

Subject Name: Source Code Management

Subject Code: 22CS003

Session: 2022-23

Department: DCSE

List of

S · N o	Progra m Title	P a g e N o ·
1	Setting up of Git Client	

CHITKARA
UNIVERSITY



Programs

Submitted By:

Bhuvesh Mittal

2210991450

G5-A

Submitted To:

Dr. Garima Chopra

2	Setting up GitHub Account	
3	Generate logs	
4	Create and visualize branches	
5	Git life cycle description	
6	Add collaborators on GitHub Repo	
7	Fork and Commit	
8	Merge and Resolve conflicts created due to own activity and collaborators activity.	
9	Reset and revert	

Git

Git is a distributed version control system that tracks changes in any set of computer files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non – linear workflows.

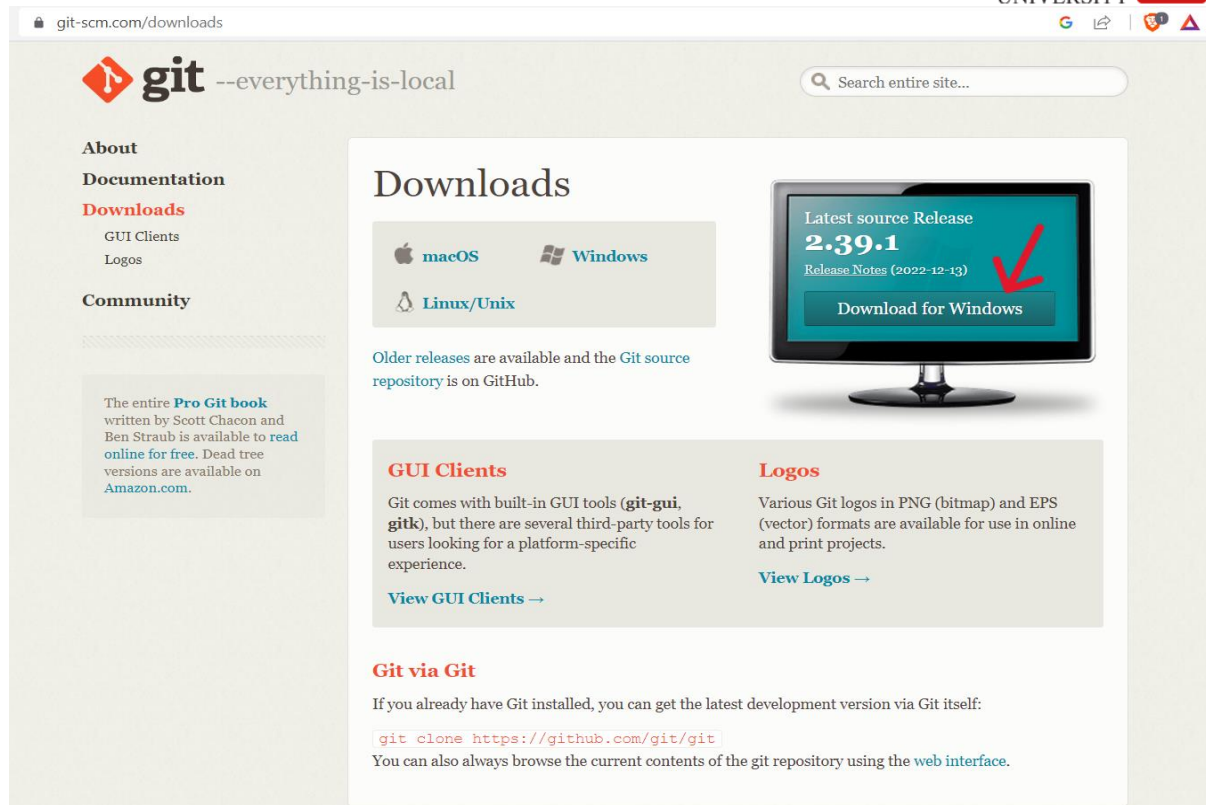
Advantages

- Performance
- Security
- Flexibility
- Wide Acceptance
- Quality open-source project

