SOURCE CODE MANAGEMENT

CODE : CSE2015

SOLT : L3 L4

NAME : SUHAS. G

Enroll no :A86605224341

Faculty : Dr Monit Kapoor

INDEX

INTRODUCTION :

Source Code Management:

SCM is the practice of tracking and controlling changes to software code throughout the development process. It helps developers manage versions of code files, collaborate with team members, and maintain a history of all modifications.

Key purposes of SCM include:

- Version Control: Keeps a history of every change made to the code so you can review, compare, or revert to previous versions if needed.
- Collaboration: Allows multiple developers to work on the same project simultaneously without overwriting each other's changes.
- Branching and Merging: Enables developers to create separate branches to work on new features or fixes independently and merge them back into the main codebase when ready.
- Backup: Acts as a safety net by storing code in remote repositories (e.g., GitHub) to prevent data loss.

Tools for Source Code Management :

1. **GitHub**
   It is a web-based platform that hosts Git repositories and provides tools for version control, collaboration, and project management. It allows

developers to store their code online, track changes, collaborate with others, and manage software projects efficiently.

- GitHub supports features like branching, pull requests, issue tracking, and code reviews, making it a powerful tool for both individual developers and teams.

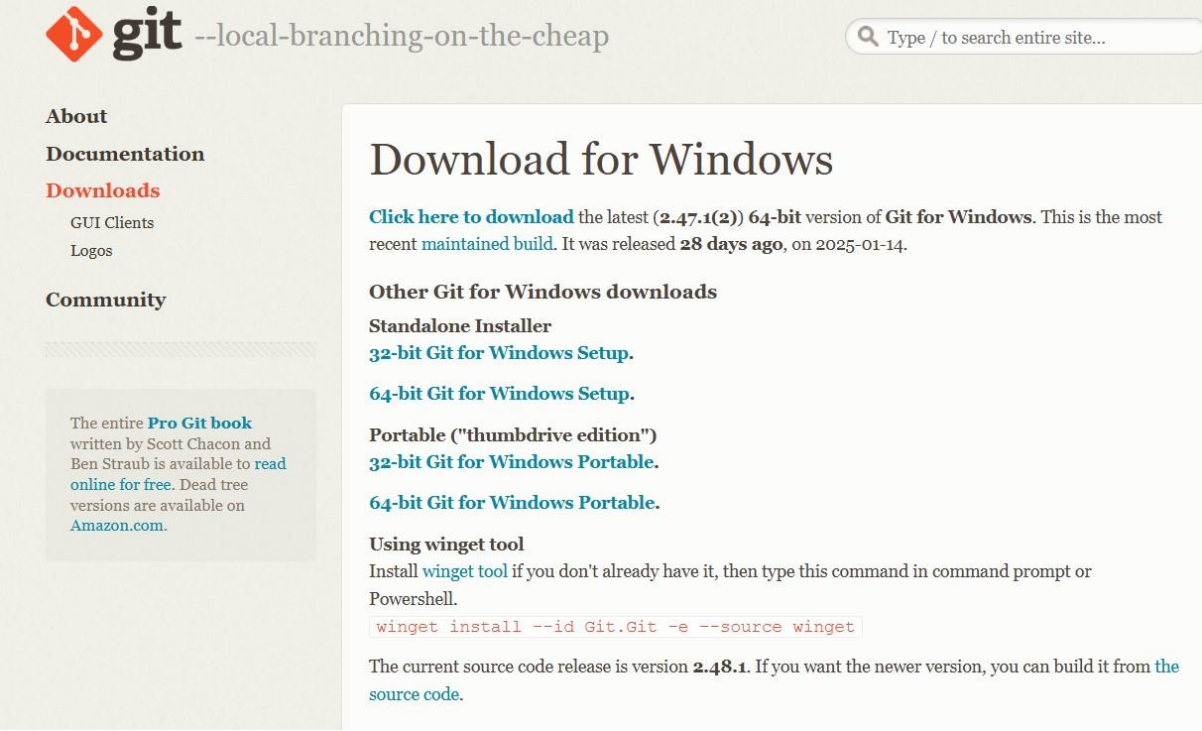
2. Git Bash

It is a command-line interface that brings the powerful Git version control system to Windows. It provides a Bash shell environment, which lets users run Unix-style commands and Git commands on Windows machines.

