

GIT ASSIGNMENT SUBMISSION

Name: Vikram

Assignment: Git – Git Flow Workflow and Hotfix Management

Problem Statement

You are required to demonstrate the Git Flow workflow by creating the architecture branches, performing feature development, merging code following Git Flow standards, and handling an urgent hotfix directly on the master branch.

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 Flow workflow architecture on Git

Steps Taken:

1. Initialized the Git repository with master branch:

```
git init -b master
```

2. Created the main development branch:

```
git branch develop
```

```
[ec2-user@ip-10-0-3-185 ~]$ git init -b master
warning: re-init: ignored --initial-branch=master
Reinitialized existing Git repository in /home/ec2-user/.git/
[ec2-user@ip-10-0-3-185 ~]$ git branch develop
[ec2-user@ip-10-0-3-185 ~]$ █
```

Task 2: Create all the required branches

Steps Taken:

Created the standard Git Flow branches:

Feature branch

[git branch feature/login](#)

Release branch

[git branch release/v1.0](#)

Hotfix branch

[git branch hotfix/urgent-fix](#)

```
[ec2-user@ip-10-0-3-185 ~]$ git branch feature/login
[ec2-user@ip-10-0-3-185 ~]$ git branch release/v1.0
[ec2-user@ip-10-0-3-185 ~]$ git branch hotfix/urgent-fix
[ec2-user@ip-10-0-3-185 ~]$ █
```

Task 3: Starting from the feature branch, push the branch to the master following the Git Flow architecture

Steps Taken:

1. Work on Feature Branch

[git checkout feature/login](#)

[echo "Login feature work" > login.txt](#)

[git add login.txt](#)

[git commit -m "Added login feature"](#)

```
[ec2-user@ip-10-0-3-185 ~]$ git checkout feature/login
echo "Login feature work" > login.txt
git add login.txt
git commit -m "Added login feature"
D      master.txt
D      public1.txt
D      public2.txt
Switched to branch 'feature/login'
[feature/login ed69c77] Added login feature
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 login.txt
[ec2-user@ip-10-0-3-185 ~]$ █
```

2. Merge Feature → Develop

```
git checkout develop
```

```
git merge feature/login
```

```
[ec2-user@ip-10-0-3-185 ~]$ git checkout develop
D      master.txt
D      public1.txt
D      public2.txt
Switched to branch 'develop'
[ec2-user@ip-10-0-3-185 ~]$ git merge feature/login
Updating c5c6e38..ed69c77
Fast-forward
  login.txt | 1 +
  1 file changed, 1 insertion(+)
  create mode 100644 login.txt
[ec2-user@ip-10-0-3-185 ~]$ █
```

3. Merge Develop → Release

```
git checkout release/v1.0
```

```
git merge develop
```

```
[ec2-user@ip-10-0-3-185 ~]$ git checkout release/v1.0
D      master.txt
D      public1.txt
D      public2.txt
Switched to branch 'release/v1.0'
[ec2-user@ip-10-0-3-185 ~]$ git merge develop
Updating c5c6e38..ed69c77
Fast-forward
  login.txt | 1 +
  1 file changed, 1 insertion(+)
  create mode 100644 login.txt
[ec2-user@ip-10-0-3-185 ~]$ █
```

4. Merge Release → Master

```
git checkout master
```

```
git merge release/v1.0
```

```
[ec2-user@ip-10-0-3-185 ~]$ git checkout master
D      master.txt
D      public1.txt
D      public2.txt
Switched to branch 'master'
[ec2-user@ip-10-0-3-185 ~]$ git merge release/v1.0
Updating c5c6e38..ed69c77
Fast-forward
  login.txt | 1 +
  1 file changed, 1 insertion(+)
  create mode 100644 login.txt
[ec2-user@ip-10-0-3-185 ~]$ █
```

Task 4: Push urgent.txt on master using hotfix

Steps Taken:

1. Work on Hotfix Branch

```
git checkout hotfix/urgent-fix  
echo "Urgent production fix" > urgent.txt  
git add urgent.txt  
git commit -m "Hotfix: urgent production fix"
```

```
[ec2-user@ip-10-0-3-185 ~]$ git checkout hotfix/urgent-fix  
echo "Urgent production fix" > urgent.txt  
git add urgent.txt  
git commit -m "Hotfix: urgent production fix"  
D      master.txt  
D      public1.txt  
D      public2.txt  
Switched to branch 'hotfix/urgent-fix'  
[hotfix/urgent-fix c4ab4e9] Hotfix: urgent production fix  
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 urgent.txt  
[ec2-user@ip-10-0-3-185 ~]$ █
```

2. Merge Hotfix → Master

```
git checkout master  
git merge hotfix/urgent-fix
```

```
[ec2-user@ip-10-0-3-185 ~]$ git checkout master
git merge hotfix/urgent-fix
D      master.txt
D      public1.txt
D      public2.txt
Switched to branch 'master'
Merge made by the 'ort' strategy.
 urgent.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 urgent.txt
[ec2-user@ip-10-0-3-185 ~]$ █
```

3. Merge Hotfix → Develop (Git Flow requirement)

git checkout develop

git merge hotfix/urgent-fix

```
[ec2-user@ip-10-0-3-185 ~]$ git checkout develop
D      master.txt
D      public1.txt
D      public2.txt
Switched to branch 'develop'
[ec2-user@ip-10-0-3-185 ~]$ git merge hotfix/urgent-fix
Merge made by the 'ort' strategy.
 urgent.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 urgent.txt
[ec2-user@ip-10-0-3-185 ~]$ █
```

Conclusion

Successfully created and followed the Git Flow workflow by using feature, develop, release, and hotfix branches. Demonstrated feature development, proper merging sequence, and hotfix application directly on master while keeping develop updated. This assignment confirms understanding of Git Flow and branch synchronization.