**GIT AND GITHUB**

GIT

• Git is a version control system.

• Version Control System is a tool that helps to track changes in code.

GIT-HUB

• Website that allows developers to store and manage their code using Git

1.Concepts of Git

***Version Control System (VCS)*:** Git is a distributed version control system designed to track changes in source code during software development. It allows multiple developers to work on a project simultaneously.

***Repository (Repo):*** A repository is a storage location where the project's files and version history are stored. It can be either local (on your machine) or remote (on a server).

***Commit:*** A commit is a record of changes made to the repository. It captures a snapshot of the project at a specific point in time along with a commit message describing the changes.

***Branch:*** Branches are parallel lines of development that allow developers to work on features or bug fixes independently. They help isolate changes and facilitate collaboration.

***Merge:*** Merging combines changes from one branch into another. It is used to integrate the work done in separate branches, ensuring a cohesive codebase.

***Pull Request (PR):*** A pull request is a mechanism for proposing changes and initiating a discussion around them. It is widely used in platforms like GitHub and GitLab. A pull request allows other team members to review and provide feedback on the proposed changes before merging.

***Remote:*** A remote is a version of the repository that is hosted on a server (e.g., GitHub, GitLab). Developers can push their changes to the remote repository or pull changes made by others.

***Clone:*** Cloning is the process of creating a local copy of a remote repository. This allows developers to work on the project locally, make changes, and push those changes back to the remote repository.

***Fetch and Pull:*** git fetch retrieves changes from a remote repository without merging them into your local branches. git pull is a combination of git fetch and git merge, updating your local branch with changes from the remote.

***HEAD:*** HEAD is a special pointer/reference in Git that points to the latest commit in the branch you are currently working on. It represents the latest state of your working directory.

***Fork:*** Creating a personal copy of someone else's project on a platform like GitHub. Forks allow developers to freely experiment with changes

2. Basic Commands of Git

***git init***: Initializes a new Git repository.

***git clone <repository>***: Creates a local copy of a repository.

***git add <file>***: Adds changes to the staging area.

***git commit -m "message"***: Commits changes with a descriptive message.

***git pull***: Fetches changes from a remote repository and merges them.

***git push***: Pushes local changes to a remote repository.

***git branch <branch\_name>***: Creates a new branch with the specified name.

***git checkout -b <branch\_name>***: Creates and switches to a new branch in one command.

***git branch***: Lists all branches in the repository.

***git merge <branch>***: Merges changes from a specified branch into the current branch.

***git status***: Displays the status of changes as untracked, modified, or staged.

***git fetch***: Retrieves changes from a remote repository without merging them.

3. Concepts on GitHub, GitLab, and Bitbucket

***GitHub***: A web-based platform that provides hosting for Git repositories. It includes collaboration features such as pull requests, issues, and wikis.

***GitLab***: Another web-based platform for version control that includes integrated CI/CD tools. It supports collaboration and project management features.

***Bitbucket***: Atlassian's Git repository management solution that integrates with other Atlassian products. It supports Git and Mercurial version control systems.

4. Industrial Practices of Using Git

***Branching Strategy***: Adopt a branching strategy such as Git Flow or GitHub Flow to manage the creation, merging, and deployment of branches effectively.

***Feature Branches***: Use feature branches to isolate work on new features or bug fixes, enabling parallel development and minimizing conflicts.

***Pull Requests/Merge Requests***: Encourage the use of pull requests or merge requests to propose, review, and discuss changes before merging them into the main branch.

***Code Reviews***: Establish a code review process to ensure code quality, share knowledge among team members, and catch issues early in the development cycle.

***Continuous Integration (CI)***: Implement a CI system to automate testing and build processes, ensuring that changes are continuously integrated and validated.

***Continuous Deployment (CD)***: Set up continuous deployment practices to automate the deployment process after successful testing, reducing the time between code changes and deployment.

***Git Hooks***: Use Git hooks to automate tasks such as code formatting, linting, or testing before allowing commits or pushes.

***Tagging Releases***: Tag releases with version numbers to mark stable points in the project's development and facilitate tracking and deployment.

***Documentation***: Maintain comprehensive documentation, including README files, coding standards, and contribution guidelines, to facilitate collaboration and onboarding.

***Issue Tracking***: Integrate issue tracking tools with Git repositories to manage tasks, track bugs, and coordinate development efforts effectively.

5. Cloning a Repo to Local

Cloning a repository to your local machine is a fundamental operation in Git that allows you to create a local copy of a remote repository. Here's how you can clone a repository:

***a) Navigate to the Repository:*** Go to the website of the platform hosting the repository (e.g., GitHub, GitLab, Bitbucket).

***b) Get the Repository URL***: Find the URL of the repository. You can usually find it on the repository's main page. It will look like https://github.com/username/repository.git (for GitHub).

***c) Open a Terminal or Command Prompt***: Open a terminal on your local machine. You can use the built-in terminal on your code editor or a standalone terminal application.

***d) Navigate to the Desired Directory***: Use the cd command to navigate to the directory where you want to store the local copy of the repository.

*cd path/to/your/directory*

***e) Clone the Repository***: Use the git clone command followed by the repository URL to clone the repository.

*git clone https://github.com/username/repository.git*

Replace the URL with the actual URL of the repository you want to clone.

***f) Authenticate (if required):*** If the repository is private, you may be prompted to enter your username and password or use other authentication methods (SSH keys, personal access tokens).

***g) Wait for the Clone to Complete***: Git will download the entire repository, including its history, to your local machine. Once the process is complete, you'll have a local copy of the repository in the specified directory.

6. Resources Used

* Chatgpt
* Youtube (Brototype, Apna College)
* Google