**61)Git Reflog**

* The term Réflogs is a short form for Reference Logs.
* These are just logs that git keeps us for as a record
* Git keeps a record of when the tips of branches and other references were updated in the repo.
* We can view and update these reflogs using the git reflog command.
* Git only keeps reflogs on your local activity (local repository). They are not shared with collaborators
* Reflogs also expire. Git cleans out old entries after around 90 days, though this can be configured.
* The git reflog command accepts subcommands like Show, expire, delete, and exists.
* Show is the only command used variant, and it is the default subcommand.
* git reflog show command will show the log of a specific reference (it defaults to HEAD).
* For example, to view the logs for the tip of the main branch we could run
* git reflog show main

All the reflogs will be stored in the logs sub-folder in the git folder. In that sub-folder there will be a HEAD file all the logs are stored in that file.

The reflog contains all the changes we did like commit, switch, checkout every git command we used.

git reflog show head

git reflog show newfeature

The above log contains all logs for newfeature branch.

git reflog show index

The above log contains all logs for index branch.

**62)Traversing the reflogs**

git reflog show head

We get 14 logs(1-14).

git reflog show head@{3}

We get logs(3-14).

When we do git checkout Head~2 we get to commit 0263e88.

At that point if we do online we only have 2 commits because at that point in time we had only 2 commits.

Text

Description automatically generated

git reflog show Head@{2} shows all logs from 2-end

git reflog show master shows all logs in master branch

Graphical user interface, text

Description automatically generated

**We can use for time too**

git reflog show [master@{l.day.ago}](mailto:master@%7bl.day.ago%7d)

git reflog show [Head@{l.day.ago}](mailto:Head@%7bl.day.ago%7d)

git reflog show head@{yesterday}

git diff head head@{yesterday} Difference between head current and head yesterday

git diff main main@{yesterday}

git diff main main@{3} Difference between main current and main@action3

**63)Get back lost commits in a branch with reflogs**

Let’s create a folder and do two commits in it features file, feature 1 and feature 2 file

Now when we do git reflog show master we will see three items in log.

Now we learn that feature 2 is not required so we reset it.

Text

Description automatically generated

The reset command gets the repo to the point where only feature 1 commit just happened.

Now we got an update that feature 2 is required.

To get that commit we can go to reflog

git reflog show master we get the log and it has feature 2 hash

Text

Description automatically generated

Now we can get that feature 2 hash and do

git reset 55fbe9e -- hard

Text

Description automatically generated

Below we have 4 commits and reset to 2nd commit

Text

Description automatically generated

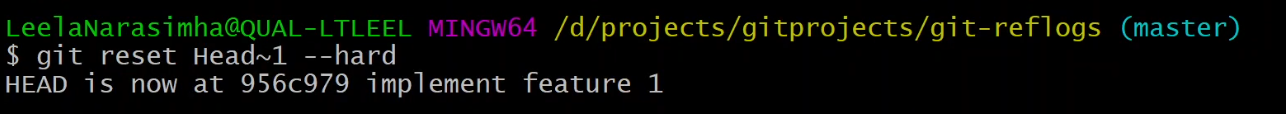
We can get feature 2 by

git reset master@{4} –hard from reflog

Text

Description automatically generated

We can get the last feature reset to undo by resetting head~1 from reflog



Text

Description automatically generated

Text

Description automatically generated

**64)Create Git Aliases for commonly used Git Commands**

The Configuration levels in Git are Project level to User level to System level

We need to set them in the aliases in the system file

System file is located at git config-- list--show-origin

Global level git config-- list--show-origin--global

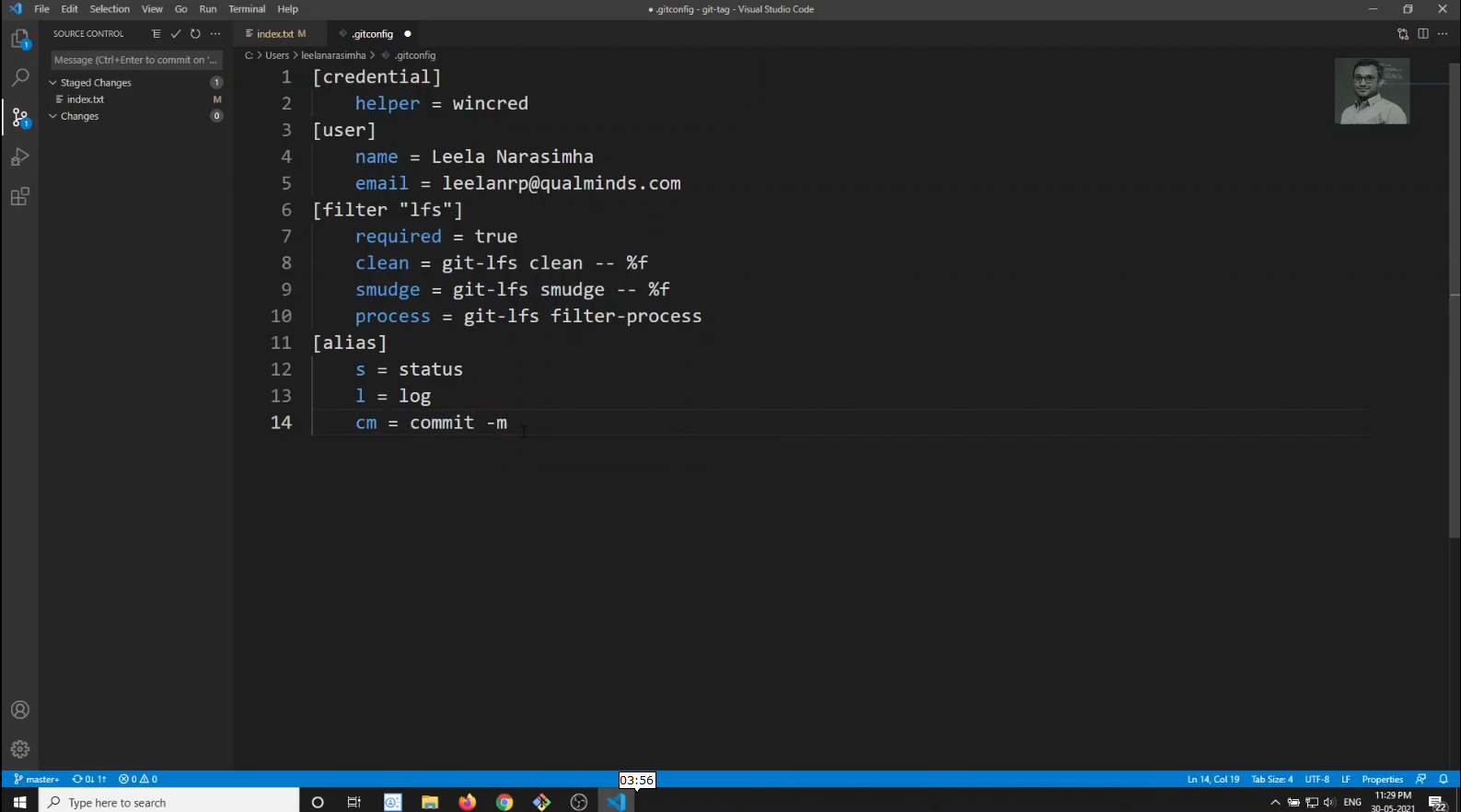
CDrive

Users

Username

.gitconfig open in visual studio code

Add in aliases like below



Then we can type git s, git l, git cm