EXPERIMENT 1 & 2

Git installation and versioning

Aim: To install git (local repository) and synchronize with github (remote repository) and perform version controlling.

Steps for installation and version control:
git config
Usage: git config –global user.name “[name]”
Usage: git config –global user.email “[email address]”
This command sets the author name and email address respectively to be used with your commits.

git init
Usage: git init [repository name]

This command is used to start a new repository.

git clone
Usage: git clone [url]
This command is used to obtain a repository from an existing URL.

git add
Usage: git add [file]
This command adds a file to the staging area.

Usage: git add *
This command adds one or more to the staging area.

git commit
Usage: git commit -m “[Type in the commit message]”
This command records or snapshots the file permanently in the version history.

Usage: git commit -a
This command commits any files you’ve added with the git add command and also commits any files you’ve changed since then.

git diff

Usage: git diff
This command shows the file differences which are not yet staged.

Usage: git diff –staged
This command shows the differences between the files in the staging area and the latest version is present.

Usage: git diff [first branch] [second branch]
This command shows the differences between the two branches mentioned.

git reset
Usage: git reset [file]

This command unstages the file, but it preserves the file contents.

Usage: git reset [commit]
This command undoes all the commits after the specified commit and preserves the changes locally.

Usage: git reset –hard [commit] This command discards all history and goes back to the specified commit.

git status
Usage: git status
This command lists all the files that have to be committed.

git rm
Usage: git rm [file]
This command deletes the file from your working directory and stages the deletion.

git log
Usage: git log
This command is used to list the version history for the current branch.

Usage: git log –follow[file]
This command lists version history for a file, including the renaming of files also.

git show
Usage: git show [commit]
This command shows the metadata and content changes of the specified commit.

git tag
Usage: git tag [commitID]
This command is used to give tags to the specific commit.

git branch
Usage: git branch
This command lists all the local branches in the current repository.

Usage: git branch [branch name]

This command creates a new branch.

Usage: git branch -d [branch name]
This command deletes the feature branch.

git checkout
Usage: git checkout [branch name]
This command is used to switch from one branch to another.

Usage: git checkout -b [branch name]
This command creates a new branch and also switches to it.

git merge
Usage: git merge [branch name]
This command merges the specified branch’s history into the current branch.

git remote
Usage: git remote add [variable name] [Remote Server Link]
This command is used to connect your local repository to the remote server.

git push
Usage: git push [variable name] master
This command sends the committed changes of master branch to your

remote repository.

Usage: git push [variable name] [branch]
This command sends the branch commits to your remote repository.

Usage: git push –all [variable name]
This command pushes all branches to your remote repository.

Usage: git push [variable name] :[branch name]
This command deletes a branch on your remote repository.

git pull
Usage: git pull [Repository Link]
This command fetches and merges changes on the remote server to your working directory.

git stash
Usage: git stash save
This command temporarily stores all the modified tracked files.

Usage: git stash pop
This command restores the most recently stashed files.

Usage: git stash list
This command lists all stashed changesets.

Usage: git stash drop
This command discards the most recently stash009564ed changeset.

Get Token
1. Log into GitHub.
2. Click on your name / Avatar in the upper right corner and select Settings.
3. On the left, click Developer settings.
4. Select Personal access tokens and click Generate new token.
5. Give the token a description/name and select the scope of the token. ...
6. Click Generate token.
7. This configures the computer to remember the complex token by enable caching of the credentials.

```
git config --global credential.helper cache
```

8. If needed, you can later clear the token from the local computer by running

```
  140  git config --global user.name "dushyant"
  141  git config --global user.email "dushyant@gmail.com"
  142  git config --list
  143  cd Desktop
  144  git config --list
  145  mkdir dev
  146  cd dev
  147  git clone https://github.com/Dushyantbhagwat/Devops.git
  148  cd Devops
  149  ls
  150  touch hello.txt
  151  nano hello.txt
  152  gedit hello.txt
  153  git status
  154  git add hello.txt
  155  git status
  156  git commit -m "new file commited"
  157  git push origin main
  158  git remote setup url https://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/Dushyantbhagwat/Devops
  159  git remote set url https://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/Dushyantbhagwat/Devops
  160  git remote set-url origin
https://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/Dushyantbhagwat/Devops
  161  git push origin main
  162  git remote set-url origin
https://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/Dushyantbhagwat/Devops
  163  git push origin main
  164  git remote set-url origin
https://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@github.com/Dushyantbhagwat/Devops
  165  git push origin main
  166  git config --global unset user.name
  167  git config --unset user.name
  168  git config --unset user.email
  169  git config --list
  170  git config --unset user.name
  171  git config --unset user.email
  172  git config --list
  173  git config --global user.name "Dushyantbhagwat"
  174  git config --global user.email "dushyantdbhagwat@gmail.com"
  175  git config --list
  176  touch new2.txt
  177  gedit new2.txt
  178  git add new2.txt
  179  git commit -m "new commit"
  180  git push origin main
  181  ls
  182  git rm hello.txt
  183  ls
  184  git status
  185  git commit -1
  186  git commit -a
  187  git status
  188  git push origin main
  189  git log
```