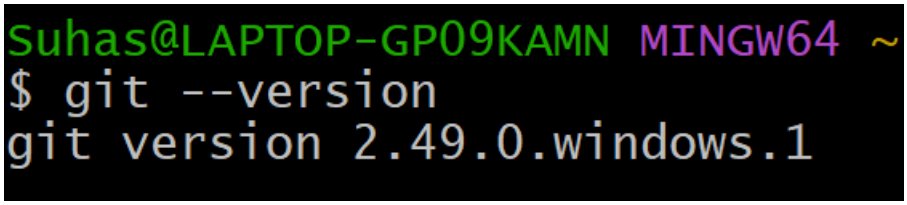
- Git Bash allows you to interact with Git repositories locally by running commands such as git init, git add, git commit, git push, and more.
- It is especially useful for managing source code versions, resolving conflicts, and coordinating work among team members.

1. Installing Git in Windows

Step 1: Visit section 1.5 of pro git document and navigate to Windows section



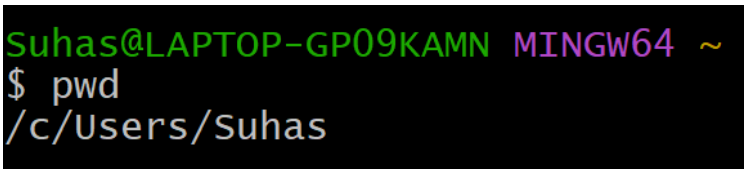
Step 2: Verify Git Installation:



Basic Commands :

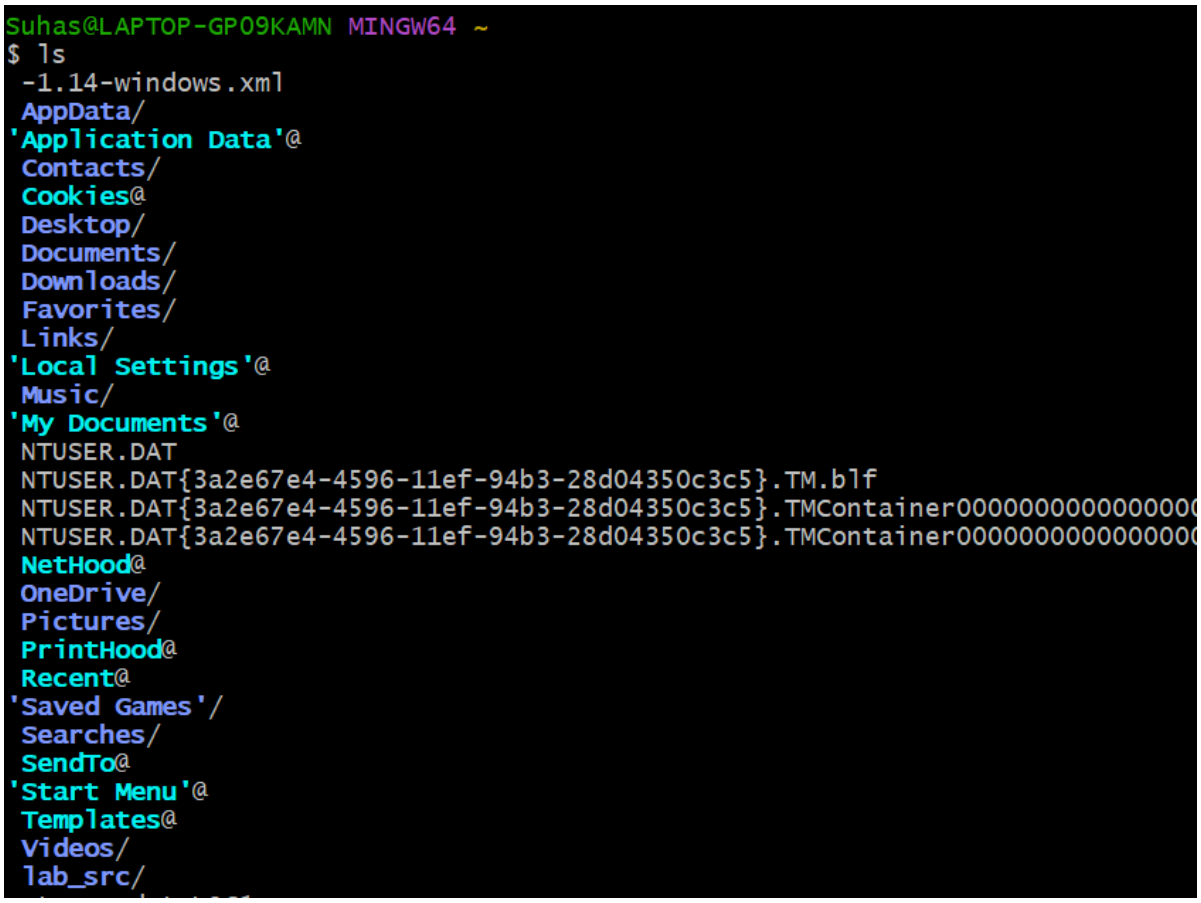
1) Command: pwd

Description: Prints the directory the user is working in.



