ASSIGNMENT – 5

COURSE : DEVOPS

Trainer : Mr . MADHUKAR
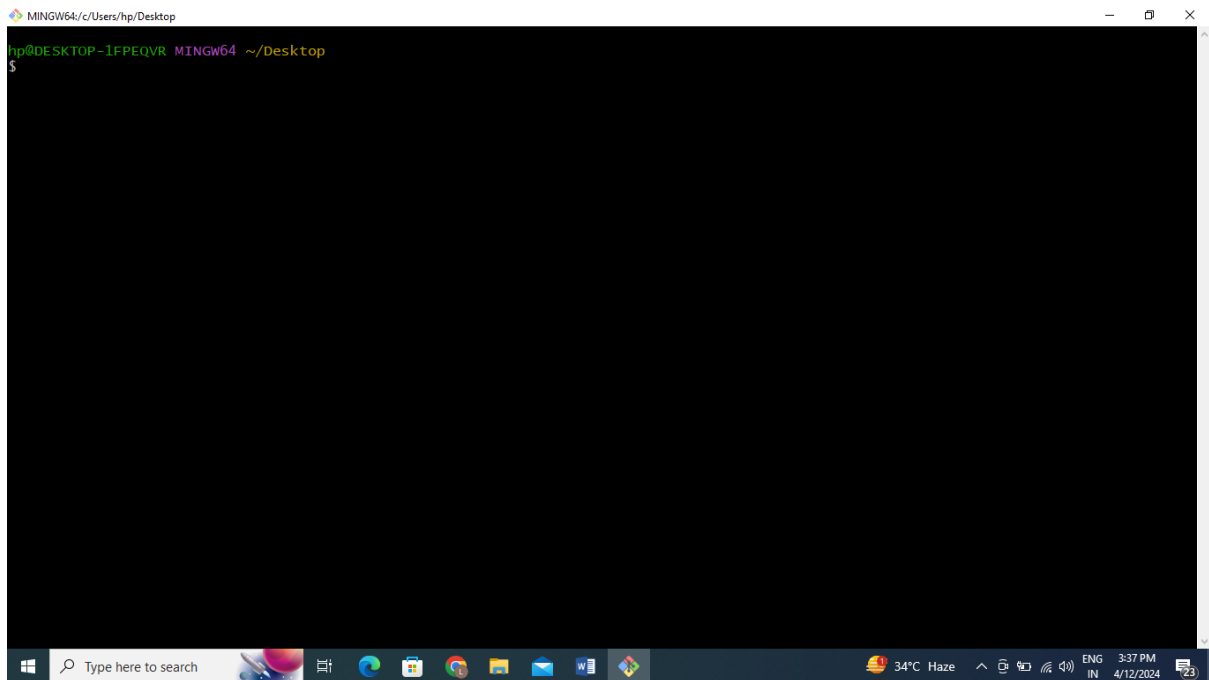
NAME:R.Lakshmi
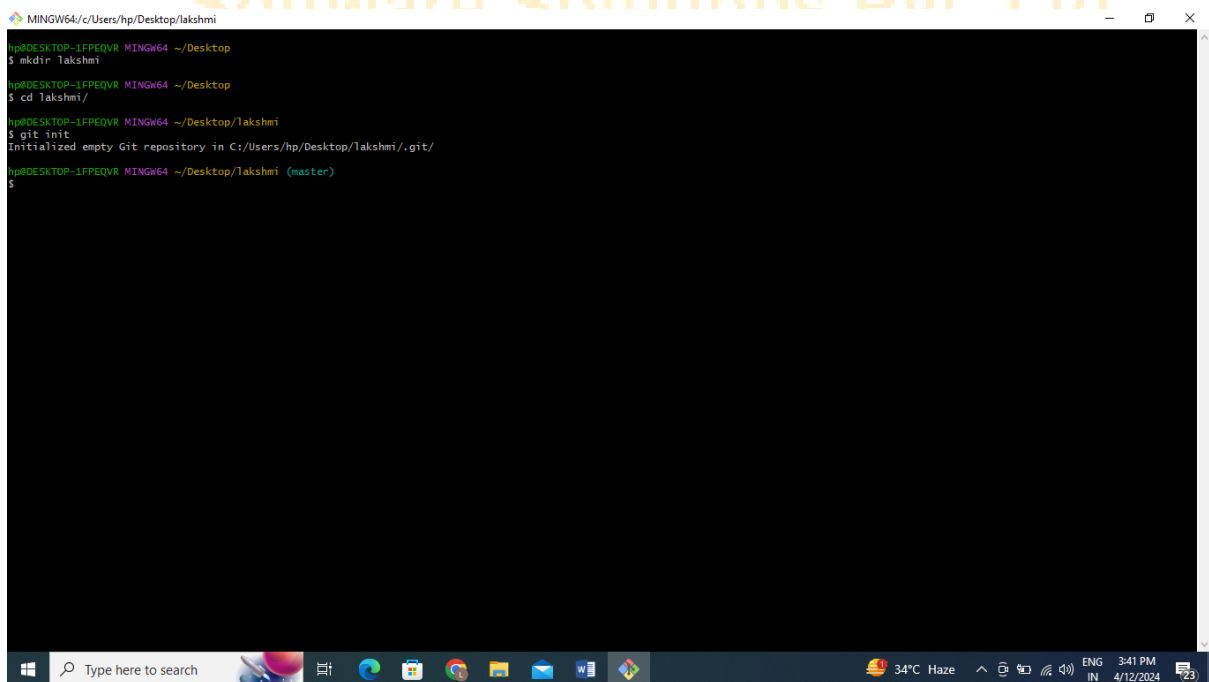
Mail id : ravurilakshmi2315@gmail.com

