Lab Exercise 3- Working with Git Rebase

Lab Exercise: Git Rebase

This exercise demonstrates the use of git rebase in a scenario where no conflicts occur.

Objective

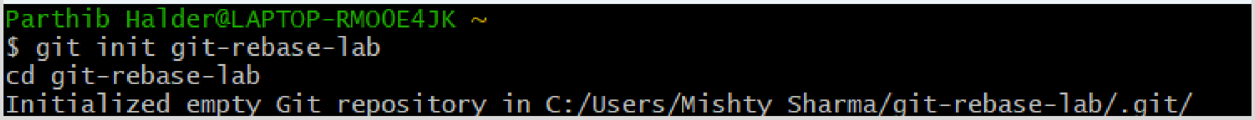
1. Learn how to rebase branches when there are no conflicting changes.

2. Understand the clean, linear history created by git rebase.

Prerequisites

1. Install Git on your system.

2. Initialize a Git repository:

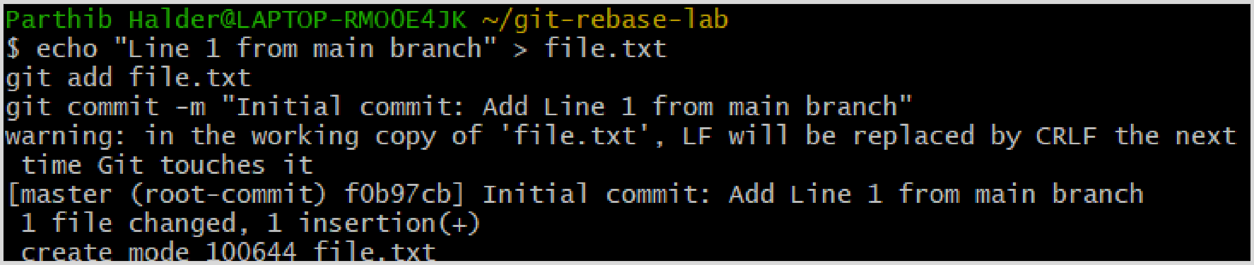


```
Parthib Halder@LAPTOP-RMOOE4JK ~
$ git init git-rebase-lab
cd git-rebase-lab
Initialized empty Git repository in C:/Users/Mishty Sharma/git-rebase-lab/.git/
```

Step-by-Step Instructions

1. Set Up the Repository

1. Create the main branch and make the initial commit:



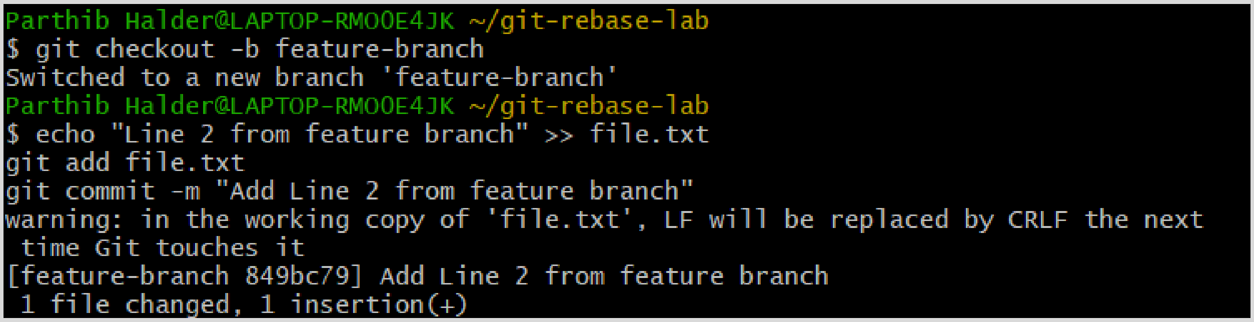
```
Parthib Halder@LAPTOP-RMOOE4JK ~/git-rebase-lab
$ echo "Line 1 from main branch" > file.txt
git add file.txt
git commit -m "Initial commit: Add Line 1 from main branch"
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next
 time Git touches it
[master (root-commit) f0b97cb] Initial commit: Add Line 1 from main branch
 1 file changed, 1 insertion(+)
 create mode 100644 file.txt
```

git commit -m "Initial commit: Add Line 1 from main branch"

