Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Academic Year: 2023_24

NAME:-PREKSHA PATEL
SAPID:-60004210126
DIV:-C2 ;BATCH:- 1
BRANCH:-COMPUTER ENGINEERING

SOFTWARE ENGINEERING

Experiment No. 9

Aim: Study of Configuration Management using GitHub

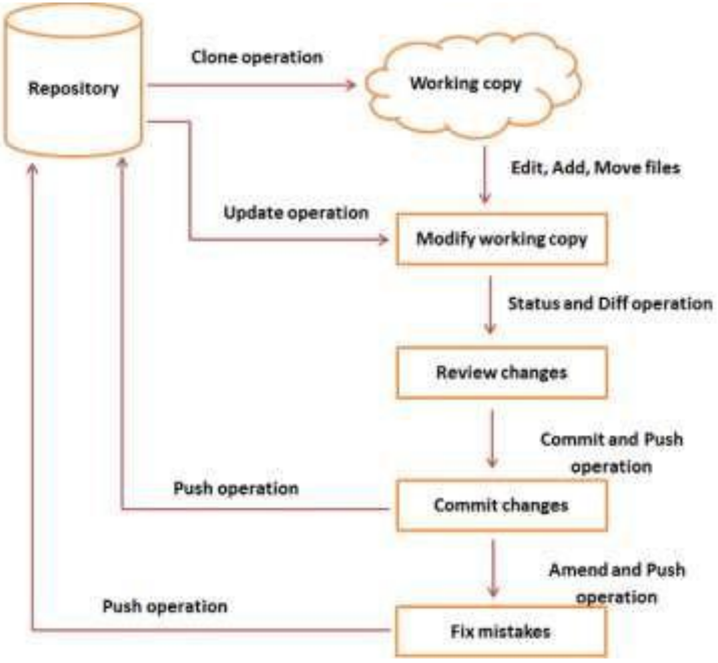
Theory:

Git is a distributed revision control and source code management system with an emphasis on speed. Git was initially designed and developed by Linus Torvalds for Linux kernel development. Git is a free software distributed under the terms of the GNU General Public License version 2.

Git Life Cycle

General workflow is as follows −

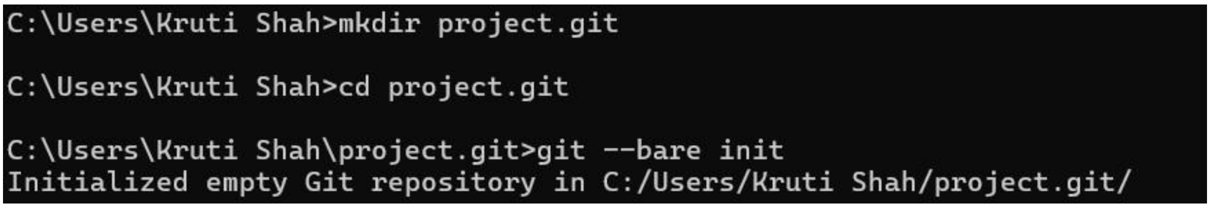
1. Clone the Git repository as a working copy.
2. Modify the working copy by adding/editing files.
3. If necessary, update the working copy by taking other developer's changes.
4. Review the changes before commit.
5. Commit changes. If everything is fine, then push the changes to the repository.
6. After committing, if something is wrong, then correct the last commit and push the changes to the repository.



