LAB MANUAL

Project Management with GitHub

Varsha Suresh, 4GW24CI060, 3RD Sem, CSE(AI&ML)

Program 1: Setting Up and Basic Commands

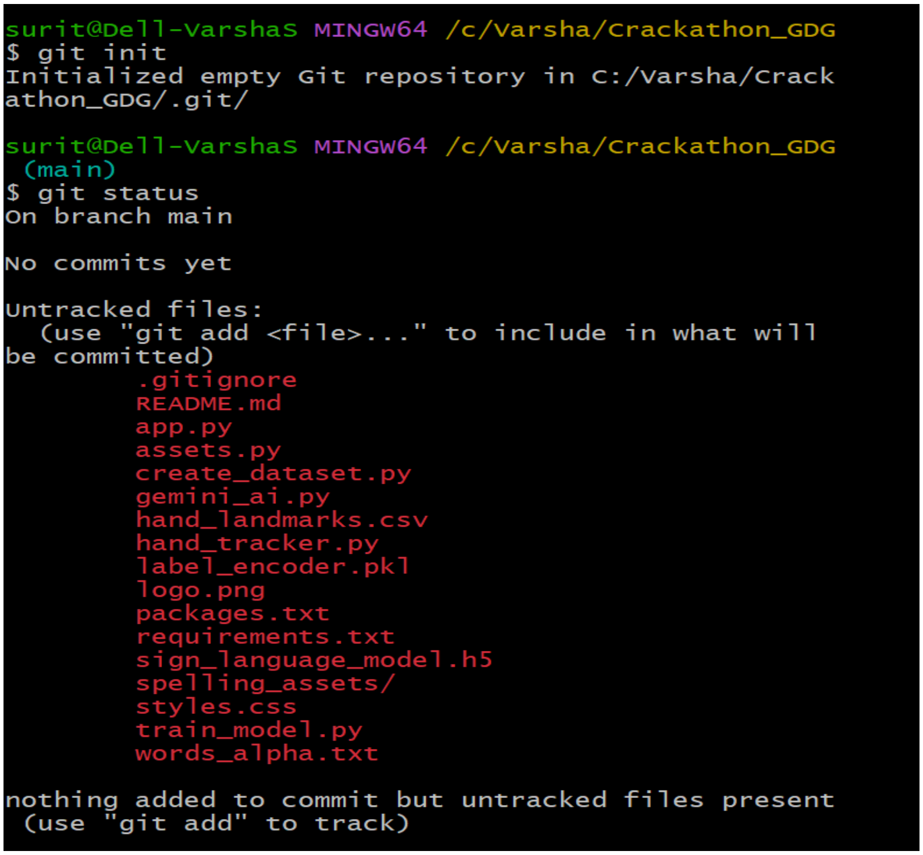
Create a new folder in a directory. Open GitBash from there.

Run the init command:
git init

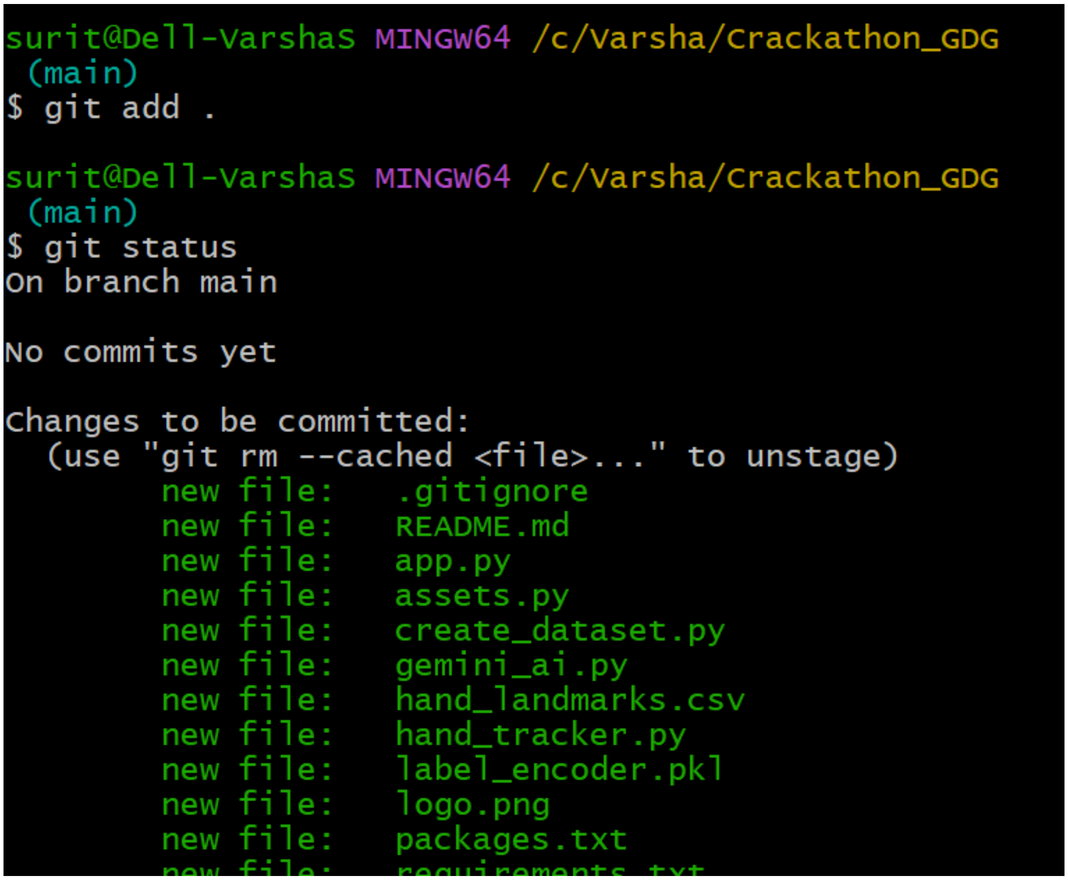
Give your identity ie., your username and your email ID.

