Note-

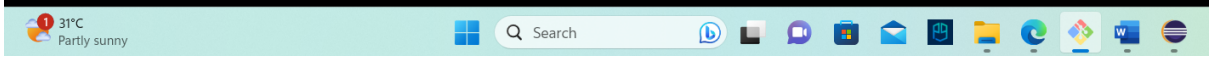
The project name in ide and the repo name need not match.

When we try to paste the name with space we need to include single quote-

```
karan@LAPTOP-H56OVJTV MINGW64 /e
$ cd 'Naveen Java Training'
```

Go to project location-

```
karan@LAPTOP-H56OVJTV MINGW64 /e
$ cd 'Naveen Java Training'

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training
$ ls
Cucumber/

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training
$ cd Cucumber/

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber
$ ls
GitHubPractice/

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber
$ cd GitHubPractice/

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice
$ ls
bin/  src/

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice
$
```




```
karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice
$ ls -alt
total 2
drwxr-xr-x 1 karan 197121   0 Jul 24 16:20 bin/
drwxr-xr-x 1 karan 197121   0 Jul 24 16:20 src/
drwxr-xr-x 1 karan 197121   0 Jul 24 16:19 ./
-rw-r--r-- 1 karan 197121 396 Jul 24 16:19 .classpath
drwxr-xr-x 1 karan 197121   0 Jul 24 16:19 .settings/
-rw-r--r-- 1 karan 197121 390 Jul 24 16:19 .project
drwxr-xr-x 1 karan 197121   0 Jul 24 16:19 ../
```

Dir command-

Shows all directories in the path.

```
karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice
$ dir
bin  src
```

Git init-

Git initialization. We need to register our local repo with ".git".

```
karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice
$ git init
Initialized empty Git repository in E:/Naveen Java Training/Cucumber/GitHubPractice/.git/
```

By default master branch will be created after this command.

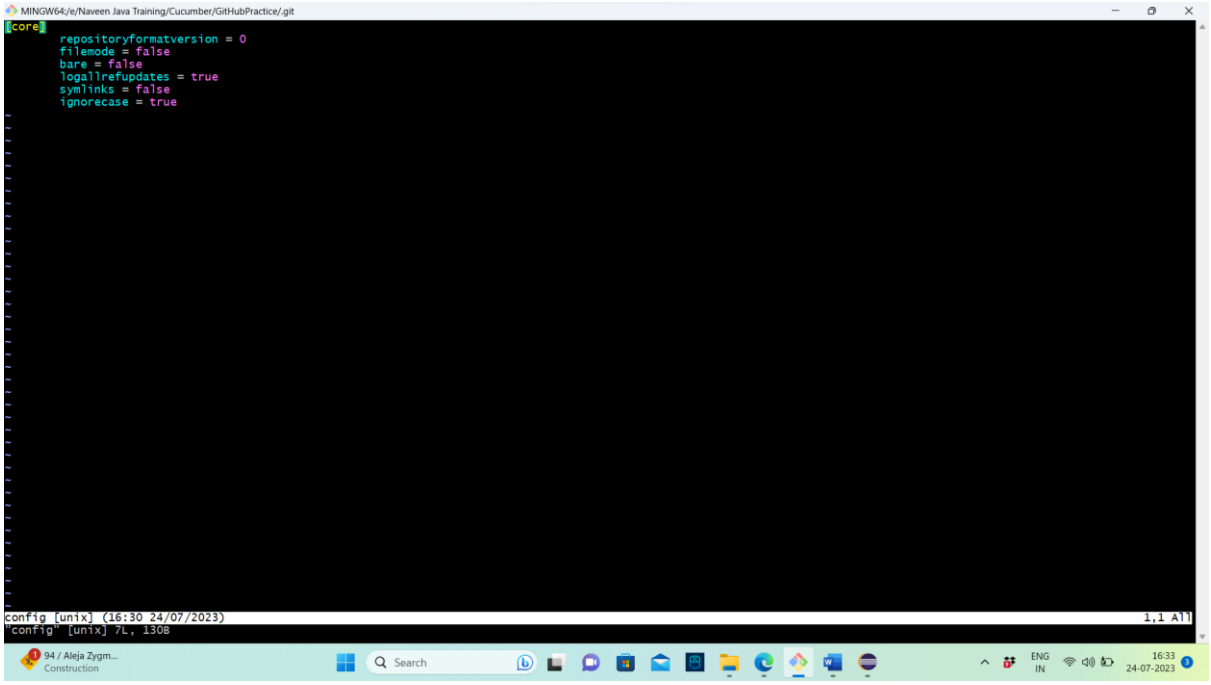
```
MINGW64:/e/Naveen Java Training/Cucumber/GitHubPractice

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice
$ git init
Initialized empty Git repository in E:/Naveen Java Training/Cucumber/GitHubPractice/.git/

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$
```

```
karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$ ls -alt
total 6
drwxr-xr-x 1 karan 197121   0 Jul 24 16:30 .git/
drwxr-xr-x 1 karan 197121   0 Jul 24 16:30 ./
drwxr-xr-x 1 karan 197121   0 Jul 24 16:20 bin/
drwxr-xr-x 1 karan 197121   0 Jul 24 16:20 src/
-rw-r--r-- 1 karan 197121 396 Jul 24 16:19 .classpath
drwxr-xr-x 1 karan 197121   0 Jul 24 16:19 .settings/
-rw-r--r-- 1 karan 197121 390 Jul 24 16:19 .project
drwxr-xr-x 1 karan 197121   0 Jul 24 16:19 ../

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$
```

```
karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$ cd .git/

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice/.git (GIT_DIR!)
$ ls
HEAD  config  description  hooks/  info/  objects/  refs/

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice/.git (GIT_DIR!)
$
```

Open config and see what is there-

All configuration present here in form of key-value pair.