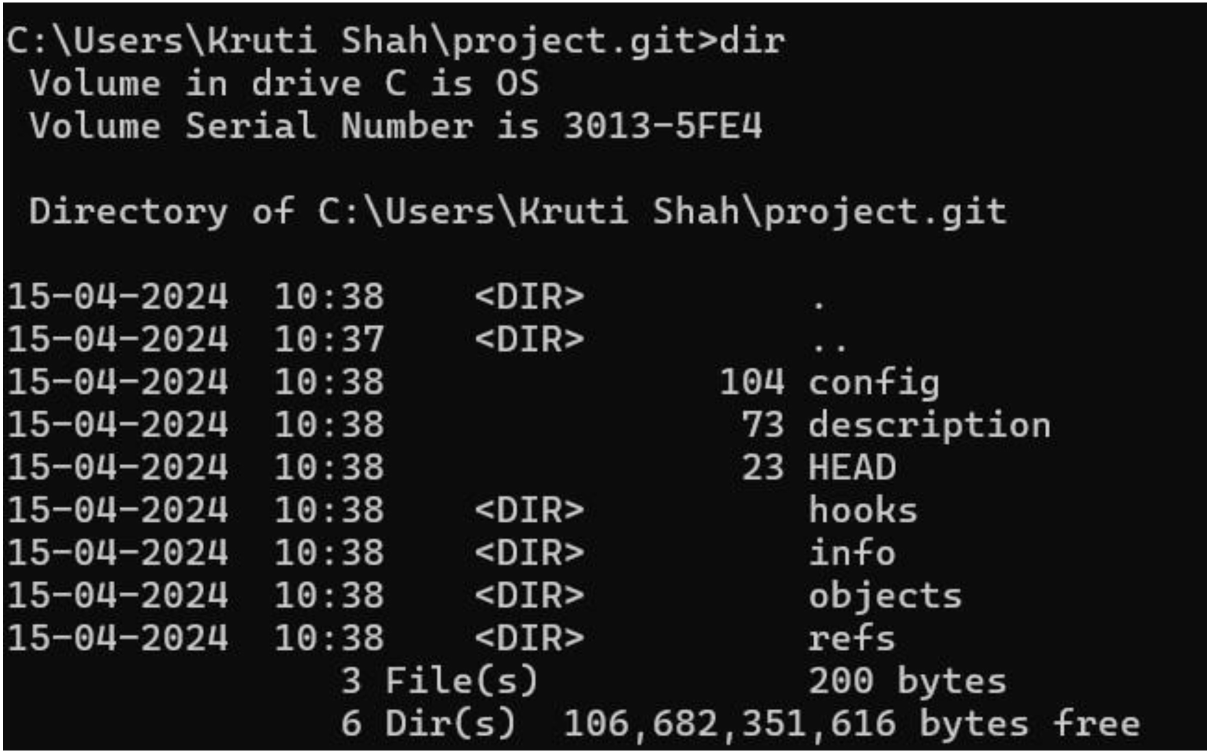
Git Life Cycle

1. Creating Git Repository

   Initialize a new repository by using **init** command followed by **--bare** option. It initializes the repository without a working directory. By convention, the bare repository must be named as **.git**.

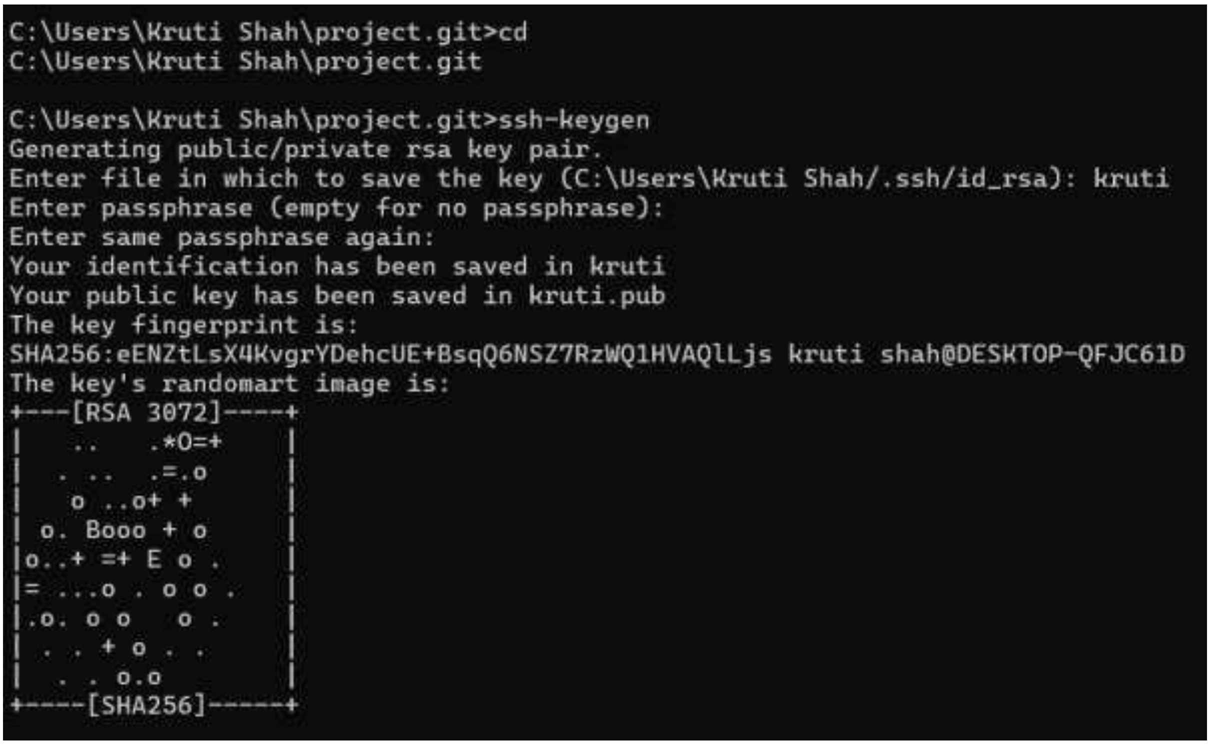


```
C:\Users\Kruti Shah>mkdir project.git

C:\Users\Kruti Shah>cd project.git

C:\Users\Kruti Shah\project.git>git --bare init
Initialized empty Git repository in C:/Users/Kruti Shah/project.git/
```



```
C:\Users\Kruti Shah\project.git>dir
 Volume in drive C is OS
 Volume Serial Number is 3013-5FE4

 Directory of C:\Users\Kruti Shah\project.git

15-04-2024  10:38    <DIR>          .
15-04-2024  10:37    <DIR>          ..
15-04-2024  10:38               104 config
15-04-2024  10:38                73 description
15-04-2024  10:38                23 HEAD
15-04-2024  10:38    <DIR>          hooks
15-04-2024  10:38    <DIR>          info
15-04-2024  10:38    <DIR>          objects
15-04-2024  10:38    <DIR>          refs
               3 File(s)            200 bytes
               6 Dir(s)  106,682,351,616 bytes free
```