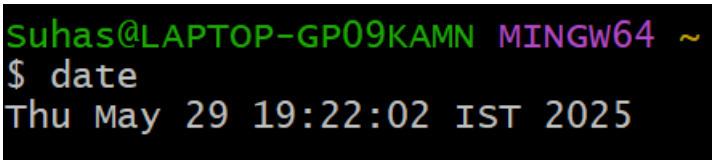
2) Command: ls

Description: Lists all files and directories in the current directory.



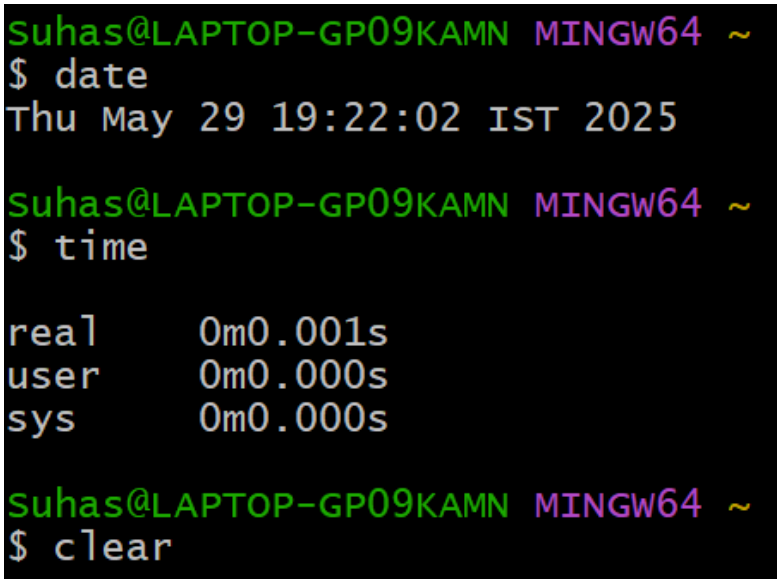
3. Command: date

Description: shows the current date and time in a standard format



4. Command: clear

Description: The `clear` command in the CLI is used to clear all the current text and output displayed in the terminal window.



5. Command: time

Description: The `time` command in the CLI is used to measure the execution time of a command or program.

```
Suhas@LAPTOP-GP09KAMN MINGW64 ~
$ time

real    0m0.001s
user    0m0.000s
sys     0m0.000s
```

6. Command: cd ‘Directory’

Description: Changes the current working directory to the desired directory.

```
Suhas@LAPTOP-GP09KAMN MINGW64 ~
$ cd documents

Suhas@LAPTOP-GP09KAMN MINGW64 ~/documents
$
```

7. Command: cd ..

Description: Goes back to the previous directory.

```
Suhas@LAPTOP-GP09KAMN MINGW64 ~
$ cd documents

Suhas@LAPTOP-GP09KAMN MINGW64 ~/documents
$ cd ..

Suhas@LAPTOP-GP09KAMN MINGW64 ~
$
```

8. Command: mkdir

Description: To create a new directory.

```
Suhas@LAPTOP-GP09KAMN MINGW64 ~/sc_files
$ mkdir new_file

Suhas@LAPTOP-GP09KAMN MINGW64 ~/sc_files
$ ls
new_file/  program.txt/
```

