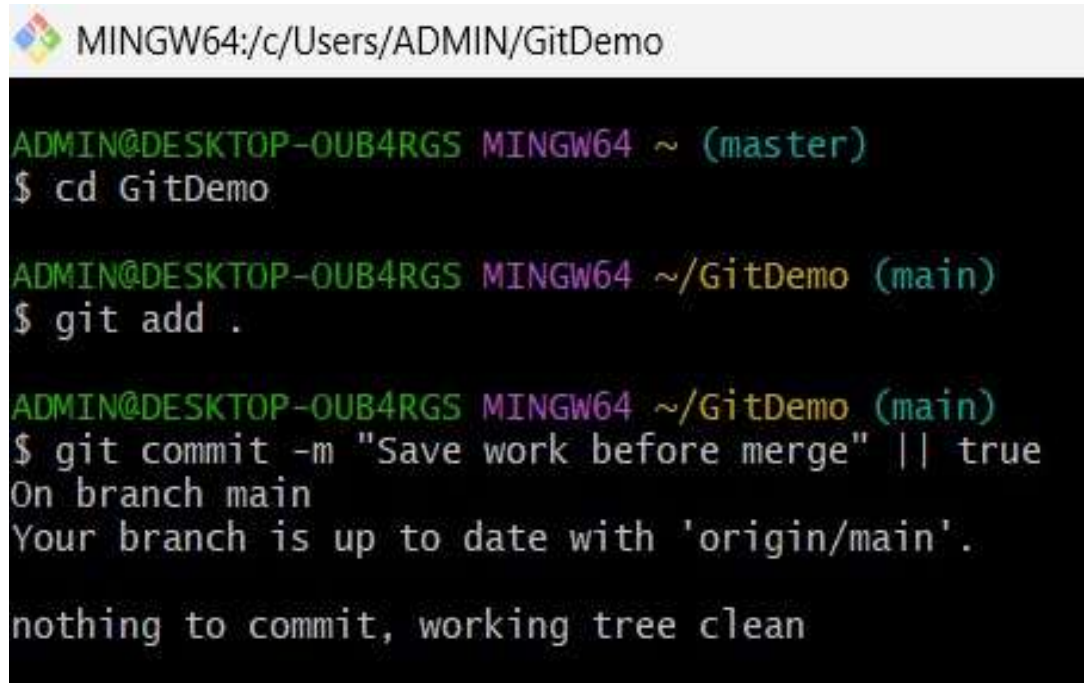## 2. Git-HOL

Create a ".log" file and a log folder in the working directory of Git. Update the .gitignore file in such a way that on committing, these files (.log extensions and log folders) are ignored.

Verify if the git status reflects the same about working directory, local repository and git repository.

```
ADMIN@DESKTOP-OUB4RGS MINGW64 ~/GitDemo (main)
$ git pull origin main --allow-unrelated-histories
From https://gitlab.com/kasthuriram116-group/gitdemo
 * branch            main         -> FETCH_HEAD
Already up to date.

ADMIN@DESKTOP-OUB4RGS MINGW64 ~/GitDemo (main)
$ printf "*.log\nlog/\n" > .gitignore

ADMIN@DESKTOP-OUB4RGS MINGW64 ~/GitDemo (main)
$ git add .gitignore
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the ne
xt time Git touches it

ADMIN@DESKTOP-OUB4RGS MINGW64 ~/GitDemo (main)
$ git commit -m "Add .gitignore to ignore *.log files and log folder"
[main bc79df2] Add .gitignore to ignore *.log files and log folder
 1 file changed, 2 insertions(+)
 create mode 100644 .gitignore

ADMIN@DESKTOP-OUB4RGS MINGW64 ~/GitDemo (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 352 bytes | 352.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://gitlab.com/kasthuriram116-group/gitdemo.git
   64b0efa..bc79df2  main -> main

ADMIN@DESKTOP-OUB4RGS MINGW64 ~/GitDemo (main)
$ touch test.log

ADMIN@DESKTOP-OUB4RGS MINGW64 ~/GitDemo (main)
$ mkdir -p log

ADMIN@DESKTOP-OUB4RGS MINGW64 ~/GitDemo (main)
$ touch log/app.log

ADMIN@DESKTOP-OUB4RGS MINGW64 ~/GitDemo (main)
$ git status --ignored
On branch main
Your branch is up to date with 'origin/main'.

Ignored files:
  (use "git add -f <file>..." to include in what will be committed)
        log/
        test.log

nothing to commit, working tree clean
```