Before we use git first time we have to follow the below commands

PS F:\java\Git\FIRSTPROJECT> git status

fatal: not a git repository (or any of the parent directories): .git

PS F:\java\Git\FIRSTPROJECT> git init -b main // -b is the branch and (main) is the name of the branch

Initialized empty Git repository in F:/java/Git/FIRSTPROJECT/.git/

PS F:\java\Git\FIRSTPROJECT> git status // after main see the status of the git with name (main

)

On branch main

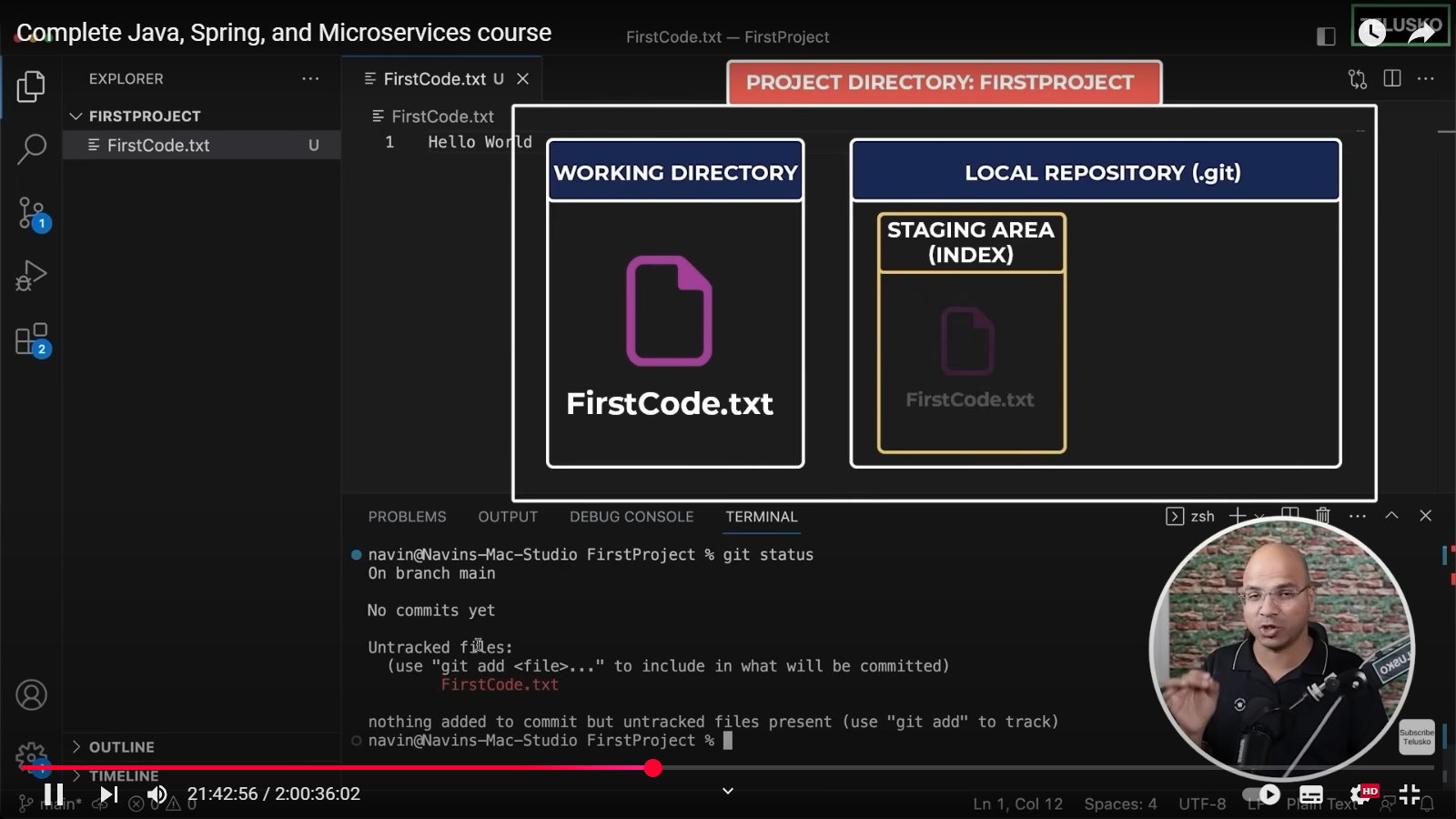
No commits yet

nothing to commit (create/copy files and use "git add" to track)

PS F:\java\Git\FIRSTPROJECT>

// create the file in FIRSTPROJECT folder (FirstCode.txt)

We will send that file to the staging area of the git to tack care of it if we use the git status without adding it shows the error likr include what is commit



PS F:\java\Git\FIRSTPROJECT> git status

On branch main

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

FirstCode.txt

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

PS F:\java\Git\FIRSTPROJECT>

// code to add the file in git

PS F:\java\Git\FIRSTPROJECT> git add FirstCode.txt (git add <filename>)

We are just sending that file to the statging environment but it is not commit yet

PS F:\java\Git\FIRSTPROJECT> git status

On branch main

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage) / to unstage that file

new file: FirstCode.txt

PS F:\java\Git\FIRSTPROJECT>

PS F:\java\Git\FIRSTPROJECT> git log // used to view all the commits commits means all the updates we done

fatal: your current branch 'main' does not have any commits yet

PS F:\java\Git\FIRSTPROJECT>

PS F:\java\Git\FIRSTPROJECT> git commit // hae to save that file in git using coming but we have to type message

hint: Waiting for your editor to close the file... unix2dos: converting file F:/java/Git/FIRSTPROJECT/.git/COMMIT\_EDITMSG to DOS format...

dos2unix: converting file F:/java/Git/FIRSTPROJECT/.git/COMMIT\_EDITMSG to Unix format...

