Git Version Control Laboratory Report

TP Git Exercises Documentation

Instructor: Imane Fouad
Report: Version Control Systems
Author: Imad Boutbaoucht
Date: September 25, 2025

Contents

1 Introduction **3**

2 Objectives **3**

3 Git Configuration **3**

3.1 Objective 3

3.2 Commands Used and Explanations 3

3.3 Results 4

4 Repository Initialization **4**

4.1 Objective 4

4.2 Commands Used and Explanations 5

4.3 Results 6

5 Branching **9**

5.1 Objective 9

5.2 Commands Used and Explanations 9

5.3 Vegetables Branch Development 10

5.4 Sauces Branch Development 14

6 Merging **14**

6.1 Objective 14

6.2 Commands Used and Explanations 15

6.3 Results 15

7 Remote Repository (GitHub) **18**

7.1 Objective 18

7.2 Steps Performed 18

7.3 Commands Used and Explanations 18

7.4 Results 19

8 Collaborative Work **20**

8.1 Objective 20

8.2 Steps Performed 20

8.3 Commands Used and Explanations 20

8.4 Results 21

8.5 Verification Process 23

9 Merge Conflicts **23**

9.1 Objective 23

9.2 Commands Used and Explanations 24

9.3 Verification and Quality Control 24

9.4 Results 25

1 Introduction

This report documents the practical exercises completed during the Git version control laboratory session. The exercises covered fundamental Git concepts including repository initialization, branching, merging, remote repository management, and collaborative workflows.

Git is a distributed version control system that enables efficient tracking of changes in software projects, facilitating collaboration among multiple developers while maintaining a complete history of modifications.

2 Objectives

The main objectives of this laboratory work were:

- Learn how to use Version Control Software effectively
- Master software project versioning techniques
- Collaborate on projects and work effectively in teams
- Understand branching strategies and merge conflicts resolution
- Practice remote repository management using GitHub

3 Git Configuration

3.1 Objective

Configure Git with user identity information to associate commits with the correct author.

3.2 Commands Used and Explanations

```
# Set global username for all repositories
git config --global user.name "Your Name"

# Set global email for all repositories
git config --global user.email your.email@domain.com

# View the configuration file contents (Windows)
type .gitconfig
```

Command Explanations:

- `git config -global`: Sets configuration values for all repositories on the system
- `user.name`: Identifies the author of commits
- `user.email`: Associates commits with an email address
- `type .gitconfig`: Displays the contents of the Git configuration file .

3.3 Results

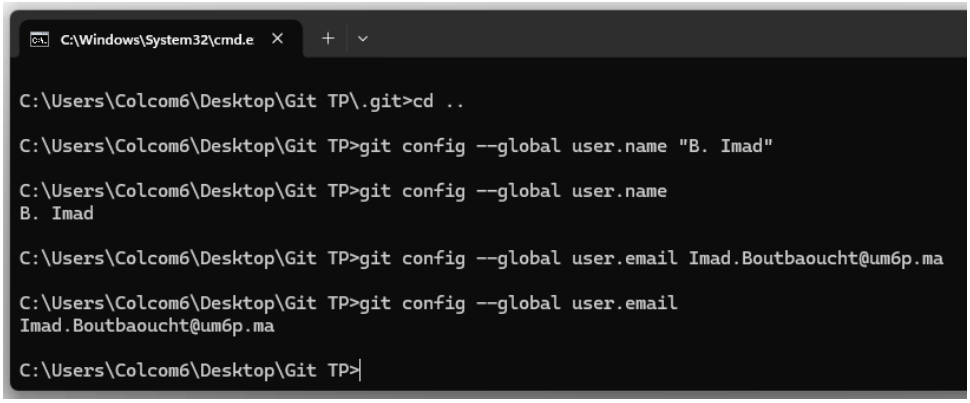

Figure 1: Git User Configuration Setup

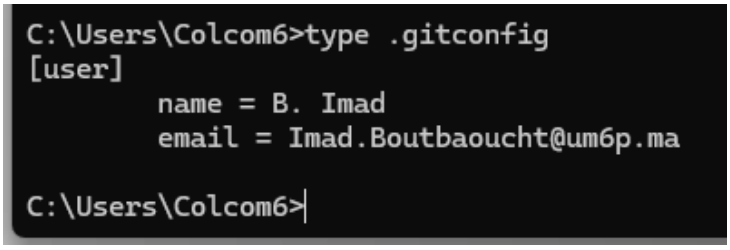

Figure 2: User Configuration Verification

The configuration ensures that all future commits are properly attributed to the correct author. The global settings apply to all Git repositories on the system.

4 Repository Initialization

4.1 Objective

Initialize a Git repository and create the initial commit structure with multiple commits adding fruits to the project.

4.2 Commands Used and Explanations

```
# Create project directory
mkdir tpgit

# Navigate to the directory
cd tpgit

# Initialize Git repository
git init

# Create empty file (Windows)
type nul > fruits.txt

# Stage the file for commit
git add fruits.txt

# Commit with descriptive message
git commit -m "Initial commit with fruits"

# Check repository status
git status

# View commit history
git log
```

Command Explanations:

- `git init`: Creates a new Git repository in the current directory
- `type nul >`: Creates an empty file (Windows command)
- `git add`: Stages files for the next commit
- `git commit -m`: Creates a commit with the staged changes and a message
- `git status`: Shows the working directory status
- `git log`: Displays the commit history

