Git for Version Control

Step 1: Install Git

First, make sure Git is installed on your system. You can download it from git-scm.com

Verify the installation by running:

```
Code -: git --version
```

Step 2: Set Up a Git Repository

Navigate to Your Project Directory:

```
Code -: cd path/to/your/project
```

Initialize the Git Repository:

```
Code -: git init
```

Create a .gitignore File:

```
Code-: touch .gitignore
```

Example .gitignore content:

```
Code -:  /node_modules/dist*.log
```

Step 3: Track Changes

Add Files to the Repository:

To start tracking files, you need to stage them using the git add command.

```
Code -:  git add .
```

This stages all files in the current directory. You can also stage specific files:

```
Code -: git add src/index.js
```

Commit Changes:

After staging the files, commit them to the repository with a message describing the changes.

Code -: git commit -m " project setup"

Check the Status:

To see the status of your files (tracked, staged, and unstaged changes), use:

Code -: git status

Step 4: Manage Versions

Create a New Branch:

Branching allows you to work on different features or fixes independently.

Code -: git branch feature-branch

Switch to the new branch:

Code -: git checkout feature-branch

Merge Changes:

After making changes in your branch and committing them, you can merge the branch back into the main branch.

First, switch to the main branch:

Code -: git checkout main

Then merge the feature branch:

Code -: git merge feature-branch

View Commit History:

To see the history of commits, use:

Code -: git log

Step 5: Collaborate with Remote Repositories

Add a Remote Repository:

Code -: git remote add origin https://github.com/username/repository.git

Push Changes to the Remote Repository:

To push your commits to the remote repository, use:

Code -: git push -u origin main

Pull Changes from the Remote Repository:

To update your local repository with changes from the remote repository, use:

Code -: git pull origin main