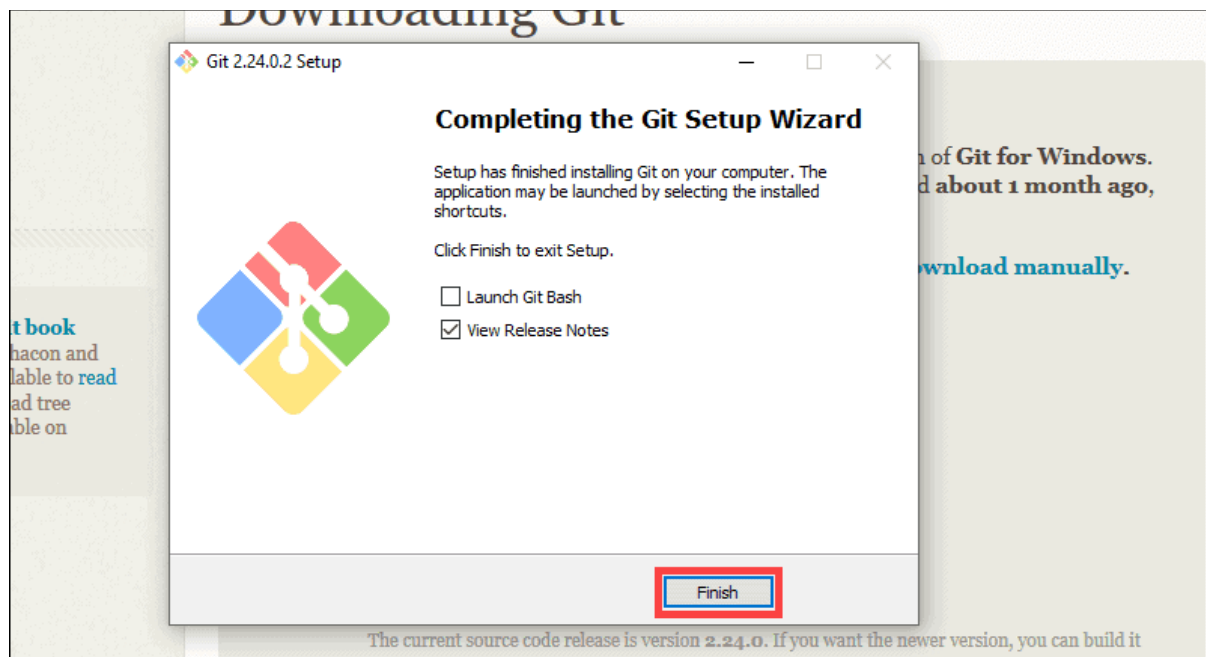
Experiment 1. Setting up of Git Client

Step 1. Open the git website: <https://git-scm.com/downloads>

Step 2. Click on *Download for Windows*



Step 3. Click on *finish* for complete installation.



Step 4. Set up **git** using the following commands:

- `$ git config --global user.email "***@gmail.com"`
- `$ git config --global user.name "***"`

```
bhuve@mittal MINGW64 /d/scm-lab2023 (master)
$ git config --global user.email "bhuves[REDACTED].com"

bhuve@mittal MINGW64 /d/scm-lab2023 (master)
$ git config --global user.name "Mi[REDACTED]"

bhuve@mittal MINGW64 /d/scm-lab2023 (master)
$ |
```

Finally, the git client is successfully setup.

