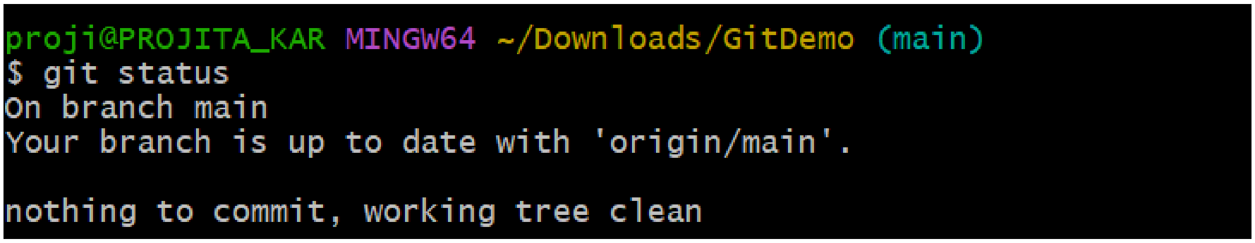
Git ASSIGNMENT 4

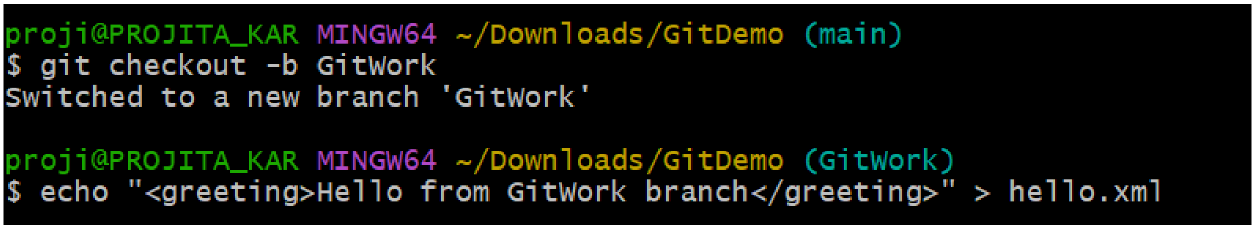
Git Repository: https://gitlab.com/projita2003kar/gitdemo

1. Verify if main is in clean state.



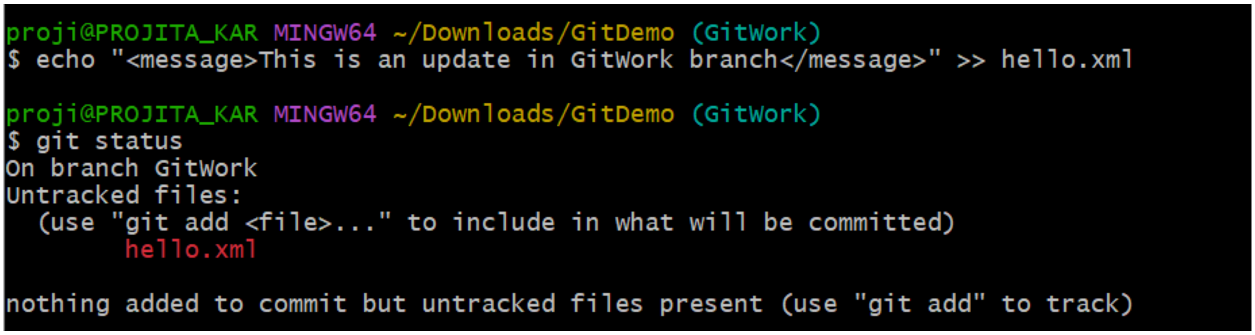
```
proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

2. Create a branch “GitWork”. Add a file “hello.xml”.



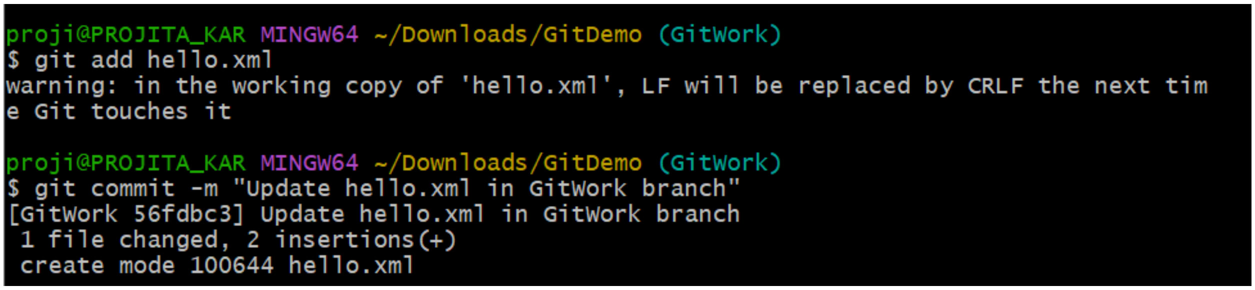
```
proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (main)
$ git checkout -b GitWork
Switched to a new branch 'GitWork'

proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (GitWork)
$ echo "<greeting>Hello from GitWork branch</greeting>" > hello.xml
```

3. Update the content of “hello.xml” and observe the status



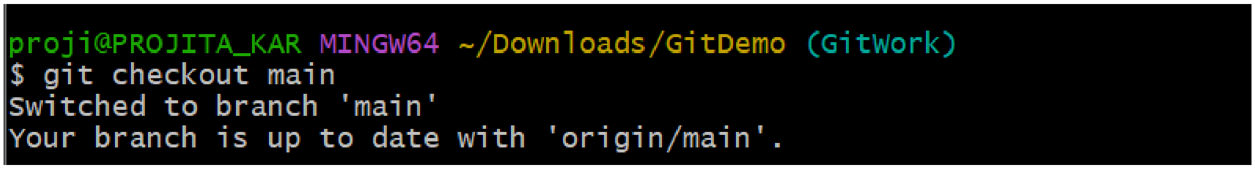
```
proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (GitWork)
$ echo "<message>This is an update in GitWork branch</message>" >> hello.xml

proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (GitWork)
$ git status
On branch GitWork
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        hello.xml

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

4. Commit the changes to reflect in the branch



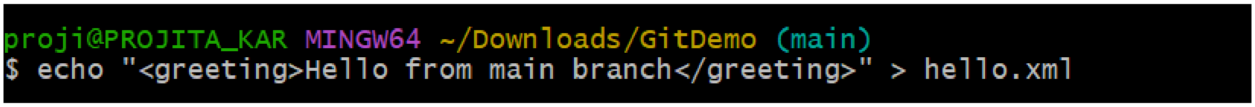
```
proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (GitWork)
$ git add hello.xml
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next tim
e Git touches it

proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (GitWork)
$ git commit -m "Update hello.xml in GitWork branch"
[GitWork 56fdbc3] Update hello.xml in GitWork branch
 1 file changed, 2 insertions(+)
 create mode 100644 hello.xml
```

