

## Experiment No. 01

**Aim:** Setting up of Git Client

### Theory:

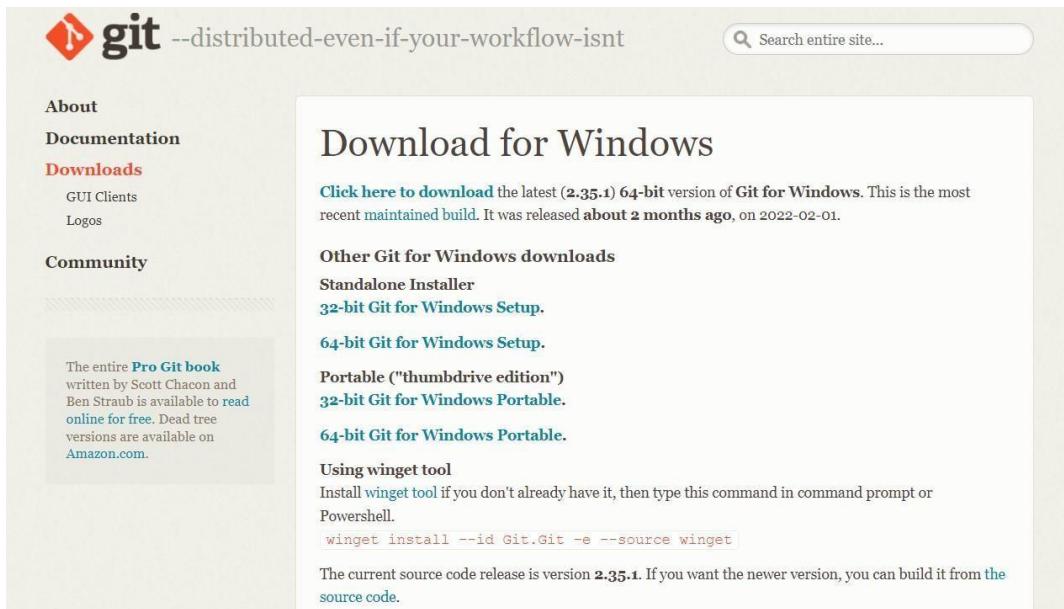
GIT → It is basically used for pushing and pulling of code. We can use git and git-hub parallelly to work with multiple members or individually. We can make, edit, recreate, copy or download any code on git hub using git.

What is GIT? → It's a Version Control System (VCS) → It is a software, or we can say a server by which we are able to track all the previous changes in the code.

Advantages of GIT →

**Procedure:** We can install Git on Windows, using the most official build which is available for download on the GIT's official website or by just typing (scmgit) on any search engine. We can go on <https://git-scm.com/download/win> and can select the platform and bitversion to download. And after clicking on your desired bit-version or ios it will start downloading automatically.

### Snapshots of download:



The screenshot shows the official Git website (<https://git-scm.com/>) with the focus on the 'Download for Windows' section. The page header includes the Git logo and the tagline '--distributed-even-if-your-workflow-isnt'. A search bar is at the top right. On the left, there's a sidebar with links for 'About', 'Documentation', 'Downloads' (which is highlighted in red), and 'Community'. Under 'Downloads', there are links for 'GUI Clients' and 'Logos'. A note about the 'Pro Git book' is also present. The main content area features a large heading 'Download for Windows' and a paragraph encouraging users to click to download the latest 64-bit version. Below this, there are sections for 'Other Git for Windows downloads' (including 'Standalone Installer', '32-bit Git for Windows Setup.', '64-bit Git for Windows Setup.', 'Portable ("thumbdrive edition")', '32-bit Git for Windows Portable.', and '64-bit Git for Windows Portable.'), 'Using winget tool' (with a command example), and a note about the current source code release being version 2.35.1.