9. Command: rmdir

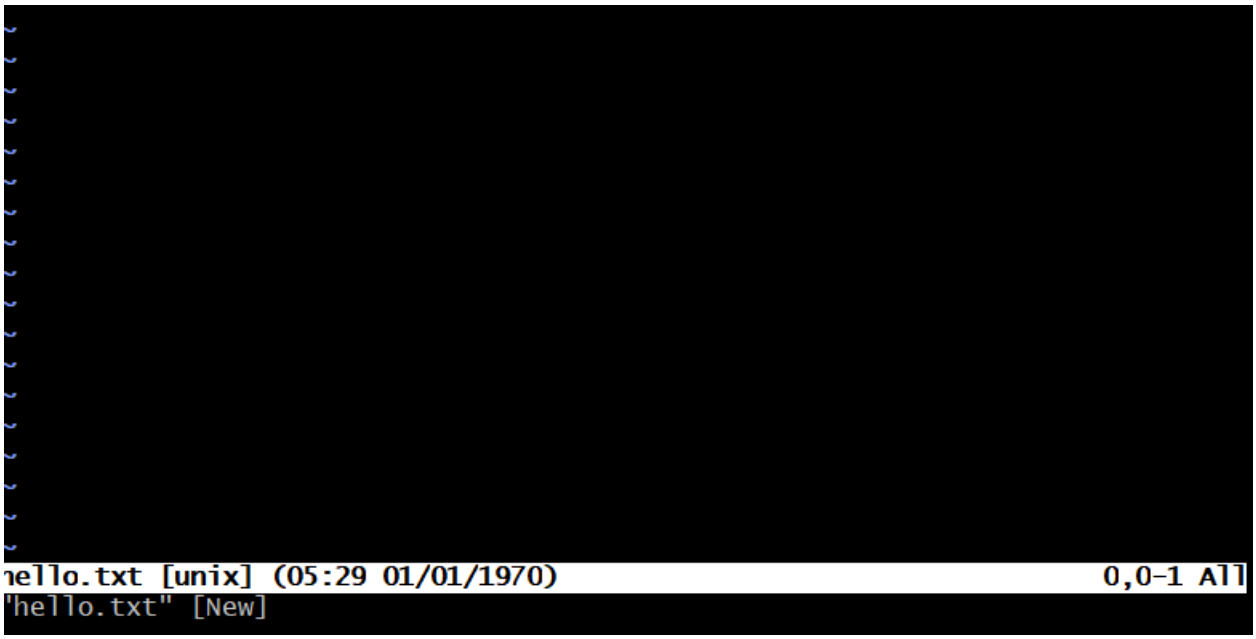
Description: To delete a directory

3. Vim Text Editor

1) Command: vi hello.txt

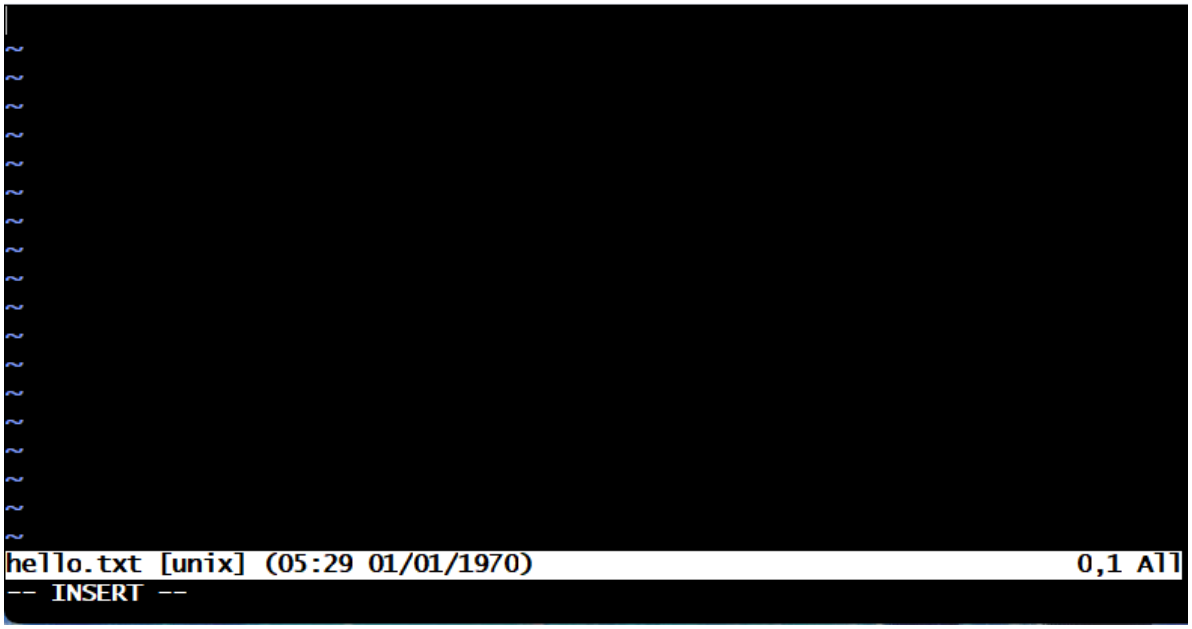
Description: Opens (or creates) the file `hello.txt` in the Vim text editor.

