Subject Name: **Source Code Management**

Subject Code: **24CSE0106**

Cluster: **Alpha**

Department: **DCSE**

Submitted By:

Name: Vikrant Garg

2410990158

G02

Submitted To:

Dr. Renu Popli

Department of Computer Science & Engineering

Chitkara University Institute of Engineering and Technology Rajpura,

Index Task 1.1

Practical No.: 1

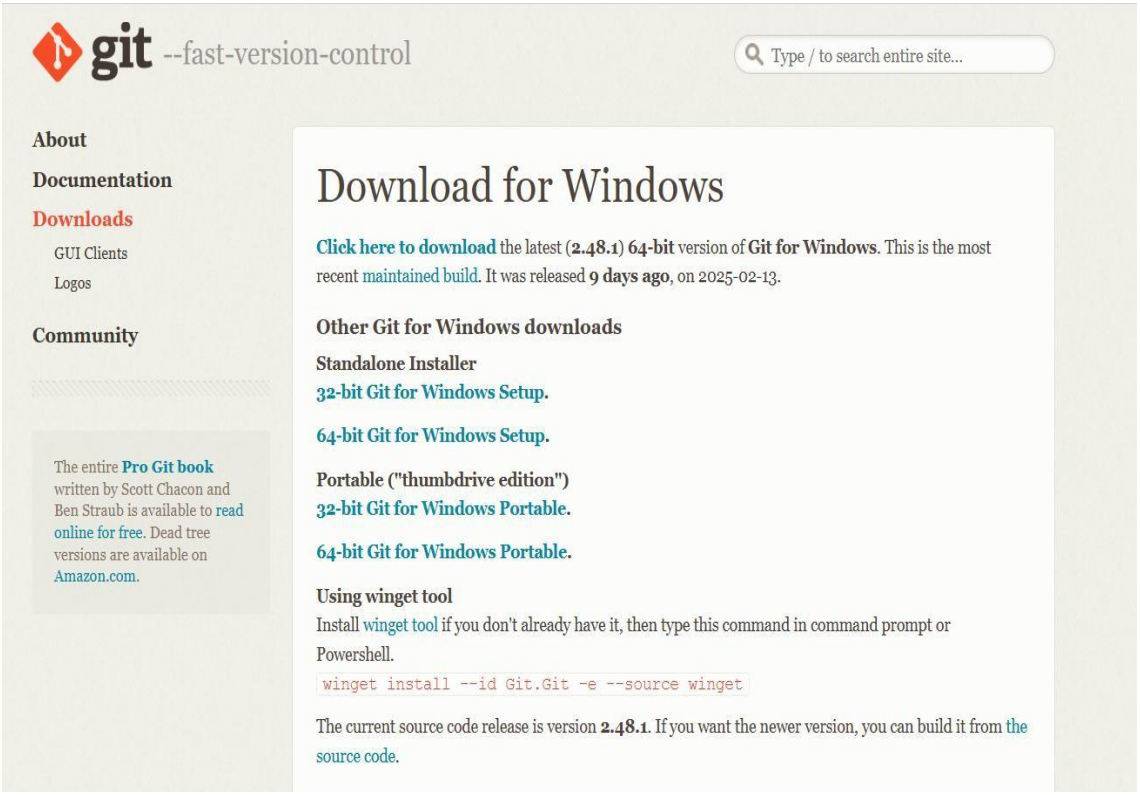
Aim: To install and configure Git Client on your local system.

Theory:

Git is a distributed version control system used to track changes in source code. This practical focuses on setting up Git on your local system for effective version control.