Aborting commit due to empty commit message.

PS F:\java\Git\FIRSTPROJECT> git commit -m "this is my first commit" // like this we enter the message for that file while commiting

[main (root-commit) d071705] this is my first commit // d071705 this is check summit the get function know any changes has completed or not ?

1 file changed, 1 insertion(+)

create mode 100644 FirstCode.txt

PS F:\java\Git\FIRSTPROJECT> git status

On branch main

nothing to commit, working tree clean

PS F:\java\Git\FIRSTPROJECT>

We can also wee the log of the commit

PS F:\java\Git\FIRSTPROJECT> git log

commit d07170527016a0a3d90d836949739a7037b0b264 (HEAD -> main)

Author: phani <batthulaphanindrakumar123@gmail.com>

Date: Mon Jan 20 13:17:33 2025 +0530

this is my first commit

PS F:\java\Git\FIRSTPROJECT>

NOW WE HAVE TO MAKE SOME CHANGES IN FILE then again we have to add that to the staging using the git add file name

PS F:\java\Git\FIRSTPROJECT> git add FirstCode.txt

And we have to commit it   
PS F:\java\Git\FIRSTPROJECT> git commit -m "this is my second commit"

[main 19be76a] this is my second commit

1 file changed, 1 insertion(+), 1 deletion(-)

PS F:\java\Git\FIRSTPROJECT>  
  
we can see the log now

PS F:\java\Git\FIRSTPROJECT> git log

commit 19be76a676b583fa63cdecadd0fa1dcecbed2d93 (HEAD -> main)

Author: phani <batthulaphanindrakumar123@gmail.com>

Date: Mon Jan 20 13:28:46 2025 +0530

this is my second commit

commit d07170527016a0a3d90d836949739a7037b0b264

Author: phani <batthulaphanindrakumar123@gmail.com>

Date: Mon Jan 20 13:17:33 2025 +0530

this is my first commit

:

We can modify file again and without staging also we can perform comit

PS F:\java\Git\FIRSTPROJECT> git commit -a -m "this is my third commit" // (-a -m)

[main 4a5ef6b] this is my third commit

1 file changed, 1 insertion(+), 1 deletion(-)

PS F:\java\Git\FIRSTPROJECT> git log // note to see the previous commits continusly click o enter

TO SEE THE WHAT ARE THE CHANGES WE MADE GOT THAT WE CAN SE THE COMMAD (git diff)

PS F:\java\Git\FIRSTPROJECT> git diff

diff --git a/FirstCode.txt b/FirstCode.txt

index 5902a57..5b11030 100644

--- a/FirstCode.txt

+++ b/FirstCode.txt

@@ -1 +1,3 @@

-Hello World !!! // previous file

\ No newline at end of file

+Hello World !!!

+take input from user // changed things

+add the values

\ No newline at end of file

// NOTE : IF WE MAKE THAT FILE STAGING WE CANT THE DIFFERENCE git diff command will not work

// to view the difference after we do file staging using (git diff –staged)

PS F:\java\Git\FIRSTPROJECT> git add FirstCode.txt

PS F:\java\Git\FIRSTPROJECT> git diff

PS F:\java\Git\FIRSTPROJECT> git diff --staged

diff --git a/FirstCode.txt b/FirstCode.txt

index 5902a57..5b11030 100644

--- a/FirstCode.txt

+++ b/FirstCode.txt

@@ -1 +1,3 @@

-Hello World !!!

\ No newline at end of file

+Hello World !!!

+take input from user

+add the values

\ No newline at end of file

:

Now we create the multiple files like credit.txt and Readme.md

// ADDING ALL THE FIES TO THE STAGING using (git add .)

PS F:\java\Git\FIRSTPROJECT> git add .

PS F:\java\Git\FIRSTPROJECT> git status

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: FirstCode.txt

new file: Readme.md

new file: credits.txt

PS F:\java\Git\FIRSTPROJECT>

// removing the file from git repostry using (git rm --cached filename)

//AFTER THE ABOVE COMMAND DELETE THAT FILE IN WORK ENVIRONMENT VS CODE

PS F:\java\Git> git commit -m "the credits.txt is deleted"

//PUSHING THE LOCAL GIT TO GITHUB

1: CREATE A ACCOUNT IN GIT HUB

2: AND CREATE NEW RPOSITRY

3: TO CONNECT OUR LOCAL GIT TO REMOTE GIT GITHUB USE THIS COMMAND IN THE SAME FOLDER   
(ssh-keygen –o) ONCE IT IS COMPLETED OPEN FILE MANAGER AND GO TO THE BELOW PATH   
(C:\Users\PHANI) AND OPEN IT WITH VS CODE AND YOU CAN ABLE TO SEE THE FOLDER (.ssh) and ENTER THE BELOW COMMAND (cd.ssh) in id\_rsa.pub