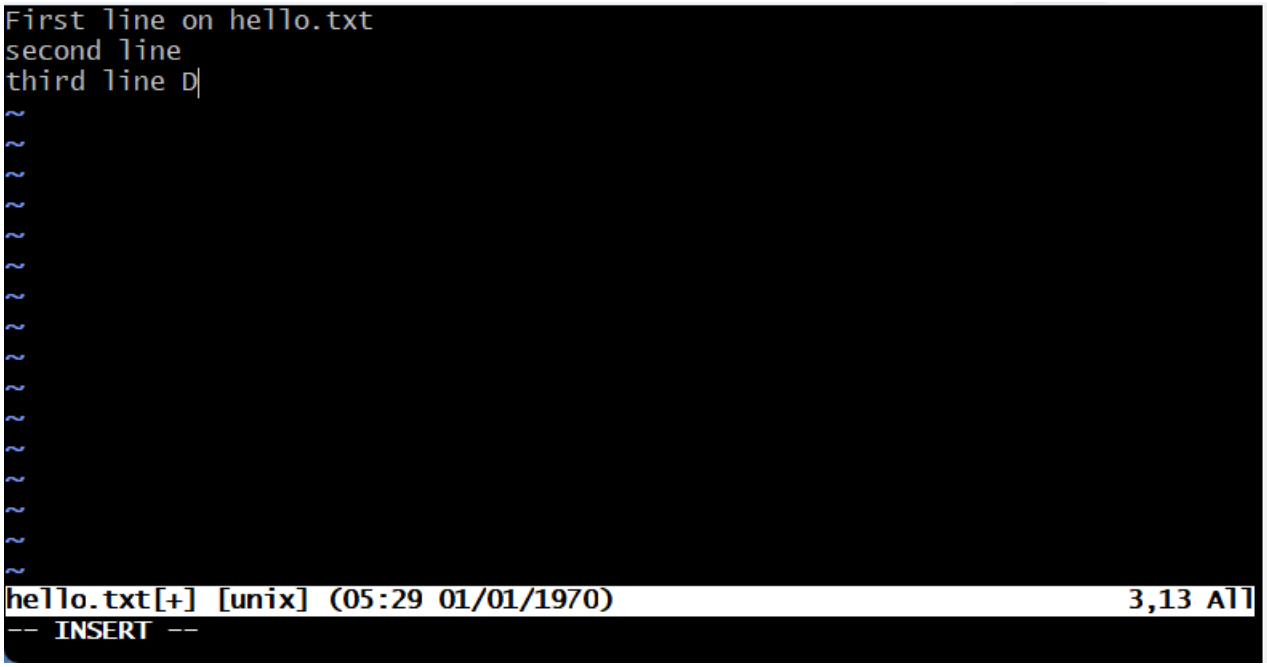
```
Suhas@LAPTOP-GP09KAMN MINGW64 ~/sc_files
$ vi hello.txt
```



2) Command: i (Insert Mode)

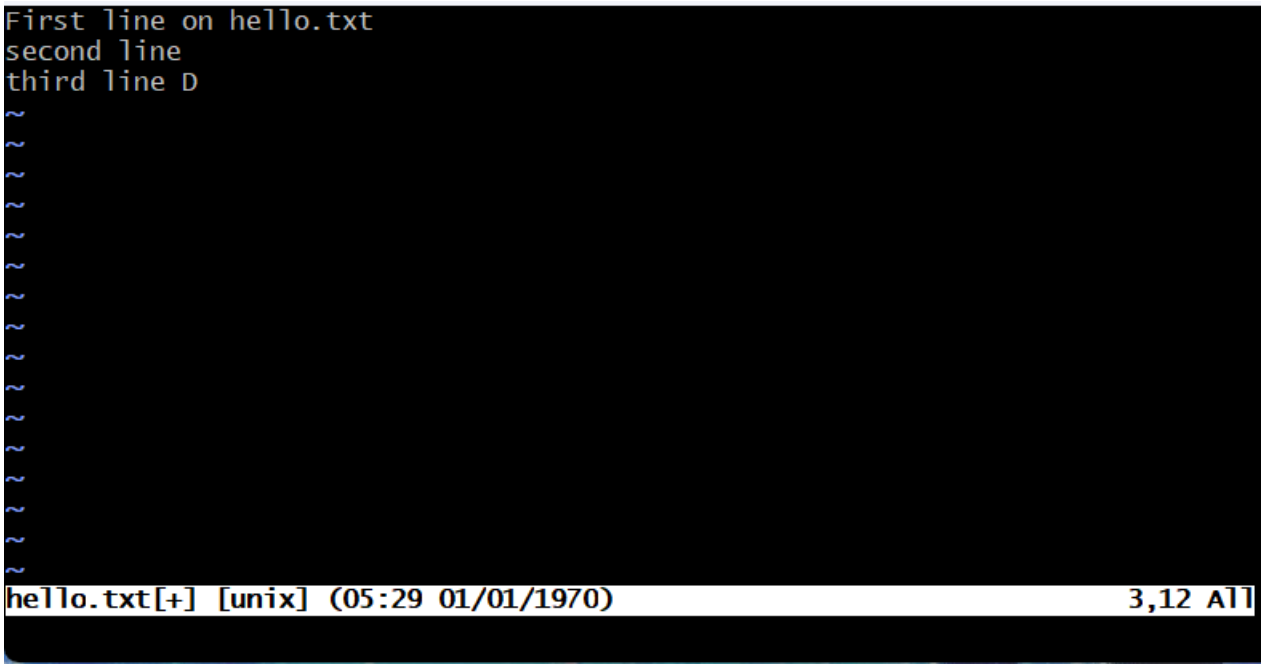
Description: Enters insert mode in Vim to allow text input.





