Practical 1: clone Repository:

1) Make Repository on GitHub.

2) clone Repository:

```
$ git clone link            [commond]

$ git clone https://.........   [excemple]
```

3] In computer C Drive:

→ open C Drive
→ users
→ admin
→ clone Repository name file

* Screen sort.
i. on GitHub ~~Account~~ created Repository.
ii. In PC created Repository
[Location in PC: c drive / user / Admin /~~Repo~~]

Practical No. 3:- use git add to the file to the repository use git commit to commit your changes.

1) Make Repository. (KBCNMU)

2) clone Repository.

3) change directory (Git bash) $ cd Repository_name,
$ cd KBCNMU, (Example)

4) new ~~add~~ File in repository.
$ ~~gi~~ touch File name.extension. [commond]

$ touch ketu.html [example]

5) commit
$ git commit -m "message," [commond]

$ git commit -m "First commit" [example]

* Screen-sort ; Local Disk (C) / User / Admin / Repository_name
i. Before File create
ii. After create File.

PC→to→online

Practical No. 4:- use git push to send your change to the primary repository. Look online to see if your changes has been pushed.

1) Make Repository

2) clone Repository `$ git clone link`

3) chand directory `$ cd Repo_name`

4) initialize Repository. `$ git init Rep_name`

5) ~~Add~~ create File to ~~re~~ repository `$ touch File_name.extension`

6) chek file make or Not. `$ git status`

7) add file to repository.

```
$ git add File_name.ext
```

same file name

8) commite

```
$ git commit -m "First commit".
```

9)

```
$ git push -u origin main
```

* Screen Sort:

i. Create Repository ii. Authentication (pop window 9)
iii. After push File on repository.

Online → to → PC

Practical 5 :

use git pull to get your pardners changes into your repositery. verify that you have the new files.

i) Create Repositer

ii) clone repositery $ git clone link

iii) change directery $ cd MCA

iv) initialize repositery. $ git init [MCA]

v) Open Repository on GitHub
Ex. MCA
In this MCA Repositery create a new file.

vi) This online file come or get in our local pc. under this "MCA" repository (ની ફાઈલ આપણા PC વરીમ રિપોઝિટરીમા આવશે)

~~$ pul~~ $ git pull link (MCA Repository link)

* Screen Sort : In PC (C Drive / user / Admin / Created Reposi.
i. Create Repository (in PC)
ii) After vi) commond repositery (in PC)

Practical 6:

Use git log or git log | less to see a list of changes to the repository.

1) create Repository

2) clone Repository.

3) Add File to repository.

4) change directory (Git Bash)

5) show commit log. `$ git log.`

6)

```
$ git log ~~ont~~ --oneline
```

7)

```
$ git log --stat
```

8)

```
$ git log --patch
```

9)

```
$ git log --graph
```

10)

```
$ git log --graph --oneline
```

11)

```
$ git log --after --2023-06-15
```

12)

```
$ git log --after "19 days ago"
```

13)

```
$ git log --author "ketenvmali"
```

14)

```
$ git log --auther "ketenvmali311@gmail.com"
```

15)

```
$ git log --grep "commit"
```

Practical 7 :-

Use git branch to create a new branch & git checkout to switch to the branch.

1) Make Repository

2) Create clone

3) Git bash (change directory)

4) Create branch

```
$ git branch branch#
```

5) switch branch

```
$ git checkout branch#
```

6) Create new File

```
$ touch ketu.html.
```

7) add File

```
$ git add ketu.html
```

8) check repository inside File create or not.

```
$ git status
```

9) Commit :

```
$ git commit -m "commit"
```

10)

```
$ git push -u origin branch#
```

Practical 8 :

Use git Merge to merge your change

~~i)~~ to the primary branch.

1) Create Repository (Git Hub)

2) clone Repository (Git Bash)

3) ~~cd~~ change directory. (Git Bash)

4) create new file

```
$ touch Demo1.txt
```

5) add file.

```
$ git add Demo1.txt
```

6) commit

```
$ git commit -m "commite1"
```

7)

```
$ git branch keru
```

8)

```
$ git checkout keru
```

9) Create another new file

~~$ git~~ $ touch Demo2.txt

10) add file

~~$git add~~ $ git add Demo2.txt

11) commit 2

```
$ git commit -m "commit 2"
```

12) check branch list

```
$ git branch --list
```

* Screen short : In Pc Repository

i) created Reposity new

ii) After practical (merge) repository files

Practical 9:

Diff commond.

1) Create Repository.

[Add somes Files in Repository & some write text. & goes offline changes messages]

2) clone Repository.

3) chang directory.
(3.1)

4)

```
$ git diff
```

5)

```
$ git diff --color-words
```

6) create branch

```
$ git branch branch2
```

7) switch branch

```
$ git checkout branch2
```

8) create new file.

```
$ touch xyz.txt
```

9) add file

```
$ git add xyz.txt,
```

10)

```
$ git commit -m " commit "
```

11) $ git diff main branch branch2

12) ~~g~~ $ git diff main branch2 --name~~t~~-only.

13) $ git diff main

14) $ git diff main...branch2

* Screen Sort: In PC Repository:

i) when Repository clone

ii) end of practical Repository.