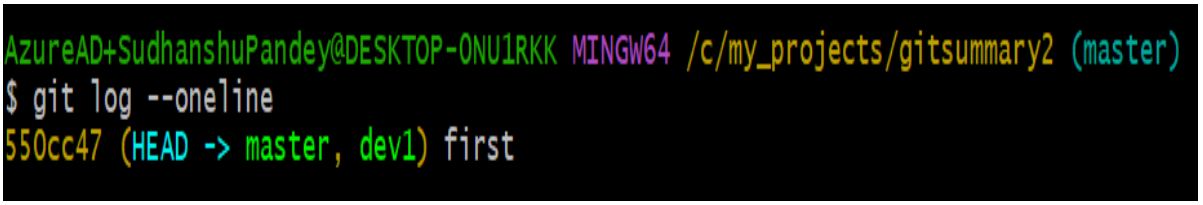
Git And GitHub Session No.05

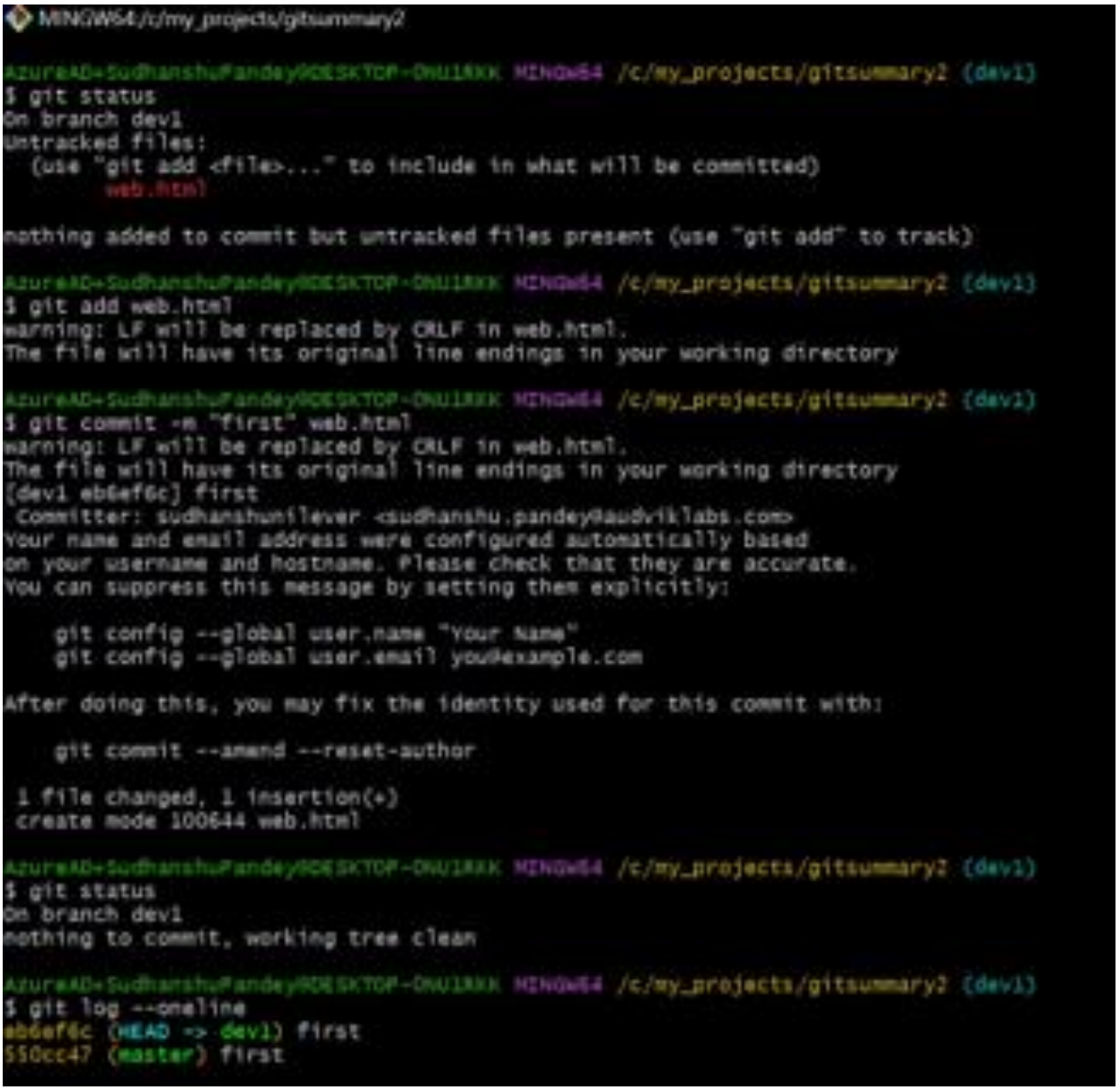
Summary 27-02-2022

- **#git cat-file -p <commit id>** -> It will just show us the data(Delta part) and parent "commit id" we had at this particular commit, Here without bringing back data to the staging area from the commit area and then further to the working area, As we have seen with "reset" and checkout" commands, It will show us the information.
- **Branching** - Branching is a concept of git, Which provides us a good way of management of our code and team, For example, we know that In a project, different developers are working on different functionalities, then making the changes directly in the master branch is the worst practice, good practice is to create respective branches and make changes there, And only after the permission of the reviewer merge it to master branch, As well as Whenever we have to do some enhancement in our project then also we create a branch we do changes there, and finally merge it with the master branch.
- **#git branch** -> It will tell us all the branches we have in the current repository And will highlight our current branch, by default it is "master" or "main"
- **#git branch --show-current** -> It shows the name of the current branch.
- **#git branch dev1** -> It will create a new branch "dev1" and it will have a whole timeline reference of the "master" branch



- **#git log –online** -> You will see both branches are pointing to the HEAD commit
- **#git checkout dev1** -> Now we will switch to the "dev1" branch