4.3 Results

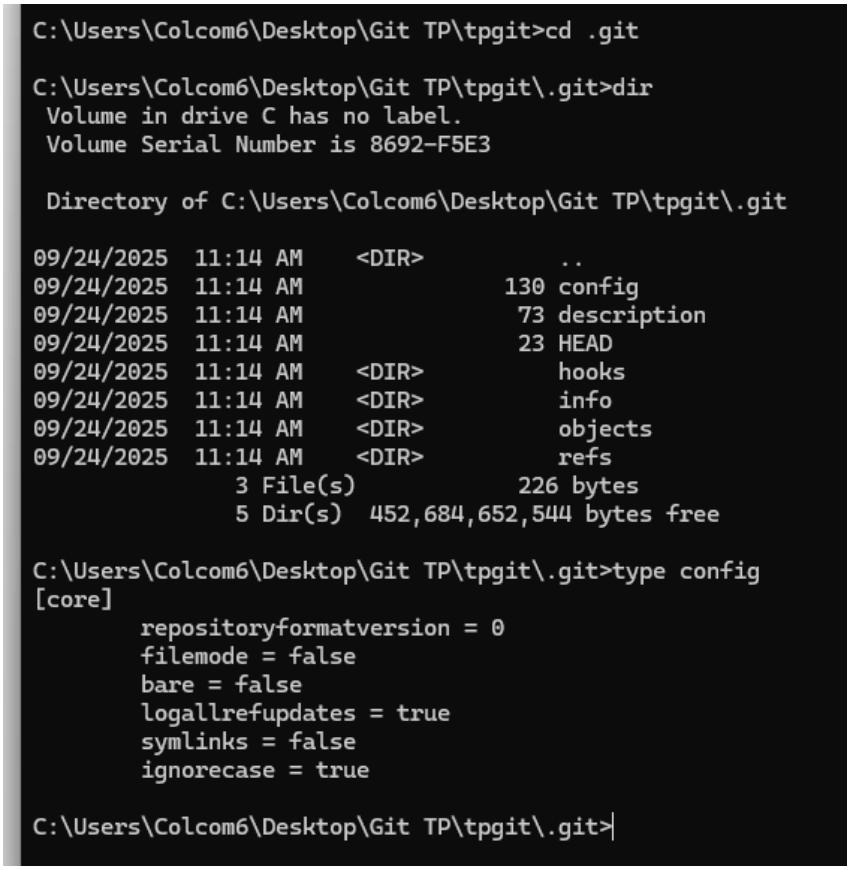

Figure 3: Inspecting the Working Directory Structure

```
C:\Users\Colcom6\Desktop\Git TP\tpgit>echo WaterMelon > fruits.txt

C:\Users\Colcom6\Desktop\Git TP\tpgit>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        fruits.txt

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

C:\Users\Colcom6\Desktop\Git TP\tpgit>git add fruits.txt

C:\Users\Colcom6\Desktop\Git TP\tpgit>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   fruits.txt


C:\Users\Colcom6\Desktop\Git TP\tpgit>git commit -m "WaterMelon Succesdfully added"
[master (root-commit) f2a1f3f] WaterMelon Succesdfully added
 1 file changed, 1 insertion(+)
 create mode 100644 fruits.txt

C:\Users\Colcom6\Desktop\Git TP\tpgit>git status
On branch master
nothing to commit, working tree clean

C:\Users\Colcom6\Desktop\Git TP\tpgit>git log
commit f2a1f3f5ae38dde54a8a9b4ffad1c8395495666a (HEAD -> master)
Author: B. Imad <Imad.Boutbaoucht@um6p.ma>
Date:   Wed Sep 24 11:33:36 2025 +0100

    WaterMelon Succesdfully added

C:\Users\Colcom6\Desktop\Git TP\tpgit>
```

Figure 4: First Commit to fruits.txt - Initial Setup

```
C:\Users\Colcom6\Desktop\Git TP\tpgit>git status
On branch master
nothing to commit, working tree clean

C:\Users\Colcom6\Desktop\Git TP\tpgit>echo Tomatooe >> fruits.txt

C:\Users\Colcom6\Desktop\Git TP\tpgit>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:   fruits.txt

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

C:\Users\Colcom6\Desktop\Git TP\tpgit>git add fruits.txt

C:\Users\Colcom6\Desktop\Git TP\tpgit>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   fruits.txt


C:\Users\Colcom6\Desktop\Git TP\tpgit>git commit -m "Tomatooe Successfully added "
[master b9d0bde] Tomatooe Successfully added
 1 file changed, 1 insertion(+)

C:\Users\Colcom6\Desktop\Git TP\tpgit>git status
On branch master
nothing to commit, working tree clean

C:\Users\Colcom6\Desktop\Git TP\tpgit>git log
commit b9d0bde23a06800300f5d2f363f3a7a1ed31c5df (HEAD -> master)
Author: B. Imad <Imad.Boutbaoucht@um6p.ma>
Date:   Wed Sep 24 11:39:59 2025 +0100

    Tomatooe Successfully added

commit f2a1f3f5ae38dde54a8a9b4ffad1c8395495666a
Author: B. Imad <Imad.Boutbaoucht@um6p.ma>
Date:   Wed Sep 24 11:33:36 2025 +0100

    WaterMelon Succesdfully added

C:\Users\Colcom6\Desktop\Git TP\tpgit>
```

Figure 5: Second Commit Adding More Fruits

```
C:\Users\Colcom6\Desktop\Git TP\tpgit>git status
On branch master
nothing to commit, working tree clean

C:\Users\Colcom6\Desktop\Git TP\tpgit>git diff fruits.txt

C:\Users\Colcom6\Desktop\Git TP\tpgit>git log
commit b9d0bde23a06800300f5d2f363f3a7a1ed31c5df (HEAD -> master)
Author: B. Imad <Imad.Boutbaoucht@um6p.ma>
Date:   Wed Sep 24 11:39:59 2025 +0100

    Tomatooe Successfully added

commit f2a1f3f5ae38dde54a8a9b4ffad1c8395495666a
Author: B. Imad <Imad.Boutbaoucht@um6p.ma>
Date:   Wed Sep 24 11:33:36 2025 +0100

    WaterMelon Succesdfully added

C:\Users\Colcom6\Desktop\Git TP\tpgit>git log --graph --oneline --decorate --all
* b9d0bde (HEAD -> master) Tomatooe Successfully added
* f2a1f3f WaterMelon Succesdfully added

C:\Users\Colcom6\Desktop\Git TP\tpgit>
```

Figure 6: Repository Status and Initial Commit Logs

The repository was successfully initialized with a complete commit history showing the sequential addition of fruits. Each commit represents a distinct snapshot of the project's state.