Name	Date modified	Type	Size
Git Bash	16-03-2022 08:51	Shortcut	2 KB
Git CMD	16-03-2022 08:51	Shortcut	2 KB
Git FAQs (Frequently Asked Questions)	16-03-2022 08:51	Internet Shortcut	1 KB
Git GUI	16-03-2022 08:51	Shortcut	2 KB
Git Release Notes	16-03-2022 08:51	Shortcut	2 KB

```
MINGW64:/c/Users/Surface/Desktop/Demo_SCM
Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/Demo_SCM (master)
$
```

## Experiment No. 02

**Aim:** Setting up GitHub Account

### Theory:

**What is GitHub** -> GitHub is a website and cloud-based service (client) that helps an individual or a developer to store and manage their code. We can also track as well as control changes to our or public code.

**Advantages of GitHub** -> GitHub's has a user-friendly interface and is easy to use .We can connect the git-hub and git but using some commands shown below in figure 001. Without GitHub we cannot use Git because it generally requires a host and if we are working for a project, we need to share it with our team members, which can only be done by making a repository. Additionally , anyone can sign up and host a public code repository for free, which makes GitHub especially popular with open-source projects.

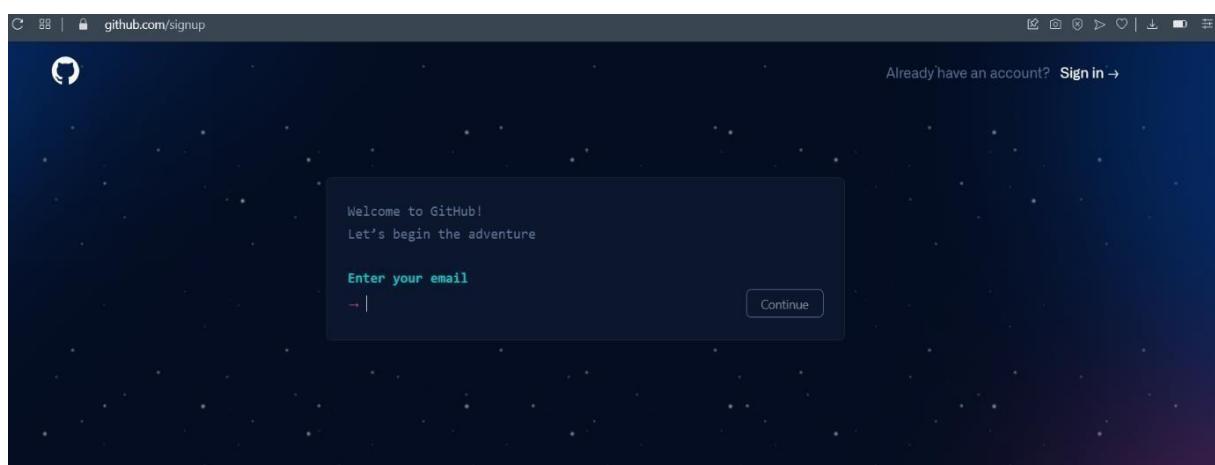
### Procedure:-

#### Step1 :-

Google (any search engine)  
Search for git-hub or (<https://github.com/signup>).

