**GIT & GITHUB**

**Document: Research on GIT & GitHub**

**1. What is GIT?**

GIT is a distributed version control system that allows multiple developers to work on a project simultaneously without overwriting each other’s changes. It tracks changes to files, allowing you to revert to previous versions and collaborate with others.

Key Features of GIT:

- Version Control: Keeps track of changes in the code.

- Branching and Merging: Allows multiple branches for development and merging them back together.

- Distributed: Each developer has the full history of the project locally.

- Speed and Efficiency: Optimized for performance.

**2. What is GitHub?**

GitHub is a web-based platform that uses GIT for version control. It provides a collaborative environment for developers to work on projects, share code, and manage their work.

Key Features of GitHub:

- Repositories: Storage space for your project, where you can keep your files and track versions.

- Pull Requests: Propose changes to a codebase and discuss them with collaborators.

- Issues Track tasks, enhancements, and bugs.

- GitHub Actions: Automate workflows directly from your repository.

**3. Key Differences Between GIT and GitHub:**

- GIT: A version control system to manage your code locally.

- GitHub: A hosting service for GIT repositories with additional features like collaboration, project management, and continuous integration.

**4. Common GIT Commands and Their Uses:**

- `git init`: Initializes a new GIT repository.

- `git clone [url]`: Clones an existing repository.

- `git add [file]`: Stages changes for the next commit.

- `git commit -m "message"`: Commits the staged changes with a message.

- `git status`: Shows the status of changes.

- `git push`: Pushes local commits to the remote repository.

- `git pull`: Fetches and merges changes from the remote repository.

**5. Setting Up a GitHub Repository:**

1. Create a Repository on GitHub:

- Go to GitHub and sign in.

- Click the “New” button in the repositories section.

- Enter a repository name and description.

- Choose to make it public or private.

- Click “Create repository”.

2. Push Code to GitHub:

- Open your terminal and navigate to your project directory.

- Initialize a new GIT repository:

git init

- Add your files:

git add

- Commit your changes:

git commit -m "Initial commit"

- Link your local repository to GitHub:

git remote add origin [your-repository-URL]

- Push your changes to GitHub:

git push -u origin master