

## WEEK-8 (Hands-on)

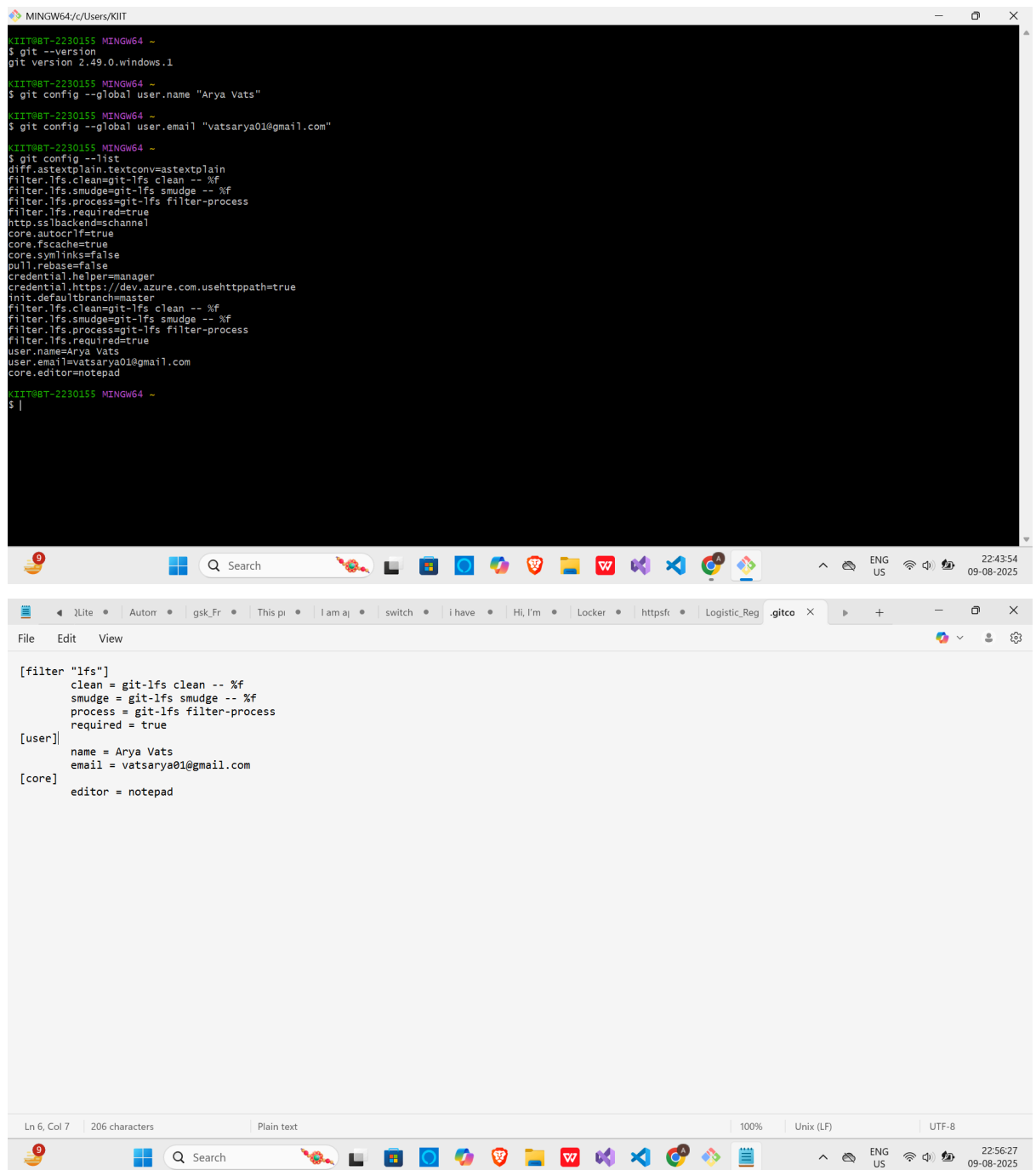
**Name - Arya Vats**

**Superset ID - 6358118**

### Assignment 1 – Git Basics

- `git --version`
- `git config --list`
- `git init`
- `ls -a`
- `echo "Welcome to Git!" > welcome.txt`
- `ls`
- `cat welcome.txt`
- `git status`
- `git add welcome.txt`
- `git commit -m "Added welcome.txt"`
- `git branch -M main`
- `git remote -v`
- `git push -u origin main`
- `git status`