5 Branching

5.1 Objective

Create and manage multiple branches to work on different features independently, maintaining separate development streams for vegetables and sauces.

5.2 Commands Used and Explanations

```
# Create a new branch for vegetables
git branch vegetables

# Switch to the vegetables branch
git checkout vegetables

# Alternative: create and switch in one command
git checkout -b vegetables

# Create sauces branch and switch to it
git checkout -b sauces

# List all branches in the repository
git branch

# Show branch structure with commits
git log --oneline --graph --all
```

Command Explanations:

- `git branch`: Lists, creates, or deletes branches
- `git checkout`: Switches between branches or restores files
- `git checkout -b`: Creates a new branch and switches to it
- `git log -oneline -graph -all`: Shows compact commit history with branch visualization

5.3 Vegetables Branch Development

```
C:\Users\Colcom6\Desktop\Git TP\tpgit>git branch vegetables

C:\Users\Colcom6\Desktop\Git TP\tpgit>git checkout vegetables
Switched to branch 'vegetables'

C:\Users\Colcom6\Desktop\Git TP\tpgit>git branch
  master
* vegetables

C:\Users\Colcom6\Desktop\Git TP\tpgit>git status
On branch vegetables
nothing to commit, working tree clean

C:\Users\Colcom6\Desktop\Git TP\tpgit>echo > vegetables.txt

C:\Users\Colcom6\Desktop\Git TP\tpgit>dir
 Volume in drive C has no label.
 Volume Serial Number is 8692-F5E3

 Directory of C:\Users\Colcom6\Desktop\Git TP\tpgit

09/24/2025  11:45 AM    <DIR>          .
09/24/2025  11:43 AM    <DIR>          ..
09/24/2025  11:39 AM                24 fruits.txt
09/24/2025  11:45 AM                13 vegetables.txt
               2 File(s)             37 bytes
               2 Dir(s)  452,620,972,032 bytes free
```