1) Create two branches and merge them , create directory and push in the repository , commit the changes to the file and use cherry pick to commit changes between branches ?

- Open the gitbash here then create a directory in desktop using command
- Command is :- mkdir <directory name>
- Initialize the created directory command is "**git init**"
- Default branch is "**master**"

- Now create a file in master branch and do the work flow (i.e., git add . & git commit)

- Now create new branch and create files in new branch and do the work flow (i.e., git add . & git commit)
- New create & switch branch commands are : -
- git branch <branch Name>
- git checkout <branch Name>

- Now using cherry-pick command we required file to be copied into one branch to another branch command is
- git cherry-pick <first 7 or more digits commit id of we required>

```
commit 4ab3d22afc9d45af5fea60bb136cf58f56b37bbd (master)
Author: Ravurilakshmi28 <ravurilakshmi2315@gmail.com>
Date:    Fri Apr 12 16:08:20 2024 +0530

    1st commit

hp@DESKTOP-1FPEQVR MINGW64 ~/Desktop/lakshmi (dev)
$ git add file3
warning: in the working copy of 'file3', LF will be replaced by CRLF the next time Git touches it

hp@DESKTOP-1FPEQVR MINGW64 ~/Desktop/lakshmi (dev)
$ git commmit -m "3rd commit"
git: 'commmit' is not a git command. See 'git --help'.

The most similar command is
        commit

hp@DESKTOP-1FPEQVR MINGW64 ~/Desktop/lakshmi (dev)
$ git commit -m "3rd commit"
[dev d4a5e40] 3rd commit
 1 file changed, 4 insertions(+)
 create mode 100644 file3

hp@DESKTOP-1FPEQVR MINGW64 ~/Desktop/lakshmi (dev)
$ git log
commit d4a5e403000ecf98f9e43d66c30695b9b019f99b (HEAD -> dev)
Author: Ravurilakshmi28 <ravurilakshmi2315@gmail.com>
Date:    Fri Apr 12 16:13:32 2024 +0530

    3rd commit

commit 6b15321044b38b0025927cba971ea92b6522244d
```