5. Switch to main.



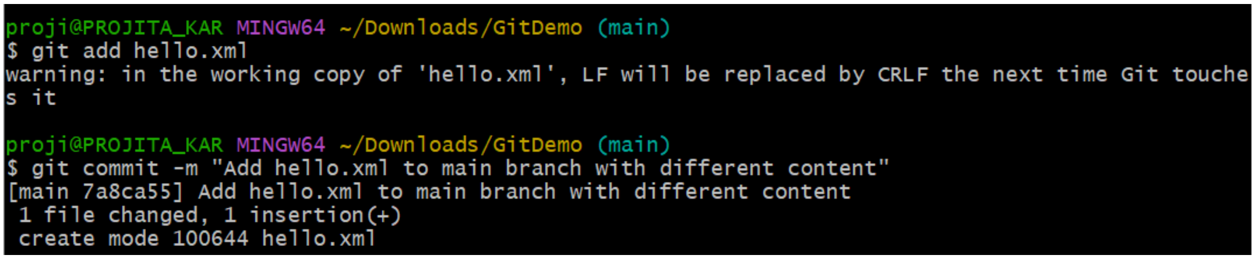
```
proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (GitWork)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

6. Add a file “hello.xml” to the main and add some different content than previous.



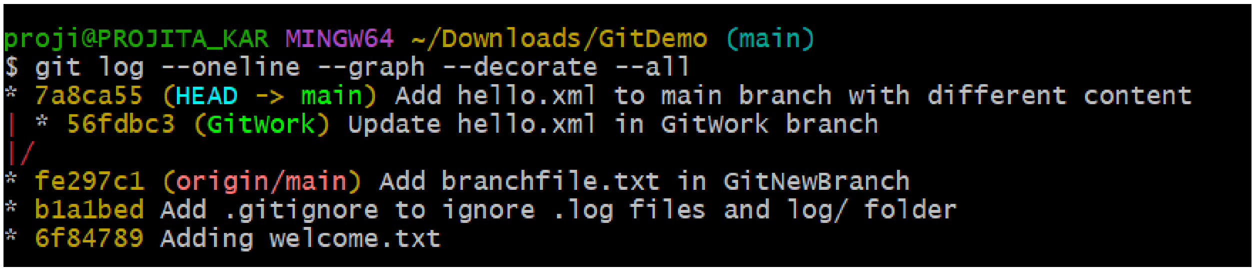
```
proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (main)
$ echo "<greeting>Hello from main branch</greeting>" > hello.xml
```

7. Commit the changes to the main



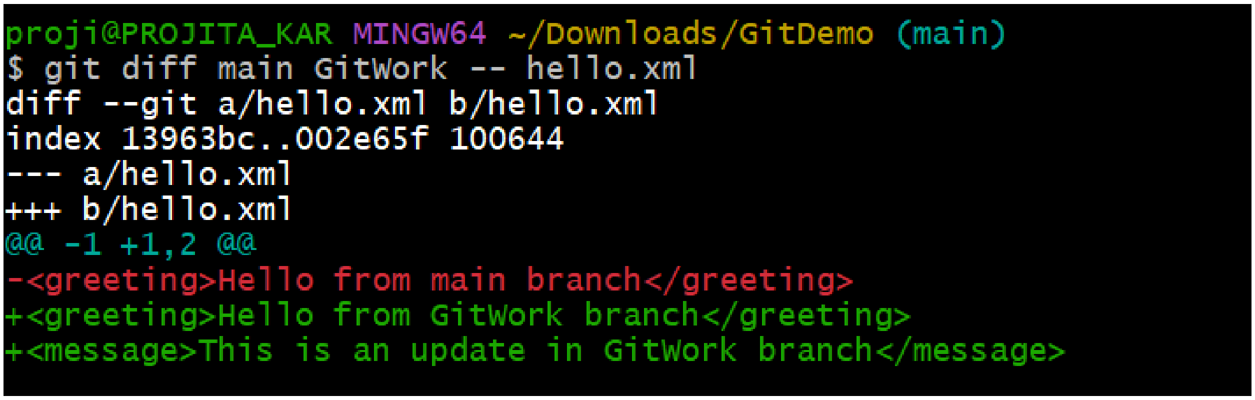
```
proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (main)
$ git add hello.xml
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next time Git touche
s it

proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (main)
$ git commit -m "Add hello.xml to main branch with different content"
[main 7a8ca55] Add hello.xml to main branch with different content
 1 file changed, 1 insertion(+)
 create mode 100644 hello.xml
```

8. Observe the log by executing “git log –oneline –graph –decorate –all”



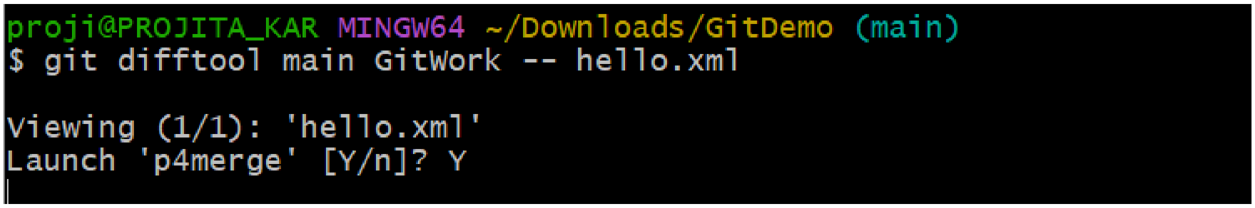
```
proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (main)
$ git log --oneline --graph --decorate --all
* 7a8ca55 (HEAD -> main) Add hello.xml to main branch with different content
| * 56fdbc3 (GitWork) Update hello.xml in GitWork branch
|/
* fe297c1 (origin/main) Add branchfile.txt in GitNewBranch
* b1a1bed Add .gitignore to ignore .log files and log/ folder
* 6f84789 Adding welcome.txt
```

9. Check the differences with Git diff tool

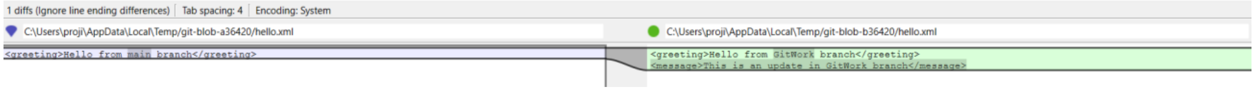


```
proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (main)
$ git diff main GitWork -- hello.xml
diff --git a/hello.xml b/hello.xml
index 13963bc..002e65f 100644
--- a/hello.xml
+++ b/hello.xml
@@ -1 +1,2 @@
-<greeting>Hello from main branch</greeting>
+<greeting>Hello from GitWork branch</greeting>
+<message>This is an update in GitWork branch</message>
```

10. For better visualization, use P4Merge tool to list out all the differences between main and branch



```
proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (main)
$ git difftool main GitWork -- hello.xml

Viewing (1/1): 'hello.xml'
Launch 'p4merge' [Y/n]? Y
```



1 diffs (Ignore line ending differences) | Tab spacing: 4 | Encoding: System

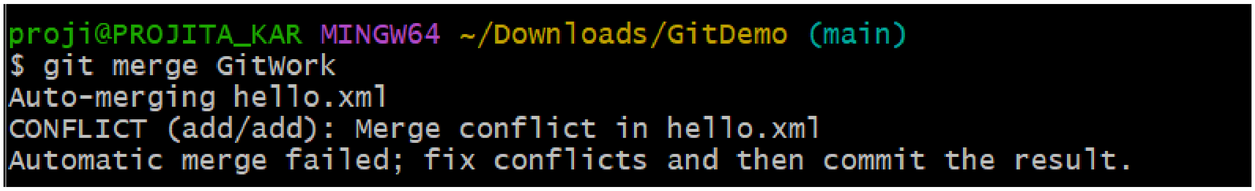
C:\Users\proji\AppData\Local\Temp/git-blob-a36420/hello.xml