To unset the git client, following commands will be used:

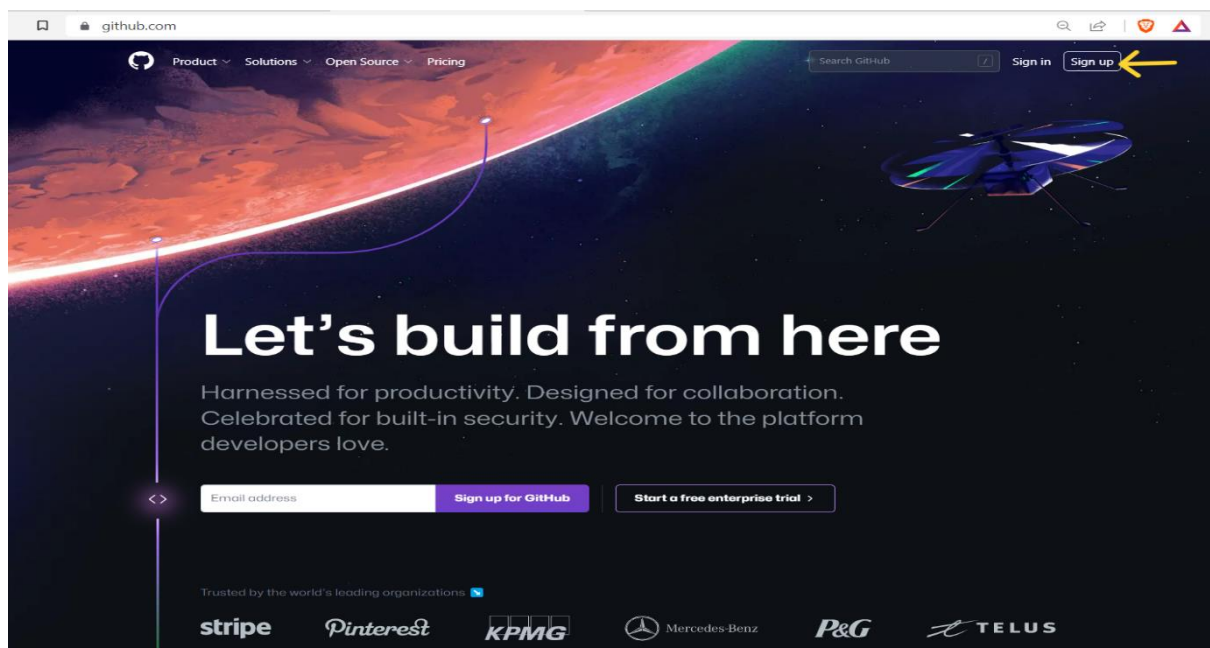
- *\$ git config --global --unset user. Email*
- *\$ git config --global --unset user.name*

Experiment 2. Setting up of Git Hub account

GitHub is an internet hosting service for software development and version control using git. It provides the distributed version control of git plus access control, bug tracking, software feature

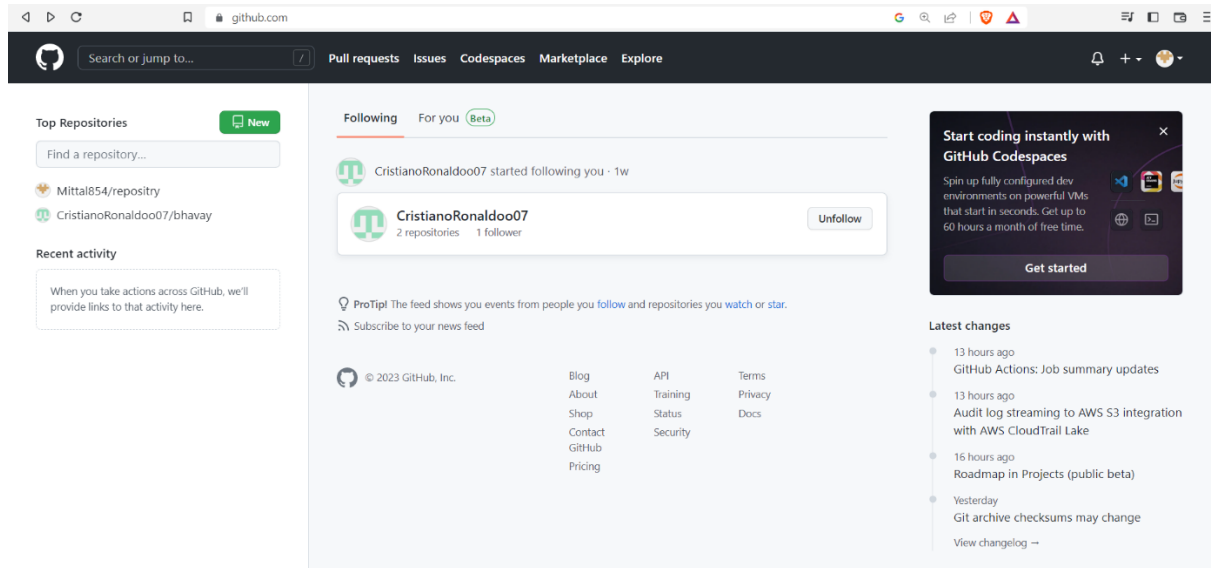
Requests, task managements, continuous integration, and wikis for every project.

