Assignment No 2 (A)

Aim: Create version control account on GitHub and using Git commands to create repository and push your code to GitHub

Performance (3)	Understanding (1)	Regularity (1)	Total (5)	Sign of Staff

Assignment 2 a

Title: Create version control account on GitHub

Problem Statement:

a. Create version control account on GitHub and using Git commands to create repository and push your code to GitHub.

Objective: Create repositories on GitHub.

Theory:

Create version control account on GitHub and using Git commands to create repository and push your code to GitHub

Git is a version control system. It helps you keep track of code changes and it is used to collaborate on code.

Git commands: git –version

```
git version 2.30.2.windows.1
```

Git and GitHub are different things.

Git is a popular version control system. It was created by Linus Torvalds in 2005, and has been maintained by Junio Hamano since then.

It is used for:

- Tracking code changes
- Tracking who made changes
- Coding collaboration

What does Git do?

- Manage projects with **Repositories**
- **Clone** a project to work on a local copy
- Control and track changes with **Staging** and **Committing**
- **Branch** and **Merge** to allow for work on different parts and versions of a project
- **Pull** the latest version of the project to a local copy
- **Push** local updates to the main project

Working with Git

- Initialize Git on a folder, making it a **Repository**
- Git now creates a hidden folder to keep track of changes in that folder
- When a file is changed, added or deleted, it is considered **modified**
- You select the modified files you want to **Stage**
- The **Staged** files are **Committed**, which prompts Git to store a **permanent** snapshot of the files
- Git allows you to see the full history of every commit.

- You can revert back to any previous commit.
- Git does not store a separate copy of every file in every commit, but keeps track of changes made in each commit!

What is GitHub?

- Git is not the same as GitHub.
- GitHub makes tools that use Git.
- GitHub is the largest host of source code in the world, and has been owned by Microsoft since 2018.

Steps to create version control account on GitHub and using Git commands to create repository and push your code to GitHub.

- Sign up for a GitHub account
- Install Git
- Create a new repository
- Initialize a Git repository
- Add files to the repository
- Commit changes
- Link your local repository to your GitHub repository
- Push changes to GitHub
- Using these steps, you can easily set up version control for your projects, keep track of changes to your code, and collaborate with others.

Steps to Push and PULL version control repository to GitHub

Step No	Command	Description
1	Git Installation	Download Git from the website: https://www.git-scm.com/
2	Command line >git –version	If Git is installed, it should show something like git version X.Y
3	git config --global user.name "w3schools-test" git config --global user.email "test@w3schools.com"	Configure Git Change the user name and e-mail address to your own
4	mkdir myproject cd myproject	*Creating Git Folder*
5	git init	*Initialize Git* *Initialized empty Git repository in /Users/user/myproject/.git/*

6	git status	*To check the status*
7	git add index.html	*Add file to staging environment*
8	git add --all	*add all files in the current directory to the Staging Environment:*
9	git commit -m "First release of Hello World!"	*The committ command performs a commit, and the -m "message" adds a message.*
10	git commit -a -m "Updated index.html with a new line"	*Skips staging environment*
11	git log	*To view the history of commits for a repository, you can use the log command*
12	git *command* -help	*See all the available options for the specific command*
13	git help --all	*See all possible commands*
14	git commit -help	*See help for specific command*
15	git branch hello-world-images	*a branch is a new/separate version of the main repository. This command creates a new branch hello-world-images*
16	git checkout hello-world-images	*checkout is the command used to check out/ move to a branch*
17	git checkout master	*Used to switch between branches*
18	https://github.com/	*Create a new account on github*
19		*Create a Repository on GitHub*
20	git remote add origin https://github.com/w3schools-test/hello-world.git	*Push Local Repository to GitHub*
21	git push --set-upstream origin master	*push master branch to the origin url,*
22		*go back into GitHub and see that the repository has been updated:*

23	git fetch origin	*fetch gets all the change history of a tracked branch/repo*
24	git merge origin/master	*merge combines the current branch, with a specified branch.*
25	git pull origin	*pull is a combination of fetch and merge* *It is used to pull all changes from a remote repository into the branch you are working on.*

Detailed Steps:

- Sign up for a GitHub account: Go to the GitHub website and sign up for an account by providing your email, username, and password. You can use either the free version or the paid version of GitHub, depending on your needs.

- Install Git: Download and install Git on your local machine if you haven't already. Git is a free and open−source version control system that allows you to manage your code locally and push it to GitHub.

- Create a new repository: Log in to your GitHub account and click on the "New" button to create a new repository. Give it a name and description, and choose whether it will be public or private.

- Initialize a Git repository: In your local project directory, run the command "git init" to initialize a Git repository. This will create a hidden .git directory in your project directory, which will be used to track changes to your code.

- Add files to the repository: Use the command "git add ." to add all the files in your local project directory to the Git repository. Alternatively, you can add individual files by using the command "git add filename".

- Commit changes: Use the command "git commit −m 'commit message'" to commit your changes to the repository. The commit message should be a brief description of the changes you made.

- Link your local repository to your GitHub repository: Run the command "git remote add origin https://github.com/your−username/repository−name.git" to link your local Git repository to the GitHub repository you created in step 3.

- Push changes to GitHub: Use the command "git push −u origin master" to push the changes in your local repository to the GitHub repository. This will upload your code to GitHub and make it available for others to see.

Once you've completed these steps, you can continue to use Git and GitHub to manage your code changes and collaborate with others.

Conclusion:

Git and GitHub provide a powerful and flexible platform for version control and collaborative software development. By learning these basic steps, you can start to use Git and GitHub for your own projects and join a global community of developers who are using these tools to build innovativesoftware solutions.