File the key wil there fallow the below commands   
  
PS C:\Users\PHANI> cd .ssh

PS C:\Users\PHANI\.ssh> ls

Directory: C:\Users\PHANI\.ssh

Mode LastWriteTime Length Name

---- ------------- ------ ----

-a---- 21-01-2025 11:07 1831 id\_rsa

-a---- 21-01-2025 11:07 404 id\_rsa.pub

PS C:\Users\PHANI\.ssh> cat id\_rsa.pub

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCvtsSD/MNMy29tVPr9Lh+eQN6Ed7GOP1zTJCSmg4xM/mwo0dDRqA1odsR3c8Obh3kzOREjVl1De08s4+A1qsIGb4rwGNdUTSeSGlR2QXOc29JP/ri0CECCfs2BXElyfqvmbT1g6KonKl6F5W533HzQ+yHgIHP5Qd54i6zCCrxU8Sdtisjx7O62iFmqWEJ/4LbM7ipryvFcUAcwWapVsDk9ZI99ancNPcDKZVowkcvgMb0LczeBlEYJRFjONhUly2bvycwMy/WA1K7g4LdFlq8N0CCyKB97sxDuyNnHjmFjZZiTWdUYcDYOjkCtdpL5d2At+64nF20+tVYAJ7KOPJtV phani@DESKTOP-UVJTQ6C

// like above we can able to see the key copy it compltly and open github and click on settings and open SSH and GPG keys and click on new ssh and give the name and paste the key and click on save

And after that navigate to the folder of our project (Git-code)   
  
  
AFTER THAT USE THE BELOW COMMAND

PS F:\java\Git\git-course> git remote add origin [git@github.com:phanindhrakuma/Git-code.git](mailto:git@github.com:phanindhrakuma/Git-code.git)  
  
  
// THE ABOVE LINK WILL BE SEE IN THE GIT HUB THE REPOSTRY WE CREATED THAT IS BELONGS TO (ssh)

PS F:\java\Git\git-course> git push -u origin main

The authenticity of host 'github.com (20.207.73.82)' can't be established.

ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.

This key is not known by any other names.

Are you sure you want to continue connecting (yes/no/[fingerprint])? y

Please type 'yes', 'no' or the fingerprint: yes

Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.

Enumerating objects: 6, done.

Counting objects: 100% (6/6), done.

Delta compression using up to 8 threads

Compressing objects: 100% (3/3), done.

Writing objects: 100% (6/6), 539 bytes | 53.00 KiB/s, done.

Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To github.com:phanindhrakuma/Git-code.git

\* [new branch] main -> main

branch 'main' set up to track 'origin/main'.

PS F:\java\Git\git-course>

3

+

// we can know the origen te link we enter   
  
git remote -v

origin git@github.com:phanindhrakuma/Git-code.git (fetch)

origin git@github.com:phanindhrakuma/Git-code.git (push)

PS F:\java\Git\git-course>

// we can also do tagging   
  
PS F:\java\Git\git-course> git tag -a v1.0 -m "the first released"

// we can view the tags

PS F:\java\Git\git-course> git tag

v1.0

// we can see the info of that tag using show like below

PS F:\java\Git\git-course> git show v1.0

tag v1.0

Tagger: phani <batthulaphanindrakumar123@gmail.com>

Date: Tue Jan 21 12:20:38 2025 +0530

the first released

commit 48ac8878d87112659c0cccb1df1a429f3bc21893 (HEAD -> main, tag: v1.0, origin/main)

Author: phani <batthulaphanindrakumar123@gmail.com>

Date: Tue Jan 21 10:52:49 2025 +0530

userservice is created

// PUSHING OF THE TAG ALSO   
  
PS F:\java\Git\git-course> git tag v1.1 -m "21 jan 2025 realeased "

PS F:\java\Git\git-course> git tag

v1.0

v1.1

PS F:\java\Git\git-course> git push origin v1.0

Enumerating objects: 1, done.

Counting objects: 100% (1/1), done.

Writing objects: 100% (1/1), 173 bytes | 173.00 KiB/s, done.

Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To github.com:phanindhrakuma/Git-code.git

\* [new tag] v1.0 -> v1.0

PS F:\java\Git\git-course>  
  
// WE CAN EDIT THE FILE AND WE CAN SEND THAT TO GITHUG   
  
PS F:\java\Git\git-course> git add USERSERVICE.txt

PS F:\java\Git\git-course> git commit -m "processing new data text"

PS F:\java\Git\git-course> git push origin main

Enumerating objects: 5, done.