To exit vi-

Press “escape :wq!”.

```
~
~
~
config [unix] (16:30 24/07/2023)
:wq!
```

94 / Aleja Zygm...
Construction

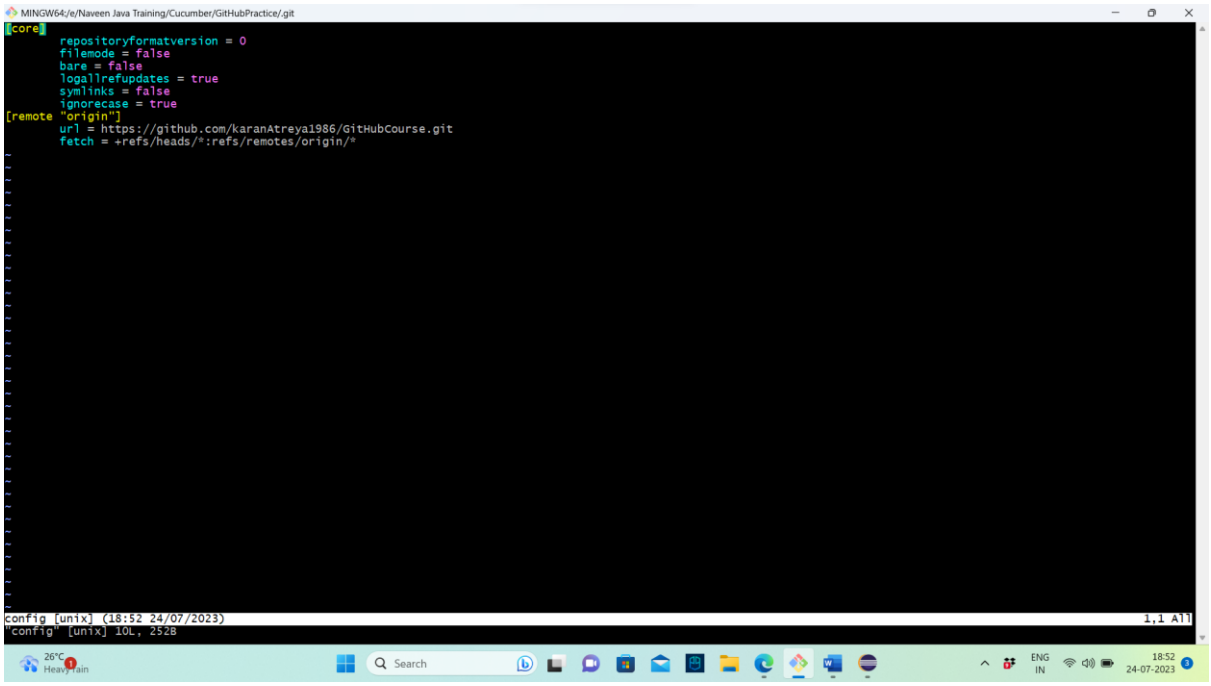
Go back to the project or one level-

```
karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice/.git (GIT_DIR!)
$ cd ..

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$
```

Add remote connection between local folder and remote –


```
karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$ git remote add origin https://github.com/karanAtreya1986/GitHubCourse.git

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$
```

Now when we open config, we get the remote urls-



Git status-

```
karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .classpath
        .project
        .settings/
        bin/
        src/

nothing added to commit but untracked files present (use "git add" to track)

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$
```

Untracked files when moved to stage or index area are not deleted from working copy.

Even files in stage/index are untracked.

Similarly copy of files is created in local repo after commit. There is no deletion happening anywhere in working copy or stage or index or local repo. These become tracked files once they are committed.

Note-

We have to always move/push the src code, not hidden files or no dot files or no garbage files, no reports, no screenshots, no npm modules etc. we have to add these files into ".gitignore".

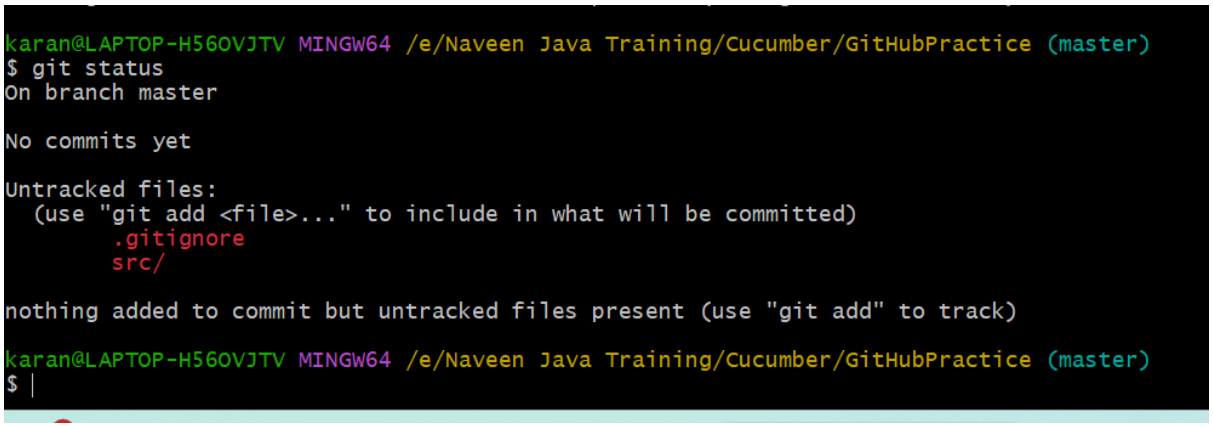
This is the complete list of all items in gitignore file-