```
<greeting>Hello from main branch</greeting>
```

C:\Users\proji\AppData\Local\Temp/git-blob-b36420/hello.xml

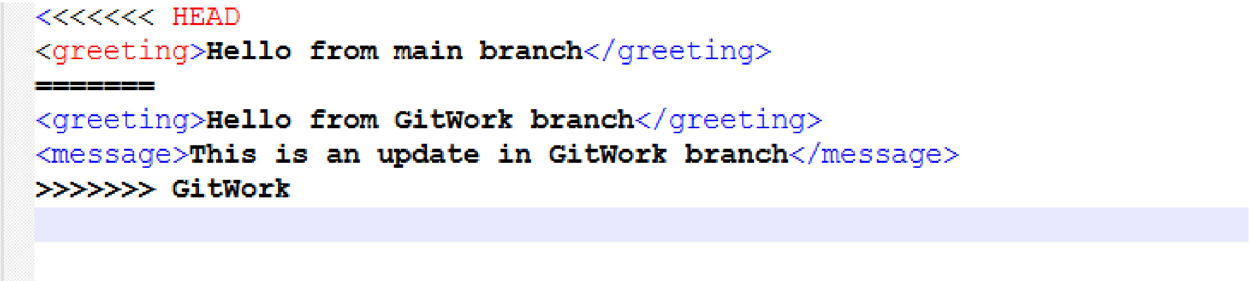
```
<greeting>Hello from GitWork branch</greeting>
<message>This is an update in GitWork branch</message>
```

11. Merge the branch to the main



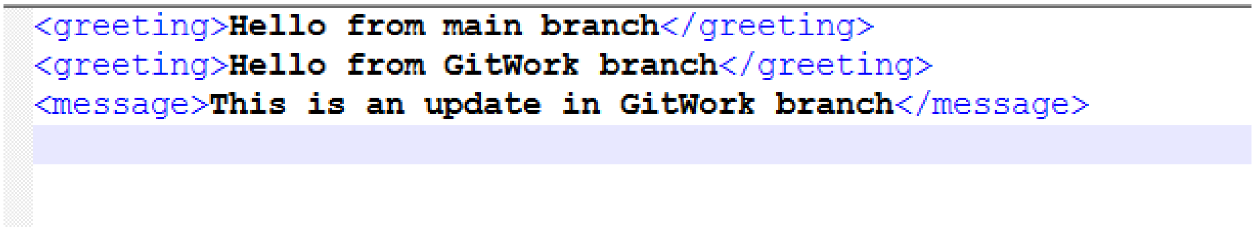
```
proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (main)
$ git merge GitWork
Auto-merging hello.xml
CONFLICT (add/add): Merge conflict in hello.xml
Automatic merge failed; fix conflicts and then commit the result.
```

12. Observe the git mark up.



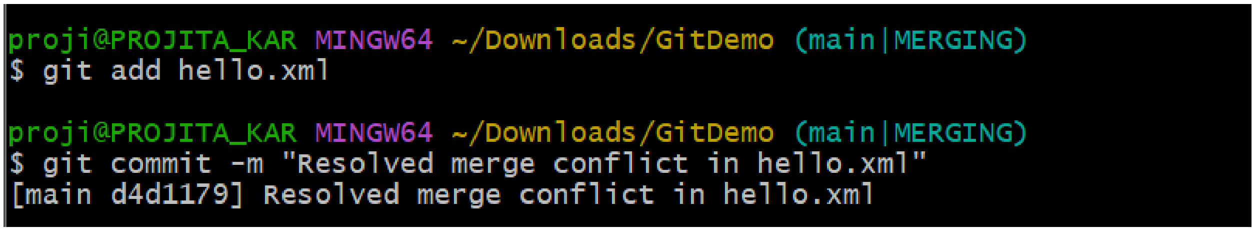
