PURBANCHAL UNIVERSITY

KHWOPA ENGINEERING COLLEGE
LIBALI-8, BHAKTAPUR

A Lab Report On

VB.NET

SUBMITTED BY

Rabin Thimi(770328)

SUBMITTED TO

Department of Computer Engineering

Date: 2081 Chaitra 8

Introduction

GIT

Git was created by Linus Torvalds in 2005 (Creator of Linux OS). It is a distributed version control system used for keeping track of changes, managing multiple versions of code and collaborating seamlessly across teams. It allows developers to work on their own local copies of a project, while still enabling them to push changes to a shared repository.

Git helps to manage and track changes in code by keeping log of commits, branching, showing changes made, and etc. The key features of GIT are,

- Version Tracking: GIT follows all adjustments done in one record, letting you revert to old releases without trouble.
- Collaboration: Different programmers can work on a similar task at the same time without clash.
- Branching: You have the option to create distinct branches for new attributes, bug repairs or tests.
- Distributed System: Every programmer has an entire version of the project implying that it is decentralized software.
- Log of Commits: With this feature, GIT maintains an account of all commit actions (changes), which makes understanding how a project has evolved over time much easier.

Basic command of GIT:

1. Setup and Configuration
    a. git config –global user.name “user_name” : Sets your name in git.
    b. git config –global user.email “user@gmail.com” : Sets your email in git.

2. Repository Management
    a. git init : Initializes a new git repository.
    b. git log : Shows commit details eg.(commit_id, author, date, branch)
    c. git clone <repository_url> : Clones an existing repository.

3. Working with Changes
    a. git add <file> : Stages a specific file for commit. but if use dot.
    b. git add . : Stages all changes in stage area.
    c. git commit -m “commit _message” : Commits staged changes with a message.

4. Branching and Merging
    a. git branch : List all branches.
    b. git branch <branch_name> : Create a new branch.
    c. git checkout <branch_name>: Switches to a specified branch also adding.all changes of working branch into staging area of destination branch.
    d. git switch <branch_name> : Just switches to specified branch.
    e. dir : Show all directories available in working branch.

f. git merge <branch> : Merges a branch into the current branch.

5. Viewing Changes and Logs
    a. git status : Shows the status of changes.
    b. git log : Displays commit history.
    c. git diff : Show difference between files.

6. Remote Repository
    a. git remote add origin “repository_url” : Adds a remote repository.
    b. git push origin -u <branch_name> : First time push branch.
    c. git push origin <branch_name> : Pushing changes to remote repository.
    d. git pull origin <branch_name> : Pulls changes from a remote repository.

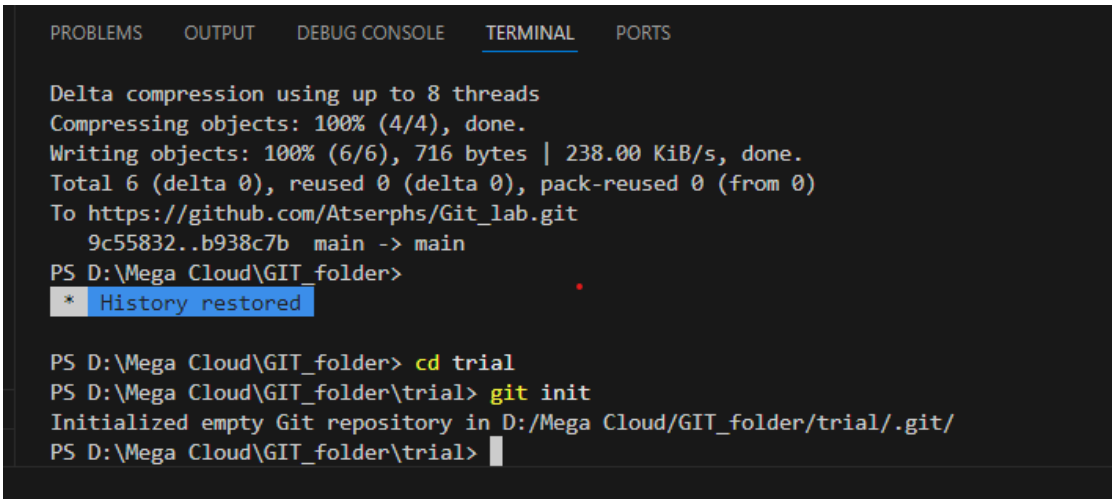
GitHub:

GitHub is a web-based platform that uses Git, a version control system, to help developers manage and track changes in their code. GitHub itself provides access control, bug tracking, software feature requests, task management, continuous integration, and wikis for project. Here are some key features of GitHub,