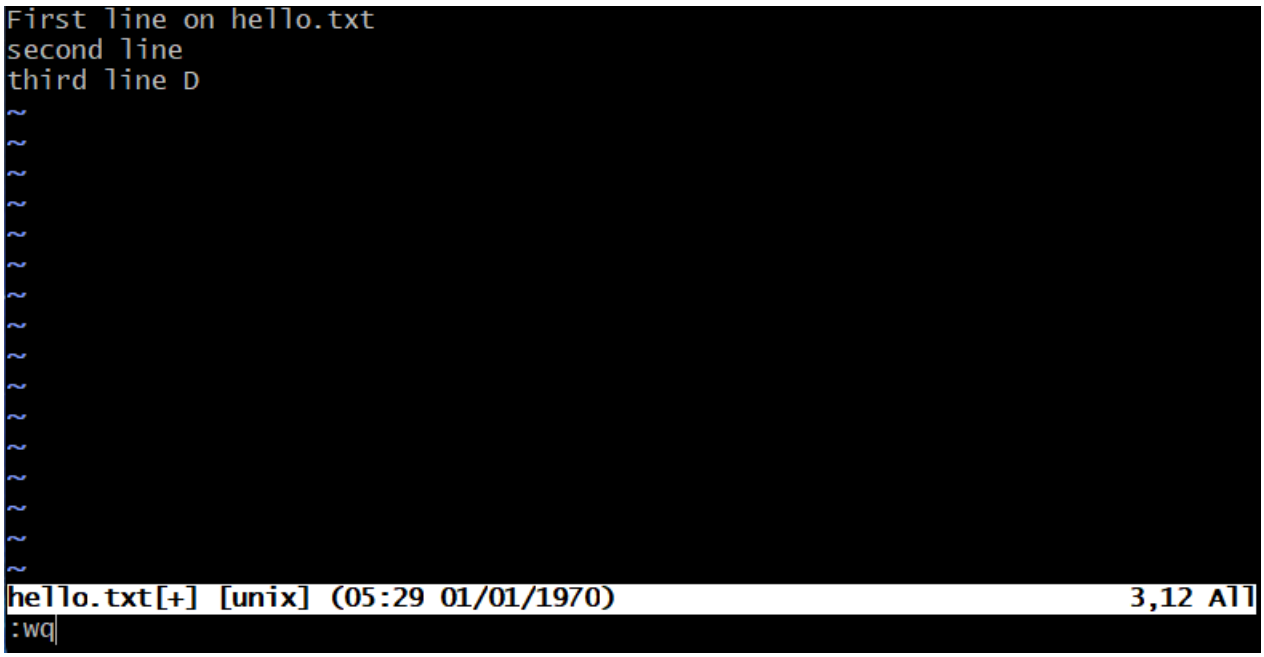
3) Command: esc

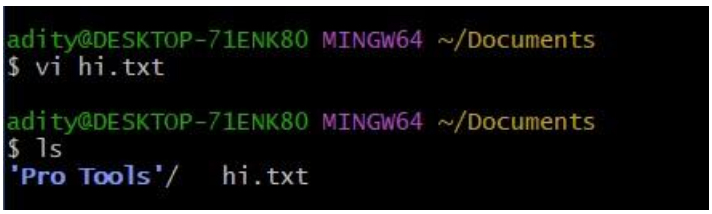
Description: Used to exit insert mode



4) Command: :wq

Description: Saves the changes and exits the Vim editor.





4. Git Commands

1. Command: git - - version

Description: The `git --version` command is used to check the installed version of Git on your system.

bash

```
Suhas@LAPTOP-GP09KAMN MINGW64 ~
$ git --version
git version 2.49.0.windows.1
```

2. Command: git init

Description: Initializes a new Git repository in the current directory.

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/Users/suhas/project_calci
$ git init
Initialized empty Git repository in C:/Users/Suhas/project_calci/.git/
```

3. Command: git status

Description: Displays the current status of the working directory and staging area.

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/Users/suhas/project_calci (master)
$ git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
```

4. Command: git add Test.c

Description: Add Test.c to the staging area in preparation for a commit.

```
Suhas@LAPTOP-GPO9KAMN MINGW64 /c/Users/suhas/project_calci (master)
$ git add index.html
warning: in the working copy of 'index.html', LF will be replaced by CRLF the
```