2. Create a new branch feature-branch:

```
git checkout -b feature-branch
```

3. Add a new line to file.txt in feature-branch:



```
Parthib Halder@LAPTOP-RMOOE4JK ~/git-rebase-lab
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'
Parthib Halder@LAPTOP-RMOOE4JK ~/git-rebase-lab
$ echo "Line 2 from feature branch" >> file.txt
git add file.txt
git commit -m "Add Line 2 from feature branch"
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next
 time Git touches it
[feature-branch 849bc79] Add Line 2 from feature branch
 1 file changed, 1 insertion(+)
```

2. Rebase feature-branch onto main

1. Switch to feature-branch:

```
git checkout feature-branch
```

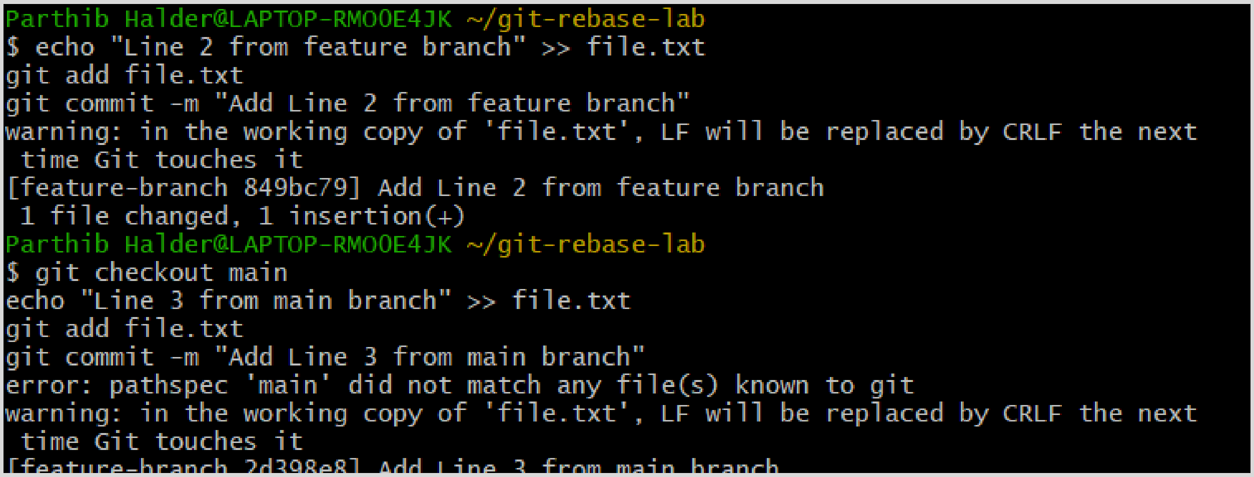
2. Rebase feature-branch onto main:

```
git rebase main
```

3. Git will replay the commit from feature-branch onto the main branch. Since there are no conflicts, the rebase completes automatically.