- Now two branches merge and push into the repository
- By git merge <branch Name> this command use merge two branches


```
hp@DESKTOP-1FPEQVR MINGW64 ~/Desktop/lakshmi (dev)
$ git log
commit d4a5e403000ecf98f9e43d66c30695b9b019f99b (HEAD -> dev)
Author: Ravurilakshmi28 <ravurilakshmi2315@gmail.com>
Date:    Fri Apr 12 16:13:32 2024 +0530

    3rd commit

commit 6b15321044b38b0025927cba971ea92b6522244d
Author: Ravurilakshmi28 <ravurilakshmi2315@gmail.com>
Date:    Fri Apr 12 16:12:20 2024 +0530

    2nd commit

commit 4ab3d22afc9d45af5fea60bb136cf58f56b37bbd (master)
Author: Ravurilakshmi28 <ravurilakshmi2315@gmail.com>
Date:    Fri Apr 12 16:08:20 2024 +0530

    1st commit

hp@DESKTOP-1FPEQVR MINGW64 ~/Desktop/lakshmi (dev)
$ git checkout master
Switched to branch 'master'

hp@DESKTOP-1FPEQVR MINGW64 ~/Desktop/lakshmi (master)
$ git cherry-pick 6b153210
[master d4bde90] 2nd commit
 Date: Fri Apr 12 16:12:20 2024 +0530
 1 file changed, 4 insertions(+)
 create mode 100644 file2

hp@DESKTOP-1FPEQVR MINGW64 ~/Desktop/lakshmi (master)
$ git log
```

- Now go to GitHub account and create repository

This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. Learn more about licenses.

ⓘ You are creating a public repository in your personal account.
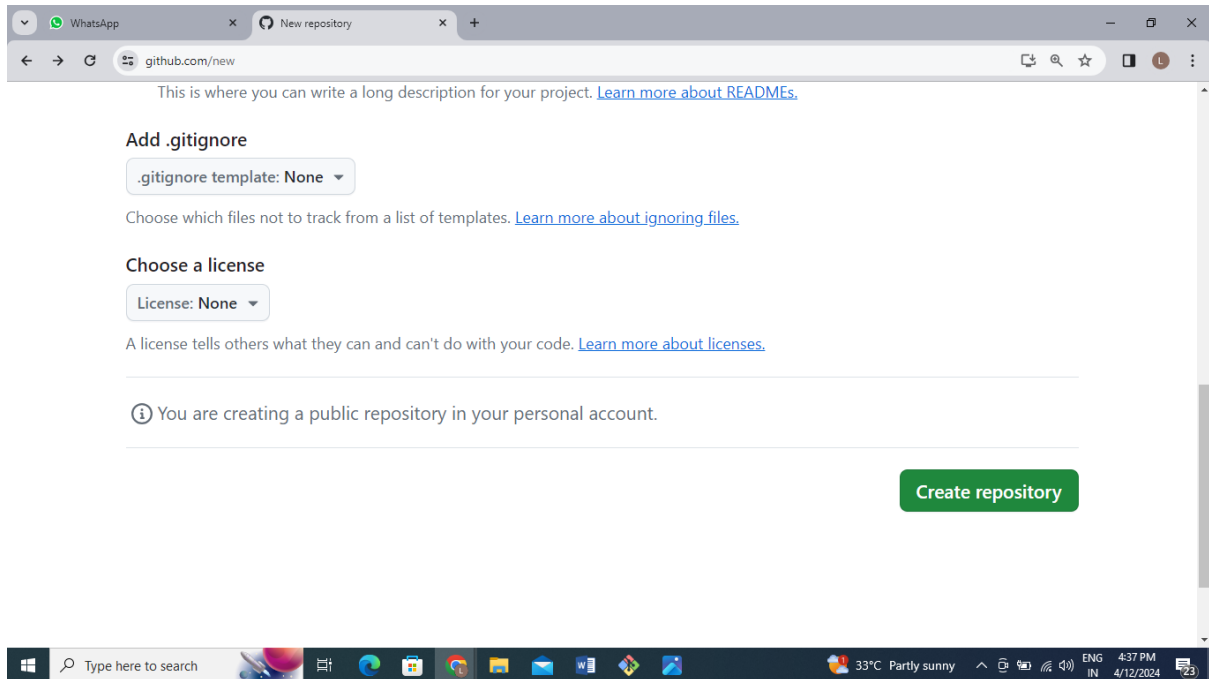
Create repository

- And copy the git remote add URL paste into the gitbash

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# lakshmi12" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Ravurilakshmi28/lakshmi12.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/Ravurilakshmi28/lakshmi12.git
git branch -M main
git push -u origin main
```

...or import code from another repository

- **"git push origin master"** using this command files are upload into the repository
- Now we can see the files in the Repository