature-branch)

$ git commit -m "Update index.html in feature branch"

* How do you switch back to the main branch in the "project1" repository?

git checkout main / git checkout master

* How do you merge the changes from the "feature-branch" into the main branch?

Consultant@LAPTOP-1L97QQ2-GBOKIRIKA MINGW64 ~/Documents/project1 (main)

$ git commit -m "Merge changes from feature-branch into main"

* How do you resolve any merge conflicts that occur during the merge process?

When encountering merge conflicts during the Git merge process, follow these steps to resolve them:

Identify conflicted files: Git will provide a message or use git status to identify the files with conflicts.

Open conflicted files: Use a text editor to open the conflicted files. Conflict markers like "<<<<<<< HEAD", "=======", and ">>>>>>> branch-name" will highlight the conflicting sections.

Analyze and resolve conflicts: Review the conflicting sections and decide on the desired outcome. Edit the file, removing conflict markers, and make the necessary changes to resolve conflicts.

Save resolved files: Save the modified file after resolving conflicts.

Mark resolved files: Use git add to mark each resolved file as resolved, one by one.

Repeat for multiple files: If there are multiple files with conflicts, repeat steps 2-5 for each file until all conflicts are resolved and marked as resolved using git add.

Complete the merge commit: Run git commit to open the commit message editor. The editor contains merge information by default. Modify the message if needed, save, and close the editor to complete the merge commit.

Review and test: Thoroughly review and test the changes after conflict resolution to ensure the desired outcome and functionality.

* How do you view the changes introduced by the merge commit?

git show <merge-commit-hash>

* How do you create a new tag named "v1.0" for the current commit in the "project1" repository?

git tag v1.0

* How do you push the "project1" repository to a remote repository named "origin"?

Consultant@LAPTOP-1L97QQ2-GBOKIRIKA MINGW64 ~/Documents/project1 (main)

$ git remote -v

Consultant@LAPTOP-1L97QQ2-GBOKIRIKA MINGW64 ~/Documents/project1 (main)

$ git push origin

fatal: The current branch main has no upstream branch.

To push the current branch and set the remote as upstream, use

git push --set-upstream origin main

To have this happen automatically for branches without a tracking

upstream, see 'push.autoSetupRemote' in 'git help config'.

Consultant@LAPTOP-1L97QQ2-GBOKIRIKA MINGW64 ~/Documents/project1 (main)

$ git push --set-upstream origin main

* How do you clone the "project1" repository from a remote repository to another machine?

Consultant@LAPTOP-1L97QQ2-GBOKIRIKA MINGW64 ~/Documents/project1 (main)

$ git clone <https://github.com/arana011/project1.git>

Cloning into 'project1'...

warning: You appear to have cloned an empty repository.

* How do you fetch the latest changes from the remote repository to your local "project1" repository?

git fetch origin

* How do you pull the latest changes from the remote repository into your current branch in the "project1" repository?

git pull origin main

* How do you create a new branch named "bug-fix" in the "project1" repository based on a specific commit?

Consultant@LAPTOP-1L97QQ2-GBOKIRIKA MINGW64 ~/Documents/project1 (main)

$ git branch bug-fix main

Consultant@LAPTOP-1L97QQ2-GBOKIRIKA MINGW64 ~/Documents/project1 (main)

$ git checkout bug-fix

Switched to branch 'bug-fix'

* How do you revert the last commit made in the "project1" repository?

git revert HEAD