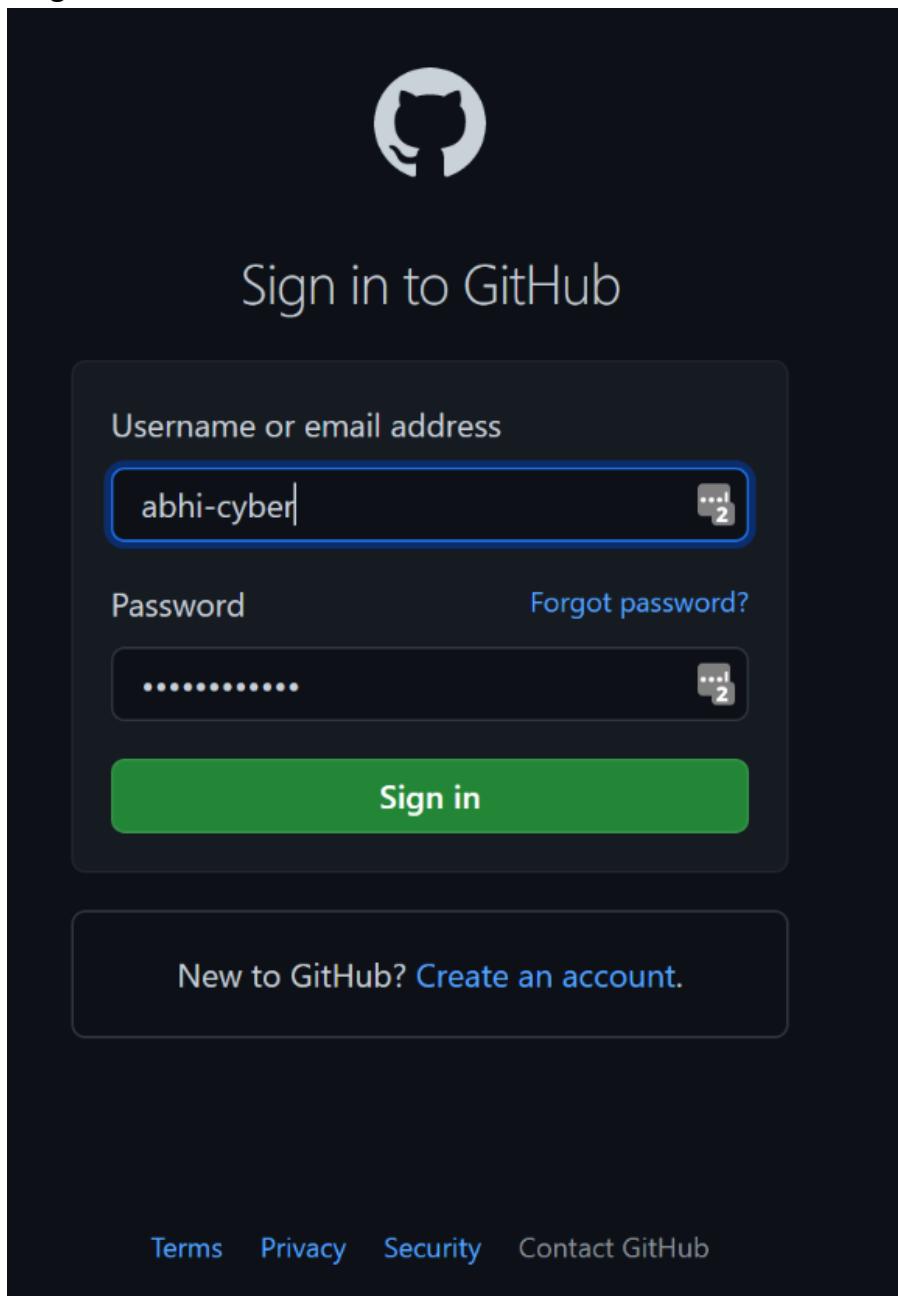
#### Step2 :-

### Snapshots –



After visiting the link this type of interface will appear, if you already have account you can sign in and if not you can create.

**Sign in into GitHub :-**



## Interface of GitHub :-

The screenshot shows a GitHub profile for the user 'abhi-cyber'. The profile picture is a circular image of a person with a camouflage beanie and a dark jacket. The bio section contains a message from the user and a list of interests and facts:

Hey, I'm Abhiraj! 🙋

- 💻 I'm currently working on Improving my programming skills.
- 🐍 I'm currently learning DSA from Udemy.
- 🤝 I'm looking to collaborate on projects related to JS/React.
- 💬 Ask me about how to start as a programmer.
- ✉️ How to reach me: [Linkedin](#)
- 😄 Pronouns: He/His
- ⚡ Fun fact: I love to spend time with my friends.
- 💡 Note: To look at some of my work do visit my personal [website](#).

