Git and Github



Assignment 2

**1.What do you understand by the term “ version control system”?**

**A version control system (VCS) is a software tool that is used to manage and track changes made to files, typically source code for software development, but it can be used for any type of files.**

* **Lets understand what is version control system with real life      example:**

**A real-life example of how a version control system works would     be a group of writers working on a book together. Each writer would have their own copy of the book's manuscript, and they would make changes to their copy as they write. When they're ready to share their changes with the rest of the team, they would use the version control system to "commit" their changes, which would then be "merged" into the main copy of the manuscript. The version control system would keep track of all the changes made to the manuscript, so that the team can easily revert to an older version if they need to.**

**In this example, the manuscript represents the codebase, the           writers represent the developers, and the version control system represents the tool that is used to manage and track changes to the manuscript.**

**Another example would be a website that undergoes multiple updates and changes over time, the version control system allows to keep track of those changes, revert to a previous version, and collaborate   with multiple developers on the same website.**

**2.What is github?**

**GitHub is a web-based platform for version control and collaboration that uses Git as its version control system. It provides a centralized place for developers to store and share their code, collaborate on projects, and track and manage changes to the codebase.**

**Real life example for github:**

**A real-life example of how GitHub is used would be a team of developers working on an open-source project. They would create a repository on GitHub to store their code, and as they work on the project, they would "commit" their changes to the repository. Other developers can then "clone" the repository, make their own changes, and "pull request" them to the original repository. The maintainers can review the changes, discuss them with the team, and merge them into the main codebase.**

**3.Mention some popular git hosting services.**

**There are following popular git hosting services :**

**GitHub: One of the most popular Git hosting services, GitHub is a web-based platform that provides a centralized place for developers to store and share their code, collaborate on projects, and track and manage changes to the codebase. It has a large community and many tools to help developers collaborate and manage their projects.**

**GitLab: Another popular Git hosting service, GitLab is similar to GitHub but it provides more flexibility and control over the hosting infrastructure. It offers both cloud-based and self-hosted options, and it includes many of the same features as GitHub, such as issue tracking, code review, and project management.**

**Bitbucket: A Git hosting service owned by Atlassian, Bitbucket is geared towards small teams and individual developers. It offers free plans for small teams and it provides many of the same features as GitHub and GitLab, such as issue tracking and code review.**

**SourceForge: A web-based platform for managing and sharing open-source software, SourceForge offers Git hosting for open-source projects and it also provides many tools for collaboration and project management.**

**These are some of the most popular Git hosting services, but there are many other options available depending on your needs.**

**4.Different types of version control systems.**

* **Centralized Version Control Systems (CVCS):**

**In a centralized version control system, a single central repository stores all the versions of the codebase, and developers must connect to the central repository to check out, check in, and merge changes. Examples of CVCS include Subversion (SVN) and Perforce.**

* **Distributed Version Control Systems (DVCS):**

**In a distributed version control system, each developer has a local copy of the entire codebase, including all the versions and changes. This allows for offline development and fast local branching and merging. Examples of DVCS include Git and Mercurial.**

**5.What benefits come with using git?**

**There are many benefits to using Git as a version control system for software development projects. Some of the key benefits include:**

**Distributed version control**

**local branching and merging**

**Flexible branching and merging**

**Staging and committing**

**Speed**

**Open-source and community support**

**Remote repositories**

**Security**

**Compatibility**

**These benefits make Git an ideal choice for software development projects.**

6.What is a git repository?

**A Git repository is a directory or folder that contains all the files, folders, and history of changes for a specific project or codebase. It is the core concept of Git and it is used to store and track the changes made to the codebase over time.**

* **Real life example for git repository:**

**A real-life example of how a Git repository can be used in software development would be a team of developers working on a new mobile app. The team would create a new repository on GitHub to store all the files, folders, and code for the app.**

**Each developer would then "clone" the repository to their own local machine, which would create a copy of the entire codebase and the entire history of changes. As they work on the app, they would make changes to the code and test them on their local machine. When they're ready to share their changes with the rest of the team, they would "commit" their changes to their local repository, and then "push" them to the remote repository on GitHub.**

**The remote repository on GitHub would then be the central place where all the developers can access the latest version of the codebase, view the history of changes, and collaborate on the project.**

**7.How can you initialize a repository in git?**

**There are several ways to initialize a repository in Git, but the most common method is to use the "git init" command. This command creates a new, empty repository in the current directory.**

**Now i am going to write steps to initialize a new repository in Git:**

* **Open a terminal or command prompt window and navigate to the directory where you want to create the new repository.**

* **Type the command "git init" and press enter. This will initialize a new, empty repository in the current directory.**

* **To check the status of the repository and confirm that it has been initialized, type the command "git status" and press enter. This will show the current status of the repository, including any untracked files.**

* **To add files to the repository, you can use the command "git add" followed by the file name or "." to add all the files in the directory.**

* **To commit the changes, you can use the command "git commit -m "message"" where message is a brief description of the changes you made.**

**If you want to connect your local repository with a remote repository, you can use the command "git remote add origin <url>" where url is the link to the remote repository.**

**Name : vinay kumar**

**Mob: 9931860964**

**Github: vinaykumar860964**