Can make this as standard list which should always be ignored.

```
allure-results/
screenshots/
```

```
screenshot/
test-output/
build/

application.log
##############################
## Java
##############################
.mtj.tmp/
*.class
*.jar
*.war
*.ear
*.nar
hs_err_pid*
activityLog.log
##############################
## Maven
##############################
target/
pom.xml.tag
pom.xml.releaseBackup
pom.xml.versionsBackup
pom.xml.next
pom.xml.bak
release.properties
dependency-reduced-pom.xml
buildNumber.properties
.mvn/timing.properties
.mvn/wrapper/maven-wrapper.jar

##############################
## IntelliJ
##############################
out/
.idea/
.idea_modules/
*.iml
*.ipr
*.iws
##############################
## Eclipse
##############################
.settings/
bin/
tmp/
.metadata
.classpath
.project
*.tmp
*.bak
*.swp
*~.nib
local.properties
.loadpath
.factorypath

## OS X
##############################
.DS_Store
```

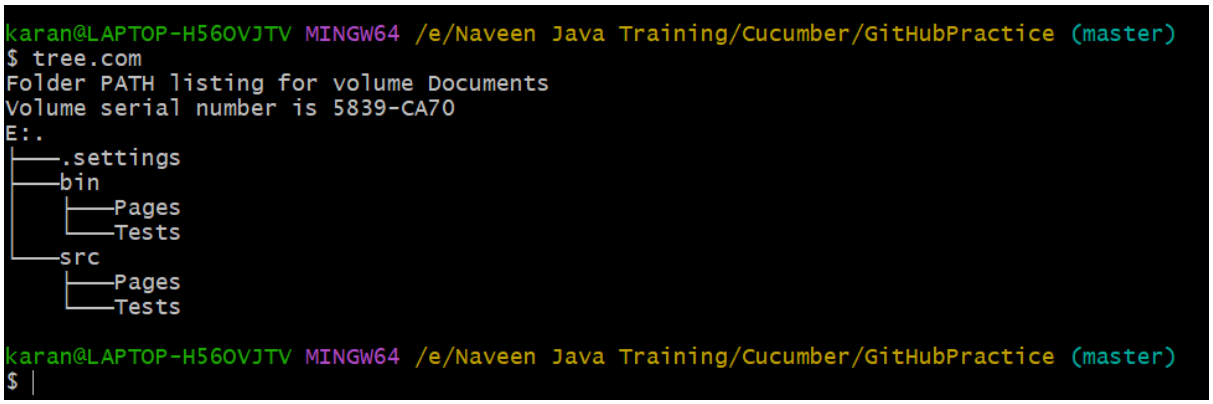
So basically, only the code from src folder should be added and pushed into git. Rest all will go into git ignore.



We can see all the files are present in ".gitignore" and only wanted files from "src" are there. All the files inside ".gitignore" wont participate in git processes.

Tree-

Tree command is little different in windows. Type few characters like "tre" and then Tab and then we get the "tree.com".



This will show the folder level.

The below command will show the files and folders-

```
karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$ tree.com //f
Folder PATH listing for volume Documents
Volume serial number is 5839-CA70
E:.
|   .classpath
|   .gitignore
|   .project
|
+---.settings
|       org.eclipse.core.resources.prefs
|       org.eclipse.jdt.core.prefs
|
+---bin
|   |   module-info.class
|   |
|   +---Pages
|   |       HomePage.class
|   |       LoginPage.class
|   |
|   \---Tests
|           HomePageTest.class
|           LoginPageTest.class
|
\---src
    |   module-info.java
    |
    +---Pages
    |       HomePage.java
    |       LoginPage.java
    |
    \---Tests
            HomePageTest.java
            LoginPageTest.java


karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$
```

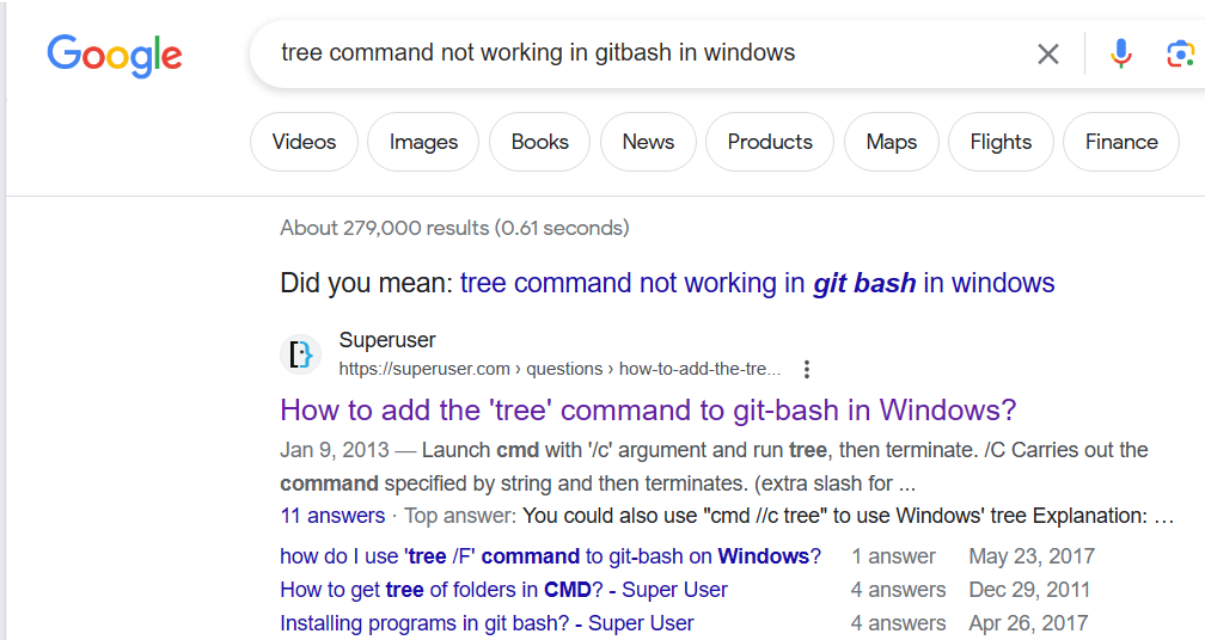
We are only concerned about ".java" files.

