**Q1: Making a repository in Github Console:** **“InnovationPython\_Namrata”**

A screenshot of a computer screen

Description automatically generated

**Q2. Read about the difference between Git and Git Hub**

Ans: Git is a version control system that lets the developer manage the source code history and tracks all the changes that are made while Github is s cloud-based hosting service that helps an individual to manage git repositories.

**Q3. Read about Git Workflow**

Ans: Git workflow is a recommendation for how to use Git to do the task in a better and more consistent manner. It is important to evaluate which workflow needs to be taken. This decision usually depends on the requirements of the team’s culture. One type of Git workflow is centralized which is great for teams transitioning to SVN.

**STEPS:**

1. Initialize the central repository
2. Host the central repository
3. Clone the central repository
4. Make changes and Commit
5. Push new commits to the central repository
6. Manage conflicts

\*The Git Workflow defines a strict branching model designed around project release.

Q3. How many types of Version control systems are there?

Ans: There are three types of version control systems:

1. **Local version control system:** it is the simplest form and has a database that kept all changes to file under revision control. RCS (Revision Control System) is an example.
2. **Centralized Version Control System:** it is the kind of system which has one repository and each user gets their own working copy. If we want to reflect on the changes, we need to commit them. It makes collaboration between developers easy.
3. **Distributed Version Control Systems:** it contains multiple repositories. Each user has their own repository and working copy. Git is one of them. It helps in avoiding the problem of a single point of failure.

**Q4. Explain branching concept in Git**

Ans: In Git, branching is part of the everyday business process. Git branches are effectively a pointer to snapshot the changes. Whenever we want to add a new feature or fix a bug we can make a new branch to encapsulate the changes. This makes it harder for unstable code to get merged into the main code and gives a chance to clean up the future’s history before merging it into the main branch. A branch is an independent line of development The command git branch command lets a user create, list, name and delete branches.

**Q5. Explain forking workflow in Git.**

Ans: The Forking Workflow is fundamentally different than other popular Git workflows. Instead of using a single server-side repository to act as the “central” codebase, it gives every developer their own server-side repository. This means that each contributor has not one, but two Git repositories: a private local one and a public server-side one. The Forking Workflow is most often seen in public open source projects .The main advantage of the Forking Workflow is that contributions can be integrated without the need for everybody to push to a single central repository. Developers push to their own server-side repositories, and only the project maintainer can push to the official repository. This allows the maintainer to accept commits from any developer without giving them write access to the official codebase. The Forking Workflow typically follows a branching model based on the [Gitflow Workflow](https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow). This means that complete feature branches will be purposed for merge into the original project maintainer's repository. The result is a distributed workflow that provides a flexible way for large, organic teams.