2. Generate Public-Private RSA Key Pair

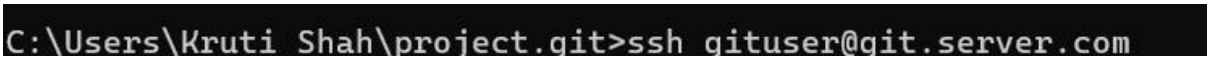


```
C:\Users\Kruti Shah\project.git>cd
C:\Users\Kruti Shah\project.git

C:\Users\Kruti Shah\project.git>ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\Kruti Shah/.ssh/id_rsa): kruti
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in kruti
Your public key has been saved in kruti.pub
The key fingerprint is:
SHA256:eENZtLsX4KvgrYDehcUE+BsqQ6NSZ7RzWQ1HVAQlLjs kruti shah@DESKTOP-QFJC61D
The key's randomart image is:
+---[RSA 3072]----+
|   ..   .*O=+    |
|  . ..  .=.o     |
|   o ..o+ +      |
| o. Booo + o     |
|o..+ =+ E o .    |
|= ...o . o o .   |
|.o. o o   o .    |
| . . + o . .     |
|  . . o.o        |
+----[SHA256]-----+
```

3. Adding keys to authorized keys

Suppose there are two developers working on a project. Both users have generated public keys.

Both add their public key to the server by using ssh-copy-id command as given below



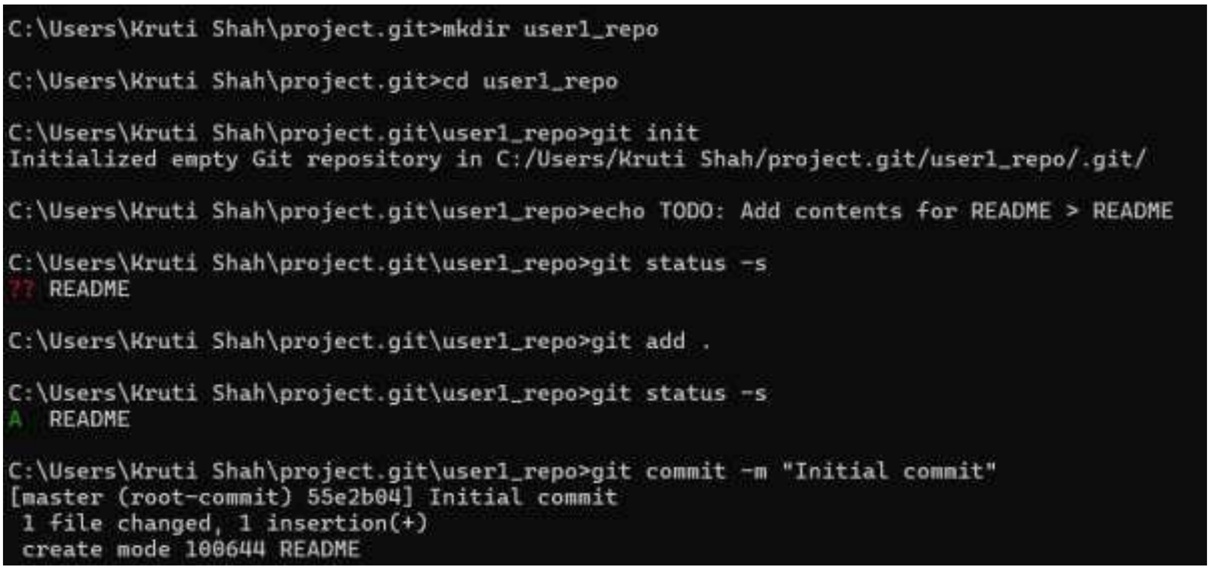
```
C:\Users\Kruti Shah\project.git>ssh gituser@git.server.com
```

4. Push changes to the repository

We have created a bare repository on the server and allowed access for two users. Both users can push their changes to the repository by adding it as a remote.