The below command is similar-

```
karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$ tree.com //a
Folder PATH listing for volume Documents
Volume serial number is 5839-CA70
E:.
+---.settings
+---bin
|   +---Pages
|   \---Tests
\---src
    +---Pages
    \---Tests

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$
```

This is to show ascii lines but not needed for us.

Go to this website for solution on tree-



[How to add the 'tree' command to git-bash in Windows? - Super User]

This post-

28

There is a tree command in windows already — only problem is it is `tree.com` and git bash will not automatically add extension `.com` and execute it.

However it will find it if you press `tab` after you type *tree* or *tre*

To see files you have to use `//f` — you have to use `//` or bash will think it is folder name

I also used `//a` to show ascii lines but you don't have to use it

Example:

```
dean@dean:~/java$ tree
bash: tree: command not found

dean@dean:~/java$ tree.com //a
Folder PATH listing for volume c
Volume serial number is 4E70-B37A
C:.
+---atom
+---sublime
\---vscode

dean@dean:~/java$ tree.com //a //f
Folder PATH listing for volume c
Volume serial number is 4E70-B37A
C:.
+---atom
|       test1
```

```
karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$ git add .

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .gitignore
        new file:   src/Pages/HomePage.java
        new file:   src/Pages/LoginPage.java
        new file:   src/Tests/HomePageTest.java
        new file:   src/Tests/LoginPageTest.java
        new file:   src/module-info.java


karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$
```

I deleted the module-info manually and again did git add to keep it updated. This is not mandatory step, you can delete the file, do git status, it will show file removed in untracked area, then we can just commit to local repo and do git status.

```
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    src/module-info.java


karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$ git add .

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .gitignore
        new file:   src/Pages/HomePage.java
        new file:   src/Pages/LoginPage.java
        new file:   src/Tests/HomePageTest.java
        new file:   src/Tests/LoginPageTest.java


karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$
```

All yellow highlighted are important ones-

```
karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$ git commit -m "first commit"
[master (root-commit) 9fe238f] first commit
 5 files changed, 96 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 src/Pages/HomePage.java
 create mode 100644 src/Pages/LoginPage.java
 create mode 100644 src/Tests/HomePageTest.java
 create mode 100644 src/Tests/LoginPageTest.java

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$
```

Commit id to master branch.

How many files changed.

How many insertions done into file.

Random memory or index allocated to different files.

Once all committed then the branch is clean-

```
karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$ git status
On branch master
nothing to commit, working tree clean

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$
```

Git branch-

To see how many branches are there.

```
karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$ git branch
* master

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$
```

Git push-

By default the first code we write is always on local master branch in local repo. First push will always be from local master to remote master.

```
karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$ git push origin master
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 1.16 KiB | 395.00 KiB/s, done.
Total 10 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/karanAtreya1986/GitHubCourse.git
 * [new branch]      master -> master

karan@LAPTOP-H56OVJTV MINGW64 /e/Naveen Java Training/Cucumber/GitHubPractice (master)
$
```

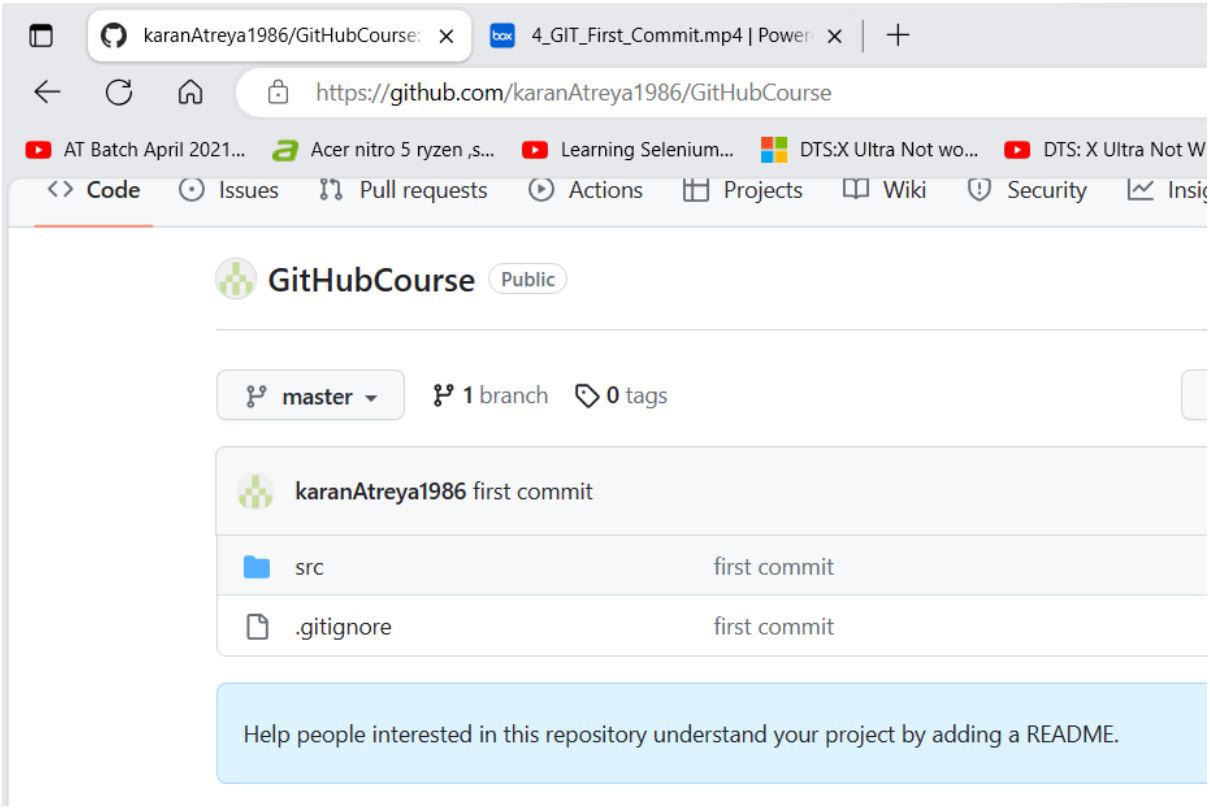
Compression done.