Step 1. Browse the GitHub website: <https://github.com/> & click on **sign up**.



Step 2. Enter your personal details for the account.

Step 3. Successfully, account is created.



Experiment 3. Generate Logs

It displays all the commits being made in that repository in multiple lines along with the commit id, author name, date and commit message. Logs helps us to check the changes made in code or files and by whom. It also contains time at which change was made.

The command to generate logs is:

➤ \$ git log

```
bhuve@mittal MINGW64 /d/scm-lab2023 (master)
$ git commit -m "Hello"
[master 216ae47] Hello
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 b.txt

bhuve@mittal MINGW64 /d/scm-lab2023 (master)
$ vi b.txt

bhuve@mittal MINGW64 /d/scm-lab2023 (master)
$ git add b.txt
warning: in the working copy of 'b.txt', LF will be replaced by CRLF the next time Git touches it

bhuve@mittal MINGW64 /d/scm-lab2023 (master)
$ git commit -m "commit committed"
[master 95fa442] commit committed
1 file changed, 3 insertions(+)

bhuve@mittal MINGW64 /d/scm-lab2023 (master)
$ git log
commit 95fa442ba3c96fa7de9bfa9902d8945f083bdb4d (HEAD -> master)
Author: Mittal854 <bhuveshmittal854@gmail.com>
Date: Wed Feb 1 15:32:52 2023 +0530

    commit committed

commit 216ae478c03d3b5b497d830243d831caf66b8db7
Author: Mittal854 <bhuveshmittal854@gmail.com>
Date: Wed Feb 1 15:27:35 2023 +0530

    Hello

commit 9f59baba2cd488b5af2bd922a677fb14d46b47e7
Author: Mittal854 <bhuveshmittal854@gmail.com>
Date: Thu Jan 12 19:04:32 2023 +0530

    hiii

commit c69984372df9987fc481fdc7ce5d8764fccd8aa7
Author: Mittal854 <bhuveshmittal854@gmail.com>
Date: Tue Jan 10 10:34:09 2023 +0530

    Hello
```

Experiment 4. Create & visualize Branches

In Git, a branch is a new/separate version of the main repository. Branches allow you to work on different parts of a project without impacting the main branch. When the work is

complete, a branch can be merged with the main project. You can even switch between branches and work on different projects without them interfering with each other.

