NAME:P.S.V MOURYA;
ROLL NUMBER:122CS0563;

Q1)Explain very briefly the difference between Git and GitHub.

ANS:

1.Git:

- It is very popular version control system that is used to manage projects. A version control system is software that is used to track changes made to our project and it stores these modifications as different versions of our project.
- Git manages the different versions of our project by maintaining a series of snapshots of the project. Whenever we want to commit or store a change permanently a new snapshot of our current version is taken and stored by Git. Any project managed using Git is called a Git Repository.
- A major advantage of using Git, or any other version control system, is that we can easily compare the current version of our project with previous versions and rectify errors. We can also roll back to the previous versions.
- It is a distributed version control system so that anyone who clones our repository will have access to the entire commit history.
- Git also provides excellent branching support. A branch is just a pointer to a commit. They are used to create an independent working environment for developers where they can experiment with different things without worrying about affecting the project.

2.Github:

- GitHub is a Git Repository hosting service that is owned by the tech giant Microsoft. We can upload our local Git repository to GitHub and access it anywhere.
- GitHub is very useful for collaborative purposes. A person can share his code with other developers by uploading the Git

Repository to GitHub. We can also authorize other developers to make changes to our remote repository. GitHub can also be used as a backup for our repository.

- GitHub provides all the features of Git and it is also more user-friendly and intuitive to use.
- There are other Git hosting services like Bitbucket or GitLab, but GitHub is the most popular and the most widely used one.

GIT	GITHUB
Git is a version control system.	GitHub is a Git repository hosting service.
Git is installed on our local machine and we don't need internet access to use Git.	GitHub is completely cloud-based and an internet connection is needed to use GitHub.
Git is maintained by the Linux Foundation.	GitHub is maintained by Microsoft.
Git is software and used as a command-line tool.	GitHub provides a service and provides a Graphical Interface to its users.

Q2) Explain very briefly any 5 basic git commands.

Ans}

5 basic git commands:

1.git int:

The git init command creates a new Git repository. It can be used to convert an existing, unversioned project to a Git repository or initialize a new, empty repository. Most other Git commands are not available outside of an initialized repository, so this is usually the first command you'll run in a new project.

```
git init <directory>
```

Create an empty Git repository in the specified directory. Running this command will create a new subdirectory called containing nothing but the .git subdirectory.

2.git add:

The git add command adds a change in the working directory to the staging area. It tells Git that you want to include updates to a particular file in the next commit. However, git add doesn't really affect the repository in any significant way—changes are not actually recorded until you run git commit.

```
git add <file>
```

Stage all changes in <file> for the next commit.

Git add is the first command in a chain of operations that directs Git to "save" a snapshot of the current project state, into the commit history. When used on its own, git add will promote pending changes from the working directory to the staging area.

3.git commit:

- The commit command makes sure that the changes are saved to the local repository

The git commit command captures a snapshot of the project's currently staged changes. Committed snapshots can be thought of as “safe” versions of a project—Git will never change them unless you explicitly ask it to. Prior to the execution of git commit, The git add command is used to promote or 'stage' changes to the project that will be stored in a commit. These two commands git commit and git add are two of the most frequently used.

```
git commit
```

Commit the staged snapshot. This will launch a text editor prompting you for a commit message. After you’ve entered a message, save the file and close the editor to create the actual commit.

```
git commit -a
```

Commit a snapshot of all changes in the working directory. This only includes modifications to tracked files (those that have been added with git add at some point in their history).

```
git commit -m "commit message"
```

A shortcut command that immediately creates a commit with a passed commit message. By default, git commit will open up the locally configured text editor, and prompt for a commit message to be entered. Passing the -m option will forgo the text editor prompt in-favor of an inline message.

4.git status:

The git status command displays the state of the working directory and the staging area. It lets you see which changes have been staged, which haven't, and which files aren't being tracked by Git. Status output does not show you any information regarding the committed project history. For this, you need to use git log.

```
git status
```

List which files are staged, unstaged, and untracked.

The command provides the current working branch. If the files are in the staging area, but not committed, it will be shown by the git status. Also, if there are no changes, it will show the message no changes to commit, working directory clean.

5.git config:

The git config command is used initially to configure the user.name and user.email. This specifies what email id and username will be used from a local repository.

When git config is used with --global flag, it writes the settings to all repositories on the computer.

```
git config user.email
```

Email is a child property of the user configuration block. This will return the configured email address, if any, that Git will associate with locally created commits.

GIT CONFIG LEVELS AND FILES

`--LOCAL`

By default, git config will write to a local level if no configuration option is passed. Local level configuration is applied to the context repository git config gets invoked in. Local configuration values are stored in a file that can be found in the repo's .git directory: .git/config

`--global`

Global level configuration is user-specific, meaning it is applied to an operating system user. Global configuration values are stored in a file that is located in a user's home directory

`--system`

System-level configuration is applied across an entire machine. This covers all users on an operating system and all repos. The system level configuration file lives in a gitconfig file off the system root path. $(prefix)/etc/gitconfig on unix systems. On windows this file can be found at C:\Documents and Settings\All Users\Application Data\Git\config on Windows XP, and in C:\ProgramData\Git\config on Windows Vista and newer.

Thus the order of priority for configuration levels is: local, global, system. This means when looking for a configuration value, Git will start at the local level and bubble up to the system level.

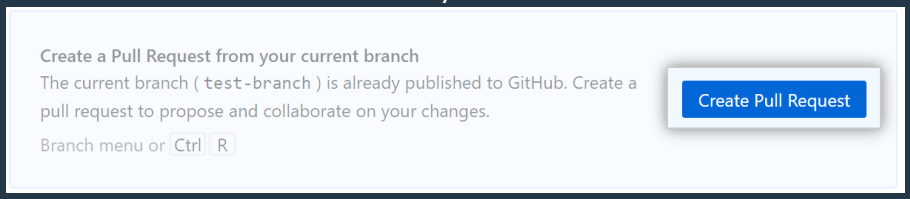
Q3)How to make a pull request?

Ans:

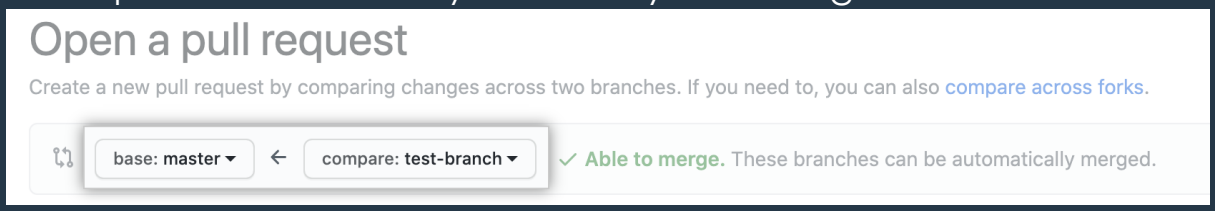
Steps to create a pull request:

1. Switch to the branch that you want to create a pull request for

2. Click Create Pull Request. GitHub Desktop will open your default browser to take you to GitHub



3. On GitHub, confirm that the branch in the base: drop-down menu is the branch where you want to merge your changes. Confirm that the branch in the compare: drop-down menu is the topic branch where you made your changes.



4. Type a title and description for your pull request.
5. To create a pull request that is ready for review, click Create Pull Request. To create a draft pull request, use the drop-down and select Create Draft Pull Request, then click Draft Pull Request.

THANK YOU……