1.Опишіть для чого використовують git, які основні дії та команди в ньому виконують.

**Git** is a version control system designed to track changes in project files and facilitate collaboration within a team. The main tasks and commands in Git include:

**Repository Initialization:** The git init command creates a new Git repository in the current directory or initializes an existing project as a repository.

**Commit Creation:** The git commit command allows you to save the current state of the project in its history. Each commit includes a message explaining what changes were made.

**Adding Files to the Index:** The git add command adds changes in files to the "index" (or "staging area"). Files in the index will be included in the next commit.

**Checking Status:** The git status command displays the current status of files in the working directory, index, and repository.

**Branch Switching:** The git checkout command allows you to switch between different branches in the repository. Branches enable you to separate the project's history and develop it in parallel.

**Synchronization with Remote Repository:** The git push and git pull commands are used to interact with remote repositories. git push sends changes to the server, while git pull updates your local repository with changes made by others.

**Commit History:** The git log command lets you view the commit history of the repository. It displays a list of commits, their hash sums, authors, dates, and messages.

**Branching and Merging:** Git allows you to create branches for working on specific features or tasks and then merge them into the main branch using the git merge command.

2.Що таке "комміт" як він дозволяє відслідковувати зміни у файлах?

**A commit** is the fundamental unit for storing the history of changes in Git. Each commit represents a snapshot of the current state of the project. Each commit includes the following information:

**Changes in Files**: A commit includes the changes made to files since the previous commit. This means you can track which lines of code were modified, added, or deleted.

**Commit Message**: You can add a brief message that explains the nature of the changes made in this commit. This helps other developers, and yourself understand what was done.

**Author and Date:** Each commit contains information about the author of the commit and the date it was created. This is crucial information for tracking who made the changes and when.

Commits form a chain of changes that can be viewed and analyzed. They enable developers to observe the project's history, mark branching points, perform merges, and track development progress.