MIT WORLD PEACE UNIVERSITY

TYBTECH, Sem- 1

Full Stack Development (FSD)

ASSIGNMENT 1-

Version control with Git

PB-45

Paras Raut

BATCH-B2

# Aim:

Version control with Git.

# Objectives:

1. To introduce the concepts and software behind version control, using the example of Git.
2. To understand the use of &#39;version control&#39; in the context of a coding project.
3. To learn Git version control with Clone, commit to, and push, pull from a git repository.

# Theory:

**1. What is Git? What is Version Control?**

"Git, a distributed version control system (VCS), plays a pivotal role in modern software development. Introduced by Linus Torvalds in 2005, Git has risen to prominence as a widely adopted version control system, revolutionizing the way developers manage source code and collaborate on software projects.

In essence, version control systems empower multiple individuals, typically software developers, to collaboratively work on a project concurrently. They serve as the backbone for tracking file modifications over time, offering a comprehensive history of alterations and facilitating seamless teamwork within development teams. The significance of version control cannot be overstated, particularly in the realm of software development.

Git offers a rich set of capabilities that distinguishes it from the rest:

1. **Change Monitoring**: Git meticulously records every change made to a project's files, providing an unambiguous account of additions, modifications, or deletions at any given juncture.
2. **Branching Flexibility**: Git's branch feature empowers developers to create separate lines of development. This means that different team members can independently tackle various features or bug fixes without interfering with the main codebase. Later, these branches can be seamlessly merged back into the primary codebase.
3. **Facilitating Collaboration**: Git fosters collaboration by facilitating concurrent work on the same project. It equips developers with tools to distribute and amalgamate changes made by different team members.
4. **Historical Insight and Rollback**: Git meticulously maintains an exhaustive historical log of changes, simplifying the process of reverting to a previous code state when necessary. This functionality is indispensable for troubleshooting and rectifying issues stemming from recent modifications.
5. **Decentralized Structure**: As a distributed version control system, Git endows each developer with a comprehensive project history copy on their local machine. This decentralized structure ensures resilience and adaptability, catering to both small-scale and large-scale development teams.
6. **Open Source Nature**: Git embraces an open-source ethos, making it freely available for use and customization. It boasts a vibrant and engaged community of users and contributors, ensuring its continuous growth and improvement.

In conclusion, Git stands as a formidable instrument for overseeing and tracing transformations within software projects. Its role in enhancing collaborative development, code maintenance, and the overall integrity of software over time cannot be overstated."

# 2. How to use Git for version controlling?

Using Git for version control involves several key steps and commands. Here's a step- by-step guide on how to use Git for version control:

1. \*\*Install Git:\*\*

- If Git is not already installed on your system, you can download and install it from the official Git website (https://git-scm.com/).

1. \*\*Configure Git:\*\*

- Before you start using Git, you should configure your name and email address to associate with your commits. Use the following commands to set your Git identity:

```

git config --global user.name "Your Name"

git config --global user.email "[your.email@example.com](mailto:your.email@example.com)"

```

1. \*\*Initialize a Git Repository: \*\*

- To start version controlling a project, navigate to the project's root directory in your terminal and run the following command to initialize a Git repository:

```

git init

```

1. \*\*Add Files to the Repository: \*\*

- You can add files to the staging area using the `git add` command. For example, to add all files in the current directory and its subdirectories, use:

```

git add .

```

1. \*\*Commit Changes:\*\*

- Once your changes are staged, you can commit them with a descriptive message using the `git commit` command:

```

git commit -m "Your commit message here"

```

1. \*\*Create Branches (Optional):\*\*

- If you're working on a new feature or bug fix, it's a good practice to create a new branch for your work. Use the following commands to create and switch to a new branch:

```

git checkout -b new-feature-branch

```

1. \*\*Work on Your Code:\*\*

- Make changes to your code within your branch.

1. \*\*Stage and Commit Changes:\*\*

- As you make changes, continue to use `git add` and `git commit` to save your progress with meaningful commit messages.