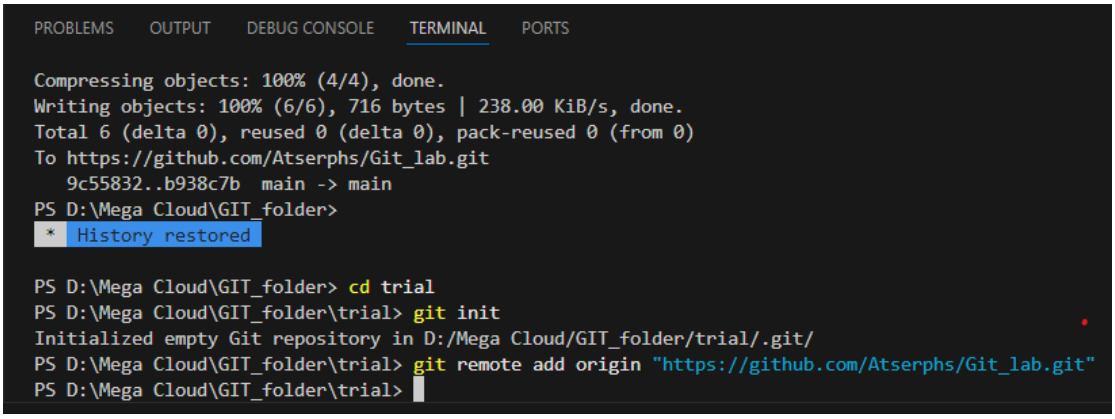
- Code Review: Inline commenting and feedback system for improving code quality collaboratively.
- Issues Tracking: Organize and track tasks, bugs, and features requests with labels, milestones, and assignees.
- Pull Request: A structured way for developers to propose, discuss and review changes before me arranging them.
- Actions: Automate workflows like testing, or deployment.
- Projects: A Kanban-style project board to visualize workflows and task progress.
- Wiki: Serves as a collaborative space for documentation. It’s especially useful for projects that require comprehensive instructions, guidelines, or other important information.
- Security: It have Dependabot and Code Scanning to ensure the no security compromised.
    a. Dependabot: Automatically updates dependencies to reduce security risks.
    b. Code Scanning: Detect vulnerabilities and potential issues in codebases.
- Insights: Provides valuable analytics about your repository to track activity and improve project management
- Code Hosting and Sharing: Share codes in repository. Also host static websites directly from repository.

Observation:

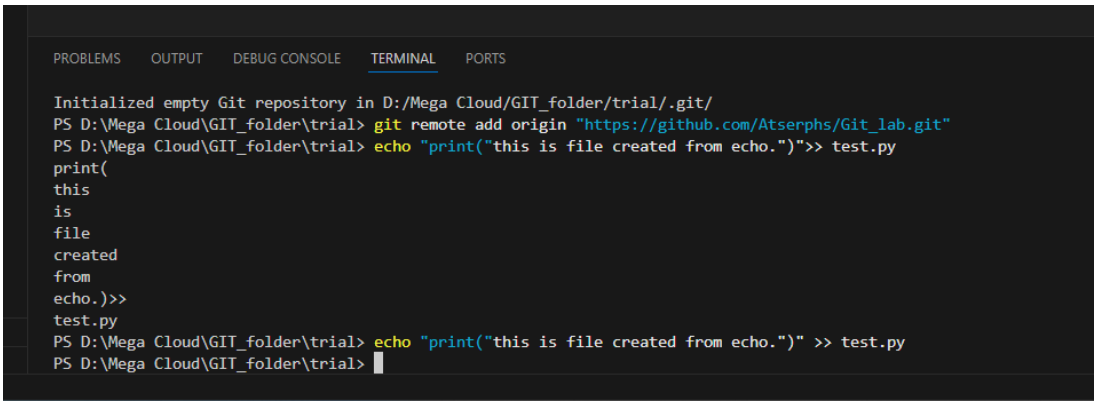
1. Initializing git in folder named trial.



2. Adding remote origin.



3. Creating file in trail folder.



4. Viewing status of GIT.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Mega Cloud\GIT_folder\trial> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        hit
        test.py