We have pushed our code from local master to remote master.

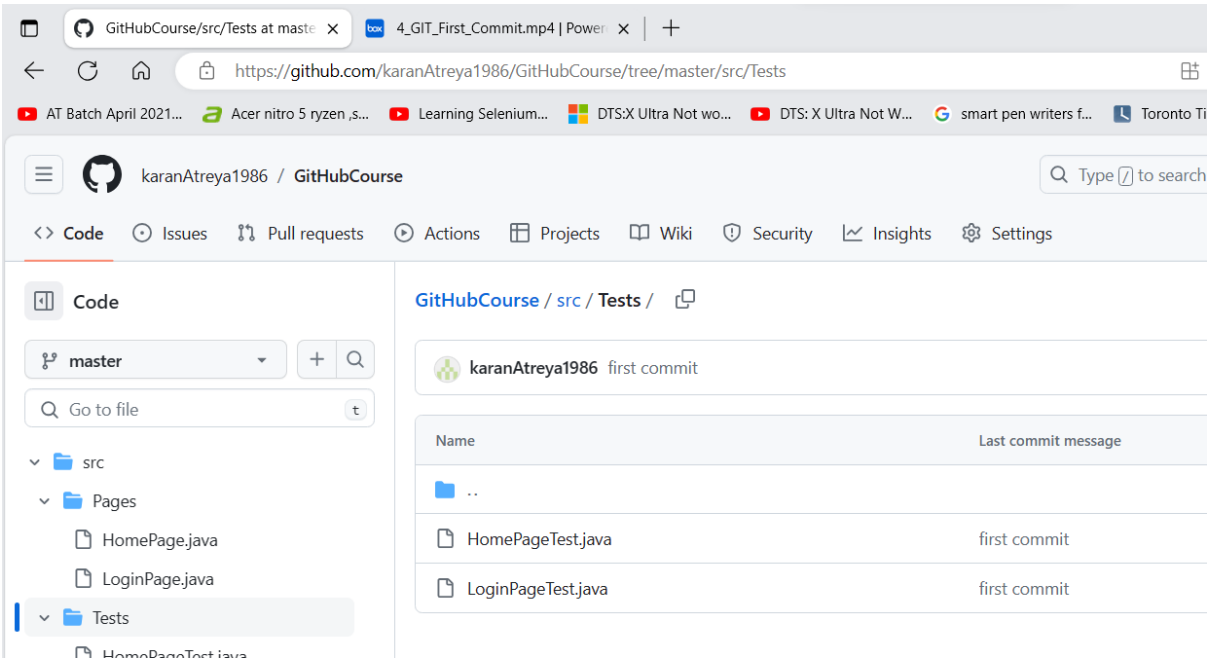
url for remote git is also present.

We are pushing code into the origin (origin also known as remote side) to a branch called master.