Below the bio, there are sections for 'Popular repositories' and 'Customize your pins'. The 'Popular repositories' section shows four repositories: 'Portfolio' (Public), 'Free-Education' (Public), 'Demo\_SCM' (Public), and 'JS\_Basics' (Public). The 'Free-Education' repository has a description: "This provides links to free resources and books where one can find all their best resources in one place without having to browse here and there."

## To link GitHub account with Git bash –

### For username:-

```
git config --global user.name "username in git-hub"
```

### For user email:-

```
git config --global user.email "your email in git-hub"
```

### To verify:-

```
git config user.name
```

```
git config user.email
```

**Snapshot :-**

**Theory: -**

**Logs ->** Logs are nothing but the history which we can see in git by using the code git log. It contains all the past commits, insertions and deletions in it which we can see any time.

**Why logs ->** Logs helps to check that what were the changes in the code or any other file and by whom. It also contains the number of insertions and deletions including at which time it was changed.

**Snapshots –**

```
MINGW64:/c/Users/Surface/Desktop/Demo_SCM
Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/Demo_SCM (master)
$ git log
commit 99f53016c3fbf773078f76144cd0e2392ac86c2b (HEAD -> master, origin/master,
origin/HEAD)
Author: abhi-cyber <abhirajchattrath@gmail.com>
Date:   Wed Mar 16 14:06:18 2022 +0530

    Added college group

commit 32eef1b37a0316a6da77d8bb849f026e0a0c247b
Author: abhi-cyber <abhirajchattrath@gmail.com>
Date:   Wed Mar 16 14:05:18 2022 +0530

    Added college name

commit baa426699fb4e3611702bcc7de03cee1c6319ca8
Author: abhi-cyber <abhirajchattrath@gmail.com>
Date:   Wed Mar 16 14:04:26 2022 +0530

    Added Name

commit f1f7f8a75488b9fba7af76f3dce813bbc79c9662
Author: abhi-cyber <abhirajchattrath@gmail.com>
Date:   Wed Mar 16 14:03:42 2022 +0530
```

## Experiment No. 04

**Aim:** Create and visualize branches

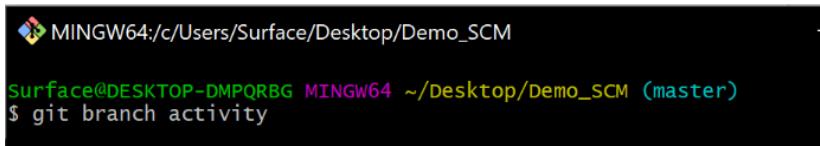
### Create branches:-

The main branch in git is called as master branch. But we can make branches out of this main master branch. All the files present in master can be shown in branch but the file which are created in branch are not shown in master branch. We can also merge both the parent (master) and child (other branches).

### Syntax: -

1. For creating a new branch. git  
branch name of branch

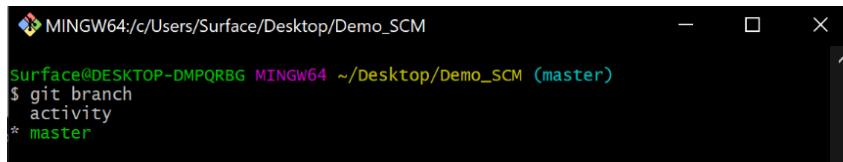
### Snapshots –



```
MINGW64:/c/Users/Surface/Desktop/Demo_SCM
Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/Demo_SCM (master)
$ git branch activity
```

2. We can also check how many branches we have.

git branch

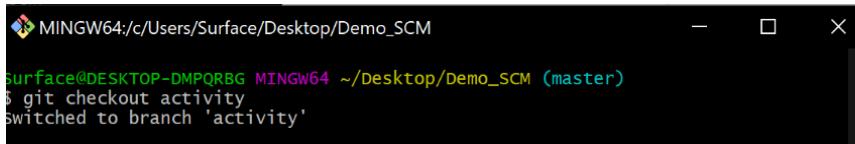


```
MINGW64:/c/Users/Surface/Desktop/Demo_SCM
Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/Demo_SCM (master)
$ git branch
activity
* master
```

3. To change the present working branch.

git checkout name of branch.