Figure 7: Creating and Switching to Vegetables Branch

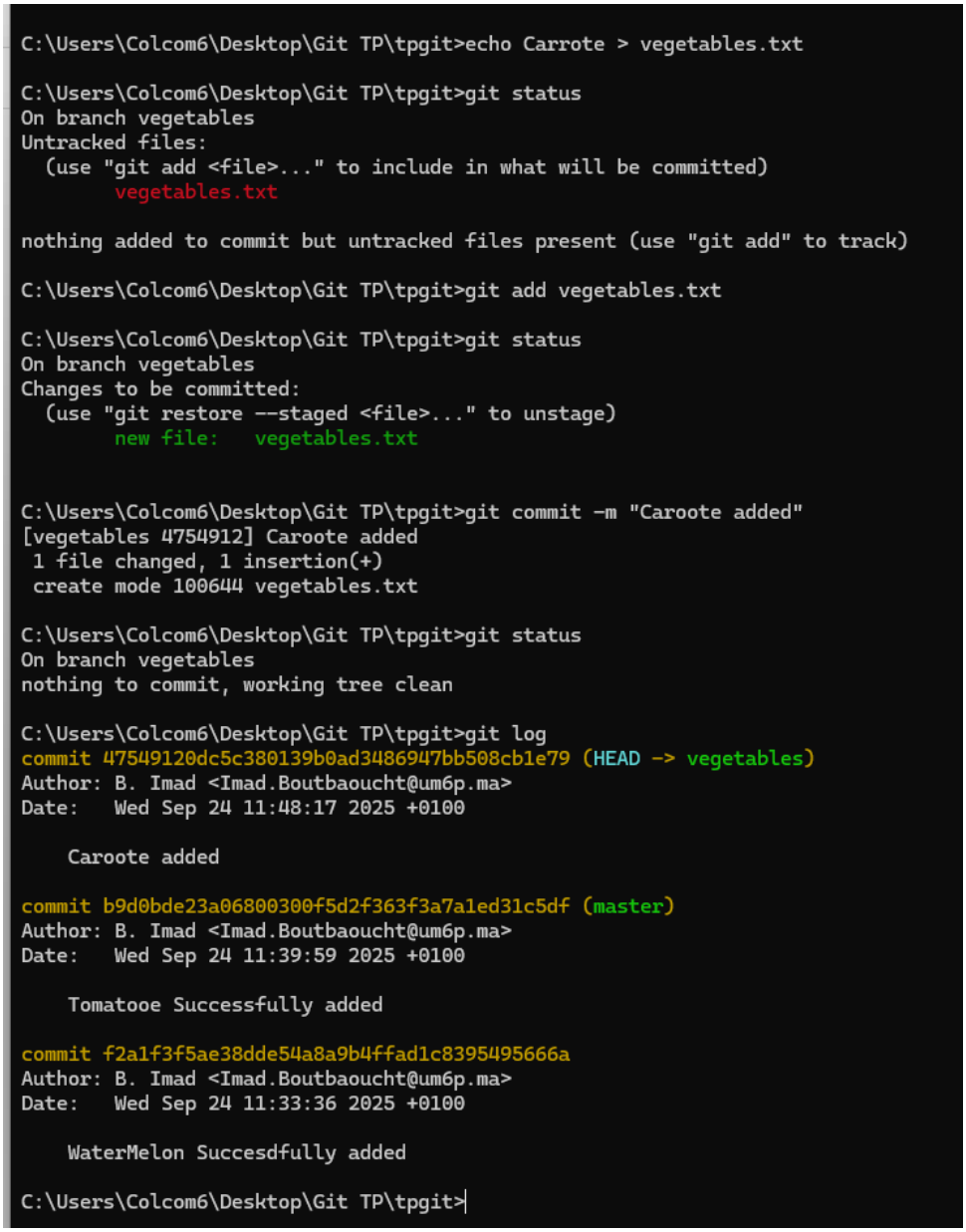

Figure 8: Working on Vegetables Branch - File Creation

```
C:\Users\Colcom6\Desktop\Git TP\tpgit>echo Potatoes >> vegetables.txt

C:\Users\Colcom6\Desktop\Git TP\tpgit>git status
On branch vegetables
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:   vegetables.txt

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

C:\Users\Colcom6\Desktop\Git TP\tpgit>git add vegetables.txt

C:\Users\Colcom6\Desktop\Git TP\tpgit>git commit -m "Potatooooooes :: "
[vegetables 531deba] Potatooooooes ::
 1 file changed, 1 insertion(+)

C:\Users\Colcom6\Desktop\Git TP\tpgit>git log
commit 531deba1a91ca288f4cf2656df6856429f44cf2a (HEAD -> vegetables)
Author: B. Imad <Imad.Boutbaoucht@um6p.ma>
Date:   Wed Sep 24 11:53:20 2025 +0100

    Potatooooooes ::

commit 2e68723261b05fd153c950248e27352243c98660
Author: B. Imad <Imad.Boutbaoucht@um6p.ma>
Date:   Wed Sep 24 11:50:22 2025 +0100

    Onion Onion

commit 47549120dc5c380139b0ad3486947bb508cb1e79
Author: B. Imad <Imad.Boutbaoucht@um6p.ma>
Date:   Wed Sep 24 11:48:17 2025 +0100

    Caroote added

commit b9d0bde23a06800300f5d2f363f3a7a1ed31c5df (master)
Author: B. Imad <Imad.Boutbaoucht@um6p.ma>
Date:   Wed Sep 24 11:39:59 2025 +0100

    Tomatooe Successfully added

commit f2a1f3f5ae38dde54a8a9b4ffad1c8395495666a
Author: B. Imad <Imad.Boutbaoucht@um6p.ma>
Date:   Wed Sep 24 11:33:36 2025 +0100

    WaterMelon Succesdfully added

C:\Users\Colcom6\Desktop\Git TP\tpgit>
```

Figure 9: Committing Changes to Vegetables Branch

```
C:\Users\Colcom6\Desktop\Git TP\tpgit>echo Onion >> vegetables.txt

C:\Users\Colcom6\Desktop\Git TP\tpgit>git status
On branch vegetables
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:   vegetables.txt

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

C:\Users\Colcom6\Desktop\Git TP\tpgit>git add vegetables.txt

C:\Users\Colcom6\Desktop\Git TP\tpgit>git status
On branch vegetables
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   vegetables.txt


C:\Users\Colcom6\Desktop\Git TP\tpgit>git commit -m "Onion Onion"
[vegetables 2e68723] Onion Onion
 1 file changed, 1 insertion(+)

C:\Users\Colcom6\Desktop\Git TP\tpgit>git status
On branch vegetables
nothing to commit, working tree clean

C:\Users\Colcom6\Desktop\Git TP\tpgit>git log
commit 2e68723261b05fd153c950248e27352243c98660 (HEAD -> vegetables)
Author: B. Imad <Imad.Boutbaoucht@um6p.ma>
Date:   Wed Sep 24 11:50:22 2025 +0100

    Onion Onion

commit 47549120dc5c380139b0ad3486947bb508cb1e79
Author: B. Imad <Imad.Boutbaoucht@um6p.ma>
Date:   Wed Sep 24 11:48:17 2025 +0100

    Caroote added

commit b9d0bde23a06800300f5d2f363f3a7a1ed31c5df (master)
Author: B. Imad <Imad.Boutbaoucht@um6p.ma>
Date:   Wed Sep 24 11:39:59 2025 +0100

    Tomatooe Successfully added

commit f2a1f3f5ae38dde54a8a9b4ffad1c8395495666a
Author: B. Imad <Imad.Boutbaoucht@um6p.ma>
Date:   Wed Sep 24 11:33:36 2025 +0100
```

Figure 10: Vegetables Branch Commit History

5.4 Sauces Branch Development

```
C:\Users\Colcom6\Desktop\Git TP\tpgit>git branch sauces

C:\Users\Colcom6\Desktop\Git TP\tpgit>git checkout sauces
Switched to branch 'sauces'

C:\Users\Colcom6\Desktop\Git TP\tpgit>echo yassa > sauces.txt

C:\Users\Colcom6\Desktop\Git TP\tpgit>git add sauces.txt

C:\Users\Colcom6\Desktop\Git TP\tpgit>git commit -m "Yassa Sauce"
[sauces 10ddb3b] Yassa Sauce
 1 file changed, 1 insertion(+)
 create mode 100644 sauces.txt

C:\Users\Colcom6\Desktop\Git TP\tpgit>echo Biggy >> sauces.txt

C:\Users\Colcom6\Desktop\Git TP\tpgit>git add sauces.txt

C:\Users\Colcom6\Desktop\Git TP\tpgit>git commit -m "Biggy sauce"
[sauces 0d125d3] Biggy sauce
 1 file changed, 1 insertion(+)

C:\Users\Colcom6\Desktop\Git TP\tpgit>git log
commit 0d125d35195449a14a2168734633a1add7fac2fd (HEAD -> sauces)
Author: B. Imad <Imad.Boutbaoucht@um6p.ma>
Date:   Wed Sep 24 11:57:03 2025 +0100

    Biggy sauce

commit 10ddb3b971ff00ac312d5cf7cbdbf7bc4f1cf2c8
Author: B. Imad <Imad.Boutbaoucht@um6p.ma>
Date:   Wed Sep 24 11:56:17 2025 +0100

    Yassa Sauce

commit 531deba1a91ca288f4cf2656df6856429f44cf2a (vegetables)
Author: B. Imad <Imad.Boutbaoucht@um6p.ma>
Date:   Wed Sep 24 11:53:20 2025 +0100

    Potatooooooes ::
```

Figure 11: Sauces Branch Creation and Development

Each branch maintained independent development streams while keeping the main branch stable.