5. Command: git commit -m “add file one”

Description: Commits the stage changes with the message “add file one”.

```
Suhas@LAPTOP-GPO9KAMN MINGW64 /c/Users/suhas/project_calci (master)
$ git commit -m "Basic calculator HTML and JavaScript logic"
[master (root-commit) 0a99e1c] Basic calculator HTML and JavaScript
 1 file changed, 33 insertions(+)
 create mode 100644 index.html
```

6. Command: git log

Description: Display the commit history of the repository.

```
Suhas@LAPTOP-GPO9KAMN MINGW64 /c/Users/suhas/project_calci (master)
$ git log
commit 0a99e1c39d389db43ff4ca0a2216b55cf3467044 (HEAD -> master)
Author: suhasg1 <suhas.g@s.amity.edu>
Date:   Thu May 29 21:02:56 2025 +0530

    Basic calculator HTML and JavaScript logic
```

7. Command: git log --oneline

Description: For generating shorter commit ID.

```
Suhas@LAPTOP-GPO9KAMN MINGW64 /c/Users/suhas/project_calci (master)
$ git log --oneline
78cd05f (HEAD -> master) Final code of simple calculator
e74fce5 (footer-section) Footer section with copyright
5d549fb (js-validation) Input validation to prevent NaN operations
06613f1 (style-update) add styling for background and input layout
0a99e1c (CAL/master) Basic calculator HTML and JavaScript logic
```

8. Command: git remote add “Variable”

Description: To connect with the Users GitHub account.

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/Users/suhas/project_calci (master)
$ git remote add CAL https://github.com/Suhasg1/simple_calculator.git
```

9. Command: git remote

Description: To check the status of the repositories connected with the Users account.

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/Users/suhas/project_calci (master)
$ git remote
CAL
```

10. Command: git push -u “Variable” master

Description: To push all the files to the Users account.

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/Users/suhas/project_calci (master)
$ git push -u CAL master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 677 bytes | 677.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Suhasg1/simple_calculator.git
 * [new branch]      master -> master
branch 'master' set up to track 'CAL/master'.
```

11. Command: git merge “File_Name” -m “comment”

Description: To merge a branch with main branch.

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/Users/suhas/project_calci (master)
$ git merge style-update -m "Merge: style-update"
Updating 0a99e1c..06613f1
Fast-forward (no commit created; -m option ignored)
 index.html | 12 ++++++++++++
 1 file changed, 12 insertions(+)
```

12. Command: git clone

Description: To obtain a copy of an existing Git repository.

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/Users/suhas/test
$ git clone https://github.com/Suhasg1/project-1

Cloning into 'project-1'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), done.
```

13. Command: git diff

Description: To compare two files.

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/Users/suhas/proje
$ git diff 0a99e1c 78cd05f
diff --git a/index.html b/index.html
index 1d5870e..abfdb4a 100644
--- a/index.html
+++ b/index.html
@@ -1,8 +1,19 @@
-<!-- Initial calculator HTML -->
 <!DOCTYPE html>
 <html>
 <head>
   <title>Simple Calculator</title>
```

5.SCM Project

The project is to make a repository in GitHub, make 3 branches and merge it with the main branch and access all 4 team-mate's repositories, fork it, clone it, make some changes and merge them.

First, make your own repositories and make 3 branches and add files and merge with the main branch.

1) **Go to the directory on your computer**

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas
$ pwd
/c/SCM_Project/suhas
```

2) **Initialize the repo**

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas
$ git init
Initialized empty Git repository in C:/SCM_Project/suhas/.git/
```

3) **Create And add your file**

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (master)
$ vi main.cpp

Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (master)
$ git add .
warning: in the working copy of 'main.cpp', LF will be replaced by CRLF the ne
```

4) **Make a Initial commit**

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (master)
$ git commit -m "Basic structure"
[master (root-commit) 3137a89] Basic structure
 1 file changed, 44 insertions(+)
 create mode 100644 main.cpp
```

5) **Create 3 Branches**

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (master)
$ git branch feature/account

Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (master)
$ git branch feature/transaction

Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (master)
$ git branch feature/search_acc
```

6) **Checkout to 1st branch**

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (master)
$ git checkout feature/account
Switched to branch 'feature/account'
```

7) **Create and add file**

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (feature/account)
$ vi main.cpp
```

8) **Add and commit it**

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (feature/account)
$ git add .
warning: in the working copy of 'main.cpp', LF will be replaced by CRL

Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (feature/account)
$ git commit -m "Added Account creation feature"
[feature/account 66caceb] Added Account creation feature
 1 file changed, 26 insertions(+), 4 deletions(-)
```

9) **Checkout to 2nd branch**

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (feature/account)
$ git checkout feature/transaction
Switched to branch 'feature/transaction'
```

10) Create and add file

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (feature/transaction)
$ vi main.cpp
```

11) Add and commit it

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (feature/transaction)
$ git add .

Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (feature/transaction)
$ git commit -m "added depsit , withdraw , balance function "
[feature/transaction 5f0f8aa] added depsit , withdraw , balance function
 1 file changed, 80 insertions(+), 7 deletions(-)
```

12) Checkout to 3rd branch

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (feature/transaction)
$ git checkout feature/search_acc
Switched to branch 'feature/search_acc'
```

13) Add file in it

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (feature/search_acc)
$ vi main.cpp
```

14) Add and commit

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (feature/search_acc)
$ git add .

Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (feature/search_acc)
$ git commit -m "Added searh accound feature "
[feature/search_acc ab34c10] Added searh accound feature
 1 file changed, 96 insertions(+), 8 deletions(-)
```

15) Merge the 1st branch to master branch

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (master)
$ git merge feature/account
Updating 3137a89..66caceb
Fast-forward
 main.cpp | 30 ++++++++++++++++++++++++----
 1 file changed, 26 insertions(+), 4 deletions(-)
```

16) Merge it same for 2nd and 3rd branch

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas.g (master)
$ git merge feature/transaction
Updating ee47361..3b275fa
Fast-forward
 main.cpp | 57 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++---
 1 file changed, 54 insertions(+), 3 deletions(-)
```

17) Use "git log –oneloine" to see all commits

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas.g (master)
$ git log --oneline
38f9f0a (HEAD -> master, origin/master) Merged all files to main file
2a1b2f5 (origin/feature/search_acc, feature/search_acc) Added account search feature
3b275fa (origin/feature/transaction, feature/transaction) Added deposit,withdraw functions
ee47361 (origin/feature/account, feature/account) Added new account feature
112a113 simple structure
```

18) Using git remote to add Github Repository

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (master)
$ git remote add origin https://github.com/Suhasg1/SCM_project.git
```

19) Pushing all commits and files to Github repository

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/suhas (master)
$ git push -u origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 1.01 KiB | 1.01 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Suhasg1/SCM_project.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

(so now to give contribution to other team members we should make fork of their repo and after changes we have to send them a pull request)

6. Contribution :

-Prajwal

1) First we have to fork the repo in their repo

2) Go to the directory where to save the folder

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/prajwal
$ pwd
/c/SCM_Project/prajwal
```

3) Clone the forked repository

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/prajwal
$ git clone https://github.com/Suhasg1/SCM_fork1.git
Cloning into 'SCM_fork1'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 9 (delta 3), reused 6 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (9/9), done.
Resolving deltas: 100% (3/3), done.
```

4) Go into the repositories

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/prajwal
$ cd SCM_fork1

Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/prajwal/SCM_fork1 (master)
$
```

5) create new branch and checkout into it

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/prajwal/SCM_fork1 (master)
$ git checkout -b suhas
Switched to a new branch 'suhas'
```

6) Make changes to file

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/prajwal/SCM_fork1 (suhas)
$ vi main.cpp
```

7) Add edited file

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/prajwal/SCM_fork1 (suhas)
$ git add .
```

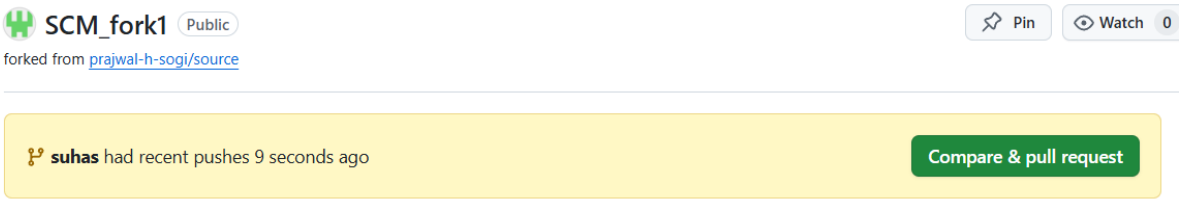
8) Commit after adding

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/prajwal/SCM_fork1 (suhas)
$ git commit -m "contribution by suhas "
[suhas 7fa8a2b] contribution by suhas
 1 file changed, 46 insertions(+), 108 deletions(-)
```

9) Push into forked repository

```
Suhas@LAPTOP-GP09KAMN MINGW64 /c/SCM_Project/prajwal/SCM_fork1 (suhas)
$ git push -u origin suhas
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.13 KiB | 1.13 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'suhas' on GitHub by visiting:
remote:      https://github.com/Suhasg1/SCM_fork1/pull/new/suhas
remote:
To https://github.com/Suhasg1/SCM_fork1.git
 * [new branch]      suhas -> suhas
branch 'suhas' set up to track 'origin/suhas'.
```

10) Create pull request in forked repository



11) team members will see and merge pull request

--- END ---