## Screenshots:-



The first screenshot shows a Windows terminal window with the following commands and output:

```
MINGW64/c/Users/KIIT
KIIT@BT-2230155 MINGW64 ~
$ git --version
git version 2.49.0.windows.1

KIIT@BT-2230155 MINGW64 ~
$ git config --global user.name "Arya Vats"

KIIT@BT-2230155 MINGW64 ~
$ git config --global user.email "vatsarya01@gmail.com"

KIIT@BT-2230155 MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=Arya Vats
user.email=vatsarya01@gmail.com
core.editor=notepad
```

The second screenshot shows a Notepad editor window with the following configuration:

```
[filter "lfs"]
  clean = git-lfs clean -- %f
  smudge = git-lfs smudge -- %f
  process = git-lfs filter-process
  required = true

[user]
  name = Arya Vats
  email = vatsarya01@gmail.com

[core]
  editor = notepad
```

```
MINGW64/c/Users/KIIT/GitDemo
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=Arya Vats
user.email=vatsarya01@gmail.com
core.editor=notepad

KIIT@BT-2230155 MINGW64 ~
$ git config --global -e
$ git init
Reinitialized existing Git repository in C:/Users/KIIT/GitDemo/.git/
KIIT@BT-2230155 MINGW64 ~
$ ls -a
./ ../ .git/ welcome.txt
KIIT@BT-2230155 MINGW64 ~
$ echo "welcome ti git!" > welcome.txt
KIIT@BT-2230155 MINGW64 ~
$ ls
welcome.txt
KIIT@BT-2230155 MINGW64 ~
$ cat welcome.txt
welcome ti git!
KIIT@BT-2230155 MINGW64 ~
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   welcome.txt

no changes added to commit (use "git add" and/or "git commit -a")
KIIT@BT-2230155 MINGW64 ~
$
```

```
MINGW64/c/Users/KIIT/GitDemo
KIIT@BT-2230155 MINGW64 ~
$ git remote set-url origin https://github.com/aryavats2/GitDemo.git
KIIT@BT-2230155 MINGW64 ~
$ git remote -v
origin https://github.com/aryavats2/GitDemo.git (fetch)
origin https://github.com/aryavats2/GitDemo.git (push)
KIIT@BT-2230155 MINGW64 ~
$ git pull origin master --allow-unrelated-histories
fatal: couldn't find remote ref master
KIIT@BT-2230155 MINGW64 ~
$ ^C
KIIT@BT-2230155 MINGW64 ~
$ git pull origin main --allow-unrelated-histories
fatal: couldn't find remote ref main
KIIT@BT-2230155 MINGW64 ~
$ git branch -M main
git pull origin main --allow-unrelated-histories
fatal: couldn't find remote ref main
KIIT@BT-2230155 MINGW64 ~
$ git branch -M main
git pull origin main --allow-unrelated-histories
fatal: couldn't find remote ref main
KIIT@BT-2230155 MINGW64 ~
$
KIIT@BT-2230155 MINGW64 ~
$ ^C
KIIT@BT-2230155 MINGW64 ~
$ git branch -M main
git push -u origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 467 bytes | 233.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/aryavats2/GitDemo.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