```
<<<<<<< HEAD
<greeting>Hello from main branch</greeting>
=======
<greeting>Hello from GitWork branch</greeting>
<message>This is an update in GitWork branch</message>
>>>>>>> GitWork
```

13. Use 3-way merge tool to resolve the conflict



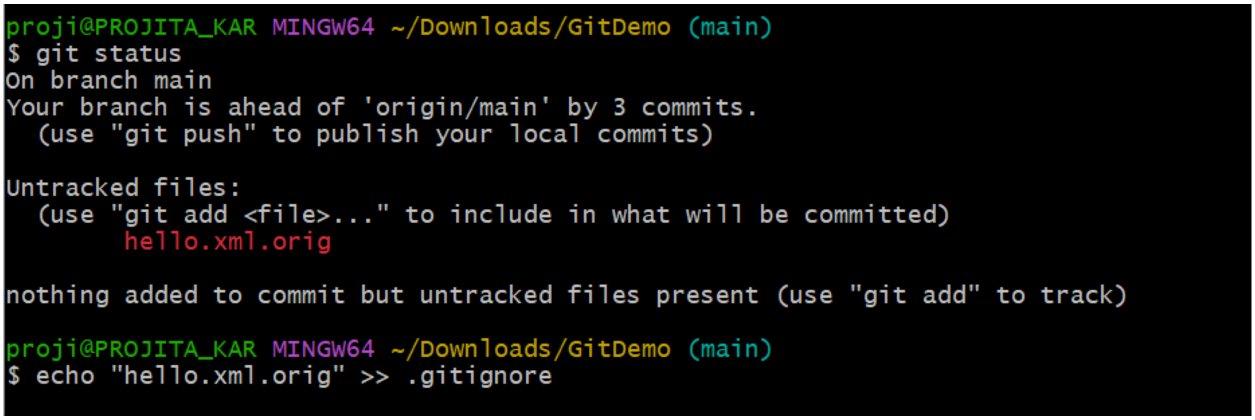
```
<greeting>Hello from main branch</greeting>
<greeting>Hello from GitWork branch</greeting>
<message>This is an update in GitWork branch</message>
```

14. Commit the changes to the main, once done with conflict



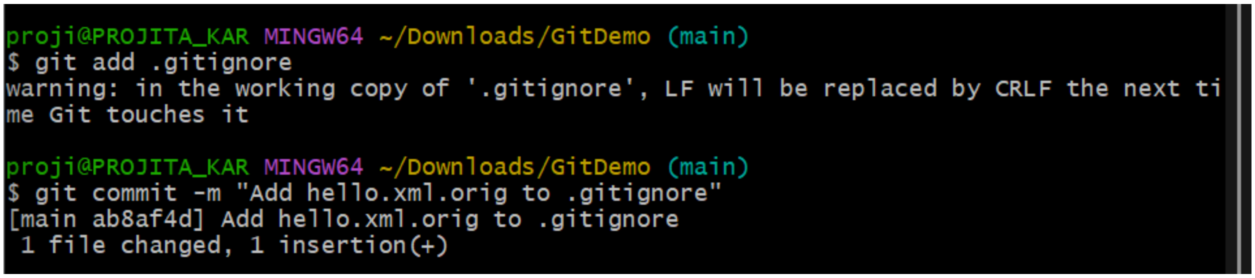
```
proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (main|MERGING)
$ git add hello.xml

proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (main|MERGING)
$ git commit -m "Resolved merge conflict in hello.xml"
[main d4d1179] Resolved merge conflict in hello.xml
```

15. Observe the git status and add backup file to the .gitignore file.



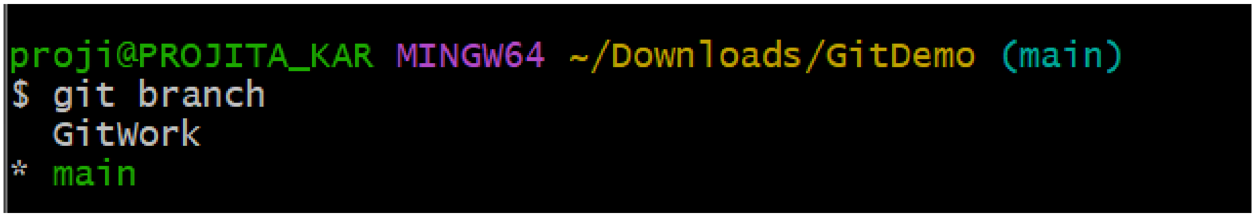
```
proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 3 commits.
  (use "git push" to publish your local commits)

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

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

proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (main)
$ echo "hello.xml.orig" >> .gitignore
```

16. Commit the changes to the .gitignore



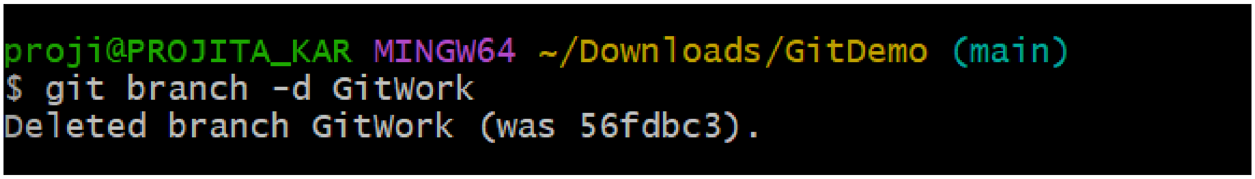
```
proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (main)
$ git add .gitignore
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next ti
me Git touches it

proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (main)
$ git commit -m "Add hello.xml.orig to .gitignore"
[main ab8af4d] Add hello.xml.orig to .gitignore
 1 file changed, 1 insertion(+)
```

17. List out all the available branches



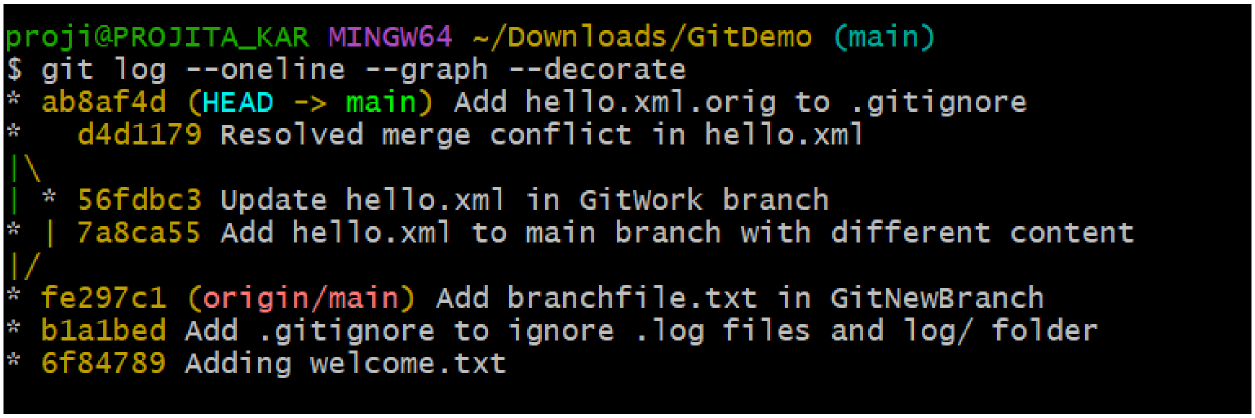
```
proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (main)
$ git branch
  GitWork
* main
```

18. Delete the branch, which merge to main.



```
proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (main)
$ git branch -d GitWork
Deleted branch GitWork (was 56fdbc3).
```

19. Observe the log by executing “git log –oneline –graph –decorate”



```
proji@PROJITA_KAR MINGW64 ~/Downloads/GitDemo (main)
$ git log --oneline --graph --decorate
* ab8af4d (HEAD -> main) Add hello.xml.orig to .gitignore
*   d4d1179 Resolved merge conflict in hello.xml
|\
| * 56fdbc3 Update hello.xml in GitWork branch
* | 7a8ca55 Add hello.xml to main branch with different content
|/
* fe297c1 (origin/main) Add branchfile.txt in GitNewBranch
* b1a1bed Add .gitignore to ignore .log files and log/ folder
* 6f84789 Adding welcome.txt
```