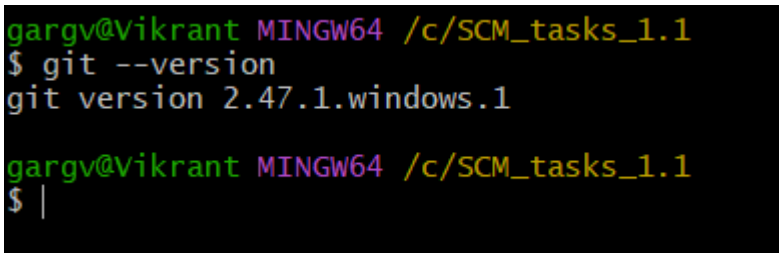
Procedure:

1. Download Git from [git-scm.com](git-scm.com) .

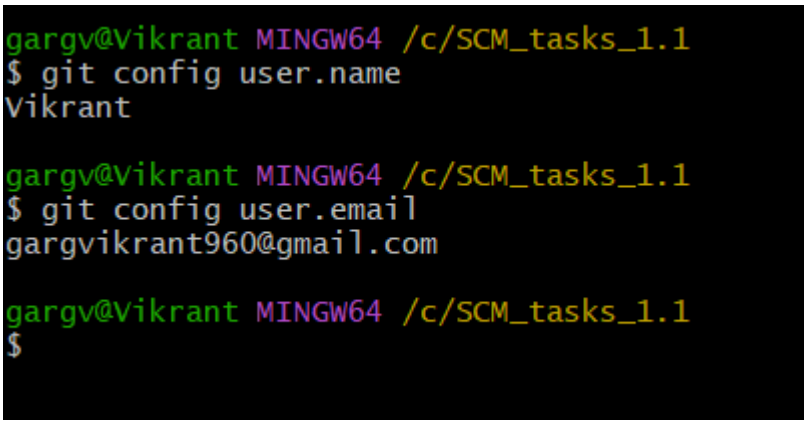


2. Install Git by following the setup wizard.

3. Open Git Bash and verify installation using the command: git --version.



4. Configure user details using the commands:
    - git config --global user.name "Your Name"
    - git config --global user.email "Your Email"



Practical No.: 2

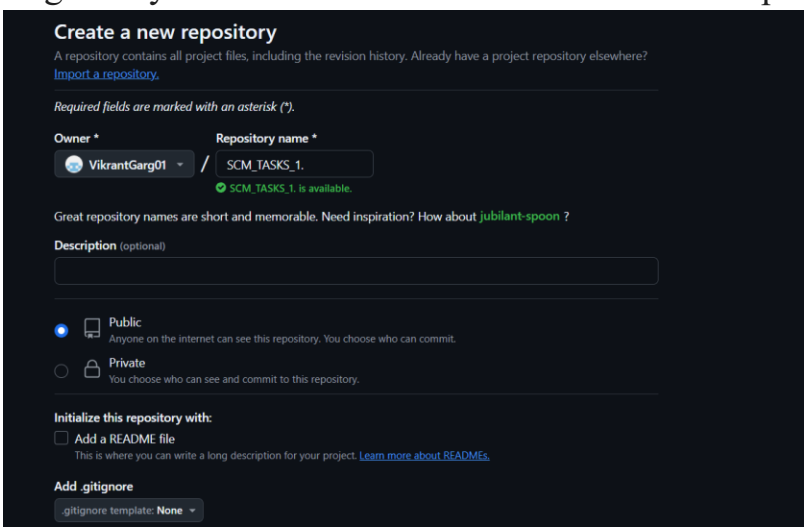
Aim: To add collaborators to a GitHub repository for collaborative work.

Theory:

Collaborators are individuals with write access to a repository. They can contribute to the project by pushing changes and merging pull requests.