- **For creating a new branch:** git branch “name of the branch”
- **To change the present working branch:** git checkout “name of the branch”
- **To check how many branches:** git branch

```
bhuve@mittal MINGW64 /d/scm-lab2023 (bhuvesh)
$ git branch "online"

bhuve@mittal MINGW64 /d/scm-lab2023 (bhuvesh)
$ git checkout "online"
Switched to branch 'online'

bhuve@mittal MINGW64 /d/scm-lab2023 (online)
$ git branch
  bhuvesh
  master
* online
```

Visualization branches:

To visualize, we have to create a new file in the new branch “online” instead of the master branch. After this we have to do three step architecture i.e., working directory, staging area and git repository.



```

bhuve@mittal MINGW64 /d/scm-lab2023 (online)
$ git branch
  bhuvesh
  master
* online

bhuve@mittal MINGW64 /d/scm-lab2023 (online)
$ touch f6.txt

bhuve@mittal MINGW64 /d/scm-lab2023 (online)
$ git add --a

bhuve@mittal MINGW64 /d/scm-lab2023 (online)
$ git status
On branch online
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   f6.txt

bhuve@mittal MINGW64 /d/scm-lab2023 (online)
$ git commit -m "visualized branch"
[online fc0867e] visualized branch
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 f6.txt

bhuve@mittal MINGW64 /d/scm-lab2023 (online)
$ git log
commit fc0867effc5344fc43ee292dadda0ea121b71a8e (HEAD -> online)
Author: Mittal854 <bhuveshmittal854@gmail.com>
Date:   Wed Feb 1 16:33:27 2023 +0530

    visualized branch

commit 95fa442ba3c96fa7de9bfa9902d8945f083bdb4d (master, bhuvesh)
Author: Mittal854 <bhuveshmittal854@gmail.com>
Date:   Wed Feb 1 15:32:52 2023 +0530

    commit committed

commit 216ae478c03d3b5b497d830243d831caf66b8db7
Author: Mittal854 <bhuveshmittal854@gmail.com>
Date:   Wed Feb 1 15:27:35 2023 +0530

    Hello

commit 9f59baba2cd488b5af2bd922a677fb14d46b47e7
Author: Mittal854 <bhuveshmittal854@gmail.com>
Date:   Thu Jan 12 19:04:32 2023 +0530

    hiii

commit c69984372df9987fc481fdc7ce5d8764fccd8aa7
Author: Mittal854 <bhuveshmittal854@gmail.com>
Date:   Tue Jan 10 10:34:09 2023 +0530

    Hello

bhuve@mittal MINGW64 /d/scm-lab2023 (online)
$

```



```

bhuve@mittal MINGW64 /d/scm-lab2023 (online)
$ git checkout "master"
Switched to branch 'master'

bhuve@mittal MINGW64 /d/scm-lab2023 (master)
$ git log
commit 95fa442ba3c96fa7de9bfa9902d8945f083bdb4d (HEAD -> master, bhuvesh)
Author: Mittal854 <bhuveshmittal854@gmail.com>
Date:   Wed Feb 1 15:32:52 2023 +0530

    commit committed

commit 216ae478c03d3b5b497d830243d831caf66b8db7
Author: Mittal854 <bhuveshmittal854@gmail.com>
Date:   Wed Feb 1 15:27:35 2023 +0530

    Hello

commit 9f59baba2cd488b5af2bd922a677fb14d46b47e7
Author: Mittal854 <bhuveshmittal854@gmail.com>
Date:   Thu Jan 12 19:04:32 2023 +0530

    hiii

commit c69984372df9987fc481fdc7ce5d8764fccd8aa7
Author: Mittal854 <bhuveshmittal854@gmail.com>
Date:   Tue Jan 10 10:34:09 2023 +0530

    Hello

bhuve@mittal MINGW64 /d/scm-lab2023 (master)
$ |

```

Experiment 5. Git life cycle description

When a directory is made a git repository, there are mainly 3 states which make the essence of Git Version Control System. The three states are –

- Working Directory
- Staging Area
- Git Directory

1. Working Directory

Whenever we want to initialize our local project directory to make it a git repository, we use the *git init* command. After this command, git becomes aware of the files in the project although it doesn't track the files yet. The files are further tracked in the staging area.