Delta compression using up to 8 threads

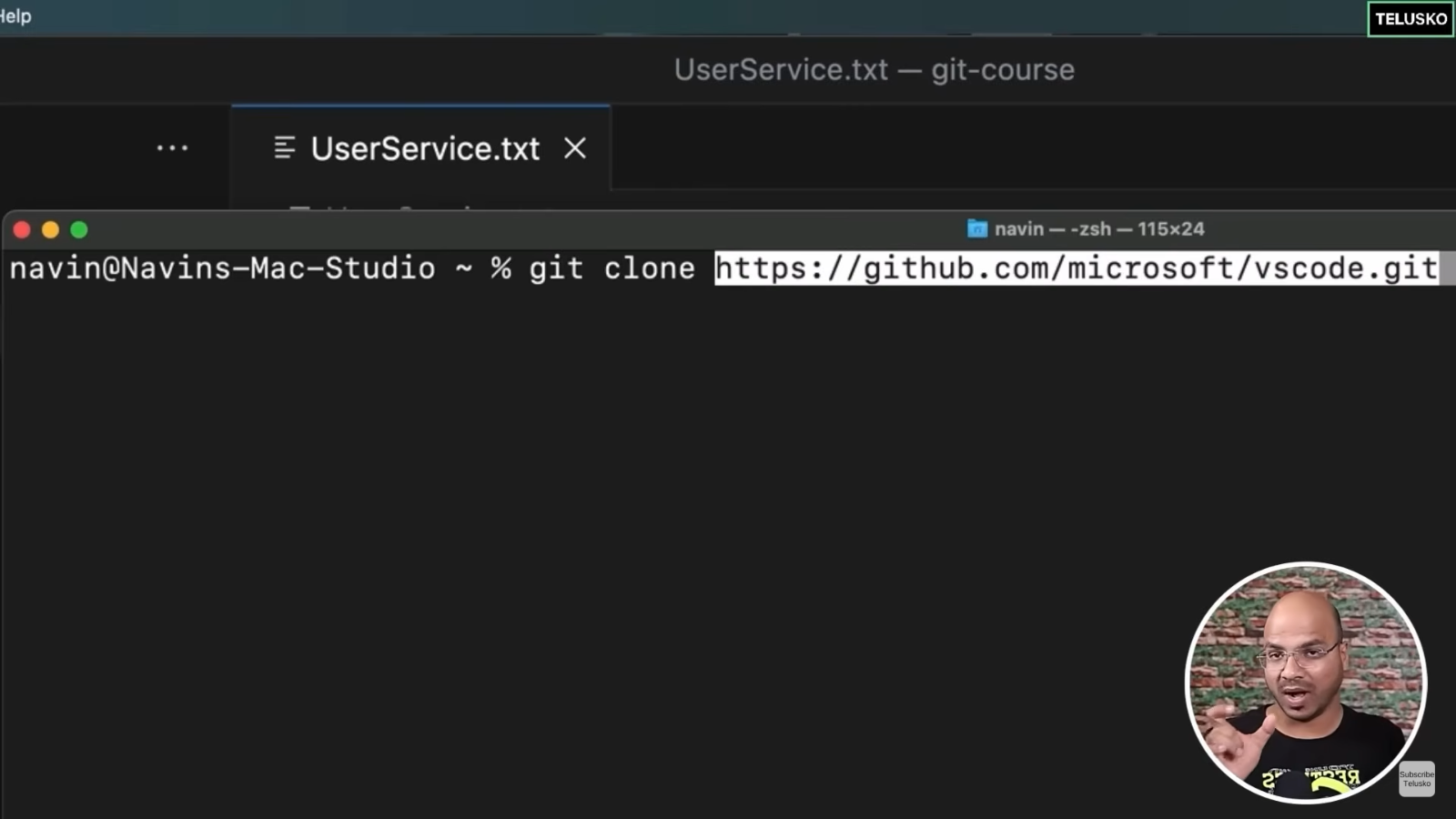
Compressing objects: 100% (2/2), done.

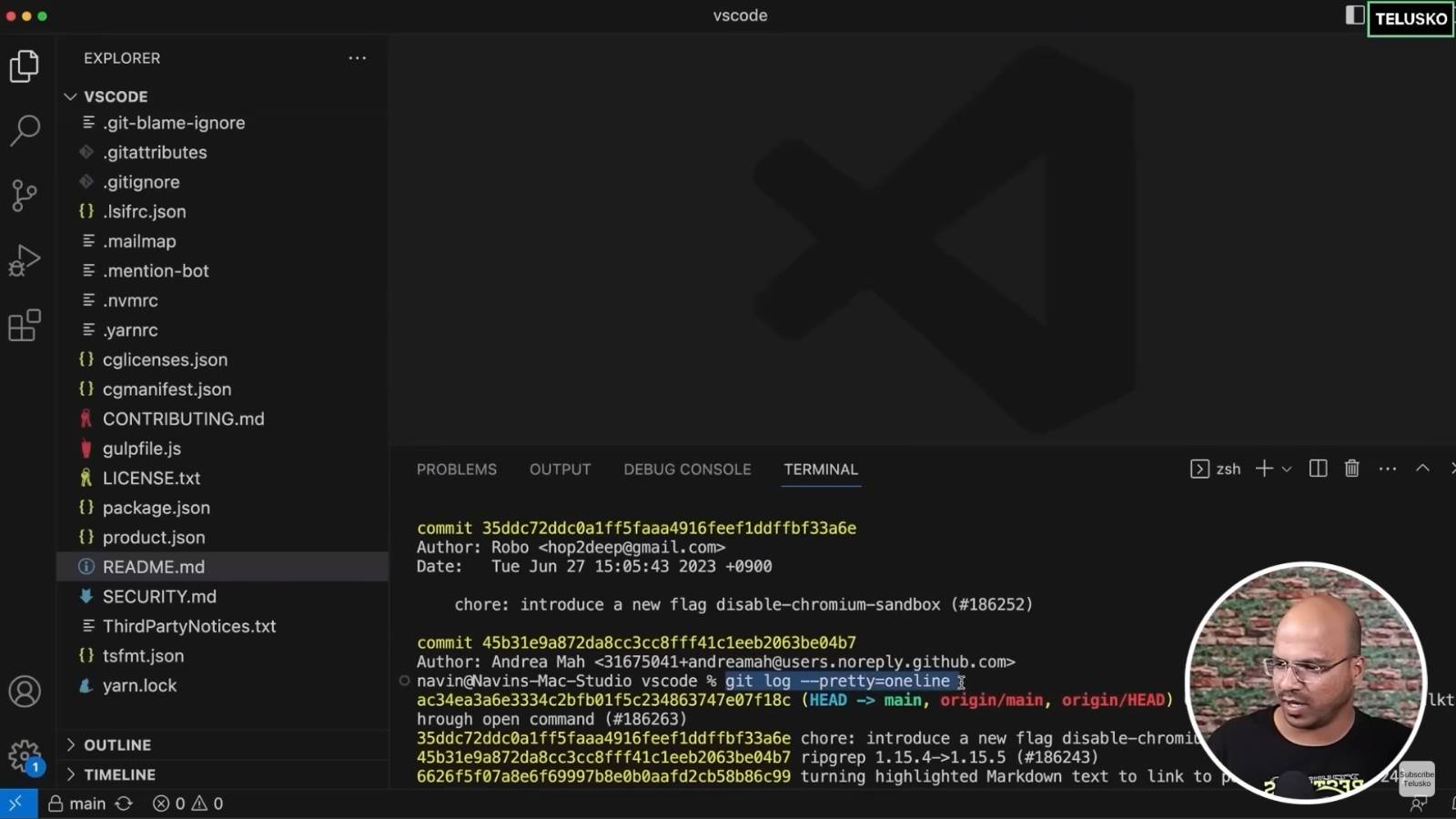
Writing objects: 100% (3/3), 334 bytes | 11.00 KiB/s, done.