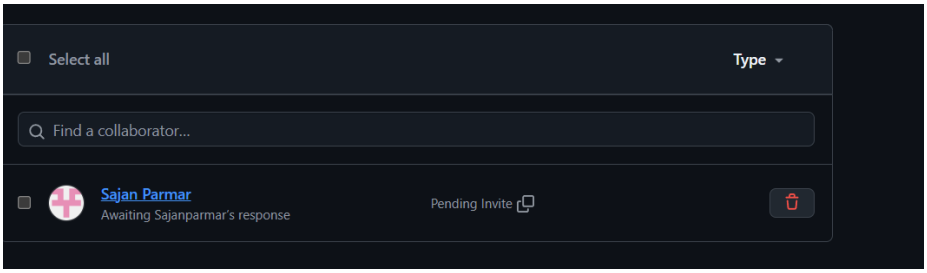
Procedure:

1. Log in to your GitHub account and create a new repository.

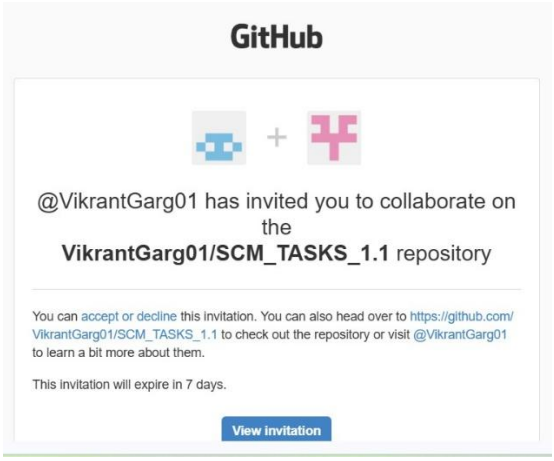


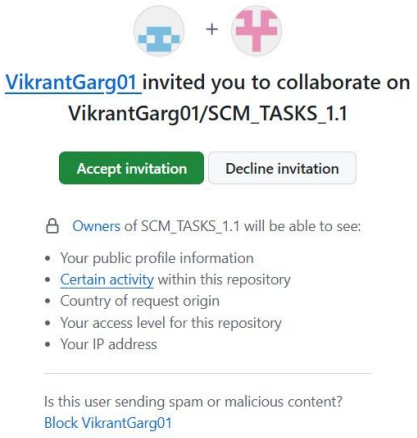
2. Navigate to Settings > Manage Access in the repository.

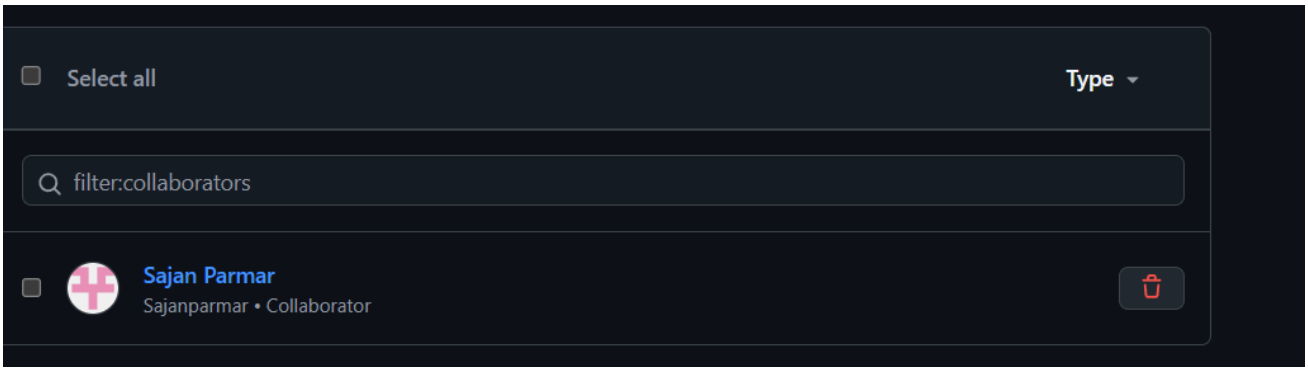
3. Add collaborators by their GitHub usernames.



4. Collaborators will receive an invitation email, which they must accept.







Practical No.: 3

Aim: To merge two branches within a Git repository.

Theory:

Merging branches in Git allows you to combine changes from one branch into another. It is a fundamental process in collaborative workflows, ensuring all contributions are integrated into a single codebase.