6 Merging

6.1 Objective

Integrate changes from feature branches back into the main branch using merge operations while preserving commit history.

6.2 Commands Used and Explanations

```
# Switch to master branch
git checkout master

# Merge vegetables branch into master
git merge vegetables

# Merge sauces branch into master
git merge sauces

# Verify merge results
git log --oneline --graph --all
```

Command Explanations:

- `git merge`: Combines changes from different branches
- Fast-forward merge: When the target branch hasn't diverged
- Three-way merge: When branches have diverged, creating a merge commit

6.3 Results

```
C:\Users\Colcom6\Desktop\Git TP\tpgit>git checkout vegetables
Switched to branch 'vegetables'

C:\Users\Colcom6\Desktop\Git TP\tpgit>dir
 Volume in drive C has no label.
 Volume Serial Number is 8692-F5E3

 Directory of C:\Users\Colcom6\Desktop\Git TP\tpgit

09/24/2025  12:01 PM    <DIR>          .
09/24/2025  11:59 AM    <DIR>          ..
09/24/2025  11:58 AM             1,266 f
09/24/2025  11:39 AM                24 fruits.txt
09/24/2025  11:52 AM                29 vegetables.txt
               3 File(s)          1,319 bytes
               2 Dir(s)  452,586,889,216 bytes free

C:\Users\Colcom6\Desktop\Git TP\tpgit>
```

Figure 12: Initial branch status before merging operations

Figure 13: Merging vegetables branch into master

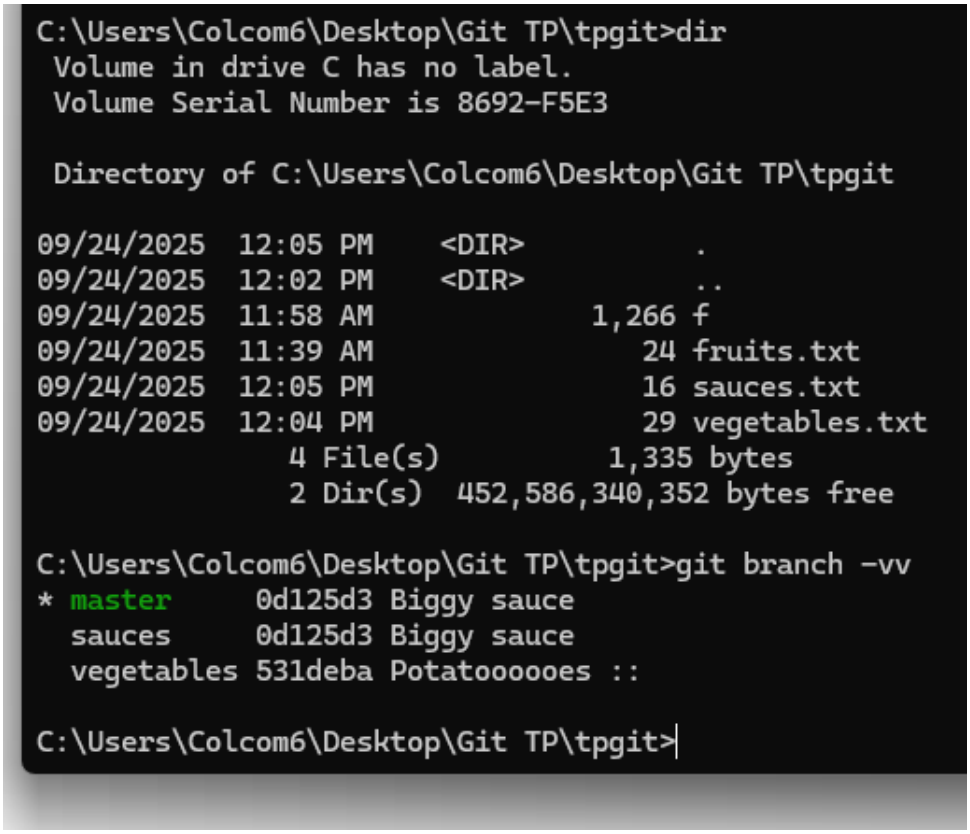

Figure 14: Branch tracking status showing vegetables branch not up to date with remote

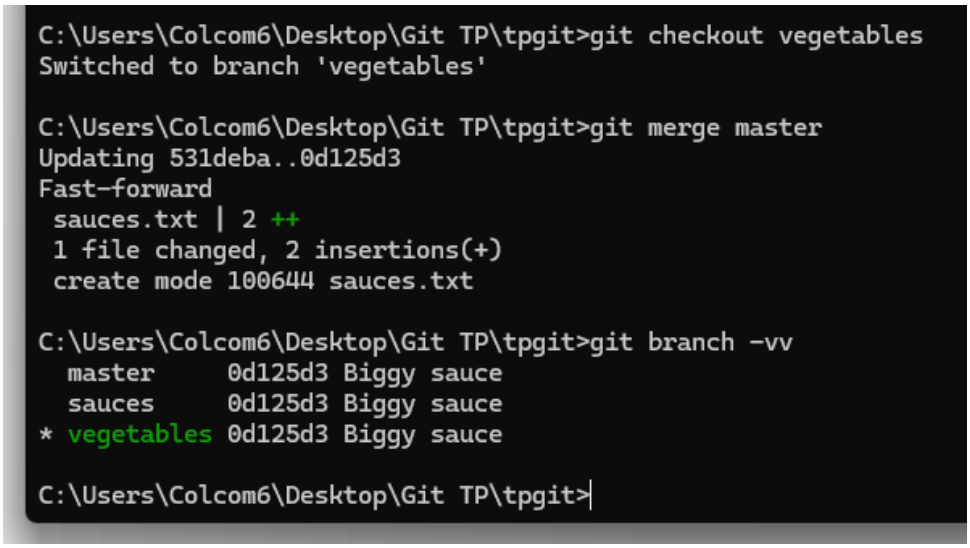

Figure 15: Final branch status after successful merges and synchronization

All features were successfully integrated into the main branch while preserving commit history.