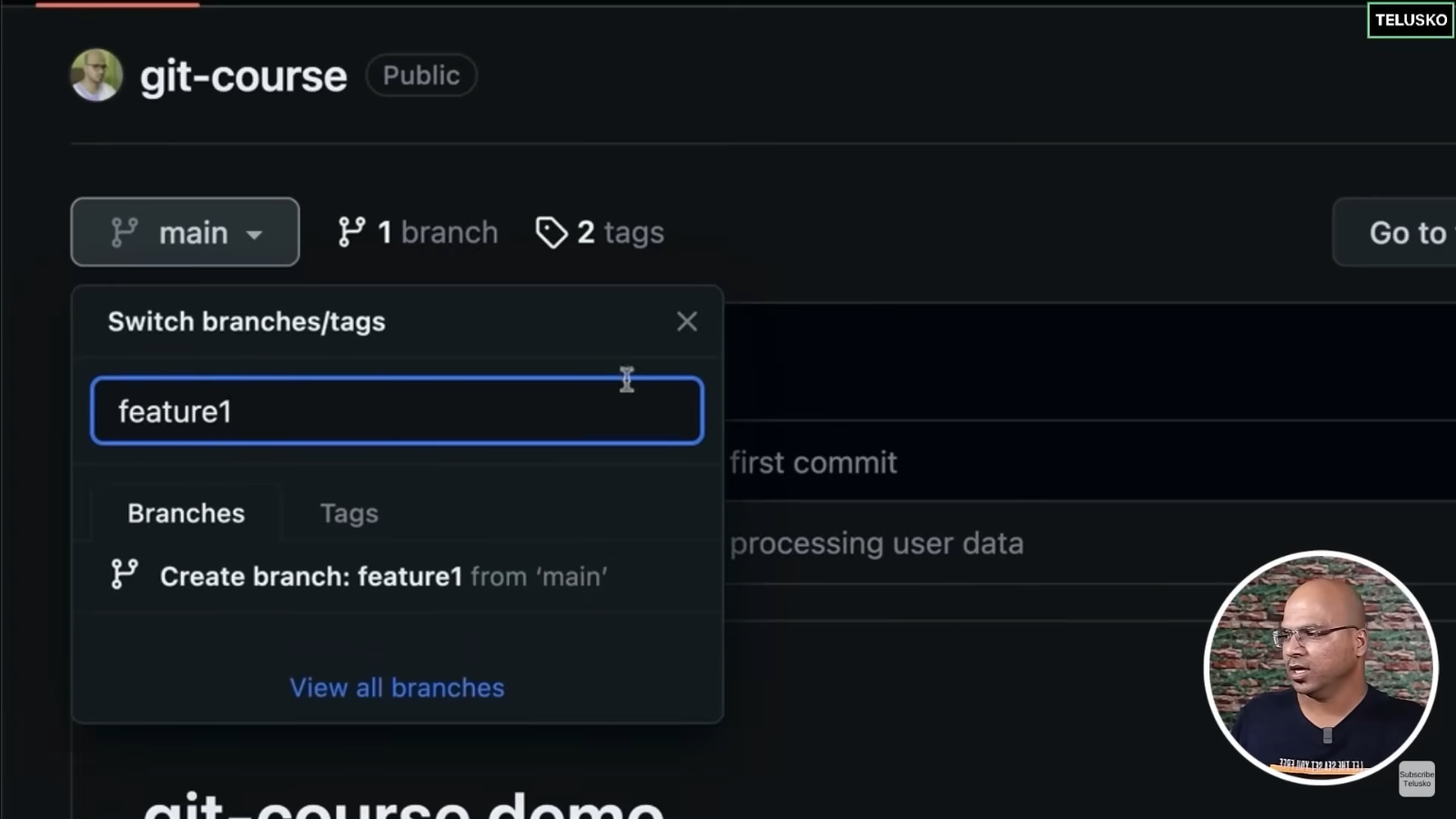
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To github.com:phanindhrakuma/Git-code.git

