Mohd Farman (MST03-0075)

Assignment 6

Q.1 Describe Git and GitHub

Git:

- Git is a free and open-source distributed version control system designed to handle everything from small to huge projects with speed and efficiency.
- It is easy to learn and has a tiny footprint with lightning-fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and Clear Case with features like cheap local branching, convenient staging areas, and multiple workflows.
- Features of Git:
  - Tracks history
  - Free and open source
  - Supports non-linear development
  - Creates backups
  - Scalable
  - Supports collaboration
  - Branching is easier
  - Distributed development

GitHub:

- GitHub is an online software development platform. It's used for storing, tracking, and collaborating on software projects.
- It makes it easy for developers to share code files and collaborate with fellow developers on open-source projects. GitHub also serves as a social networking site where developers can openly network, collaborate, and pitch their work.
- Features of GitHub:
  - Collaboration
  - Integrated issue and bug tracking
  - Graphical representation of branches
  - Git repositories hosting
  - Project management
  - Team management
  - Code hosting
  - Track and assign tasks
  - Conversations
  - Wikisc

Q.2 Difference between Git and GitHub

Git	GitHub
• It's a software	• It's a service
• It is installed locally on the system	• It hosted on the web
• It is the command-line tool	• It provides a graphic interface
• It is open source licensed	• Includes a free tier and pay-for-user tier
• Maintain by Linux	• Maintain by Microsoft
• It's a tool to manage different versions of edit, made to file in a Git repository	• It's a space to upload a copy of the Git repository
• It provides functions like system version control system source code management	• It provides functions like VCS, and source code management as well as adds a few of its own features.

Q.3 Important commands of Git

1. Git config

This command allows you to specify the username and email address that will be used with your commits.

```
# sets up Git with your name

Git config --global user.name "<Your-Full-Name>"

# sets up Git with your email

Git config --global user.email "<your-email-address>"
```

2. Git init

Use the Git init command to create a new Git repository.

```
$ Git init
```

3. Git clone

The Git clone command takes a path (typically a URL) to the Git repository you want to clone.

```
$ Git clone https://Github.com/<repo-url>
```

4. Git status

It will inform us of Git's thoughts and the state of our repository as seen by Git

```
$ Git status
```

5. Git add

To move files from the Working Directory to the Staging Index, use the Git add command.

```
$ Git add <file1> <file2> … <fileN>
```

6. Git commit

This command saves a log message along with the commit ID of the modifications made to the Git repository. The modifications are saved in your local repository with a Git commit.

```
$ Git commit –m "<Type your commit message here>"
```

7. Git push

This command pushes the contents of your local repository to the remote repository you've added. This pushes your master branch's commits to the newly added remote repository

```
$ Git push
```

8. Git branch

Add a new branch to an existing branch, list all existing branches, and delete a branch with a Git branch. This command is used to perform operations on a branch that has been specified

```
$ Git branch <branch-name>
```

9. Git checkout

We can use the Git checkout command to switch to an existing branch or to create and switch to a new one.

```
$ Git checkout <branch-name>
```

10. Git merge

The command Git merge joins your branch to the parent branch. Depending on your process, the parent branch can be either a development or a master branch.

```
$ Git merge <name-of-branch-to-merge-in>
```

11. Git pull

Git pull pulls the most recent changes from the remote server into the local repository, ensuring you have the most up-to-date information from your coworkers.

```
$ Git pull
```

12. Git log

The Git log command is used to show all of a repository's commits. This command displays a log of all commits made to the current branch so far.

```
$ Git log
```