7 Remote Repository (GitHub)

7.1 Objective

Push the local Git repository to GitHub and configure remote collaboration.

7.2 Steps Performed

1. Created GitHub repository on the GitHub platform
2. Added remote origin to connect local repository to GitHub
3. Pushed local repository to GitHub
4. Verified the synchronization and explored GitHub features

7.3 Commands Used and Explanations

```
# Add GitHub as a remote repository named 'origin'
git remote add origin https://github.com/username/tpgit.git

# Verify the remote connection was configured correctly
git remote -v

# Push local master branch to GitHub and set upstream tracking
git push -u origin master
```

Command Explanations:

- `git remote add origin [URL]`: Creates connection between local and remote repository
- `git remote -v`: Displays all configured remote repositories
- `git push -u origin master`: Pushes local commits to remote repository

7.4 Results

```
E:\Software Projects\Git TP\Software-Engineering-Projects\tpgit>git remote add origin https://github.c
ngineering-Projects.git
error: remote origin already exists.

E:\Software Projects\Git TP\Software-Engineering-Projects\tpgit>git remote -v
origin  https://github.com/IBoutbaoucht/Software-Engineering-Projects.git (fetch)
origin  https://github.com/IBoutbaoucht/Software-Engineering-Projects.git (push)

E:\Software Projects\Git TP\Software-Engineering-Projects\tpgit>git push -u origin master
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 4 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (21/21), 1.75 KiB | 105.00 KiB/s, done.
Total 21 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/IBoutbaoucht/Software-Engineering-Projects/pull/new/master
remote:
To https://github.com/IBoutbaoucht/Software-Engineering-Projects.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

E:\Software Projects\Git TP\Software-Engineering-Projects\tpgit>
```

Figure 16: Adding GitHub as remote origin and first push

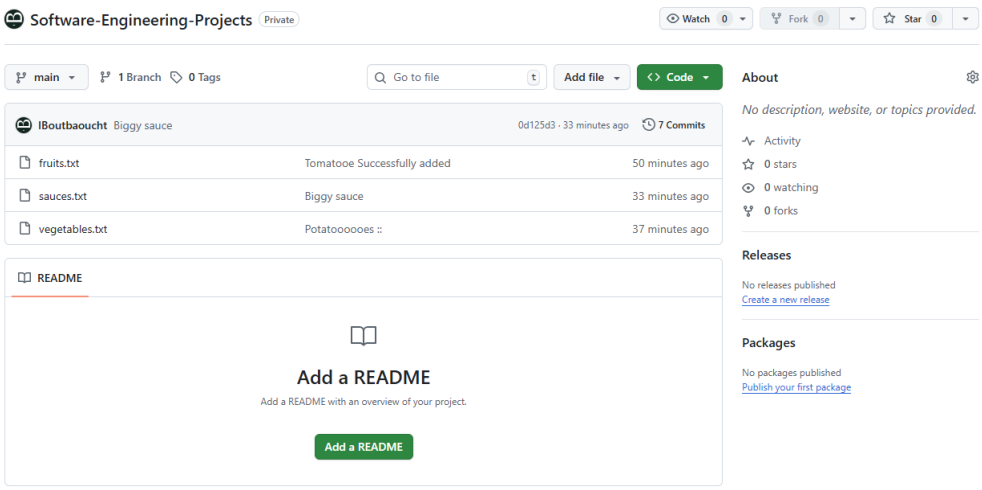

Figure 17: GitHub repository interface after successful push

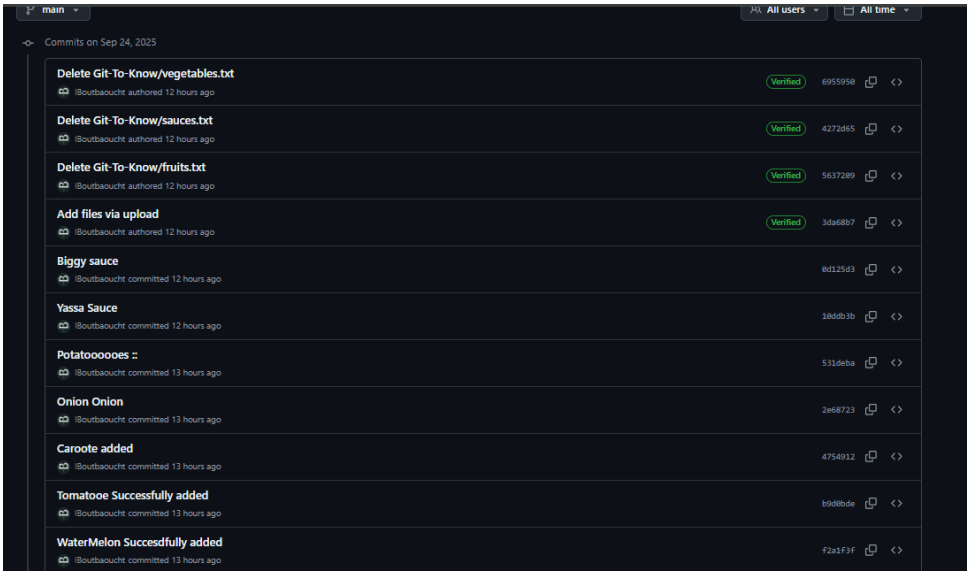

Figure 18: Commit history displayed on GitHub web interface

8 Collaborative Work

8.1 Objective

Practice collaborative workflows by cloning a teammate's repository and managing synchronization.