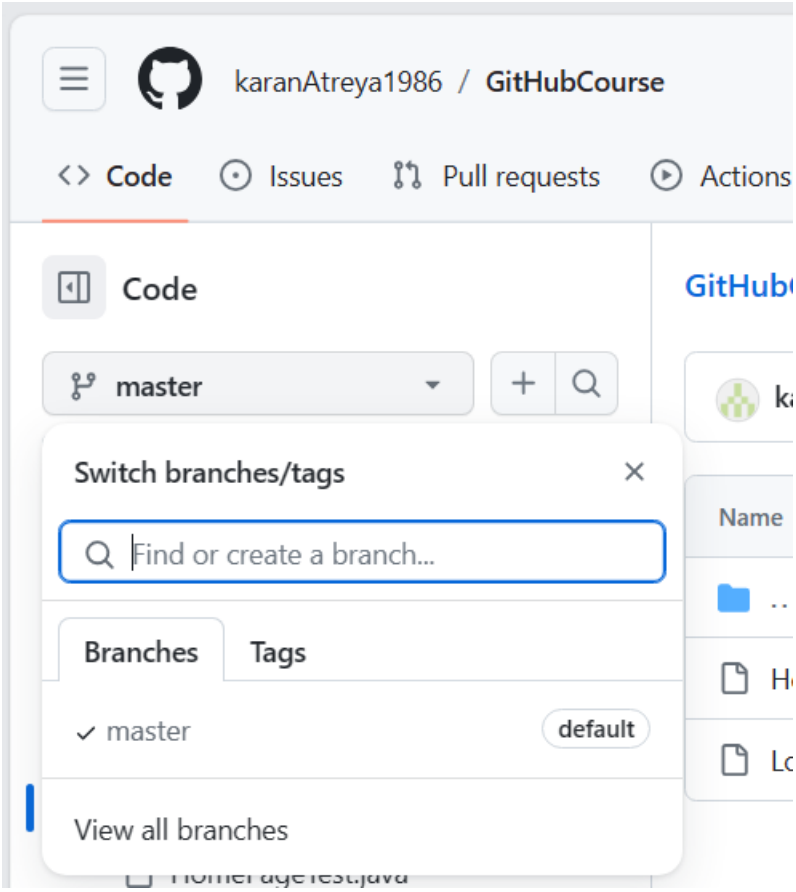
Code pushed-



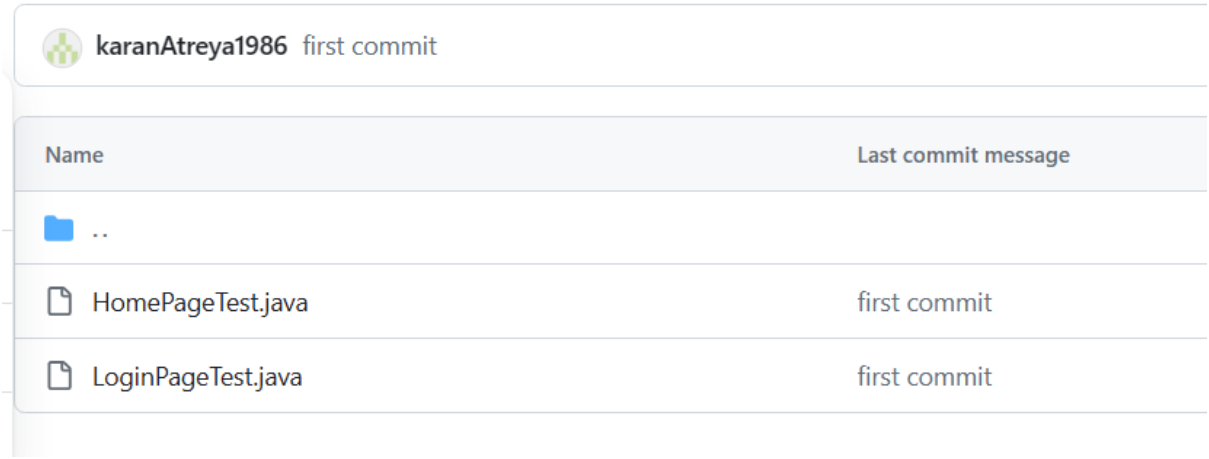
Play around.



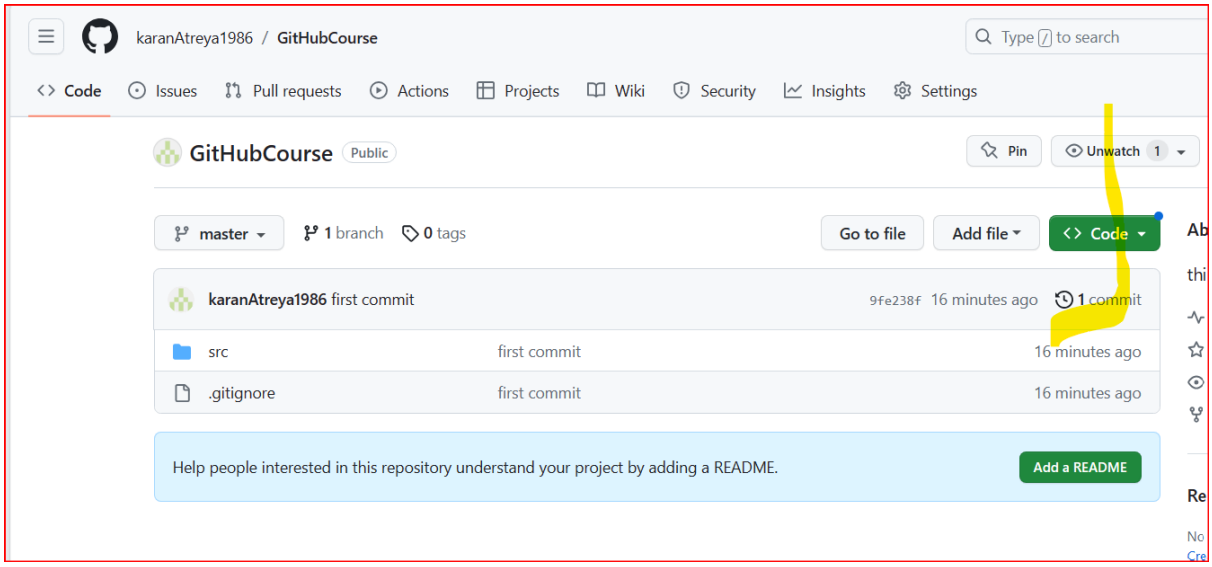
Number of branches can be found from the drop down-



