PW SKILLS

JAVA With DSA & System Design

Assignment-Git and GitHub

Part -2

---

1. **How to check if git is available on your system?**
   Ans: To check if git is available on the system or installed, search and launch the "Git Bash" command line from startup menu, then check the Git Version by the following command:

   ```
   $ git –version
   ```

   If the output shows the currently installed version in system means Git is successfully installed, otherwise it gives an error message that indicates the Git software.

2. **How to initialize a new Git repository?**
   Ans: In GIT, a repository can be initialized in the following format-
   If a file is shared on a version control system and control it with Git. Then start the git command line in Git Bash. To initialize a new repository, run the following command:
   TYPE> **$ git init**
   This above command initializes or creates an empty Git repository named. git that holds all necessary repository files on local workspace (project folder).

   If we want to start version controlling on it, we should have to track these files to stage area/ index area/ staging area from a git add command followed by the filename:
   TYPE> **$ git add<filename>**

   If all files are ready to store in local repository, then it will send to local repository by the git commit command:

   ```
   $ git commit -m "commit message…"
   ```

   If we want to share this file to others then we should have to push this file form local repository to the remote repository server by the git

   ```
   $ git push <remote> <local>
   ```

3. **How to tell git about your name and email?**
   Ans: Git supports a command called **git config** that helps to get and set configuration variables and used to set Git configuration values on a global or local place.

   - Setting username

   The username is used by Git for each commit.

   ```
   $ git config --global user.name "Shadab"
   ```

- Setting email id

  The Git uses this email id for each commit.

  ```
  $ git config --global user. email “mscoder@gmail.com"
  ```

4. **How to add a file to the staging area?**

   Ans: The **git add** command is used **t**o add files into the staging area. It shows the tracked files which are ready to commit for local repository.

   ```
   $ git add.   //for all file in one go to staging area
   $ git add <filename> // particular file go to staging area
   ```

5. **How to remove a file from the staging area?**

   Ans: To remove a file from a staging area by the **git rm –cached** command. This command is made possible to un-stage the files from staged or index area to workplace. It specifies that the removal operation will only act on the staging index, not on the repository. The syntax is:

   ```
   $ git rm –cached <filename>
   ```

6. **How to make a commit?**

   Ans: The commit is used to record or save the changes in the local repository, after the git add command which sends the file to the staging area. From the staging area, the files are further moved to the local repository by the **git commit** command. The commit command will commit the changes and generate a commit-id. It will run as follows:

   ```
   $git commit -m “commit message”
   ```

7. **How to send your changes to a remote repository?**

   Ans: To send files from local repository to remote repository by the ‘Push’ command. In the pushing method, after committing, the file the local repository file transfers to the remote repository which updates the remote rfs with local rfs and saves the history or record changes on the remote server. There are the commands for pushing file into the local server:

   - **git init**
   - **git add<filename>**
   - **git commit -m “commit message”**
   - **git branch -m main**
   - **git remote add origin <remote server url>**
   - **git push -u origin main**

8. **What is the difference between clone and pull?**

   Ans:

   Git Fetch -

- It gives the information of a new change from a remote repository without merging in the current branch.
- Repository data is updated in the. git directory.
- It reviews the commits and changes can be done.
- It doesn't have the possibility of merge conflicts.
- Syntax:

```
git fetch <remote>
```

Git Pull -

- It brings the copy of all file changes from a remote repository to local repository.
- The local repository takes updated files directly.
- Updates the changes to the local repository immediately.
- Here, the merge conflicts can be possible if the remote and local repositories have done changes at the same place.
- Syntax:

```
git pull<remote><branch>
```