1. \*\*Merge Branches (Optional):\*\*

- If you've created a feature branch and your work is complete, you can merge it back into the main branch (e.g., `main` or `master`) using the following commands:

```

git checkout main

git merge new-feature-branch

```

1. \*\*Push to a Remote Repository (Optional):\*\*

- If you're collaborating with others or using a remote Git hosting service (e.g., GitHub, GitLab), you can push your changes to the remote repository:

```

git push origin main

```

1. \*\*Pull Changes from a Remote Repository (Optional):\*\*

- To update your local repository with changes made by others, use the `git pull` command:

```

git pull origin main

```

1. \*\*Review History:\*\*

- You can view the commit history and changes using `git log`:

```

git log

```

1. \*\*Tagging (Optional):\*\*

- You can create tags to mark specific points in your project's history, such as release versions:

```

git tag -a v1.0 -m "Version 1.0"

```

These are the essential steps and commands for using Git for version control. Git provides powerful tools for tracking changes, collaborating with others, and maintaining a history of your project's development. Depending on your project's complexity and team workflow, you may explore more advanced Git features and workflows.

# FAQ’s:

1. **What is branching in Git?**

Branching within Git serves as a cornerstone concept, empowering developers to pursue distinct avenues of development within a single repository. Each branch embodies an autonomous trajectory of modifications, commonly employed for introducing new features, addressing bugs, or conducting experimentation on code alterations, all while safeguarding the integrity of the primary codebase. This branching mechanism offers a degree of isolation, enabling developers to concurrently engage in multiple tasks.

Git's branching mechanism is characterized by its lightweight and streamlined design, rendering it an essential asset for fostering collaborative and well-structured software development.

1. **How to create and merge branches in Git? Write the commands used.** Creating and merging branches in Git involves a series of commands. Here are the steps with the corresponding Git commands:

*\*\*To create a new branch:\*\**

Use the `git checkout -b` command to create and switch to a new branch simultaneously. Replace `branch-name` with the desired branch name:

```bash

git checkout -b branch-name

```

This command creates a new branch and switches to it, allowing you to start making changes within the new branch.

*\*\*To switch between branches:\*\**

Use the `git checkout` command followed by the branch name to switch between branches:

```bash

git checkout branch-name

```

This command allows you to move between different branches in your Git repository.

*\*\*To list all branches:\*\**

To see a list of all branches in your repository and see which branch you are currently on, use:

```bash git branch

```

The branch with an asterisk (`\*`) next to it is the currently checked out branch.

*\*\*To merge a branch into another branch:\*\**

To merge changes from one branch into another (e.g., merging a feature branch into the main branch), follow these steps:

1. First, ensure you are on the branch you want to merge changes into. For example, to merge a feature branch into the main branch:

```bash

git checkout main

```

1. Then, use the `git merge` command followed by the branch name you want to merge:

```bash

git merge branch-name

```

This command merges the changes from `branch-name` into the currently checked out branch (in this case, `main`).

1. If there are no conflicts between the branches, Git will perform an automatic merge. If there are conflicts, you will need to manually resolve them and commit the changes.

*\*\*To delete a branch (after merging):\*\**

Once you've merged a branch and no longer need it, you can delete it using the `-d` option:

```bash

git branch -d branch-name

```

Or, if you want to forcefully delete the branch without checking if changes are unmerged (use with caution):

```bash

git branch -D branch-name

```

These commands cover the basics of creating, switching between, merging, and deleting branches in Git. Branching is a powerful feature that helps organize and manage development workflows effectively.

# Output:

# The following is the copied text of the git bash command line.

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~

$ mkdir FSDLab1

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~

$ ls

'3D Objects'/

AppData/

'Application Data'@

'Cisco Packet Tracer 6.2sv'/

Contacts/

Cookies@

Desktop/

Documents/

Downloads/

FSDLab1/

Favorites/

GNS3/

Histogram.ipynb

IntelGraphicsProfiles/

'Lab1-Data cleaning PB16.ipynb'

Links/

'Local Settings'@

MicrosoftEdgeBackups/

Music/

'My Documents'@

NTUSER.DAT

NTUSER.DAT{4ce2c1ad-5d59-11ec-931c-ecb1d75edcf2}.TM.blf

NTUSER.DAT{4ce2c1ad-5d59-11ec-931c-ecb1d75edcf2}.TMContainer00000000000000000001.regtrans-ms

NTUSER.DAT{4ce2c1ad-5d59-11ec-931c-ecb1d75edcf2}.TMContainer00000000000000000002.regtrans-ms

NetHood@

OneAPI.ipynb

OneDrive/

'PRIYAL DWDM ASSIGNMENT 1.ipynb'

Pictures/

PrintHood@

'Priyal DWDM.ipynb'

