

## **Lab Exercise 3- Working with Git Rebase**

**Name- Garvit janu (B3 DevOps)**

**SAP ID - 500119646**

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**

```
git init git-rebase-lab
```

```
cd git-rebase-lab
```

```
C:\Users\manis>git init git-rebase-lab
Initialized empty Git repository in C:/Users/manis/git-rebase-lab/.git/
C:\Users\manis>cd git-rebase-lab
```

---

## Step-by-Step Instructions

### 1. Set Up the Repository

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

```
echo "Line 1 from main branch" > file.txt  
  
git add file.txt
```

```
C:\Users\manis\git-rebase-lab>echo "line 1 from main branch" >file.txt  
  
C:\Users\manis\git-rebase-lab>git add file.txt  
  
C:\Users\manis\git-rebase-lab>git commit -m "initial commit add line1 from main branch"  
[master (root-commit) 551c7c6] initial commit add line1 from main branch  
 1 file changed, 1 insertion(+)  
 create mode 100644 file.txt
```

2. Create a new branch feature-branch:

```
git checkout -b feature-branch  
  
C:\Users\manis\git-rebase-lab>git checkout -b feature-branch switch to a new  
branch 'feature-branch'
```

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

```
echo "Line 2 from feature branch" >> file.txt  
  
git add file.txt  
  
git commit -m "Add Line 2 from feature branch"
```

```
C:\Users\manis\git-rebase-lab>echo "line 2 from main branch" >>file.txt  
C:\Users\manis\git-rebase-lab>git add file.txt  
C:\Users\manis\git-rebase-lab>git commit -m "add line1 from feature branch"  
[master be11101] add line1 from feature branch  
1 file changed, 1 insertion(+)
```

4. Switch back to the main branch and add another line:

```
git checkout main  
  
echo "Line 3 from main branch" >> file.txt  
  
git add file.txt  
  
git commit -m "Add Line 3 from main branch"  
  
C:\Users\manis\git-rebase-lab>git checkout main  
error: pathspec 'main' did not match any file(s) known to git  
C:\Users\manis\git-rebase-lab>echo "line 3 from main branch" >>file.txt  
C:\Users\manis\git-rebase-lab>git add file.txt  
C:\Users\manis\git-rebase-lab>git commit -m "initial commit add line3 from ma  
in branch"  
[master f062664] initial commit add line3 from main branch  
1 file changed, 1 insertion(+)
```

---

## 2. Rebase feature-branch onto main

1. Switch to feature-branch:

```
git checkout feature-branch  
  
C:\Users\manis\git-rebase-lab>git checkout feature-branch  
error: pathspec 'feature-branch' did not match any file(s) known to git
```

2. Rebase feature-branch onto main:

```
git rebase main
```

```
C:\Users\manis\git-rebase-lab>git rebase main
fatal: invalid upstream '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:

```
git log --oneline
```

```
git log --oneline --graph
```

```
C:\Users\manis\git-rebase-lab>git log--oneline
git: 'log--oneline' is not a git command. See 'git --help'.

C:\Users\manis\git-rebase-lab>git--help
'git--help' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\manis\git-rebase-lab>git log--oneline-graph
git: 'log--oneline-graph' is not a git command. See 'git --help'.
```

2. Check the contents of file.txt:

```
cat file.txt
```

```
C:\Users\manis\git-rebase-lab>cat file.txt
cat' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\manis\git-rebase-lab>
```