Procedure:

1. Create a new branch and switch to it: • git checkout -b new-branch

```
gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (main)
$ git checkout -b duplicate
Switched to a new branch 'duplicate'

gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (duplicate)
$ git branch -a
* duplicate
  main
  remotes/origin/main

gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (duplicate)
$
```

2. Make changes to a file in the new branch and commit them:
    - echo "New content" > file.txt
    - git add file.txt
    - git commit -m "Add changes in new branch"

```
gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (duplicate)
$ echo "git allows you to combine branches">> note.txt

gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (duplicate)
$ git add note.txt
warning: in the working copy of 'note.txt', LF will be replaced by CRLF the next time Git touches it

gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (duplicate)
$ git commit -m "Add changes in new branch"
[duplicate 5ec289c] Add changes in new branch
 1 file changed, 1 insertion(+)
 create mode 100644 note.txt

gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (duplicate)
$
```

3. Switch back to the main branch:
    - git checkout main

```
gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (duplicate)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (main)
$
```

4. Modify another file in the main branch and commit the changes:
    - echo "Main branch changes" > another-file.txt
    - git add another-file.txt
    - git commit -m "Modify file in main branch"

```
gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (main)
$ echo "We will merge main and duplicate branch">> do.txt

gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (main)
$ git add do.txt
warning: in the working copy of 'do.txt', LF will be replaced by CRLF the next time Git touches it

gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (main)
$ git commit -m "Add do.txt"
[main 67426a9] Add do.txt
 1 file changed, 1 insertion(+)
 create mode 100644 do.txt

gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (main)
$
```

5. Merge the new branch into the main branch:
    - git merge new-branch

```
Merge made by the 'ort' strategy.
 note.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 note.txt
```

Git log :

```
gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (main)
$ git log
commit 67426a93c62a75ba596f9f07116b0d3fe1028899 (HEAD -> main)
Author: Vikrant <gargvikrant960@gmail.com>
Date:   Thu Feb 27 19:32:37 2025 +0530

    Add do.txt

commit 43e82a4bc527ca52cb481e736ecf814f4f2cc531 (origin/main)
Author: Vikrant Garg <gargvikrant960@gmail.com>
Date:   Thu Feb 27 19:00:13 2025 +0530

    Initial commit

gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (main)
$
```

Practical No.: 4

Aim: To demonstrate push and pull operations in Git.

Theory:

Push transfers committed changes from the local repository to the remote repository, while pull retrieves updates from the remote repository.

Procedure:

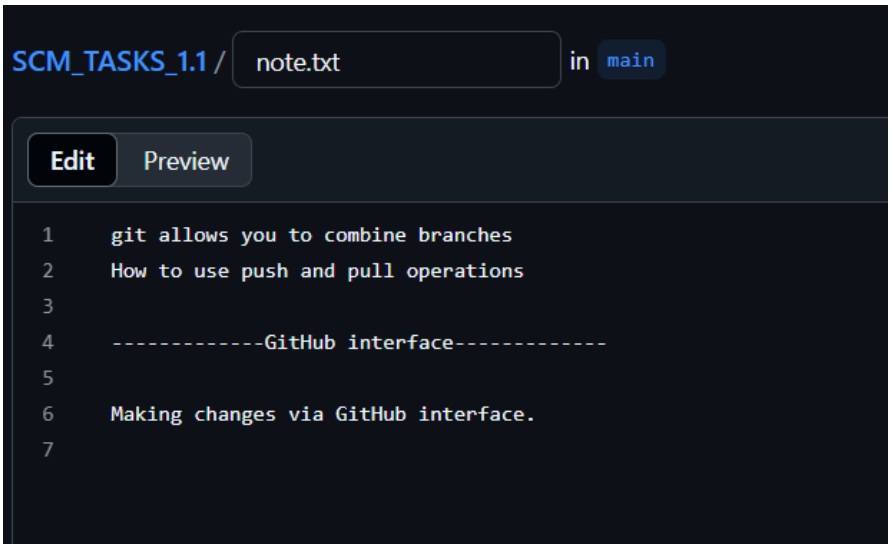
1. Make changes in the local repository and commit them.