```
MINGW64/c/Users/KIIT/GitDemo
KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$
KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git pull origin master --allow-unrelated-histories
fatal: couldn't find remote ref master
KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean
KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git add filename.txt
git commit -m "Added new file"
git push
fatal: pathspec 'filename.txt' did not match any files
On branch main
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean
Everything up-to-date
KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git add welcome.txt
git commit -m "Added welcome.txt"
git push
On branch main
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean
Everything up-to-date
KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ ^[[200~git pull origin main
bash: $'\E[200~git': command not found
KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git pull origin main
From https://github.com/aryavats2/GitDemo
* branch      main       -> FETCH_HEAD
Already up to date.
KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$
```

---

## Assignment 2 – Git Ignore

- `cd GitDemo`
- `echo "This is a log file" > test.log`
- `mkdir log`
- `echo "Log folder file" > log/info.txt`
- `echo "*.log" > .gitignore`
- `echo "log/" >> .gitignore`
- `git status` (should only show `.gitignore`)
- `git add .gitignore`
- `git commit -m "Added .gitignore to ignore .log files and log folder"`
- `git push`

- git status

## Screenshots:-

```

MINGW64/c/Users/KIIT/GitDemo
KIIT@BT-2230155 MINGW64 ~
$ cd GitDemo

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ echo "This is a log file" > test.log
mkdir log
echo "Log Folder file" > log/info.txt

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ echo "*.log" > .gitignore
echo "log/" >> .gitignore

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

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

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git add .gitignore
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git commit -m "Added .gitignore to ignore .log files and log folders"
[main 9463c13] Added .gitignore to ignore .log files and log folders
 1 file changed, 2 insertions(+)
 create mode 100644 .gitignore

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 315 bytes | 315.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/aryavats2/GitDemo.git
 f08414f..9463c13  main -> main

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$

```

## Assignment 3 – Branching & Merging

- git branch GitNewBranch
- git branch -a
- git checkout GitNewBranch
- echo "This file is in GitNewBranch" > branchfile.txt
- git add branchfile.txt
- git commit -m "Added branchfile.txt in GitNewBranch"
- git status
- git checkout main
- git diff main GitNewBranch

- `git merge GitNewBranch`
- `git log --oneline --graph --decorate`
- `git branch -d GitNewBranch`
- `git status`

### Screenshots:-

```

MINGW64/c/Users/KIIT/GitDemo
$ git status
On branch GitNewBranch
nothing to commit, working tree clean

KIIT@BT-2230155 MINGW64 ~/GitDemo (GitNewBranch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git diff main GitNewBranch
diff --git a/branchfile.txt b/branchfile.txt
new file mode 100644
index 0000000..be68bfa
--- /dev/null
+++ b/branchfile.txt
@@ -0,0 +1 @@
+This file is in GitNewBranch

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git merge GitNewBranch
Updating 9463c13..fb649fa
Fast-forward
 branchfile.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 branchfile.txt

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git log --oneline --graph --decorate
* fb649fa (HEAD -> main, GitNewBranch) Added branchfile.txt in GitNewBranch
* 9463c13 (origin/main) Added .gitignore to ignore .log files and log folders
* f08414f Added welcome.txt
* 169f178 added a file

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git branch -d GitNewBranch
Deleted branch GitNewBranch (was fb649fa).

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ |

```

```

MINGW64/c/Users/KIIT/GitDemo
KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git branch GitNewBranch

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git branch -a
* main
  remotes/origin/main

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git checkout GitNewBranch
Switched to branch 'GitNewBranch'

KIIT@BT-2230155 MINGW64 ~/GitDemo (GitNewBranch)
$ echo "This file is in GitNewBranch" > branchfile.txt
$ git add branchfile.txt
$ git commit -m "Added branchfile.txt in GitNewBranch"
warning: in the working copy of 'branchfile.txt', LF will be replaced by CRLF the
next time Git touches it
[GitNewBranch fb649fa] Added branchfile.txt in GitNewBranch
 1 file changed, 1 insertion(+)
 create mode 100644 branchfile.txt