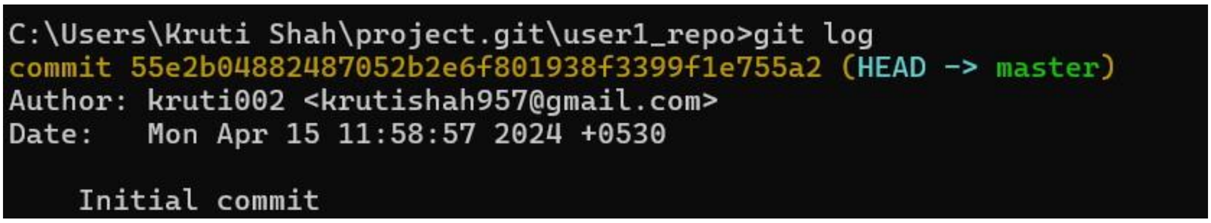
Git init command creates **.git** directory to store metadata about the repository every time it reads the configuration from the **.git/config** file.

User1 creates a new directory, adds README file, and commits his change as initial commit. After commit, he verifies the commit message by running the **git log** command.



```
C:\Users\Kruti Shah\project.git>mkdir user1_repo

C:\Users\Kruti Shah\project.git>cd user1_repo

C:\Users\Kruti Shah\project.git\user1_repo>git init
Initialized empty Git repository in C:/Users/Kruti Shah/project.git/user1_repo/.git/

C:\Users\Kruti Shah\project.git\user1_repo>echo TODO: Add contents for README > README

C:\Users\Kruti Shah\project.git\user1_repo>git status -s
?? README

C:\Users\Kruti Shah\project.git\user1_repo>git add .

C:\Users\Kruti Shah\project.git\user1_repo>git status -s
A  README

C:\Users\Kruti Shah\project.git\user1_repo>git commit -m "Initial commit"
[master (root-commit) 55e2b04] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 README
```

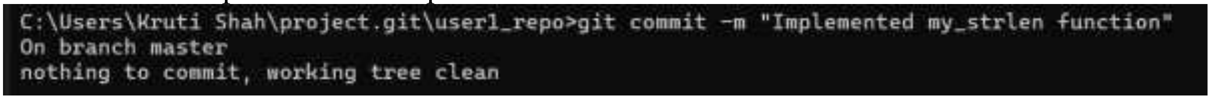
5. Checking log message by executing the git log command.



```
C:\Users\Kruti Shah\project.git\user1_repo>git log
commit 55e2b04882487052b2e6f801938f3399f1e755a2 (HEAD -> master)
Author: kruti002 <krutishah957@gmail.com>
Date:   Mon Apr 15 11:58:57 2024 +0530

    Initial commit
```

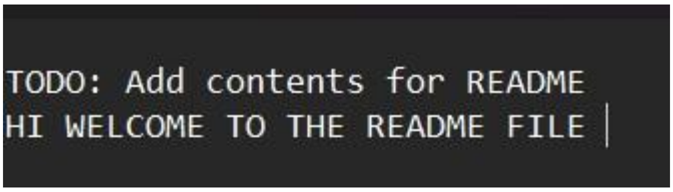
6. Commit changes

To commit the changes, he used the git commit command followed by –m option. If we omit –m option. Git will open a text editor where we can write multiline commit message

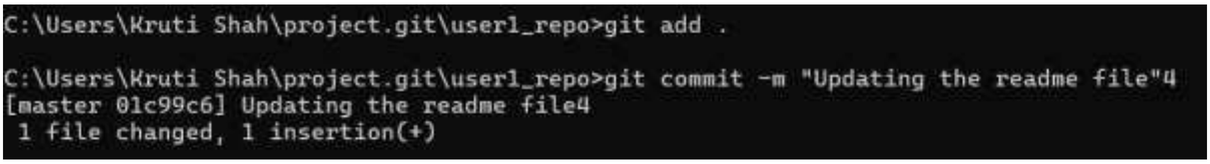


```
C:\Users\Kruti Shah\project.git\user1_repo>git commit -m "Implemented my_strlen function"
On branch master
nothing to commit, working tree clean
```

UPDATING THE README FILE



```
TODO: Add contents for README
HI WELCOME TO THE README FILE
```



```
C:\Users\Kruti Shah\project.git\user1_repo>git add .

C:\Users\Kruti Shah\project.git\user1_repo>git commit -m "Updating the readme file"4
[master 01c99c6] Updating the readme file4
 1 file changed, 1 insertion(+)
```

Performance:

1. Perform all the commands using Git
2. Take screenshots of each of the command and respective output
3. Explore the commands for merging the documents and show the screenshots.

Conclusion:

Git is a version control system that can keep several versions on a local workstation or integrate with a remote file management system. To perform commands, we used Git bash cmd and saved the local files to the GitHub files management server. We also saw the git log command, which stores the git activity log.