Introduction

1. What is GIT and why is it used?

Git is a DevOps tool used for source code management. Git is software for tracking changes in any set of files. It is a free and open-source version control system used to handle small to very large projects efficiently.
Git is used to tracking changes in the source code, enabling multiple developers to work together on non-linear development
Git is an example of a DVCS (hence Distributed Version Control System).

2. What is GITHUB?

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

3. What is Repository?

A repository contains all of your project's files and each file's revision history. You can discuss and manage your project's work within the repository. A Git repository is the .git/ folder inside a project. This repository tracks all changes made to files in your project, building a history over time. Meaning, if you delete the .git/ folder, then you delete your project's history.

4. What is Version Control System (VCS)?

Version Control Systems are the software tools for tracking/managing all the changes made to the source code during the project development. It keeps a record of every single change made to the code. It also allows us to turn back to the previous version of the code if any mistake is made in the current version. Without a VCS in place, it would not be possible to monitor the development of the project.

5. Types of VCS

- Local Visual Control System
- Centralized Version Control System
- Distributed Version Control System

- **Local Visual Control System**

  Local Version Control System is located in your local machine. If the local machine crashes, it would not be possible to retrieve the files, and all the information will be lost. If anything happens to a single version, all the versions made after that will be lost.

- **Centralized Version Control System**

  The Centralized Version Control Systems, there will be a single central server that contains all the files related to the project, and many collaborators checkout files from this single server (you will only have a working copy). The problem with the Centralized Version Control Systems is if the central server crashes, almost everything related to the project will be lost.

- **Distributed Version Control System**

  In a distributed version control system, there will be one or more servers and many collaborators similar to the centralized system. But the difference is, not only do they check out the latest version, but each collaborator will have an exact copy of the main repository on their local machines. Each user has their own repository and a working copy. This is very useful because even if the server crashes we would not lose everything as several copies are residing in several other computers.