nothing added to commit but untracked files present (use "git add" to track)
PS D:\Mega Cloud\GIT_folder\trial>
```

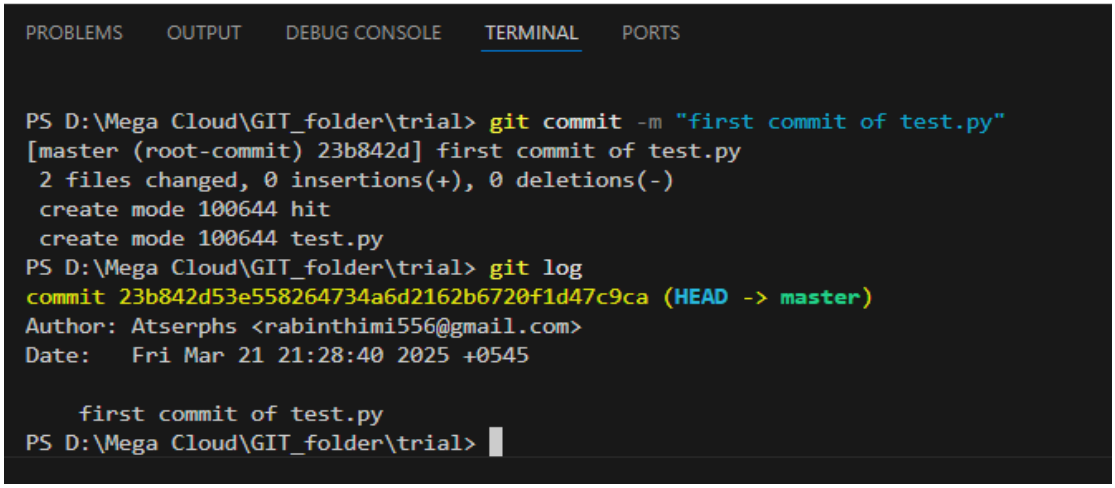
5. Changes added to staging area.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

nothing added to commit but untracked files present (use "git add" to track)
PS D:\Mega Cloud\GIT_folder\trial> git add .
PS D:\Mega Cloud\GIT_folder\trial> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   hit
        new file:   test.py

PS D:\Mega Cloud\GIT_folder\trial>
```

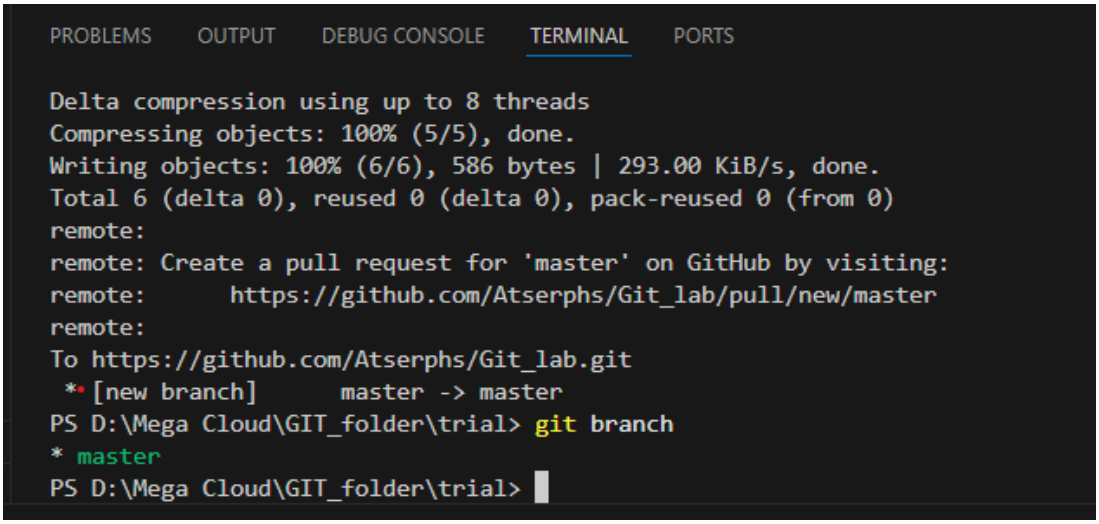
6. Committing changes to repository.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   hit
        new file:   test.py