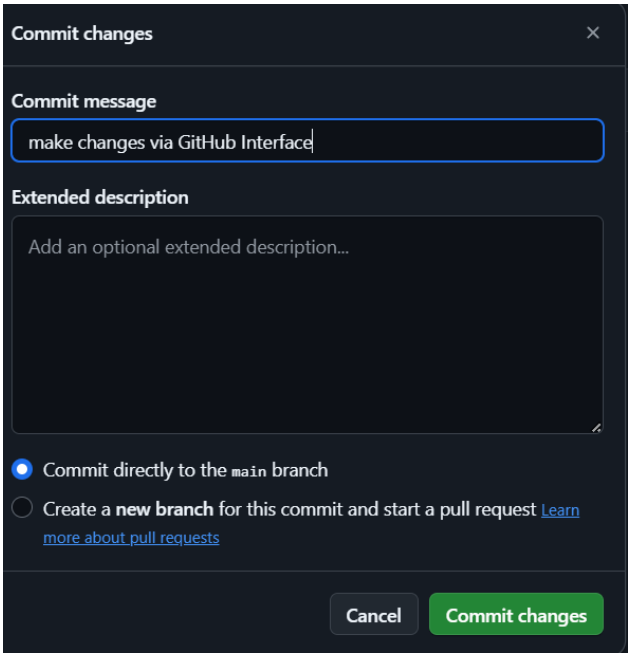
```
gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (main)
$ echo "How to use push and pull operations">>note.txt

gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (main)
$ git add .
warning: in the working copy of 'note,txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'note.txt', LF will be replaced by CRLF the next time Git touches it

gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (main)
$ git commit -m "add text in note.txt"
[main 6b3c1da] add text in note.txt
 2 files changed, 2 insertions(+)
 create mode 100644 note,txt
 create mode 100644 note.txt

gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (main)
$
```

2. Push the changes to the remote repository using git push.

```
gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (main)
$ git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (7/7), 634 bytes | 634.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/VikrantGarg01/SCM_TASKS_1.1.git
   43e82a4..6b3c1da  main -> main
branch 'main' set up to track 'origin/main'.

gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (main)
$
```

3. Make changes directly on the remote repository (e.g., via GitHub interface).





4. Pull the changes to the local repository using git pull.

```
gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (main)
$ git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 1007 bytes | 67.00 KiB/s, done.
From https://github.com/VikrantGarg01/SCM_TASKS_1.1
 * branch            main       -> FETCH_HEAD
   6b3c1da..f4527e2  main       -> origin/main
Updating 6b3c1da..f4527e2
Fast-forward
 note.txt | 5 +++++
 1 file changed, 5 insertions(+)

gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (main)
$
```

Git log :

```
gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (main)
$ git log
commit f4527e2747b5affb3e3152fe5b91fe3f48a23482 (HEAD -> main, origin/main)
Author: Vikrant Garg <gargvikrant960@gmail.com>
Date:   Thu Feb 27 19:43:49 2025 +0530

    make changes via GitHub Interface

commit 6b3c1da7fbc5fa6f72c2e42ae06dc2c5fa363e52
Author: Vikrant <gargvikrant960@gmail.com>
Date:   Thu Feb 27 19:38:33 2025 +0530

    add text in note.txt

commit 67426a93c62a75ba596f9f07116b0d3fe1028899
Author: Vikrant <gargvikrant960@gmail.com>
Date:   Thu Feb 27 19:32:37 2025 +0530

    Add do.txt

commit 43e82a4bc527ca52cb481e736ecf814f4f2cc531
Author: Vikrant Garg <gargvikrant960@gmail.com>
Date:   Thu Feb 27 19:00:13 2025 +0530

    Initial commit

gargv@Vikrant MINGW64 /c/SCM_tasks_1.1 (main)
$
```