KIIT@BT-2230155 MINGW64 ~/GitDemo (GitNewBranch)
$ git status
On branch GitNewBranch
nothing to commit, working tree clean

KIIT@BT-2230155 MINGW64 ~/GitDemo (GitNewBranch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git diff main GitNewBranch
diff --git a/branchfile.txt b/branchfile.txt
new file mode 100644
index 0000000..be68bfa
--- /dev/null
+++ b/branchfile.txt
@@ -0,0 +1 @@
+This file is in GitNewBranch

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git merge GitNewBranch
Updating 9463c13..fb649fa
Fast-forward
 branchfile.txt | 1 +

```

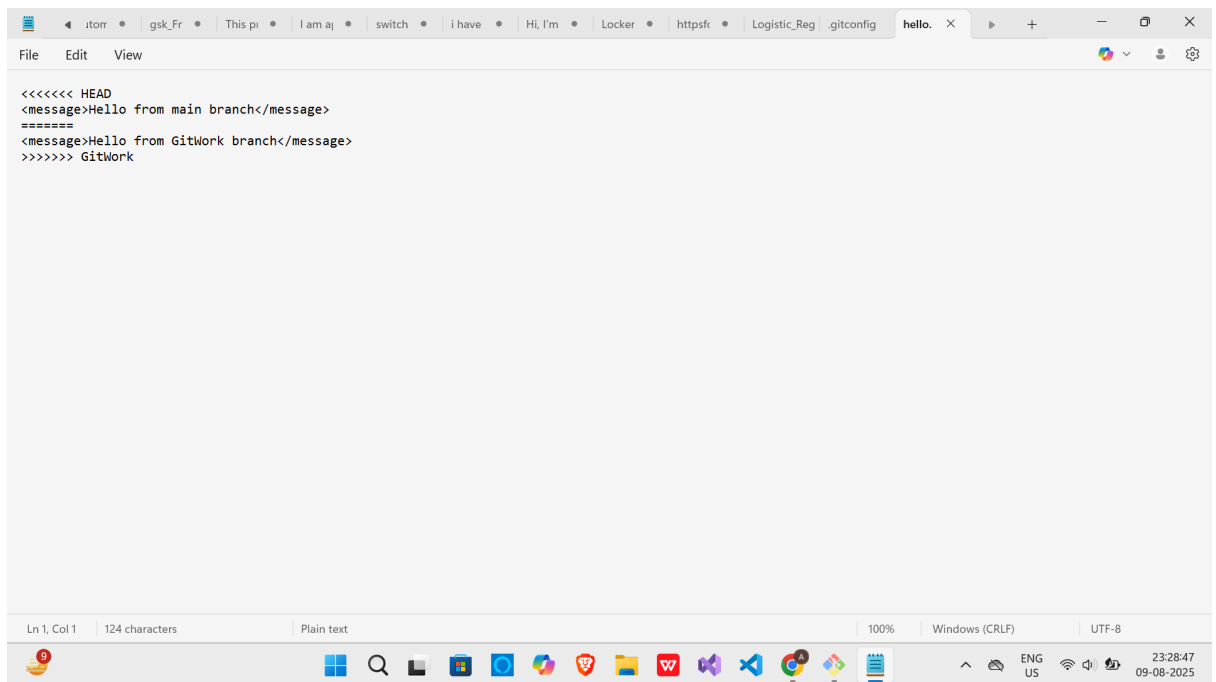
---

## Assignment 4 – Merge Conflict Resolution

- `git checkout main`
- `git status`
- `git branch GitWork`
- `git checkout GitWork`
- `echo "<message>Hello from GitWork branch</message>" > hello.xml`
- `git add hello.xml`
- `git commit -m "Added hello.xml in GitWork"`
- `git checkout main`
- `echo "<message>Hello from main branch</message>" > hello.xml`
- `git add hello.xml`
- `git commit -m "Added hello.xml in main"`
- `git log --oneline --graph --decorate --all`
- `git merge GitWork` (*screenshot showing conflict*)
- `notepad hello.xml` (*screenshot optional, show before edit with conflict markers*)
- **Edit to final version** → save & close
- `git add hello.xml`
- `git commit -m "Resolved merge conflict in hello.xml"`
- `echo "*~" >> .gitignore`
- `git add .gitignore`

- `git commit -m "Ignore backup files"`
- `git branch -d GitWork`
- `git log --oneline --graph --decorate`

### Screenshots:-



The screenshot shows a Windows Notepad application window with the title bar 'hello. x'. The menu bar includes 'File', 'Edit', and 'View'. The text area contains the output of the 'git log' command, which shows two commits. The first commit is the HEAD, with a message 'Hello from main branch'. The second commit is on the 'GitWork' branch, with a message 'Hello from GitWork branch'. The status bar at the bottom indicates 'Ln 1, Col 1', '124 characters', 'Plain text', '100%', 'Windows (CRLF)', and 'UTF-8'. The Windows taskbar is visible at the bottom with various icons and a system clock showing 23:28:47 on 09-08-2025.