8.2 Steps Performed

1. Pair 2 cloned Pair 1's GitHub repository locally
2. Both pairs created separate feature branches (spices and herbs)
3. Developed features independently with multiple commits
4. Merged feature branches into master
5. Synchronized changes through pull/push operations
6. Verified successful integration of all changes

8.3 Commands Used and Explanations

```
# Pair 2: Clone Pair 1's repository
git clone https://github.com/pair1-username/tpgit.git tp-git-binome

# Both pairs: Create and work on feature branches
git checkout -b spices    # Pair 1
```

```
git checkout -b herbs      # Pair 2

# Synchronization commands
git pull origin master     # Get latest changes
git push origin master     # Share changes
```

Command Explanations:

- `git clone [URL] [dirname]`: Creates local copy of remote repository
- `git checkout -b [branch]`: Creates new branch and switches to it
- `git pull origin master`: Fetches and merges latest changes
- `git push origin master`: Uploads local commits to remote repository

8.4 Results

```
E:\Software Projects\Git TP\Software-Engineering-Projects>git clone https://github.com/zakariy
a-sab/tpgit tp-git-binome
Cloning into 'tp-git-binome'...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 27 (delta 1), reused 27 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (27/27), done.
Resolving deltas: 100% (1/1), done.

E:\Software Projects\Git TP\Software-Engineering-Projects>cd tp-git-binome

E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>dir
 Le volume dans le lecteur E s'appelle Data
 Le numéro de série du volume est 62EE-BD60

 Répertoire de E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome

25/09/2025  08:55    <DIR>          .
25/09/2025  08:55    <DIR>          ..
25/09/2025  08:55                34 fruits.txt
25/09/2025  08:55                18 sauces.txt
25/09/2025  08:55                39 vegetables.txt
               3 fichier(s)              91 octets
               2 Rép(s)   3 583 258 624 octets libres

E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>git log
commit 2b83d1dcbf407a84ba5af52ca7fc7bf83c68310b (HEAD -> main, origin/main, origin/HEAD)
Author: zakariya sabri <zakariya.sabri@um6p.ma>
Date:   Wed Sep 24 12:02:29 2025 +0100

    add sauce_2

commit 07142d2206eba407521aef797b5adb7727c3dd85
Author: zakariya sabri <zakariya.sabri@um6p.ma>
Date:   Wed Sep 24 12:01:33 2025 +0100

    add sauce_1

commit 8e07a6842a5f406cbee35d7c2960f302aff165c7
```

Figure 19: Pair 2 cloning Pair 1's repository

```
asus@DESKTOP-EQUOP7V MINGW64 ~/Desktop/tpgit (main)
$ echo spice1 > spices.txt

asus@DESKTOP-EQUOP7V MINGW64 ~/Desktop/tpgit (main)
$ git add spices.txt
warning: in the working copy of 'spices.txt', LF will be replaced by CRLF the next time Git touches it

asus@DESKTOP-EQUOP7V MINGW64 ~/Desktop/tpgit (main)
$ git commit -m "Spice 1 !!!! "
git commit -m "Spice 1 git add spices.txt git add spices.txt  "
[main 499586b] Spice 1 git add spices.txt git add spices.txt
 1 file changed, 1 insertion(+)
 create mode 100644 spices.txt

asus@DESKTOP-EQUOP7V MINGW64 ~/Desktop/tpgit (main)
$ echo spice2 >> spices.txt

asus@DESKTOP-EQUOP7V MINGW64 ~/Desktop/tpgit (main)
$ git add spices.txt
warning: in the working copy of 'spices.txt', LF will be replaced by CRLF the next time Git touches it

asus@DESKTOP-EQUOP7V MINGW64 ~/Desktop/tpgit (main)
$ git commit -m "Spice2 Added !! "
git commit -m "Spice2 Added git add spices.txt  "
[main 61129c2] Spice2 Added git add spices.txt
 1 file changed, 1 insertion(+)

asus@DESKTOP-EQUOP7V MINGW64 ~/Desktop/tpgit (main)
$ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
```

Figure 20: Pair 1 developing spices feature branch

```
E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>git branch
* main

E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>git checkout -b herbs
Switched to a new branch 'herbs'

E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>git branch
* herbs
  main

E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>echo > herbs.txt

E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>echo Basil > herbs.txt

E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>git status
On branch herbs
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        herbs.txt

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

E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>git add herbs.txt

E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>git commit -m "Basil Added"
[herbs 56902c1] Basil Added
 1 file changed, 1 insertion(+)
 create mode 100644 herbs.txt

E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>echo Anise >> herbs.txt

E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>git add herbs.txt

E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>git commit -m "Asise Joined"
[herbs baf3e62] Asise Joined
 1 file changed, 1 insertion(+)

E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>
```