### Snapshots –



```
MINGW64:/c/Users/Surface/Desktop/Demo_SCM
surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/Demo_SCM (master)
$ git checkout activity
switched to branch 'activity'
```

### Visualizing branches :-

To visualize I have created a new file in a new branch activity 1 instead of master branch.

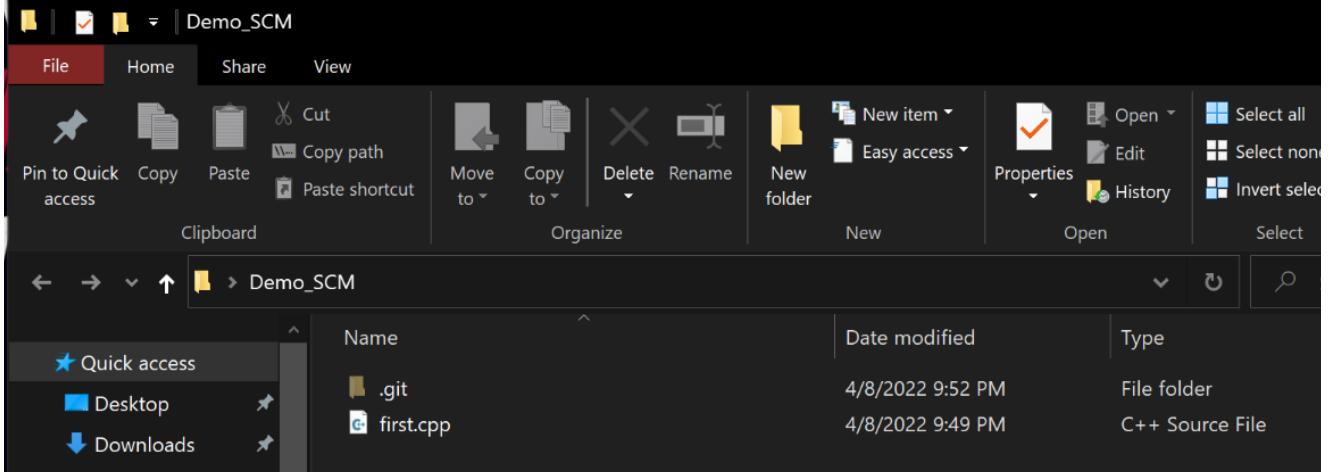
```
MINGW64:/c/Users/Surface/Desktop/Demo_SCM
Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/Demo_SCM (master)
$ pwd
/c/users/surface/Desktop/Demo_SCM

Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/Demo_SCM (master)
$ git branch features1

Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/Demo_SCM (master)
$ git branch
  feature
  features1
* master

Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/Demo_SCM (master)
$ git checkout features1
switched to branch 'features1'

surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/Demo_SCM (features1)
$
```



Name	Date modified	Type
.git	4/8/2022 9:52 PM	File folder
first.cpp	4/8/2022 9:49 PM	C++ Source File