Recent@

'Saved Games'/

Searches/

SendTo@

'Start Menu'@

Templates@

Untitled1.ipynb

Untitled2.ipynb

Videos/

WorldCupMatches\_new.csv

adult.csv

anaconda3/

combined\_csv.csv

diamonds.csv

eclipse/

eclipse-workspace/

heatmap\_pearson.png

lab/

'lab 3.12.21.ipynb'

myproject/

myproject1/

ntuser.dat.LOG1

ntuser.dat.LOG2

ntuser.ini

sdpod\_agent/

sharan/

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~

$ cd FSDLab1/

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1

$ echo "Hello World"

Hello World

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1

$ echo "Hello World" >>sampleFile.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1

$ cat sampleFile.txt

Hello World

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1

$ git init

Initialized empty Git repository in C:/Users/computer/FSDLab1/.git/

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git add sampleFile.txt

warning: in the working copy of 'sampleFile.txt', LF will be replaced by CRLF th

e next time Git touches it

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git status

On branch master

No commits yet

Changes to be committed:

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

new file: sampleFile.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ echo

'

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ echo "Hello World" >> sampleFile2.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git status

On branch master

No commits yet

Changes to be committed:

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

new file: sampleFile.txt

Untracked files:

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

sampleFile2.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git add sampleFile2.txt

warning: in the working copy of 'sampleFile2.txt', LF will be replaced by CRLF t

he next time Git touches it

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git status

On branch master

No commits yet

Changes to be committed:

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

new file: sampleFile.txt

new file: sampleFile2.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git commit -m "Two new Files Added"

[master (root-commit) ecd6b01] Two new Files Added

Committer: computer <computer@WPUDESK48422.mitwpu.edu.in>

Your name and email address were configured automatically based

on your username and hostname. Please check that they are accurate.

You can suppress this message by setting them explicitly. Run the

following command and follow the instructions in your editor to edit

your configuration file:

git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

2 files changed, 2 insertions(+)

create mode 100644 sampleFile.txt

create mode 100644 sampleFile2.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git status

On branch master

nothing to commit, working tree clean

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ echo "Bye Bye World" >> sampleFile2.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git status

On branch master

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: sampleFile2.txt

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

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git add sampleFile2.txt

warning: in the working copy of 'sampleFile2.txt', LF will be replaced by CRLF t

he next time Git touches it

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git status

On branch master

Changes to be committed:

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

modified: sampleFile2.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git commit -m "Second File Added"

[master 7e792aa] Second File Added

Committer: computer <computer@WPUDESK48422.mitwpu.edu.in>

Your name and email address were configured automatically based

on your username and hostname. Please check that they are accurate.

You can suppress this message by setting them explicitly. Run the

following command and follow the instructions in your editor to edit

your configuration file:

git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

1 file changed, 1 insertion(+)

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git status

On branch master

nothing to commit, working tree clean

Aborting commit due to empty commit message.

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git commit -m "third commit"

[master 8cde9f0] third commit

Committer: computer <computer@WPUDESK48422.mitwpu.edu.in>

Your name and email address were configured automatically based

on your username and hostname. Please check that they are accurate.

You can suppress this message by setting them explicitly. Run the

following command and follow the instructions in your editor to edit

your configuration file:

git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

1 file changed, 1 insertion(+)

create mode 100644 sampleFile3.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git status

On branch master

nothing to commit, working tree clean

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git log

commit 8cde9f0a4459e8b7d635fd39ec36093d0c74f7b6 (HEAD -> master)

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 06:02:41 2023 +0530

third commit

commit 7e792aa83e086c16e45673aeebf65d4e0743f230

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 05:57:07 2023 +0530

Second File Added

commit ecd6b018488b6889640e74dc8e285eaa10599780

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 05:53:52 2023 +0530

Two new Files Added

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git checkout -b "7e792aa83e086c16e45673aeebf65d4e0743f230"

Switched to a new branch '7e792aa83e086c16e45673aeebf65d4e0743f230'

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (7e792aa83e086c16e45673aeebf65d4e0743f230)

$ git status

On branch 7e792aa83e086c16e45673aeebf65d4e0743f230

nothing to commit, working tree clean

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (7e792aa83e086c16e45673aeebf65d4e0743f230)

$ git log

commit 8cde9f0a4459e8b7d635fd39ec36093d0c74f7b6 (HEAD -> 7e792aa83e086c16e45673aeebf65d4e0743f230, master)