Figure 21: Pair 2 working on herbs branch simultaneously

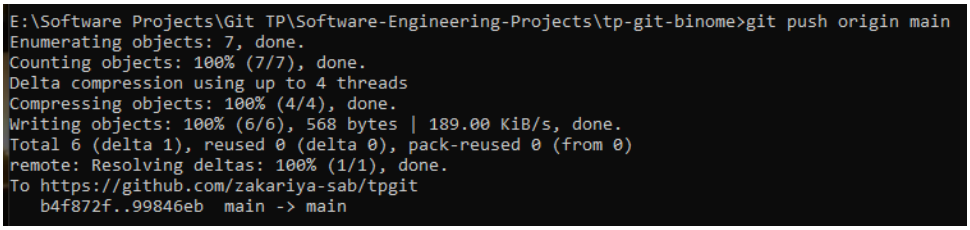

Figure 22: Automatic merge completion

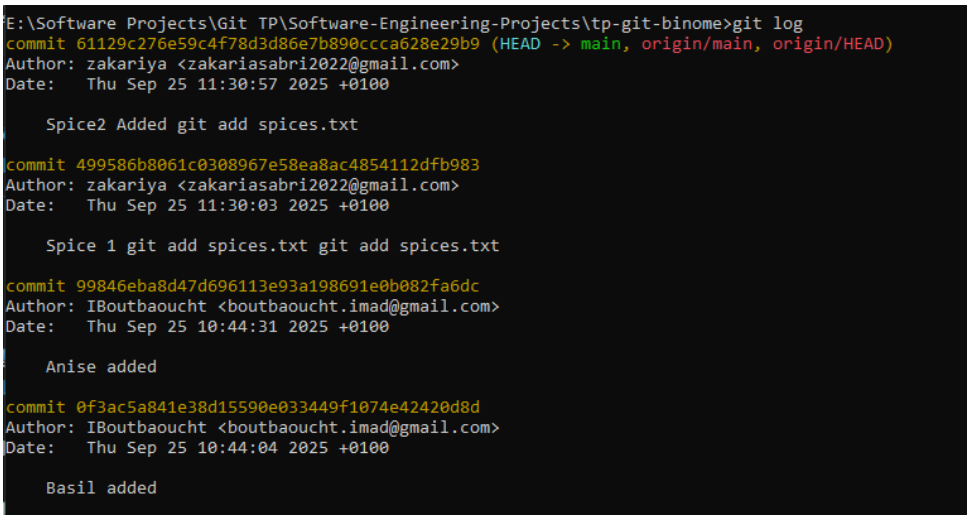

Figure 23: Final repository state showing both spices.txt and herbs.txt files integrated

8.5 Verification Process

To confirm successful collaboration:

```
# Verify both feature files exist (Windows)
dir spices.txt herbs.txt

# Check combined commit history
git log --oneline --graph --all

# Verify synchronization status
git status
git branch -vv
```

9 Merge Conflicts

9.1 Objective

Intentionally create and resolve merge conflicts to understand how Git handles simultaneous changes.

9.2 Commands Used and Explanations

```
# Generate conflicting changes
# Both pairs edit same lines in fruits.txt

# Pair 1 pushes first
git add fruits.txt
git commit -m "Simplify fruits list to only orange"
git push origin master

# Pair 2 encounters conflict
git push origin master
# Conflict error occurs
git pull origin master

# Conflict resolution process
git status
# Manually edit conflicted file
git add fruits.txt
git commit -m "Resolve merge conflict in fruits.txt"
git push origin master

# Abort merge if needed
git merge --abort
```

Command Explanations:

- `git status`: Shows files with conflicts and their resolution status
- `git merge -abort`: Cancels the merge process and returns to pre-merge state
- `git add [file]`: Marks conflict as resolved after manual editing
- Conflict markers: `«««<`, `=======`, `»»»>` show conflicting sections

9.3 Verification and Quality Control

Post-Resolution Checks:

```
# Verify no remaining conflict markers (Windows)
findstr /n "<<<<<<<" fruits.txt
findstr /n ">>>>>>>" fruits.txt

# Verify final content matches expected resolution
type fruits.txt

# Confirm both pairs' intended changes are preserved
```

9.4 Results

```
E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>git add fruits.txt

E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>git commit -m "Changed Fruit1 to Ananas"
[main db54afe] Changed Fruit1 to Ananas
 1 file changed, 1 insertion(+), 1 deletion(-)

E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>git add fruits.txt

E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>git commit -m "Changed banana to WaterMelon"
[main 8d930b9] Changed banana to WaterMelon
 1 file changed, 1 insertion(+), 1 deletion(-)

E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>
```

Figure 24: Initial conflict setup showing divergent changes

```
asus@DESKTOP-EQUOP7V MINGW64 ~/Desktop/tpgit (main)
$ cat fruits.txt
orange
banana
fruit_1
fruit_2

asus@DESKTOP-EQUOP7V MINGW64 ~/Desktop/tpgit (main)
$ echo orange > fruits.txt

asus@DESKTOP-EQUOP7V MINGW64 ~/Desktop/tpgit (main)
$ git add fruits.txt
warning: in the working copy of 'fruits.txt', LF will be replaced by CRLF the next time Git touches it

asus@DESKTOP-EQUOP7V MINGW64 ~/Desktop/tpgit (main)
$ git commit -m "delet all fruits except orange"
[main 4a4afb5] delet all fruits except orange
 1 file changed, 3 deletions(-)
```

Figure 25: Conflict markers displayed in fruits.txt

```
E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 272 bytes | 20.00 KiB/s, done.
From https://github.com/zakariya-sab/tpgit
   61129c2..4a4afb5  main       -> origin/main
Auto-merging fruits.txt
CONFLICT (content): Merge conflict in fruits.txt
Automatic merge failed; fix conflicts and then commit the result.

E:\Software Projects\Git TP\Software-Engineering-Projects\tp-git-binome>git diff
diff --cc fruits.txt
index 170119e,f8c1e5c..0000000
--- a/fruits.txt
+++ b/fruits.txt
@@@ -1,4 -1,1 +1,7 @@@
  orange
++<<<<<<< HEAD
 +WaterMelon
 +Ananas
 +fruit_2
++=======
++>>>>>>> 4a4afb564476b122d810c40ba56970a0048b5b4f
```

Figure 26: Merge conflict error when Pair 2 attempts to push

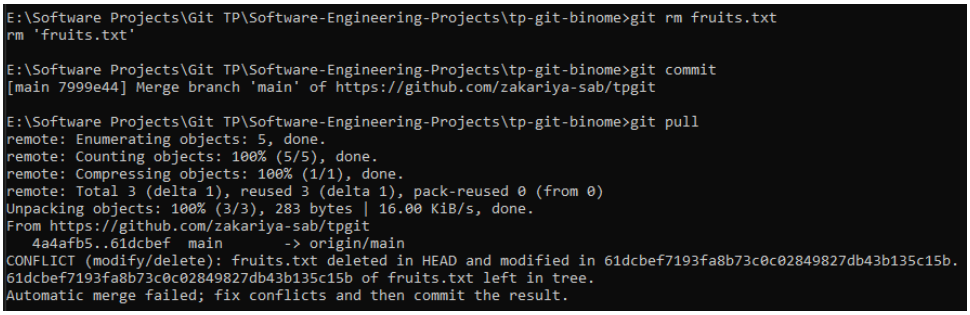

Figure 27: Successful conflict resolution