```
<<<<<< HEAD
<message>Hello from main branch</message>
=====
<message>Hello from GitWork branch</message>
>>>>>> GitWork
```



```
MINGW64/c/Users/KIIT/GitDemo
KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git checkout main
git status
Already on 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(Use "git push" to publish your local commits)
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(Use "git push" to publish your local commits)

nothing to commit, working tree clean

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git branch Gitwork
git checkout Gitwork
Switched to branch 'Gitwork'

KIIT@BT-2230155 MINGW64 ~/GitDemo (Gitwork)
$ echo "<message>Hello from Gitwork branch</message>" > hello.xml
git add hello.xml
git commit -m "Added hello.xml in Gitwork"
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next ti
me Git touches it
[Gitwork 395cbc4] Added hello.xml in Gitwork
1 file changed, 1 insertion(+)
create mode 100644 hello.xml

KIIT@BT-2230155 MINGW64 ~/GitDemo (Gitwork)
$ git checkout main
echo "<message>Hello from main branch</message>" > hello.xml
git add hello.xml
git commit -m "Added hello.xml in main"
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(Use "git push" to publish your local commits)
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next ti
me Git touches it
[main 6f3ff94] Added hello.xml in main
1 file changed, 1 insertion(+)
create mode 100644 hello.xml

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git log --oneline --graph --decorate --all
* 6f3ff94 (HEAD -> main) Added hello.xml in main
/* 395cbc4 (Gitwork) Added hello.xml in Gitwork
/* fb649fa Added branchfile.txt in GitNewBranch

KIIT@BT-2230155 MINGW64 ~/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.

KIIT@BT-2230155 MINGW64 ~/GitDemo (main|MERGING)
$ notepad hello.xml

KIIT@BT-2230155 MINGW64 ~/GitDemo (main|MERGING)
$ git add hello.xml
git commit -m "Resolved merge conflict in hello.xml"
[main Self01e] Resolved merge conflict in hello.xml

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ echo "+-" >> .gitignore
git add .gitignore
git commit -m "Ignore backup files"
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next t
ime Git touches it
[main 027338a] Ignore backup files
1 file changed, 1 insertion(+)

KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git branch -d Gitwork
Deleted branch Gitwork (was 395cbc4).

