Task2

Version Control:

Version control is a system that allows you to keep track of changes made to files and directories over time. It helps you manage different versions of your code and collaborate with others effectively. Version control systems enable you to track modifications, revert to previous versions, and merge changes seamlessly.

Git and GitHub are common tools used in programming.

They help you manage different versions of your code and collaborate with other developers.Building projects is one of the core parts of being a developer. And Git and GitHub are essential tools you'll use when building projects with others.

Git

Git is a distributed version control system that allows multiple people to work on the same project simultaneously. It tracks changes made to files and directories and allows you to switch between different versions of your code. Git stores a complete history of your project, enabling you to collaborate, experiment, and undo changes easily,is a version control system that lets you manage and keep track of your source code history.

GitHub is a cloud-based hosting service that lets you manage Git repositories. If you have open-source projects that use Git, then GitHub is designed to help you better manage them

Why GitHub Works Only with Git:

GitHub is a web-based hosting service for Git repositories. It works exclusively with Git because Git is the most widely used and supported version control system. GitHub provides a platform for hosting Git repositories, managing collaboration, and offering additional features like pull requests, issue tracking, and project management tools.

Adding Git to the Project:

To add Git to your project, follow these steps:

1-Install Git: Download and install Git from the official website (https://git-scm.com/). Git provides installers for different operating system

2-Initialize Git Repository: Open the command line or terminal and navigate to your project's directory. Run the following command to initialize a new Git repository

```
git init
```

3- Track Files: Use the following command to start tracking files in your project:

git add .

4- Commit Changes: After making changes to your code, commit them to the Git repository using the following command: git commit -m "Commit message" , Replace “Commit message” with a brief description of the changes you made.

5- Connect to a Remote Repository: If you want to use GitHub as your remote repository, create a new repository on GitHub. Then, use the following command to connect your local repository to the remote repository: git remote add origin <remote_repository_url>

6- Push Changes: Finally, push your local changes to the remote repository using the following command: git push -u origin master This command pushes your changes to the master branch of the remote repository.

Main vs Branches:

In Git, the main branch (previously known as master) is the default branch that contains the main line of development. It usually represents the stable and production-ready version of the code. Other branches can be created to work on new features, bug fixes, or experiments without affecting the main branch.

Branches allow you to work on different versions of your code simultaneously. You can create a new branch using the following command: git branch <branch_name>

To switch to a branch, use the following command:

git checkout <branch_name>

You can merge changes from one branch to another using the git merge command. This helps incorporate new features or bug fixes into the main branch when they are ready.To summarize, version control with Git and GitHub provides a structured and collaborative approach to managing code changes, tracking project history, and facilitating team collaboration. By using Git, you can easily manage different versions of your code, while GitHub offers a platform for hosting Git repositories and enabling effective collaboration among team members.