

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.

```
siva4@Siva177 MINGW64 ~ (main)
$ cd git-demo

siva4@Siva177 MINGW64 ~/git-demo (main)
$ git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
  add

siva4@Siva177 MINGW64 ~/git-demo (main)
$ git add .

siva4@Siva177 MINGW64 ~/git-demo (main)
$ git commit -m " save work before merge" || true
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

siva4@Siva177 MINGW64 ~/git-demo (main)
$ git pull origin main --allow-unrelated-histories
From https://github.com/siva1707/git-demo
 * branch      main       -> FETCH_HEAD
Already up to date.

siva4@Siva177 MINGW64 ~/git-demo (main)
$ printf "%.log\nlog/\n" > .gitignore

siva4@Siva177 MINGW64 ~/git-demo (main)
$ git add .gitignore
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it

siva4@Siva177 MINGW64 ~/git-demo (main)
$ git commit -m "Add .gitignore to ignore *.log files and log folder"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

```
siva4@Siva177 MINGW64 ~/git-demo (main)
$ git push origin main
Everything up-to-date

siva4@Siva177 MINGW64 ~/git-demo (main)
$ touch test.log

siva4@Siva177 MINGW64 ~/git-demo (main)
$ mkdir -p log

siva4@Siva177 MINGW64 ~/git-demo (main)
$ touch log/app.log

siva4@Siva177 MINGW64 ~/git-demo (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)
    error.log
    log/
    test.log

nothing to commit, working tree clean

siva4@Siva177 MINGW64 ~/git-demo (main)
$ |
```