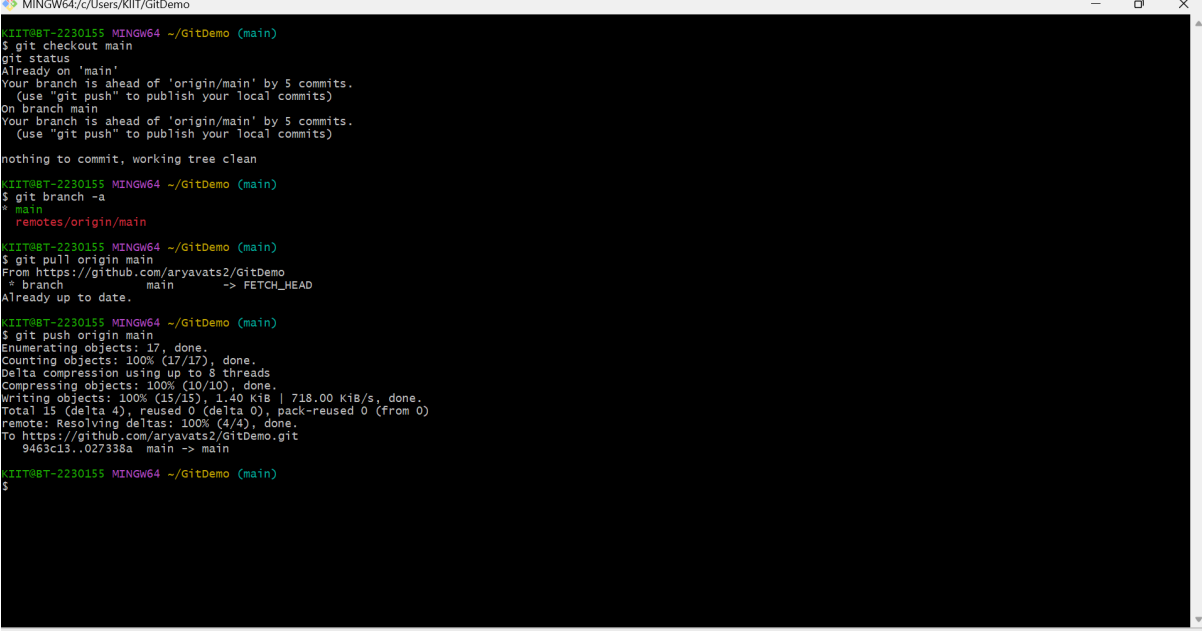
KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git log --oneline --graph --decorate
* 027338a (HEAD -> main) Ignore backup files
/* Self01e Resolved merge conflict in hello.xml
/* 395cbc4 Added hello.xml in Gitwork
/* 6f3ff94 Added hello.xml in main
/* fb649fa Added branchfile.txt in GitNewBranch
/* 9463c13 (origin/main) Added .gitignore to ignore .log files and log folders
/* f08414f Added welcome.txt
/* 169f178 added a file
```

## Assignment 5 – Clean Up & Push

- git checkout main
- git status
- git branch -a

- git pull origin main
- git push origin main

### Screenshots:-



```
MINGW64/c/Users/KIIT/GitDemo
KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git checkout main
$ git status
git status
Already on 'main'
Your branch is ahead of 'origin/main' by 5 commits.
  (use "git push" to publish your local commits)
On branch main
Your branch is ahead of 'origin/main' by 5 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git branch -a
* main
  remotes/origin/main
KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git pull origin main
From https://github.com/aryavats2/GitDemo
  * branch      main       -> FETCH_HEAD
Already up to date.
KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$ git push origin main
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 8 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (15/15), 1.40 KiB | 718.00 KiB/s, done.
Total 15 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), done.
To https://github.com/aryavats2/GitDemo.git
   9463c13..027338a  main -> main
KIIT@BT-2230155 MINGW64 ~/GitDemo (main)
$
```

The screenshot shows a Windows terminal window with the title bar "MINGW64/c/Users/KIIT/GitDemo". The terminal displays the output of several Git commands. The first command is "git checkout main", followed by "git status", which shows the current branch is 'main' and it is ahead of 'origin/main' by 5 commits. The next command is "git branch -a", which lists the local 'main' branch and the remote 'remotes/origin/main' branch. This is followed by "git pull origin main", which shows the pull is already up to date. The final command is "git push origin main", which shows the push is successful, updating the remote 'main' branch from 9463c13 to 027338a. The Windows taskbar is visible at the bottom of the screen.