VTU 2025 Scheme

Project Management (with Git)

1BCSL307A – 3rd Semester

EXPERIMENT – 10

Git History and Commit Filtering

Display Commits by Author and Date Range

Aim

To display all Git commits made by the author `JohnDoe` between January 1, 2025 and December 31, 2025 using Git log filtering options.

Objectives

After completing this experiment, the student will be able to:

- Understand Git commit history.
- Display commit history using `git log`.
- Filter commits based on author name.
- Filter commits based on a starting date.
- Filter commits based on an ending date.
- Combine author and date filters.
- Display filtered commits in a readable format.
- Understand the use of `--author`, `--since`, and `--until` options.

Software Requirements

- Git
- Windows Command Prompt / PowerShell / Git Bash
- Existing Git repository containing commit history

Theory

Git stores information about every commit created in a repository.

A commit normally contains information such as:

- Commit ID
- Author name

- Author email
- Commit date
- Commit message

The `git log` command is used to display this commit history.

In repositories containing many commits, Git log options can be used to display only the required commits.

For this experiment, the commit history is filtered using:

- Author name: `JohnDoe`
- Starting date: January 1, 2025
- Ending date: December 31, 2025

Author Filter

The `--author` option is used to display commits matching a specified author.

Syntax:

```
git log --author="author-name"
```

For this experiment:

```
git log --author="JohnDoe"
```

Date Filters

The `--since` option specifies the starting date.

Example:

```
git log --since="2025-01-01"
```

The `--until` option specifies the ending date.

Example:

```
git log --until="2025-12-31 23:59:59"
```

These options can be combined with the author filter.

Filtering Concept

The following diagram shows how Git filters commit history.

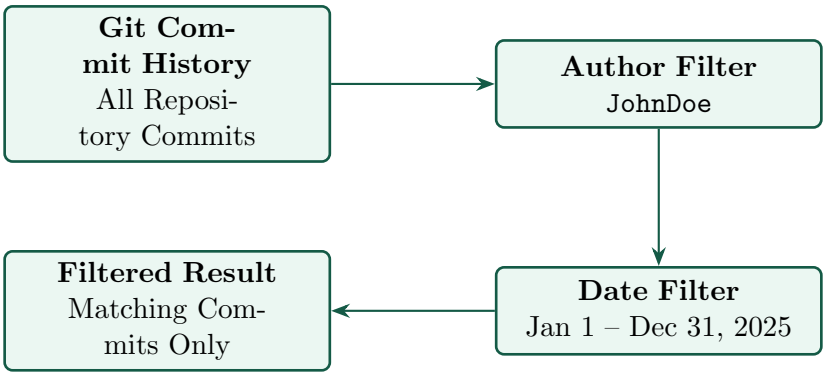


Diagram Explanation:

- Git first contains the complete commit history.
- The author filter selects commits matching `JohnDoe`.
- The date filter limits the commits to the required period.
- Only matching commits are displayed in the final result.

Important Commands

Command	Purpose
`git log`	Displays Git commit history.
`git log --oneline`	Displays commit history in compact form.
`git log --author="JohnDoe"`	Displays commits matching author JohnDoe.
`git log --since="2025-01-01"`	Displays commits after the specified starting date.
`git log --until="2025-12-31 23:59:59"`	Limits the log to the specified ending date and time.

Procedure

Step 1: Open Command Prompt, PowerShell, or Git Bash.

Step 2: Open the required Git repository.

Step 3: Display the complete commit history.

Step 4: Verify that the repository contains commit records.

Step 5: Filter the commit history using author name `JohnDoe`.

Step 6: Apply the starting date filter.

Step 7: Apply the ending date filter.

Step 8: Combine the author and date filters.

Step 9: Display the matching commits.

Step 10: Display the result in a compact readable format.

Step 11: Verify the author, date, and commit message.