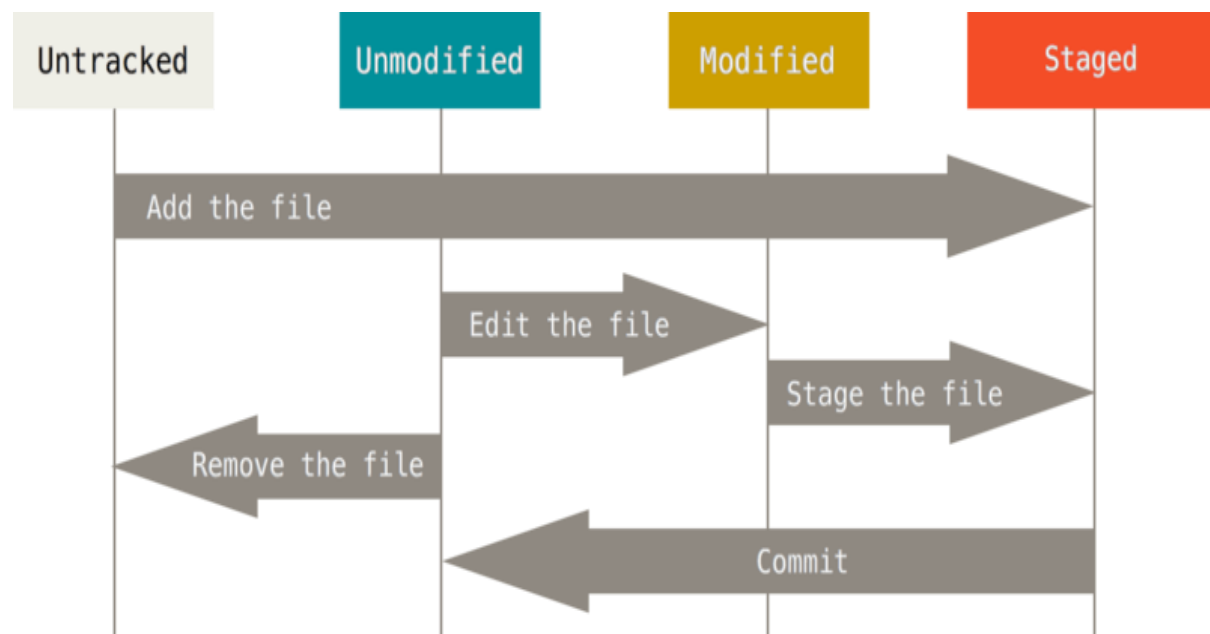
2. Staging Area

Staging area is the playground where you group, add and organise the files to be committed to Git for tracking their versions.

3. Git directory

Now that the files to be committed are grouped and ready in the staging area, we can commit these files. So, we commit this group of files along with a commit message explaining what is the commit about. Apart from commit message, this step also records the author and time of the commit. Now, a snapshot of the files in the commit is recorded by Git.

The information related to this commit is stored in the Git directory.



```

bhuve@mittal MINGW64 /d/scm-lab2023
$ git init
Initialized empty Git repository in D:/scm-lab2023/.git/

bhuve@mittal MINGW64 /d/scm-lab2023 (master)
$ git status
On branch master

No commits yet

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

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

bhuve@mittal MINGW64 /d/scm-lab2023 (master)
$ git add --a
warning: in the working copy of 'a.txt.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'b.txt', LF will be replaced by CRLF the next time Git touches it

bhuve@mittal MINGW64 /d/scm-lab2023 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   a.txt.txt
        new file:   b.txt

bhuve@mittal MINGW64 /d/scm-lab2023 (master)
$ git commit -m "hii"
[master (root-commit) 314ad64] hii
 2 files changed, 5 insertions(+)
 create mode 100644 a.txt.txt
 create mode 100644 b.txt

bhuve@mittal MINGW64 /d/scm-lab2023 (master)
$ git log
commit 314ad64f79d6dcfc3ac7fc64ebc201019078ca0c (HEAD -> master)
Author: Mittal854 <bhuveshmittal854@gmail.com>
Date:   Wed Feb 1 16:55:40 2023 +0530

hii

```