After this I have done the 3 step architecture which is tracking the file , send it to staging area and finally we can role back to any previously saved version of this file.

```

MINGW64:/c/Users/Surface/Desktop/Demo_SCM

Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/Demo_SCM (features1)
$ git status
on branch features1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   first.cpp

no changes added to commit (use "git add" and/or "git commit -a")

Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/Demo_SCM (features1)
$ git add --a

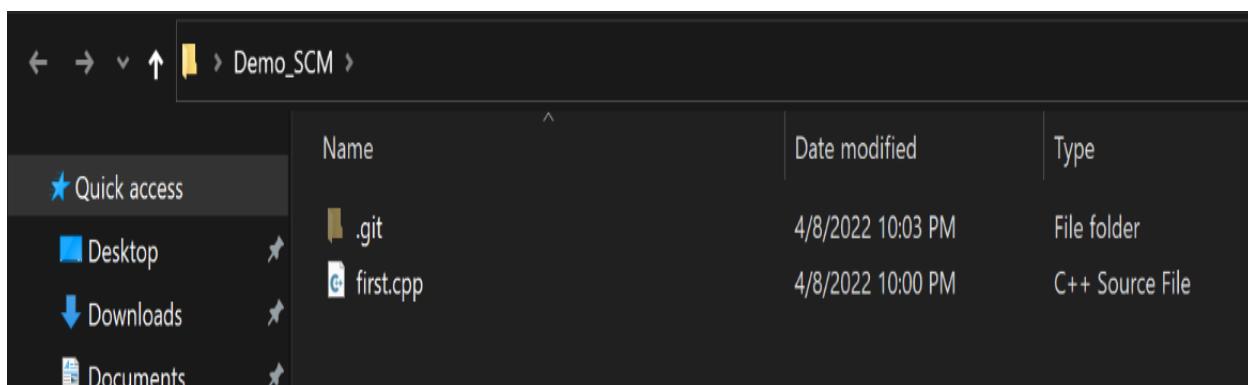
Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/Demo_SCM (features1)
$ git status
on branch features1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   first.cpp

Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/Demo_SCM (features1)
$ git commit -m "Age function added"
[features1 614c80d] Age function added
 1 file changed, 8 insertions(+)

Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/Demo_SCM (features1)
$ |

```

After this we will change the branch from activity1 to master, but when I will switch to the master branch there will not be the same file in the master , it will not show the new file in the master branch.



In this way we can create and change different branches . We can also merge the branches by using git merge command.

## Experiment No. 05

**Aim:** Git lifecycle description

### Theory:

**Stages in GIT Life Cycle ->** Files in a Git project have various stages like Creation, Modification, Refactoring, and Deletion and so on. Irrespective of whether this project is tracked by Git or not, these phases are still prevalent. However, when a project is under Git version control system, they are present in three major Git states in addition to these basic ones. Here are the three Git states:

- Working directory
- Staging area
- Git directory

#### **Working Directory ->**

Consider a project residing in your local system. This project may or may not be tracked by Git. In either case, this project directory is called your Working directory.

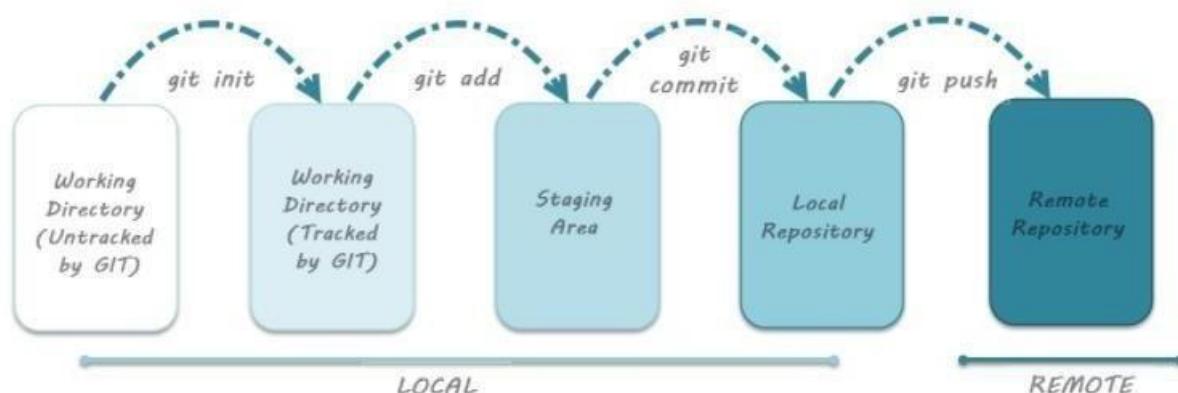
#### **Staging Area ->**

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

#### **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.

**Remote Repository->** means mirror or clone of the local Git repository in GitHub. And pushing means uploading the commits from local Git repository to remote repository hosted in GitHub.