48ac887..685d794 main -> main

// WE CAN DOWNLOAD THE CODE IN GITHUB ( COP THE HTTS )   
  


We only see that commits and messages on one line   
  


:w ca create the branch in the github   


// we can also create branch using the command

PS F:\java\Git\git-course> git status

On branch main

Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

PS F:\java\Git\git-course> git checkout -b feature1

Switched to a new branch 'feature1'  
  
// seeing the branches   
  
PS F:\java\Git\git-course> git branch

\* feature1 // star is there in the left side so this branch is active

main

PS F:\java\Git\git-course>

Make the changes in the file and save it and follow the below because of the feature1 branch is active so the changes we made will not effect another branch (main)   
  
On branch main

Your branch is up to date with 'origin/main'.

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: USERSERVICE.txt

no changes added to commit (use "git add" and/or "git commit -a")

PS F:\java\Git\git-course> git add USERSERVICE.txt

PS F:\java\Git\git-course> git commit -m "processing new data text"

PS F:\java\Git\git-course> git push origin main

Enumerating objects: 5, done.

Delta compression using up to 8 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (3/3), 334 bytes | 11.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To github.com:phanindhrakuma/Git-code.git

48ac887..685d794 main -> main

PS F:\java\Git\git-course> git tag

v1.0

v1.1

PS F:\java\Git\git-course> git push origin v1.0

Enumerating objects: 1, done.

Counting objects: 100% (1/1), done.

PS F:\java\Git\git-course> git status

On branch main

Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

PS F:\java\Git\git-course> git checkout -b feature1

Switched to a new branch 'feature1'

main

PS F:\java\Git\git-course> git status

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

modified: USERSERVICE.txt

no changes added to commit (use "git add" and/or "git commit -a")

PS F:\java\Git\git-course> git add USERSERVICE.txt

PS F:\java\Git\git-course> git commit -m " experimenting user "

[feature1 e1d7346] experimenting user

1 file changed, 3 insertions(+), 1 deletion(-)

PS F:\java\Git\git-course> git status

On branch feature1

nothing to commit, working tree clean

PS F:\java\Git\git-course> git log

commit e1d7346bb1ce361106ca41ac5924474510468b4e (HEAD -> feature1)

Author: phani <batthulaphanindrakumar123@gmail.com>

Date: Tue Jan 21 13:22:51 2025 +0530

experimenting user

commit 685d794eb4e5da79fc62db911bd1697a8da0e5f2 (tag: v1.1, origin/main, main)

Author: phani <batthulaphanindrakumar123@gmail.com>

Date: Tue Jan 21 12:27:00 2025 +0530

processing new data text

commit 48ac8878d87112659c0cccb1df1a429f3bc21893 (tag: v1.0)

Author: phani <batthulaphanindrakumar123@gmail.com>

Date: Tue Jan 21 10:52:49 2025 +0530

userservice is created

commit 3b96452d87883317e195bf9e7311daf5bc6b63d0

Author: phani <batthulaphanindrakumar123@gmail.com>

Date: Tue Jan 21 10:49:36 2025 +0530

this is my first git commit git-course

(END)  
  
// we can swich the branches like below

PS F:\java\Git\git-course> git switch main

Switched to branch 'main'

Your branch is up to date with 'origin/main'.

PS F:\java\Git\git-course> git switch feature1

Switched to branch 'feature1'

PS F:\java\Git\git-course>

// WE SWITCHING THE BRANCHES THOSE WILL BECAME ACTIVE   
  
PS F:\java\Git\git-course> git switch feature1

Switched to branch 'feature1'

PS F:\java\Git\git-course> git branch

\* feature1

main

PS F:\java\Git\git-course> git switch main

Switched to branch 'main'

Your branch is up to date with 'origin/main'.

PS F:\java\Git\git-course> git branch

feature1

\* main

PS F:\java\Git\git-course>

// CREATING THENNEW BRANCHES USING ANOTHER COMMAND

PS F:\java\Git\git-course> git switch -c feature2

Switched to a new branch 'feature2'

PS F:\java\Git\git-course> git switch -c feature3

Switched to a new branch 'feature3'

PS F:\java\Git\git-course>  
  
  
// WE CAN SEE ALL BRANCHES PRESENT IS LOCAL AND REMOTE BRANCH   
PS F:\java\Git\git-course> git branch --all

feature1

feature2

\* feature3

remotes/origin/main

Switched to branch 'feature2'

// WE CAN O=MOVE TO BREVIOUS BRANCH USING T=BELOW COMMAND

PS F:\java\Git\git-course> git switch -

Switched to branch 'feature3'

PS F:\java\Git\git-course> git witch feature1

git: 'witch' is not a git command. See 'git --help'.

The most similar command is

Switch

// WE CAN DIRECTLY SWITCH TO THE BRANCH ALSO WITH NAME

PS F:\java\Git\git-course> git switch feature1

Switched to branch 'feature1'

PS F:\java\Git\git-course> git branch

\* feature1

feature2

feature3

main

PS F:\java\Git\git-course>  
  
// DELETING THE BRANCH   
  
PS F:\java\Git\git-course> git branch -d feature3

Deleted branch feature3 (was e1d7346).