Execution

Step 1: Open the Git Repository

Open Command Prompt, PowerShell, or Git Bash.
Move into the required Git repository.
Example:

```
cd git-project
```

Step 2: Verify Repository Status

Execute:

```
git status
```

This confirms that the current directory is a Git repository.

Step 3: Display Complete Commit History

Execute:

```
git log
```

Sample Output:

```
commit a1b2c3d...
Author: JohnDoe <john@example.com>
Date:   Mon Jun 16 10:30:00 2025 +0530

    Update project documentation

commit e4f5g6h...
Author: Student <student@example.com>
Date:   Tue May 20 11:00:00 2025 +0530

    Add project module
```

The actual output depends on the repository.

Step 4: Display Compact History

Execute:

```
git log --oneline
```

Sample Output:

```
a1b2c3d Update project documentation
e4f5g6h Add project module
h7i8j9k Fix application error
```

Step 5: Filter Commits by Author

Execute:

```
git log --author="JohnDoe"
```

This displays commits whose author information matches `JohnDoe`.

Step 6: Filter by Starting Date

Execute:

```
git log --since="2025-01-01 00:00:00"
```

This limits the displayed commit history using January 1, 2025 as the starting date.

Step 7: Filter by Ending Date

Execute:

```
git log --until="2025-12-31 23:59:59"
```

This limits the displayed commit history using December 31, 2025 as the ending date.

Step 8: Combine Author and Date Filters

Execute:

```
git log --author="JohnDoe" --since="2025-01-01 00:00:00" --until="2025-12-31
    23:59:59"
```

Explanation:
This command filters the Git commit history using all three conditions:

- Author must match `JohnDoe`.
- Commit must satisfy the starting date filter.
- Commit must satisfy the ending date filter.

Step 9: Display Filtered Result in Compact Format

Execute:

```
git log --author="JohnDoe" --since="2025-01-01 00:00:00" --until="2025-12-31
    23:59:59" --oneline
```

Sample Output:

```
a1b2c3d Update project documentation
d4e5f6g Add login feature
h7i8j9k Fix database connection
```

Only commits matching the specified filters are displayed.

Step 10: Display Author, Date, and Message Clearly

For a more readable output, execute:

```
git log --author="JohnDoe" --since="2025-01-01 00:00:00" --until="2025-12-31
    23:59:59" --pretty=format:"%h | %an | %ad | %s" --date=short
```

Sample Output:

```
a1b2c3d | JohnDoe | 2025-06-16 | Update project documentation
d4e5f6g | JohnDoe | 2025-04-10 | Add login feature
h7i8j9k | JohnDoe | 2025-02-05 | Fix database connection
```

Understanding the Format Symbols

The following format is used:

```
--pretty=format:"%h | %an | %ad | %s"
```

The symbols represent:

Symbol	Meaning
%h	Short commit ID.
%an	Author name.
%ad	Author date.
%s	Commit message or subject.

Main Command for Experiment 10

The main command required for this experiment is:

```
git log --author="JohnDoe" --since="2025-01-01 00:00:00" --until="2025-12-31
    23:59:59"
```

For a simple compact output:

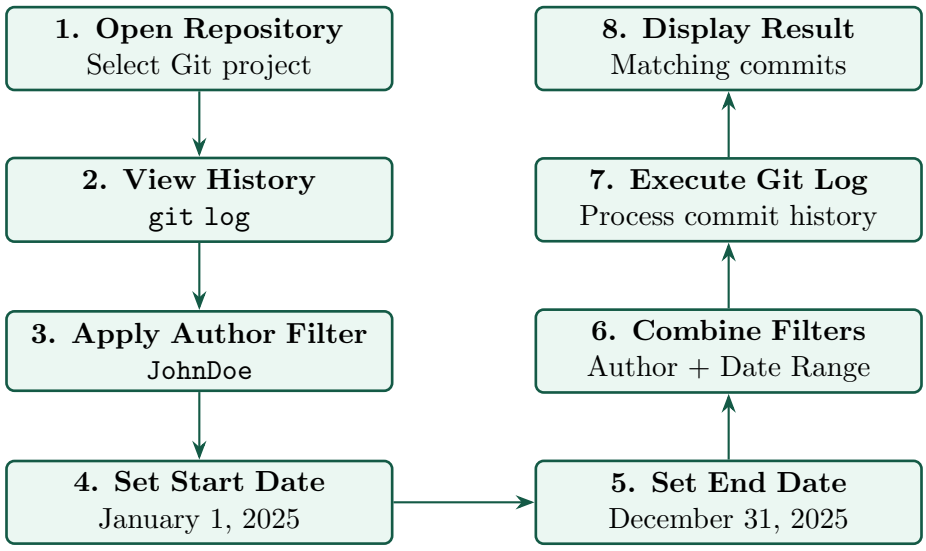
```
git log --author="JohnDoe" --since="2025-01-01 00:00:00" --until="2025-12-31
    23:59:59" --oneline
```

Complete Command Sequence

```
cd git-project

git status

git log

git log --oneline

git log --author="JohnDoe"

git log --since="2025-01-01 00:00:00"

git log --until="2025-12-31 23:59:59"

git log --author="JohnDoe" --since="2025-01-01 00:00:00" --until="2025-12-31
    23:59:59"

git log --author="JohnDoe" --since="2025-01-01 00:00:00" --until="2025-12-31
    23:59:59" --oneline
```


Experiment Workflow

The complete workflow of Experiment 10 is shown below.



Workflow Explanation:

1. Open the required Git repository.
2. Display the complete Git commit history.
3. Apply the author filter for `JohnDoe`.
4. Specify January 1, 2025 as the starting date.
5. Specify December 31, 2025 as the ending date.
6. Combine all filters in one Git log command.
7. Git searches the commit history.
8. Only matching commits are displayed.

Git Log Filtering Options

Option	Purpose
`--author="JohnDoe"`	Filters commits based on matching author information.
`--since="date"`	Sets the starting date condition for the log.
`--until="date"`	Sets the ending date condition for the log.
`--oneline`	Displays each commit in a compact single-line format.
`--date=short`	Displays dates in YYYY-MM-DD format.
`--pretty=format:"..."`	Allows a custom output format.

Result

Thus, the Git commit history was successfully filtered to display commits matching the author `JohnDoe` within the specified date range from January 1, 2025 to December 31, 2025 using Git log filtering options.

Viva Questions and Answers

1. **Which command is used to view Git commit history?**

```
git log
```

2. **What is the purpose of the –author option?**

   It filters the Git commit history based on matching author information.

3. **How do you display commits by JohnDoe?**

```
git log --author="JohnDoe"
```

4. **What is the purpose of –since?**

   It specifies the starting date condition used when filtering the commit history.

5. **What is the purpose of –until?**

   It specifies the ending date condition used when filtering the commit history.

6. **What is the author used in this experiment?**

   The author used is:

   ```
   JohnDoe
   ```

7. **What is the date range used in this experiment?**

   January 1, 2025 to December 31, 2025.

8. **Can multiple filters be used with git log?**

   Yes. Author and date filters can be combined in the same command.

9. **Which option displays commit history in compact form?**

   ```
   --oneline
   ```

10. **What does %h represent in pretty format?**

    It represents the shortened commit hash.

11. **What does %an represent?**

    It represents the author name.

12. **What does %ad represent?**

    It represents the author date.

13. **What does %s represent?**

    It represents the commit subject or commit message.

14. **How can we display dates in short format?**

    Use:

    ```
    --date=short
    ```

15. **What happens if no commit matches the filters?**

    Git displays no matching commit entries.

Conclusion

In this experiment, Git commit history filtering was demonstrated. The `git log` command was used with author and date options to display commits matching `JohnDoe` for the required period. The filtered history can also be displayed in compact or customized formats for easier analysis.