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 06:02:41 2023 +0530

third commit

commit 7e792aa83e086c16e45673aeebf65d4e0743f230

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 05:57:07 2023 +0530

Second File Added

commit ecd6b018488b6889640e74dc8e285eaa10599780

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 05:53:52 2023 +0530

Two new Files Added

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (7e792aa83e086c16e45673aeebf65d4e0743f230)

$ echo "This is the new file" >> sampleFile4.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (7e792aa83e086c16e45673aeebf65d4e0743f230)

$ git add sampleFile4.txt

warning: in the working copy of 'sampleFile4.txt', LF will be replaced by CRLF the next time Git touches it

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (7e792aa83e086c16e45673aeebf65d4e0743f230)

$ git status

On branch 7e792aa83e086c16e45673aeebf65d4e0743f230

Changes to be committed:

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

new file: sampleFile4.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (7e792aa83e086c16e45673aeebf65d4e0743f230)

$ git commit -m "Fourth File Created"

[7e792aa83e086c16e45673aeebf65d4e0743f230 5e68b04] Fourth File Created

Committer: computer <computer@WPUDESK48422.mitwpu.edu.in>

Your name and email address were configured automatically based

on your username and hostname. Please check that they are accurate.

You can suppress this message by setting them explicitly. Run the

following command and follow the instructions in your editor to edit

your configuration file:

git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

1 file changed, 1 insertion(+)

create mode 100644 sampleFile4.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (7e792aa83e086c16e45673aeebf65d4e0743f230)

$ git status

On branch 7e792aa83e086c16e45673aeebf65d4e0743f230

nothing to commit, working tree clean

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (7e792aa83e086c16e45673aeebf65d4e0743f230)

$ git log

commit 5e68b0407ff5d43aae1ae552eea341203dd2ea33 (HEAD -> 7e792aa83e086c16e45673aeebf65d4e0743f230)

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 06:07:54 2023 +0530

Fourth File Created

commit 8cde9f0a4459e8b7d635fd39ec36093d0c74f7b6 (master)

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 06:02:41 2023 +0530

third commit

commit 7e792aa83e086c16e45673aeebf65d4e0743f230

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 05:57:07 2023 +0530

Second File Added

commit ecd6b018488b6889640e74dc8e285eaa10599780

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 05:53:52 2023 +0530

Two new Files Added

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (7e792aa83e086c16e45673aeebf65d4e0743f230)

$ git checkout -b master

fatal: a branch named 'master' already exists

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (7e792aa83e086c16e45673aeebf65d4e0743f230)

$ git switch master

Switched to branch 'master'

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git log

commit 8cde9f0a4459e8b7d635fd39ec36093d0c74f7b6 (HEAD -> master)

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 06:02:41 2023 +0530

third commit

commit 7e792aa83e086c16e45673aeebf65d4e0743f230

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 05:57:07 2023 +0530

Second File Added

commit ecd6b018488b6889640e74dc8e285eaa10599780

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 05:53:52 2023 +0530

Two new Files Added

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git merge dev

merge: dev - not something we can merge

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git branch master

fatal: a branch named 'master' already exists

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git branch dev

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git switch dev

Switched to branch 'dev'

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (dev)

$ echo "last file" >> last.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (dev)

$ git add last.txt

warning: in the working copy of 'last.txt', LF will be replaced by CRLF the next time Git touches it

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (dev)

$ git commit -m "last file added"

[dev 2329345] last file added

Committer: computer <computer@WPUDESK48422.mitwpu.edu.in>

Your name and email address were configured automatically based

on your username and hostname. Please check that they are accurate.

You can suppress this message by setting them explicitly. Run the

following command and follow the instructions in your editor to edit

your configuration file:

git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

1 file changed, 1 insertion(+)

create mode 100644 last.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (dev)

$ git status

On branch dev

nothing to commit, working tree clean

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (dev)

$ git log

commit 2329345828e2fc50f3dc006fbd9d2ee07e1b2897 (HEAD -> dev)

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 06:23:41 2023 +0530

last file added

commit 8cde9f0a4459e8b7d635fd39ec36093d0c74f7b6 (master)

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 06:02:41 2023 +0530

third commit

commit 7e792aa83e086c16e45673aeebf65d4e0743f230

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 05:57:07 2023 +0530

Second File Added

commit ecd6b018488b6889640e74dc8e285eaa10599780

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 05:53:52 2023 +0530

Two new Files Added

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (dev)

$ git switch master

Switched to branch 'master'

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git merge dev

Updating 8cde9f0..2329345

Fast-forward

last.txt | 1 +

1 file changed, 1 insertion(+)

create mode 100644 last.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git log

commit 2329345828e2fc50f3dc006fbd9d2ee07e1b2897 (HEAD -> master, dev)