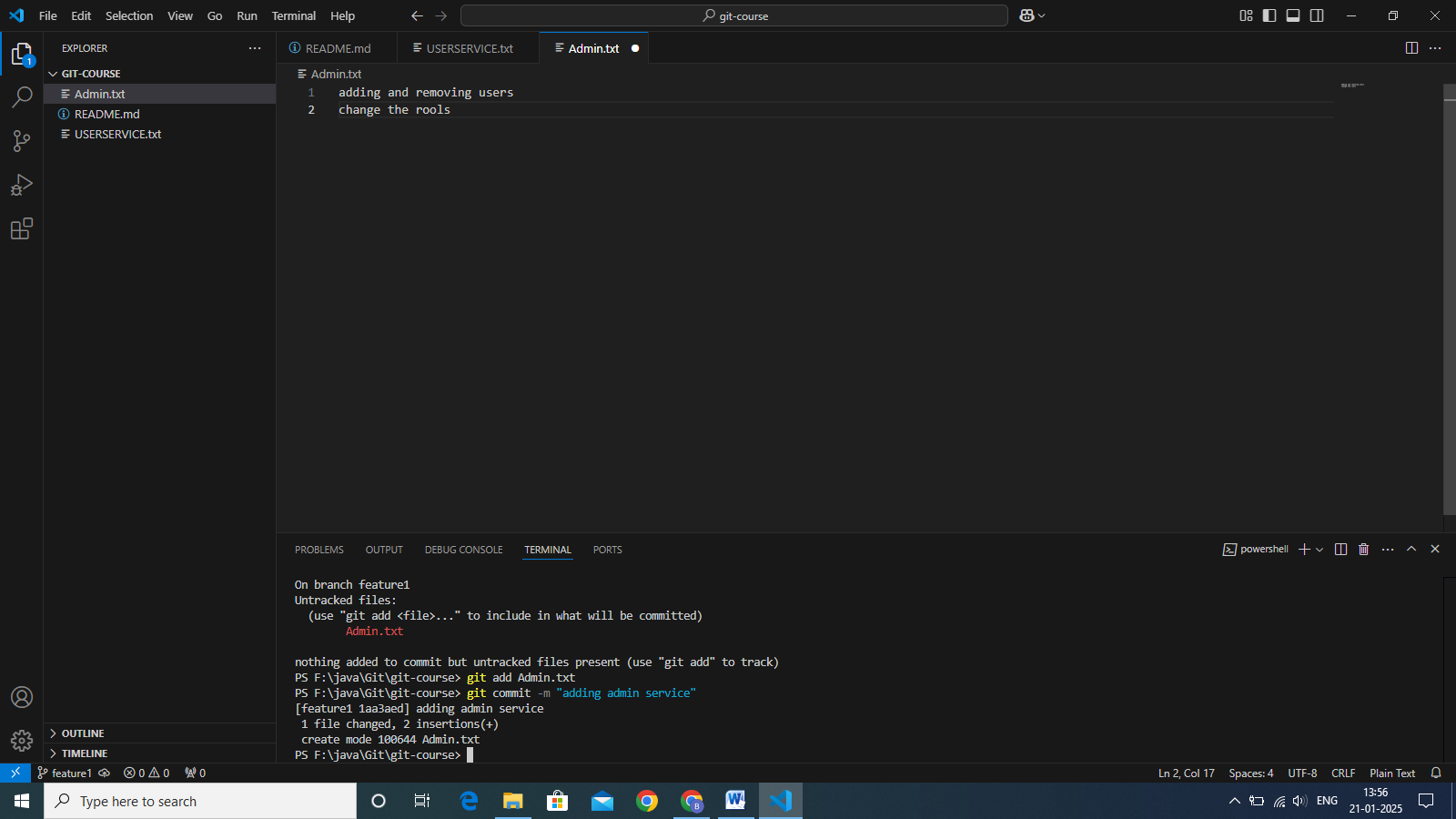
PS F:\java\Git\git-course> git branch

\* feature1

feature2

main

PS F:\java\Git\git-course>  
  
//CREATING THE NEW FILE IN FEATURE1 BRANCH ADMIN.TXT IN VS CODE



Your branch is up to date with 'origin/main'.

PS F:\java\Git\git-course> git switch feature1

\* feature1

main

PS F:\java\Git\git-course> git status

On branch feature1

Untracked files:

(use "git add <file>..." to include in what will be committed)

Admin.txt

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

PS F:\java\Git\git-course> git add Admin.txt

PS F:\java\Git\git-course> git commit -m "adding admin service"

[feature1 1aa3aed] adding admin service

1 file changed, 2 insertions(+)

create mode 100644 Admin.txt

PS F:\java\Git\git-course>

// pushing the branch   
  
PS F:\java\Git\git-course> git switch feature1

Switched to branch 'feature1'

PS F:\java\Git\git-course> git push origin feature1

Enumerating objects: 8, done.

Delta compression using up to 8 threads

Compressing objects: 100% (5/5), done.

Writing objects: 100% (6/6), 680 bytes | 680.00 KiB/s, done.

Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

remote:

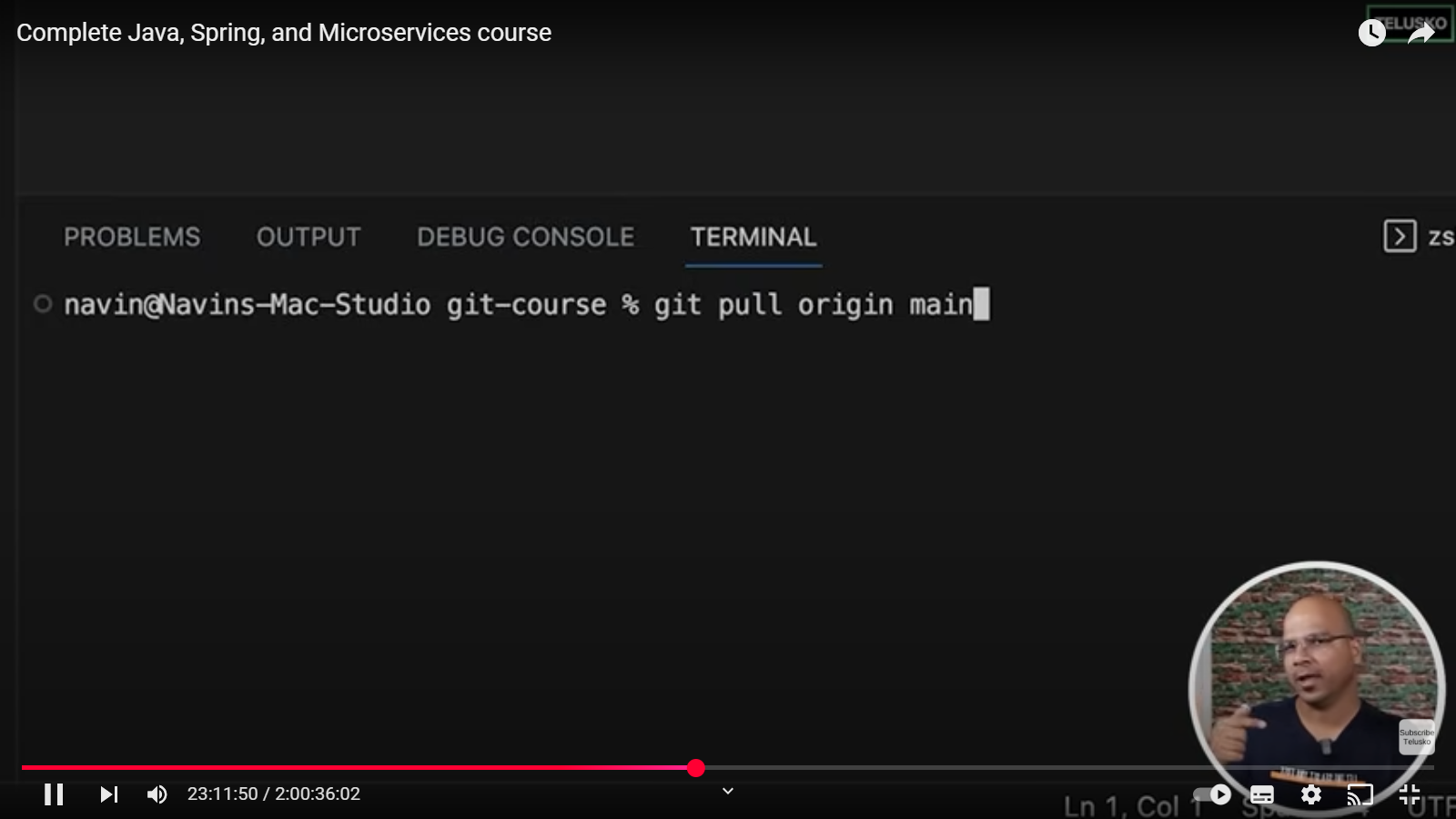
remote: Create a pull request for 'feature1' on GitHub by visiting:

remote: https://github.com/phanindhrakuma/Git-code/pull/new/feature1

remote:

To github.com:phanindhrakuma/Git-code.git

\* [new branch] feature1 -> feature1

//MERGING THE BRANCHES // merging o feature1 in main branch   
  
// before merging we have to pull the main branch   
  
// FIRST WE HAVE TO SWITCH TO THE MAIN BRANCH THEN ONLY WE MERGE   
PS F:\java\Git\git-course> git branch

\* feature1

main

PS F:\java\Git\git-course> git switch main

Switched to branch 'main'

Your branch is up to date with 'origin/main'.

PS F:\java\Git\git-course> git branch

feature1

\* main  
  
// COMMAND TO MERGE THE BRANCH

PS F:\java\Git\git-course> git merge feature1

Updating 685d794..1aa3aed

Fast-forward

Admin.txt | 2 ++

USERSERVICE.txt | 4 +++-

2 files changed, 5 insertions(+), 1 deletion(-)

create mode 100644 Admin.txt

PS F:\java\Git\git-course>

// pushing to the remote server from local   
  
PS F:\java\Git\git-course> git status

On branch main

Your branch is ahead of 'origin/main' by 2 commits.

(use "git push" to publish your local commits)

nothing to commit, working tree clean

PS F:\java\Git\git-course> git push origin main

Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To github.com:phanindhrakuma/Git-code.git

685d794..1aa3aed main -> main

PS F:\java\Git\git-course>