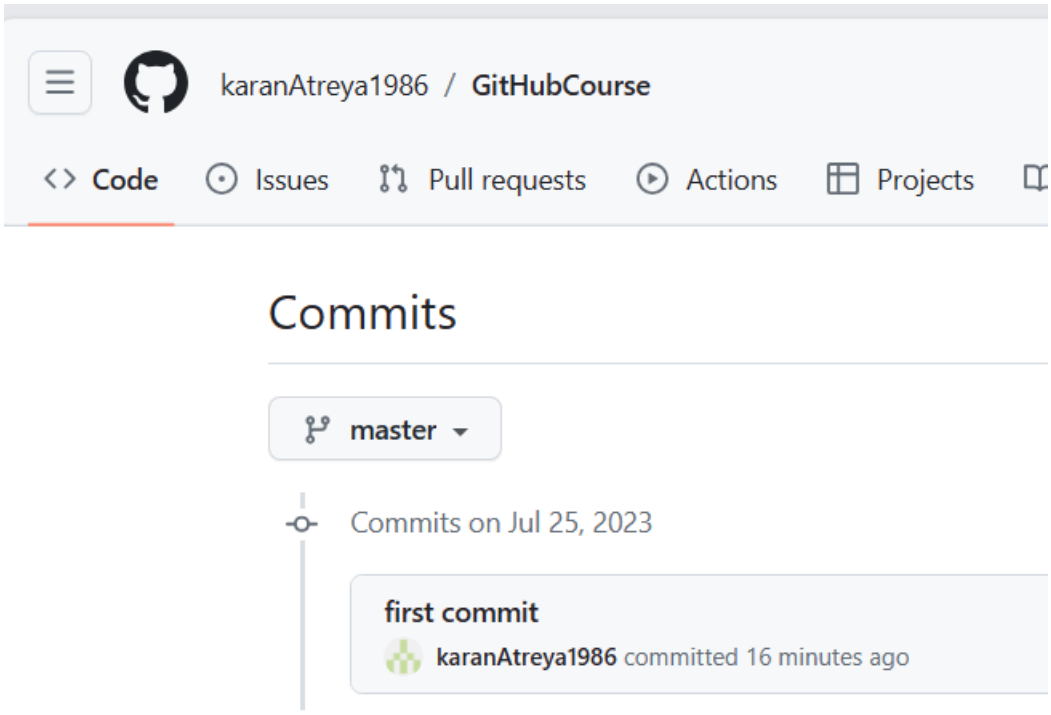
Commit reason which we wrote-



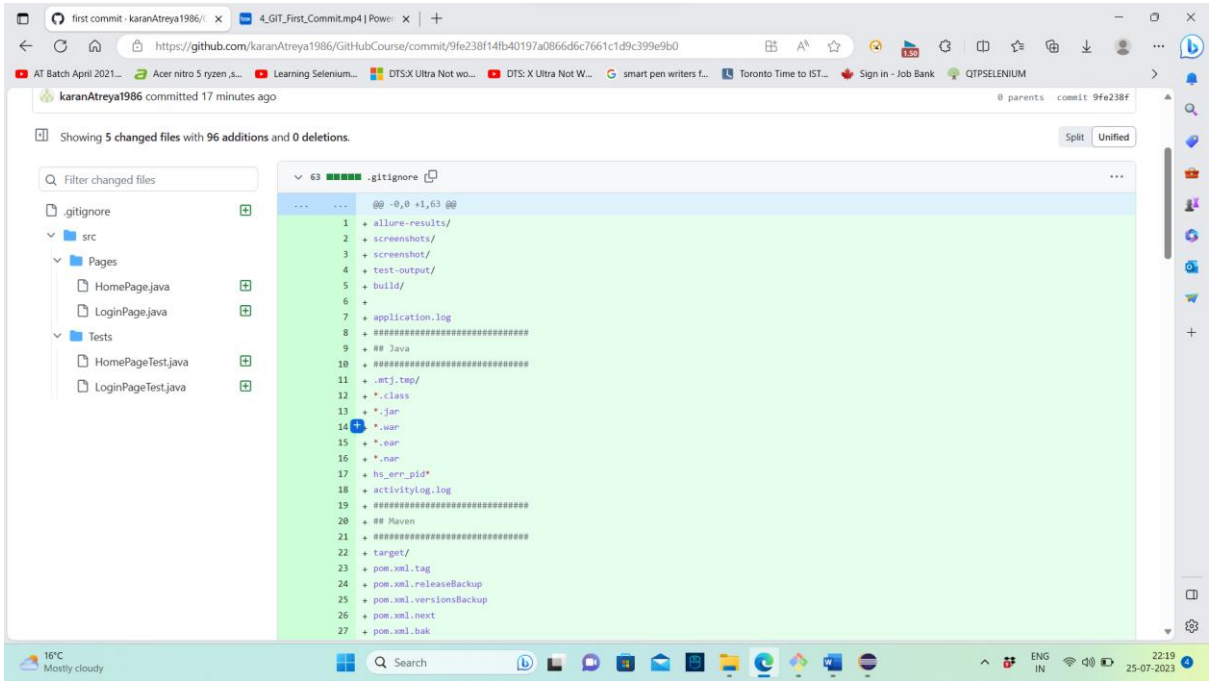
Click on commit link-



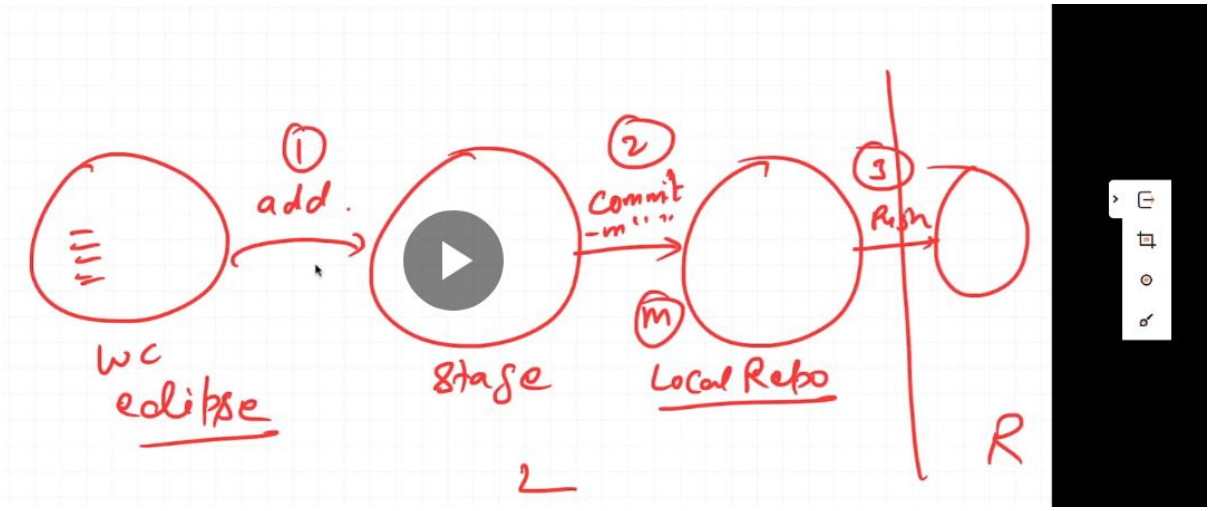
Click first commit.



You can see the changes made.



Just remember this steps and stages-

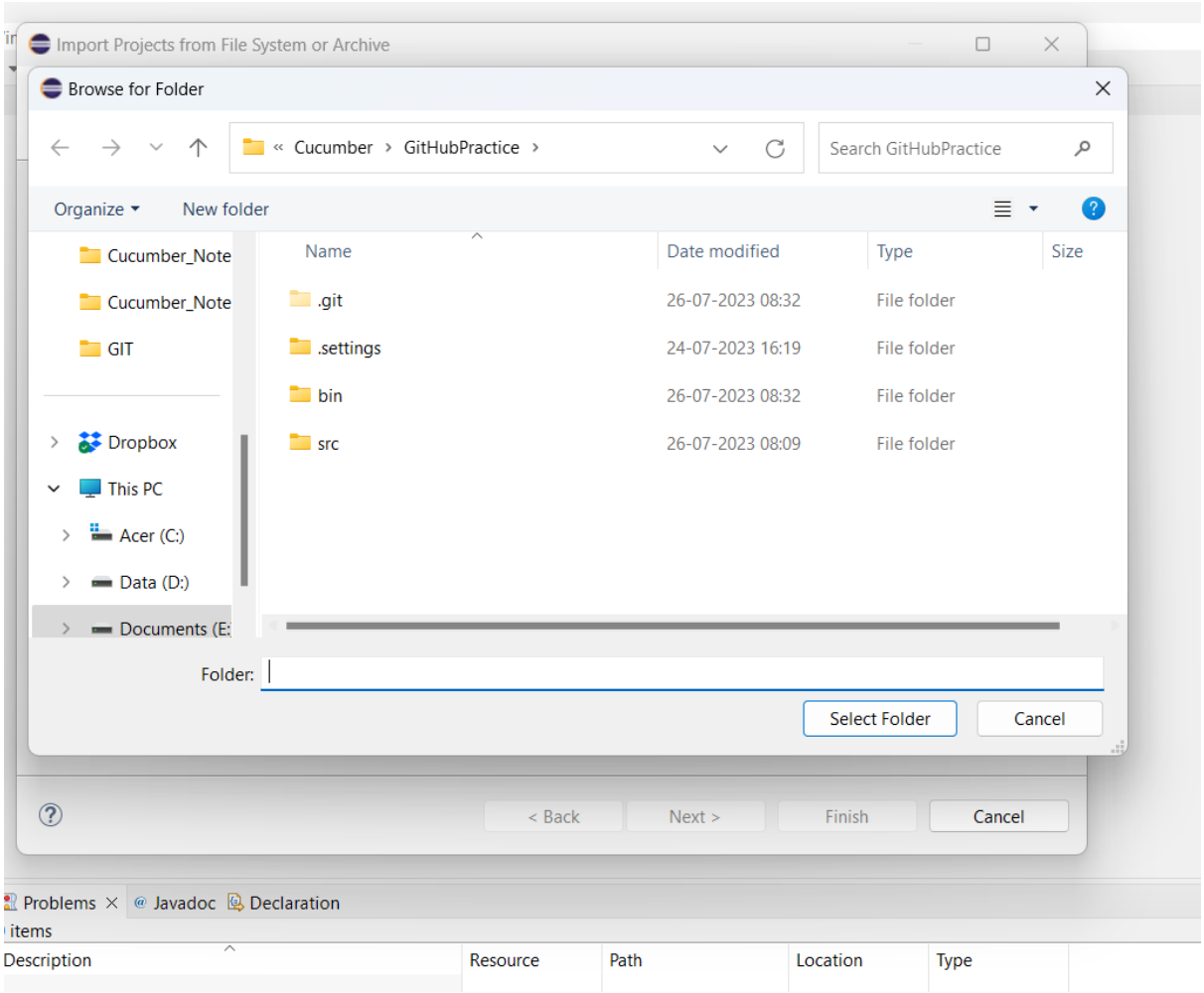


Wc means working copy of the code or the code which we write. Here we use IDE's to write the code.

By default master branch is created and pushed the first time.

To know upto which level we need to select the project-

File -> open projects from file system -> directory -> select the directory upto the first level as seen below (it should be one level above the src folder).



Eclipse workspace-

Here we need to give path till the folder under which the project has to be created.