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 06:23:41 2023 +0530

last file added

commit 8cde9f0a4459e8b7d635fd39ec36093d0c74f7b6

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 06:02:41 2023 +0530

third commit

commit 7e792aa83e086c16e45673aeebf65d4e0743f230

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 05:57:07 2023 +0530

Second File Added

commit ecd6b018488b6889640e74dc8e285eaa10599780

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 05:53:52 2023 +0530

Two new Files Added

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git branch dev2

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ echo "new branch" >> newfile.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git add newfile.txt

warning: in the working copy of 'newfile.txt', LF will be replaced by CRLF the next time Git touches it

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git commit -m "dev 2 created"

[master 13161a7] dev 2 created

Committer: computer <computer@WPUDESK48422.mitwpu.edu.in>

Your name and email address were configured automatically based

on your username and hostname. Please check that they are accurate.

You can suppress this message by setting them explicitly. Run the

following command and follow the instructions in your editor to edit

your configuration file:

git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

1 file changed, 1 insertion(+)

create mode 100644 newfile.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git status

On branch master

nothing to commit, working tree clean

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git switch dev2

Switched to branch 'dev2'

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (dev2)

$ echo "new branch file" >> dev2file.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (dev2)

$ git add dev2file.txt

warning: in the working copy of 'dev2file.txt', LF will be replaced by CRLF the next time Git touches it

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (dev2)

$ git commit -m "changes made to dev 2"

[dev2 8f2d2bb] changes made to dev 2

Committer: computer <computer@WPUDESK48422.mitwpu.edu.in>

Your name and email address were configured automatically based

on your username and hostname. Please check that they are accurate.

You can suppress this message by setting them explicitly. Run the

following command and follow the instructions in your editor to edit

your configuration file:

git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

1 file changed, 1 insertion(+)

create mode 100644 dev2file.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (dev2)

$ git status

On branch dev2

nothing to commit, working tree clean

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (dev2)

$ git switch master

Switched to branch 'master'

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git merge dev2

Merge made by the 'ort' strategy.

dev2file.txt | 1 +

1 file changed, 1 insertion(+)

create mode 100644 dev2file.txt

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git status

On branch master

nothing to commit, working tree clean

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git log

commit 55a3e44a5b8a13847620eb667f11f5e13242ea08 (HEAD -> master)

Merge: 13161a7 8f2d2bb

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 06:33:04 2023 +0530

Merge branch 'dev2'

commit 8f2d2bb534e8f823a37490a3f44d119b2fa70a0f (dev2)

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 06:32:09 2023 +0530

changes made to dev 2

commit 13161a7175201e8ff8adf49b67fb0e2d20bd0e20

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 06:30:22 2023 +0530

dev 2 created

commit 2329345828e2fc50f3dc006fbd9d2ee07e1b2897 (dev)

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 06:23:41 2023 +0530

last file added

commit 8cde9f0a4459e8b7d635fd39ec36093d0c74f7b6

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 06:02:41 2023 +0530

third commit

commit 7e792aa83e086c16e45673aeebf65d4e0743f230

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 05:57:07 2023 +0530

Second File Added

commit ecd6b018488b6889640e74dc8e285eaa10599780

!q:

/usr/bin/bash: line 1: q:: command not found

!done (press RETURN)

...skipping...

commit 8f2d2bb534e8f823a37490a3f44d119b2fa70a0f (dev2)

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 06:32:09 2023 +0530

changes made to dev 2

commit 13161a7175201e8ff8adf49b67fb0e2d20bd0e20

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 06:30:22 2023 +0530

dev 2 created

commit 2329345828e2fc50f3dc006fbd9d2ee07e1b2897 (dev)

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 06:23:41 2023 +0530

last file added

commit 8cde9f0a4459e8b7d635fd39ec36093d0c74f7b6

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 06:02:41 2023 +0530

third commit

commit 7e792aa83e086c16e45673aeebf65d4e0743f230

Author: computer <computer@WPUDESK48422.mitwpu.edu.in>

Date: Tue Aug 8 05:57:07 2023 +0530

Second File Added

commit ecd6b018488b6889640e74dc8e285eaa10599780

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$ git status

On branch master

nothing to commit, working tree clean

WPUDESK48422+computer@WPUDESK48422 MINGW64 ~/FSDLab1 (master)

$

Link For the Repo Created :