

# GIT ASSIGNMENT SUBMISSION

**Name:** Vikram

**Assignment:** Git – Branching, Stashing, and Commit Operations

---

## Problem Statement

You are required to demonstrate Git workflows by creating branches, creating files, stashing uncommitted changes, switching branches, and committing changes to the correct branches.

---

## Environment Used

All Git tasks were performed on an Amazon Linux EC2 instance using the EC2 Instance Connect terminal.

---

## Tasks Performed

---

### Task 1: Create a Git working directory with feature1.txt and feature2.txt in the master branch

#### Steps Taken:

1. Created a working directory and initialized Git:

```
mkdir git  
cd git  
git init -b master
```

2. Created the files:

```
touch feature1.txt feature2.txt
```

3. Staged and committed the files:

```
git add feature1.txt feature2.txt  
git commit -m "Added feature1.txt and feature2.txt on master"
```

```
[ec2-user@ip-10-0-3-185 ~]$ mkdir git
[ec2-user@ip-10-0-3-185 ~]$ cd git
[ec2-user@ip-10-0-3-185 git]$ git init -b master
Initialized empty Git repository in /home/ec2-user/git/.git/
[ec2-user@ip-10-0-3-185 git]$ █
```

```
[ec2-user@ip-10-0-3-185 git]$ touch feature1.txt feature2.txt
[ec2-user@ip-10-0-3-185 git]$ ls
feature1.txt  feature2.txt
[ec2-user@ip-10-0-3-185 git]$ █
```

```
[ec2-user@ip-10-0-3-185 git]$ git add feature1.txt feature2.txt
[ec2-user@ip-10-0-3-185 git]$ git commit -m "Added feature1.txt and feature2.txt on master"
[master (root-commit) b878101] Added feature1.txt and feature2.txt on master
Committer: EC2 Default User <ec2-user@ip-10-0-3-185.ec2.internal>
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, 0 insertions(+), 0 deletions(-)
create mode 100644 feature1.txt
create mode 100644 feature2.txt
[ec2-user@ip-10-0-3-185 git]$ █
```

---

## Task 2: Create three branches – develop, feature1, and feature2

### Steps Taken:

1. Created the branches:

```
git branch develop
```

```
git branch feature1
```

```
git branch feature2
```

---

```
[ec2-user@ip-10-0-3-185 git]$ git branch develop
git branch feature1
git branch feature2
[ec2-user@ip-10-0-3-185 git]$ git branch
  develop
  feature1
  feature2
* master
[ec2-user@ip-10-0-3-185 git]$ █
```

### Task 3: In the develop branch, create develop.txt without staging or committing

#### Steps Taken:

1. Switched to the develop branch:

```
git checkout develop
```

2. Created the file and added content:

```
echo "Develop branch file" > develop.txt
```

```
[ec2-user@ip-10-0-3-185 git]$ git checkout develop
Switched to branch 'develop'
[ec2-user@ip-10-0-3-185 git]$ echo "Develop branch file" > develop.txt
[ec2-user@ip-10-0-3-185 git]$ █
```

---

### Task 4: Stash the uncommitted file and checkout to feature1 branch

#### Steps Taken:

1. Stashed the untracked file:

```
git stash push -m "stash develop.txt"
```

2. Checked out to feature1 branch:

```
git checkout feature1
```

```
[ec2-user@ip-10-0-3-185 git]$ git stash push -m "stash develop.txt"
No local changes to save
[ec2-user@ip-10-0-3-185 git]$ git checkout feature1
Switched to branch 'feature1'
[ec2-user@ip-10-0-3-185 git]$ █
```

---

## Task 5: Create new.txt in feature1 branch, stage it, and commit it

### Steps Taken:

1. Created the file:

```
touch new.txt
```

2. Staged and committed:

```
git add new.txt
```

```
git commit -m "Added new.txt in feature1 branch"
```

```
ec2-user@ip-10-0-3-185 git]$ touch new.txt
ec2-user@ip-10-0-3-185 git]$ git add new.txt
ec2-user@ip-10-0-3-185 git]$ git commit -m "Added new.txt in feature1 branch"
feature1 5c6a8bb] Added new.txt in feature1 branch
Committer: EC2 Default User <ec2-user@ip-10-0-3-185.ec2.internal>
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, 0 insertions(+), 0 deletions(-)
create mode 100644 new.txt
ec2-user@ip-10-0-3-185 git]$ █
```

---

## Task 6: Checkout to develop branch, unstash the file, and commit

### Steps Taken:

1. Switched back to develop:

```
git checkout develop
```

2. Restored the stashed file:

```
git stash
```

3. Staged and committed the file:

```
git add develop.txt
```

```
git commit -m "Added develop.txt after unstash"
```

```
[ec2-user@ip-10-0-3-185 git]$ git checkout develop
Switched to branch 'develop'
[ec2-user@ip-10-0-3-185 git]$ git stash pop
No stash entries found.
[ec2-user@ip-10-0-3-185 git]$ git stash
No local changes to save
[ec2-user@ip-10-0-3-185 git]$ git add develop.txt
[ec2-user@ip-10-0-3-185 git]$ git commit -m "Added develop.txt after unstash"
[develop fb6d8a5] Added develop.txt after unstash
Committer: EC2 Default User <ec2-user@ip-10-0-3-185.ec2.internal>
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 develop.txt
[ec2-user@ip-10-0-3-185 git]$ █
```

---

## Conclusion

Successfully completed all Git tasks including branch creation, file creation, stashing uncommitted work, switching branches, and committing changes in their respective branches. This demonstrates a solid understanding of Git branching and workflow management.