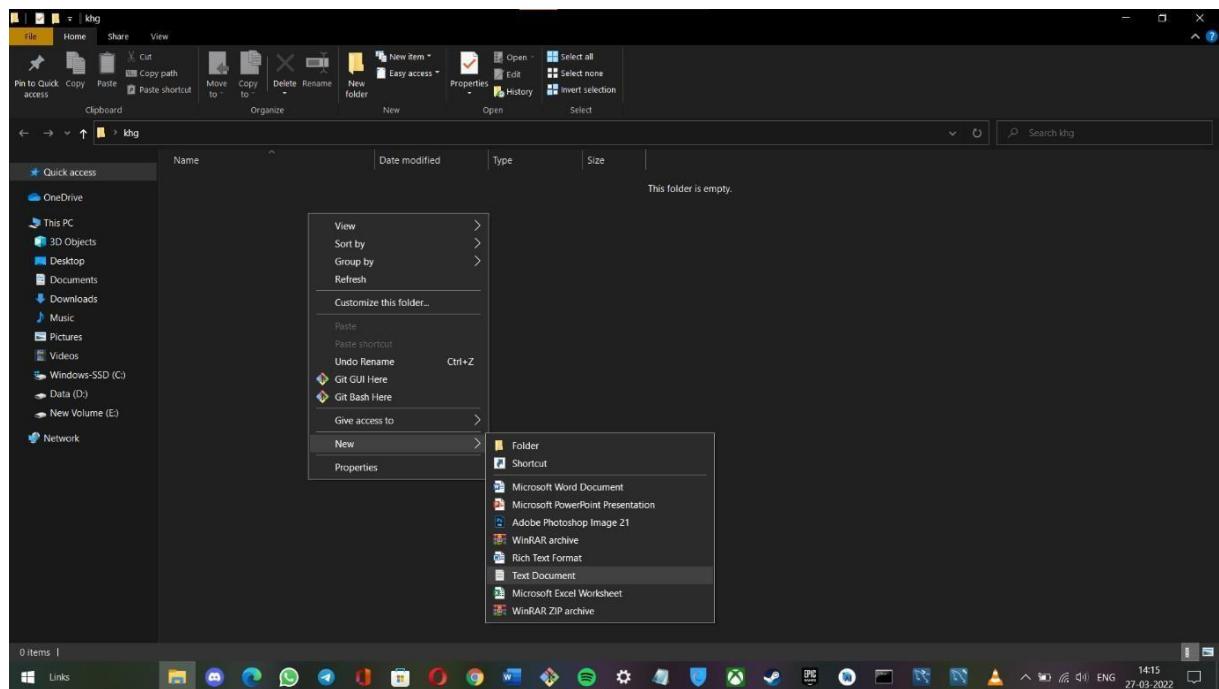


## Snapshots –

```
MINGW64:/c/Users/Surface/Desktop/SCM

Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/SCM
$ git status
fatal: not a git repository (or any of the parent directories): .git

Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/SCM
$ |
```



to	to	Organize	History	Invert Selection
d		New	Open	Select
khg				
Name		Date modified	Type	Size
git		27-03-2022 14:16	Text Document	0 KB

```
MINGW64:/c/Users/Surface/Desktop/SCM
Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/SCM
$ git status
fatal: not a git repository (or any of the parent directories): .git

Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/SCM
$ git init
Initialized empty Git repository in C:/users/surface/Desktop/SCM/.git/

Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/SCM (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/SCM (master)
$
```

```
 MINGW64:/c/Users/Surface/Desktop/SCM  
Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/SCM (master)  
$ git add --a  
  
Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/SCM (master)  
$ git status  
On branch master  
  
No commits yet  
  
nothing to commit (create/copy files and use "git add" to track)  
Surface@DESKTOP-DMPQRBG MINGW64 ~/Desktop/SCM (master)  
$ |
```