PS D:\Mega Cloud\GIT_folder\trial> git commit -m "first commit of test.py"
[master (root-commit) 23b842d] first commit of test.py
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 hit
 create mode 100644 test.py
PS D:\Mega Cloud\GIT_folder\trial>
```

7.
8. Viewing log in Git.



9. Viewing branch in Git.



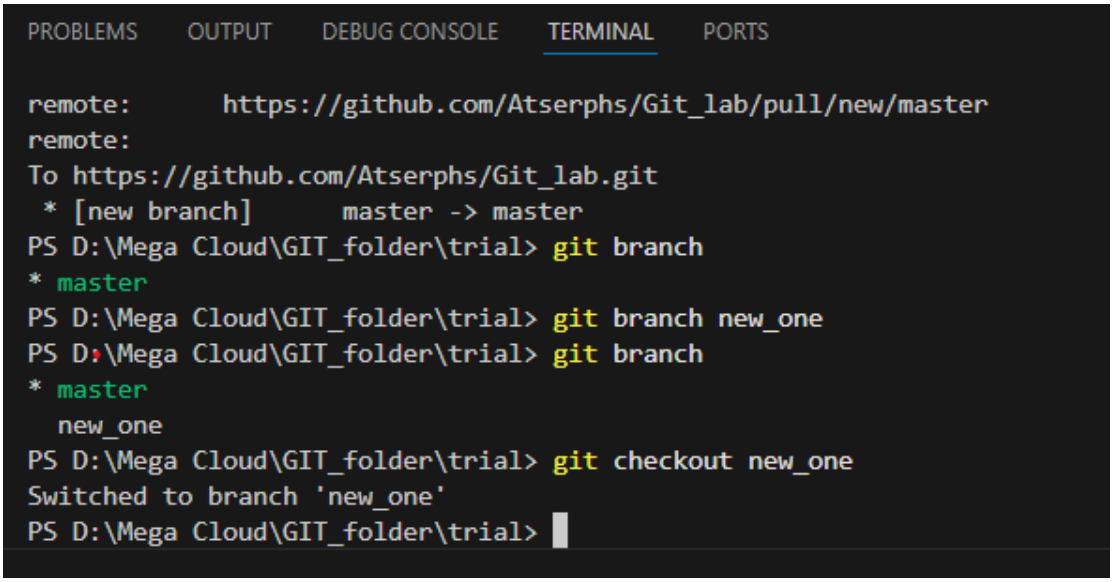
10. Creating new branch.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/Atserphs/Git_lab/pull/new/master
remote:
To https://github.com/Atserphs/Git_lab.git
 * [new branch]      master -> master
PS D:\Mega Cloud\GIT_folder\trial> git branch
* master
PS D:\Mega Cloud\GIT_folder\trial> git branch new_one
PS D:\Mega Cloud\GIT_folder\trial> git branch
* master
  new_one
PS D:\Mega Cloud\GIT_folder\trial>
```

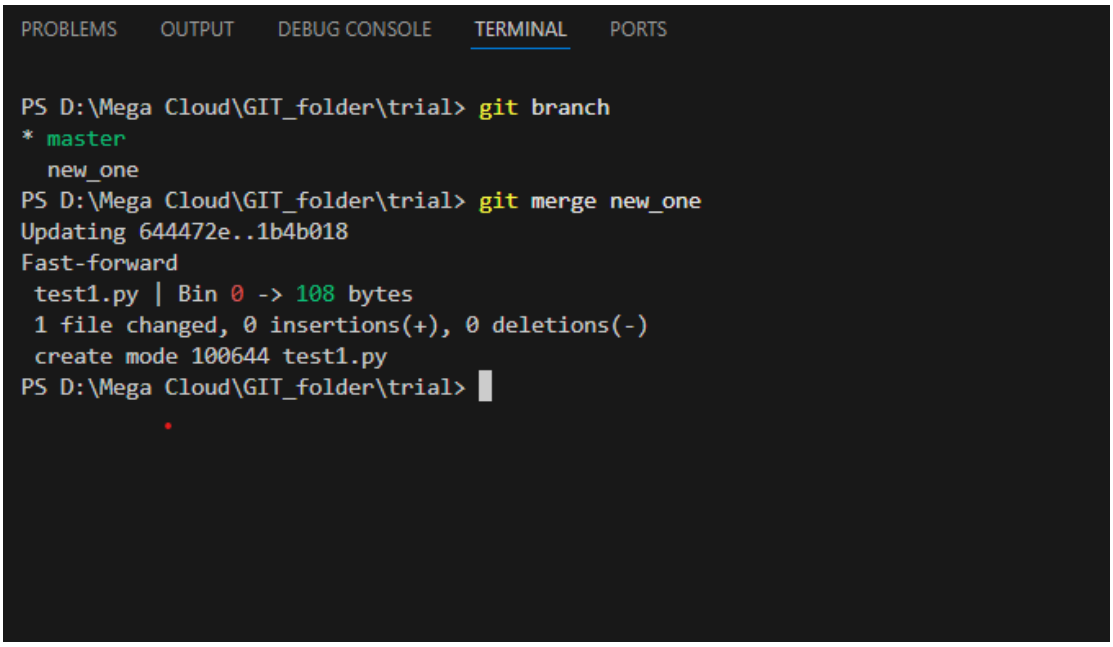
11. Switching to new branch.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Mega Cloud\GIT_folder\trial> git switch new_one
Switched to branch 'new_one'
PS D:\Mega Cloud\GIT_folder\trial>
```

12. Switching with checkout.



13. Merging between branch.



Conclusion:

In conclusion we saw how Git works by executing some basic commands. It includes initialization, setup and configuration, repository management, working with changes, branching and merging, viewing changes and logs, and adding remote repository. These commands are executed in command line interface which results are shown as above.