- Now as we are inside the "dev1" branch, here also we can **modify**, **create**, and **delete** any file data, And all these changes will be just happening in the "dev1" branch, In the master branch, nothing of these changes will reflect.

- **# vim web.html** -> Create a file with some data inside the “dev1” branch, now we can add this file to the staging area, and then commit it into the commit area, Now if we see **#git log –oneline**, We will see our dev1 branch is pointing to the new commit and our master branch is still pointing to the old commit. Hence Here our dev1 branch is one commit **ahead** of the master branch.



- **#git merge dev1** -> It will merge the dev1 branch into the master branch, So if we now do **#ls** inside the master branch now we can see the “web.html” file is present in the master branch.

```
AzureAD+SudhanshuPandey@DESKTOP-0NU1RKK MINGW64 /c/my_projects/gitsummary2 (dev1)
$ git checkout master
Switched to branch 'master'

AzureAD+SudhanshuPandey@DESKTOP-0NU1RKK MINGW64 /c/my_projects/gitsummary2 (master)
$ git merge dev1
Updating 550cc47..eb6ef6c
Fast-forward
 web.html | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 web.html
```

- Now as the “dev1” branch is merged with the “master” branch,

```
a.txt  web.html

AzureAD+SudhanshuPandey@DESKTOP-0NU1RKK MINGW64 /c/my_projects/gitsummary2 (master)
$ git log --oneline
eb6ef6c (HEAD -> master, dev1) first
550cc47 first
```

```
AzureAD+SudhanshuPandey@DESKTOP-0NU1RKK MINGW64 /c/my_projects/gitsummary2 (master)
$ git log --oneline --graph --decorate
* eb6ef6c (HEAD -> master, dev1) first
* 550cc47 first
```

- hence now both branches are pointing to the same commit.
- **#git branch -d dev1** -> It will delete the “dev1” branch.