TRAINING TR-102 DAY 19 REPORT

Date-9 July 2024

Learned about Git Command Line

Setup:

- `git config --global user.name "<Your Name>"`: Configures your name for commits.
- `git config --global user.email "<Your Email>"`: Configures your email for commits.
- `git init`: Initializes a new Git repository in the current directory.

Tracking Changes:

- `git status`: Displays the status of your working directory (modified, staged, untracked files).
- `git add <file>`: Stages a specific file for the next commit.
- `git add .`: Stages all modified files in the working directory.
- `git reset <file>`: Unstages a file while keeping the changes in the working directory.
- `git diff`: Shows the difference between the working directory and the index (unstaged changes).
- `git diff --staged`: Displays the difference between the index and the HEAD (staged changes).

Commits:

- `git commit -m "<message>"`: Creates a new commit with a specified message.

Branching and Merging:

- `git branch`: Lists all local branches.
- `git branch <branch-name>`: Creates a new branch.
- `git checkout <branch-name>`: Switches to a specified branch.
- `git merge <branch-name>`: Merges changes from another branch into the current branch.
    - Use `git merge --ff-only <branch-name>` for a fast-forward merge (no merge commit).

Undoing Changes:

- `git stash`: Temporarily saves uncommitted changes.
- `git stash pop`: Applies the most recent stash to the working directory.
- `git stash list`: Displays a list of stashes.

Viewing History:

- `git log`: Shows the commit history of the current branch.
  - `-n <number>`: Limits the number of commits shown.
  - `--oneline --graph --decorate`: Provides a compact view with commit graphs and references.

Ignoring Files:

- Create a `.gitignore` file in your project root to specify files or patterns to exclude from version control.