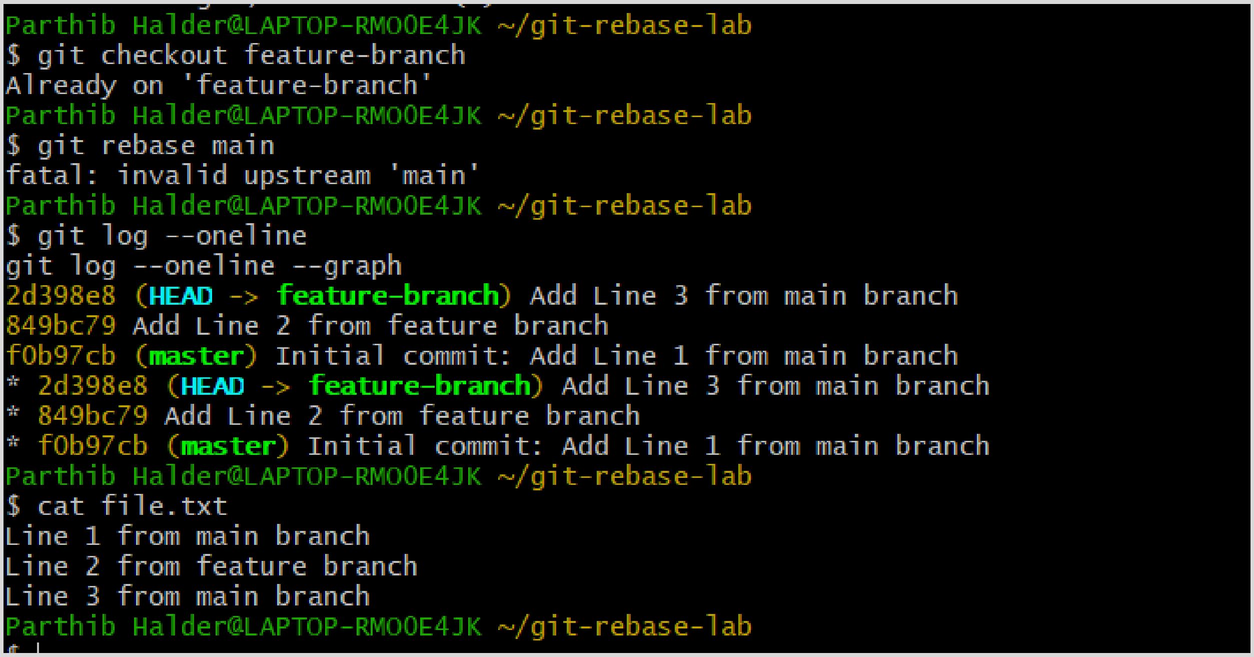
3. Verify the Rebase

1. View the commit history:



```
Parthib Halder@LAPTOP-RMOOE4JK ~/git-rebase-lab
$ echo "Line 2 from feature branch" >> file.txt
git add file.txt
git commit -m "Add Line 2 from feature branch"
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next
 time Git touches it
[feature-branch 849bc79] Add Line 2 from feature branch
 1 file changed, 1 insertion(+)
Parthib Halder@LAPTOP-RMOOE4JK ~/git-rebase-lab
$ git checkout main
echo "Line 3 from main branch" >> file.txt
git add file.txt
git commit -m "Add Line 3 from main branch"
error: pathspec 'main' did not match any file(s) known to git
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next
 time Git touches it
[feature-branch 2d398e8] Add Line 3 from main branch
```

Example output:

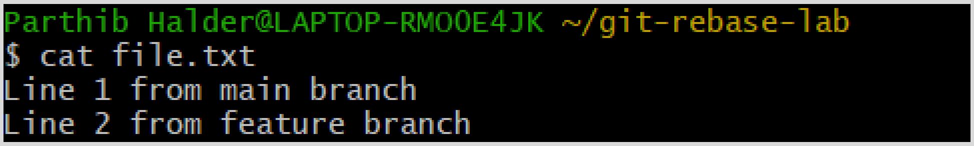


```
Parthib Halder@LAPTOP-RMO0E4JK ~/git-rebase-lab
$ git checkout feature-branch
Already on 'feature-branch'
Parthib Halder@LAPTOP-RMO0E4JK ~/git-rebase-lab
$ git rebase main
fatal: invalid upstream 'main'
Parthib Halder@LAPTOP-RMO0E4JK ~/git-rebase-lab
$ git log --oneline
git log --oneline --graph
2d398e8 (HEAD -> feature-branch) Add Line 3 from main branch
849bc79 Add Line 2 from feature branch
f0b97cb (master) Initial commit: Add Line 1 from main branch
* 2d398e8 (HEAD -> feature-branch) Add Line 3 from main branch
* 849bc79 Add Line 2 from feature branch
* f0b97cb (master) Initial commit: Add Line 1 from main branch
Parthib Halder@LAPTOP-RMO0E4JK ~/git-rebase-lab
$ cat file.txt
Line 1 from main branch
Line 2 from feature branch
Line 3 from main branch
Parthib Halder@LAPTOP-RMO0E4JK ~/git-rebase-lab
```

2. Check the contents of file.txt:

```
cat file.txt
```

Output:



```
Parthib Halder@LAPTOP-RMO0E4JK ~/git-rebase-lab
$ cat file.txt
Line 1 from main branch
Line 2 from feature branch
```