DAILY DSA | DAY-27 | Project maintainance |
-GOPALKRISHNA A

When working with project it is always good practice to maintain & manage the repositories for real-time collaboration. It encourages teams to work together in developing code and uploading content.

Repository:

- A repository (or repo) is a type of centrally located storage where you can keep all your project's files and resources.
- Any of the project's stakeholders or developers having project access can pull the repository's code (or resource) for new feature delivery or bug fixes in the product or software application.

Why do we need to use repositories:

- **Version Tracking**: A code repository allows developers to track changes made to the codebase over time. Each change is recorded as a commit, making it easy to see who made the change, when it was made, and what exactly was changed. This history can be invaluable for debugging, auditing, and understanding the evolution of the project.
- **Collaboration**: In collaborative development environments, multiple developers may be working on the same codebase simultaneously. A code repository facilitates collaboration by providing a central location where developers can push their changes, pull changes made by others, and merge them seamlessly. This helps prevent conflicts and ensures that everyone is working with the latest code.
- **Backup and Disaster Recovery**: Repositories serve as a backup of your code. In case of data loss or system crashes, you can restore your code to a previous state from the repository. This is especially important for critical projects where code loss can result in significant downtime and financial losses.
- **Testing and Quality Assurance:** Maintaining different branches in a repository allows for parallel development. Developers can create separate branches to work on new features, bug fixes, or experimental changes without affecting the stable version of the code. This facilitates testing and quality assurance efforts, as you can isolate changes and test them independently.
- **Documentation**: Commit messages in a code repository serve as a form of documentation. They provide context about why a change was made and what problem it was intended to solve.
- **Code Reviews**: Code repositories support code review processes. Developers can submit their changes as pull requests or merge requests, and team members can review the code, provide feedback, and ensure that best practices and coding standards are followed before the changes are merged into the main codebase.
- **Code Reusability**: With a well-maintained repository, you can create libraries, modules, or components that can be reused in different projects. This promotes code reusability and consistency across your organization's software projects.

Key concepts & Features:

- A **repository,** or software repository, is also known as a code repository.
- A **branch** refers to a separate line of development that initially has the same code base as the main repository. This new branch can have its own set of files and commits.
- A **commit** means saving changes made to a project's files to the repository itself.
- A **pull** from a repository simply means getting the source code from the remote repository (a cloud server) to the local machine. A pull request contains file changes (the files that are added/modified or deleted) that are to be integrated with the main repository.
- A **push** means pushing the locally built source code to a remote repository either created on GitHub, Bitbucket or similar repository management platform.
- A **pull request** refers to when we make changes in one of your branches and try to integrate those changes with the main repository.