1. what is git and what do you mean by the term ‘ version control system ‘ ?

Ans - **Git** is an **open-source distributed version control system**. It is designed to handle minor to major projects with high speed and efficiency. It is developed to co-ordinate the work among the developers. The version control allows us to track and work together with our team members at the same workspace.

Git is a modern and widely used **distributed version control** system in the world. It is developed to manage projects with high speed and efficiency. The version control system allows us to monitor and work together with our team members at the same workspace.

2 . what is github and mention some popular git hosting services ?

Ans – Github is a website and cloud based service that helps developers store and manage their code , as well as track and control changes to their code .

We’ve established that Git is a version control system, similar but better than the many alternatives available. So, what makes GitHub so special? Git is a command-line tool, but the center around which all things involving Git revolve is the hub—GitHub.com—where developers store their projects and network with like minded people.

There are many git hosting services like bitbucket , github , gitlab , perforce , codebase , sourceforge .

3 . Different types of version control system and what benefits come with using git ?

Ans -**Types of Version Control Systems:**

* Local Version Control Systems
* Centralized Version Control Systems
* Distributed Version Control Systems

**Local Version Control Systems:** It is one of the simplest forms and has a database that kept all the changes to files under revision control. RCS is one of the most common VCS tools. It keeps patch sets (differences between files) in a special format on disk. By adding up all the patches it can then re-create what any file looked like at any point in time.

**Centralized Version Control Systems:** Centralized version control systems contain just one repository globally and every user need to commit for reflecting one’s changes in the repository. It is possible for others to see your changes by updating.

Two things are required to make your changes visible to others which are:

* You commit
* They update

**Distributed Version Control Systems:** Distributed version control systems contain multiple repositories. Each user has their own repository and working copy. Just committing your changes will not give others access to your changes. This is because commit will reflect those changes in your local repository and you need to push them in order to make them visible on the central repository. Similarly, When you update, you do not get others’ changes unless you have first pulled those changes into your repository.

To make your changes visible to others, 4 things are required:

* You commit
* You push
* They pull
* They update

**Distributed Version Control Systems:** Distributed version control systems contain multiple repositories. Each user has their own repository and working copy. Just committing your changes will not give others access to your changes. This is because commit will reflect those changes in your local repository and you need to push them in order to make them visible on the central repository. Similarly, When you update, you do not get others’ changes unless you have first pulled those changes into your repository.

To make your changes visible to others, 4 things are required:

* You commit
* You push
* They pull
* They update

**Benefits of the version control system:**

* Enhances the project development speed by providing efficient collaboration,
* Leverages the productivity, expedites product delivery, and skills of the employees through better communication and assistance,
* Reduce possibilities of errors and conflicts meanwhile project development through traceability to every small change,
* Employees or contributors of the project can contribute from anywhere irrespective of the different geographical locations through this **VCS,**
* For each different contributor to the project, a different working copy is maintained and not merged to the main file unless the working copy is validated. The most popular example is **Git, Helix core, Microsoft TFS,**
* Helps in recovery in case of any disaster or contingent situation,
* Informs us about Who, What, When, Why changes have been made.

4. what is a git repository and how can you initialize a repository in git ?

Ans – A git repository  is a virtual storage of your project. It allows you to save versions of your code, which you can access when needed.

To create a new repo, you'll use the git init command. git init is a one-time command you use during the initial setup of a new repo. Executing this command will create a new .git subdirectory in your current working directory. This will also create a new main branch.