TASK 1: Repository Setup
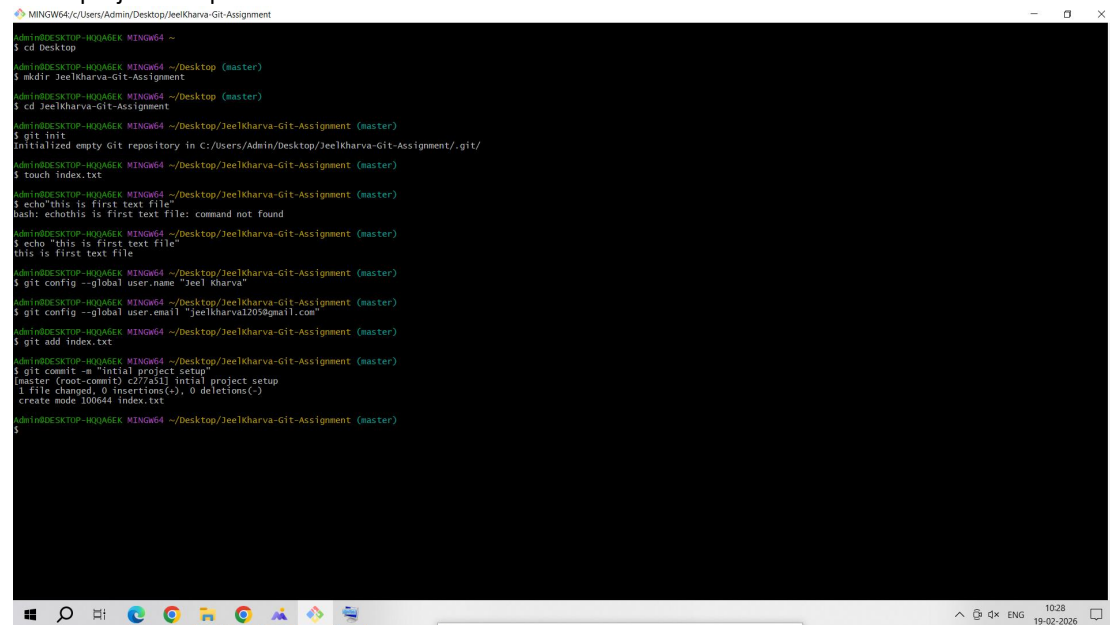Goal: Initialize Git and create first commit
Steps:
1. Create a folder named git-assignment-yourfullname
2. Navigate into the folder
3. Initialize Git repository
4. Create a file named index.txt

5. Check repository status
6. Stage the file
7. Commit with message:
Initial project setup

TASK 2: Branch Creation & Work
Goal: Work independently in a branch
Steps:
1. Create a branch named feature-header
2. Switch to feature-header
3. Create a file named header.txt
4. Stage the file
5. Commit with message:
Added header section

TASK 3: Branch Switching Observation

Goal: Observe branch isolation

Steps:

1. Switch back to master

2. List files and observe

3. Switch again to feature-header

4. List files and observe

TASK 4: Undo a Mistake
Goal: Safely undo file changes
Steps:

1. In feature-header, modify index.txt
2. Check repository status
3. Undo the change using Git
4. Verify status again

TASK 5: Merge Feature Branch
Goal: Combine work into master
Steps:
1. Switch to master
2. Merge feature-header
3. Verify merged files
4. Check commit history


```
Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment (feature-header)
$ echo "wrong line added" >> index.txt

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment (feature-header)
$ git status
On branch feature-header
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:   index.txt

no changes added to commit (use "git add" and/or "git commit -a")

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment (feature-header)
$ ^C

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment (feature-header)
$ git add
Nothing specified, nothing added.
hint: Maybe you wanted to say 'git add .'?
hint: Disable this message with "git config set advice.addEmptyPathspec false"

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment (feature-header)
$ git add index.txt
warning: in the working copy of 'index.txt', LF will be replaced by CRLF the next time Git touches it

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment ((feature-header)
$ git commit -m "updated index file"
[feature-header 8a24474] updated index file
 1 file changed, 1 insertion(+)

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment (feature-header)
$ it status
bash: it: command not found

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment (feature-header)
$ git status
On branch feature-header
nothing to commit, working tree clean

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment (feature-header)
$ git checkout -- index.txt

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment (feature-header)
$ git status
On branch feature-header
nothing to commit, working tree clean

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment (feature-header)
$ git checkout master
Switched to branch 'master'

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment (master)
$ git merge feacher-header
merge: feacher-header - not something we can merge

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment (master)
$ |
```

TASK 6: Checkout a Commit (Detached HEAD)
Goal: Understand commit-level navigation
Steps:
1. View commit history
2. Checkout any older commit
3. Check current branch status
4. Switch back safely to master


```
Date:   Thu Feb 19 10:27:43 2026 +0530

    intial project setup

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment (master)
$ git checkout <commit-id>
bash: syntax error near unexpected token `newline'

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment (master)
$ ^C

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment (master)
$ git log
commit c277a51eb1caaf1af0578d75e80cf32fb0f851d5 (HEAD -> master)
Author: Jeel Kharva <jeelkharva1205@gmail.com>
Date:   Thu Feb 19 10:27:43 2026 +0530

    intial project setup

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment (master)
$ git checkout AC

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment (master)
$ git checkout c277a51eb1caaf1af0578d75e80cf32fb0f851d5
error: pathspec 'c277a51eb1caaf1af0578d75e80cf32fb0f851d5' did not match any file(s) known to git

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment (master)
$ git checkout c277a51
Note: switching to 'c277a51'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at c277a51 intial project setup

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment ((c277a51...))
$ git status
HEAD detached at c277a51
nothing to commit, working tree clean

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment ((c277a51...))
$ git checkout master
Switched to branch 'master'

Admin@DESKTOP-HQQA6EK MINGW64 ~/Desktop/JeelKharva-Git-Assignment (master)
$
```