git config user.email “varsha.suresh.mys@gmail.com”

git config user.name "varsha-sureshh"



```
surit@Dell-Varshas MINGW64 /c/Varsha/Crackathon_GDG
$ git init
Initialized empty Git repository in C:/Varsha/Crack
athon_GDG/.git/

surit@Dell-Varshas MINGW64 /c/Varsha/Crackathon_GDG
 (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will
be committed)
        .gitignore
        README.md
        app.py
        assets.py
        create_dataset.py
        gemini_ai.py
        hand_landmarks.csv
        hand_tracker.py
        label_encoder.pkl
        logo.png
        packages.txt
        requirements.txt
        sign_language_model.h5
        spelling_assets/
        styles.css
        train_model.py
        words_alpha.txt

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



```
surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_GDG
 (main)
$ git add .

surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_GDG
 (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .gitignore
        new file:   README.md
        new file:   app.py
        new file:   assets.py
        new file:   create_dataset.py
        new file:   gemini_ai.py
        new file:   hand_landmarks.csv
        new file:   hand_tracker.py
        new file:   label_encoder.pkl
        new file:   logo.png
        new file:   packages.txt
        new file:   requirements.txt
```

git init

git status

git add <file-name>

git commit -m "Initial commit"

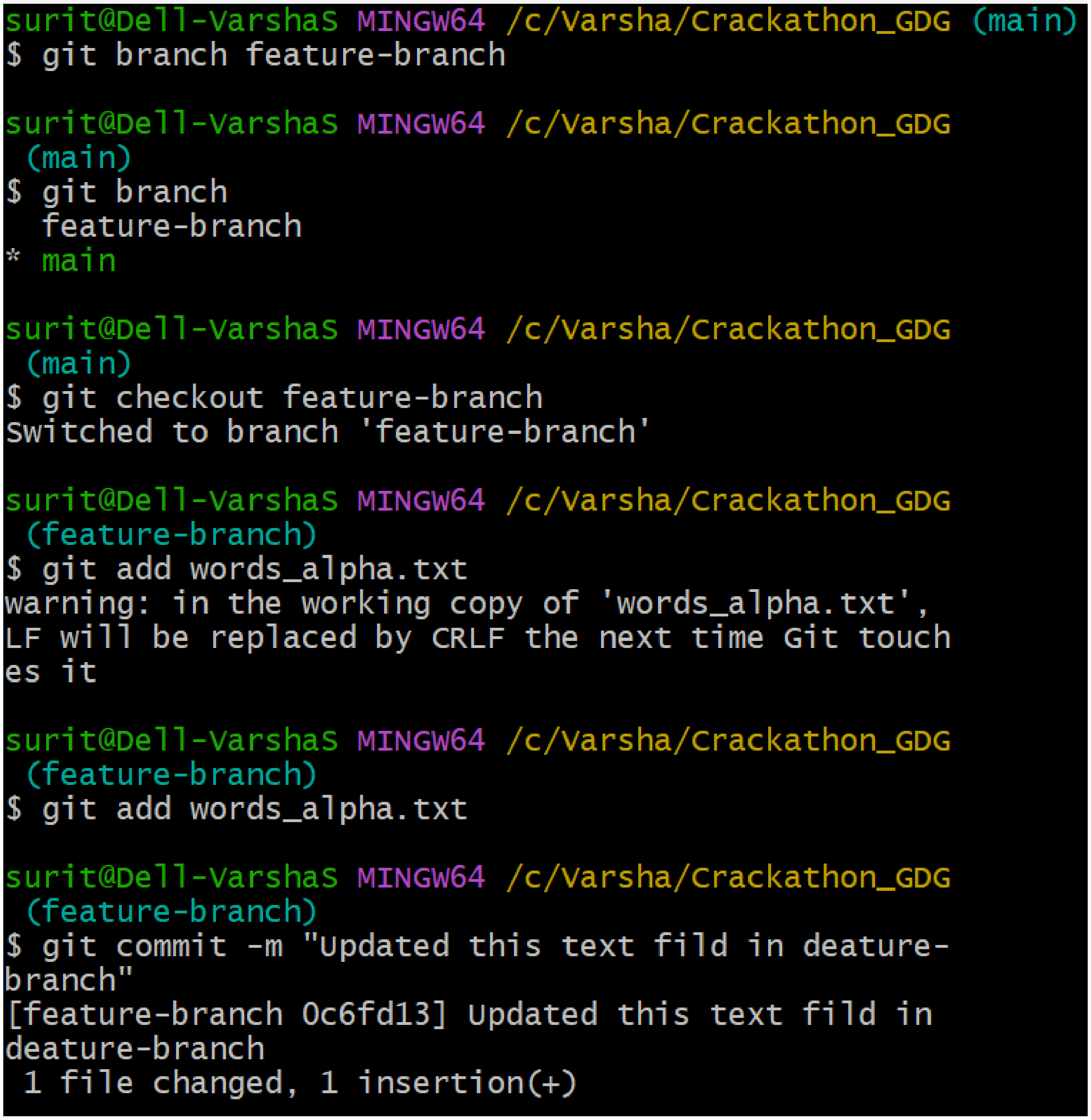
This experiment initializes a new Git repository, adds a file to the staging area, and commits the changes with an appropriate commit message.

Program 2: Creating and Managing Branches

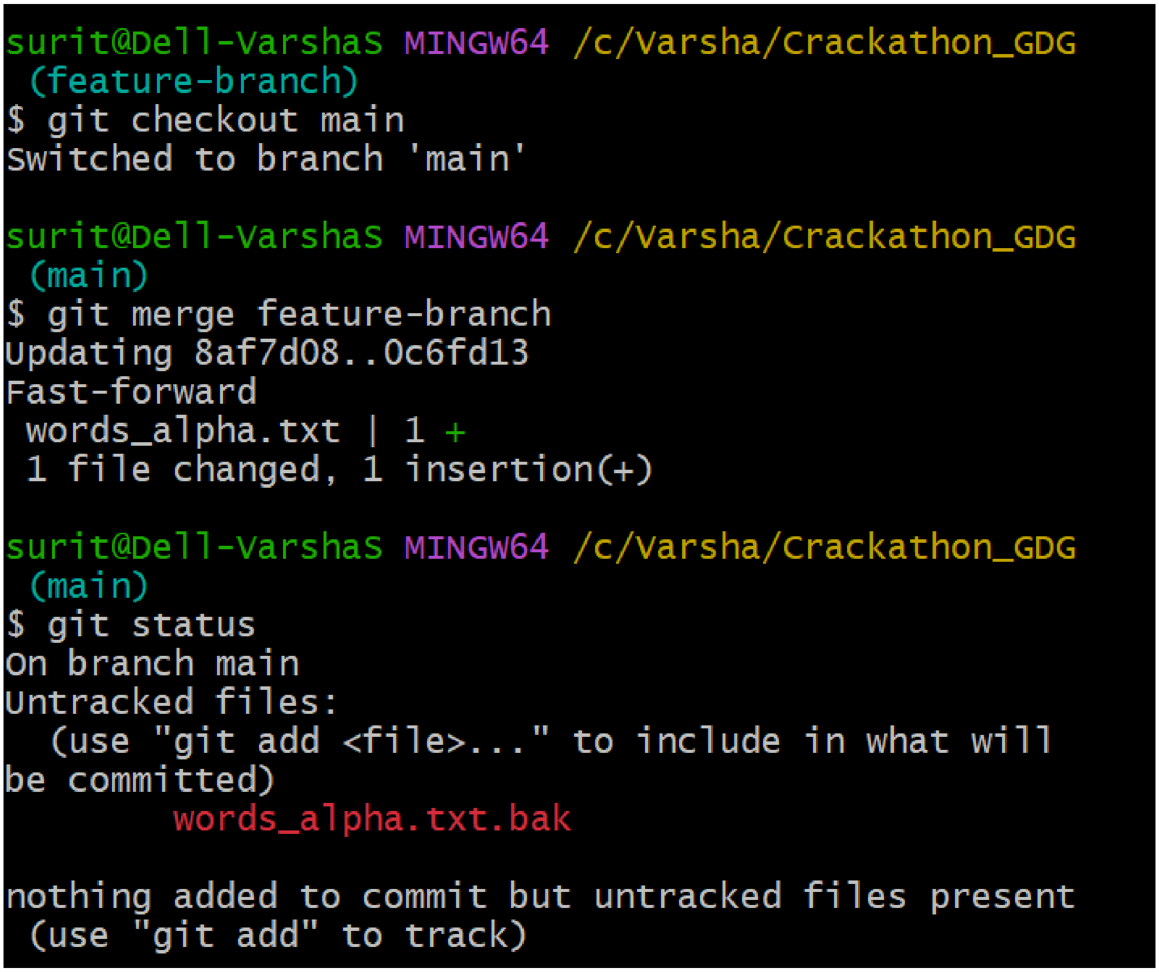
git branch feature-branch

git checkout master

git merge feature-branch



```
surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_GDG (main)
$ git branch feature-branch

surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_GDG
 (main)
$ git branch
  feature-branch
* main

surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_GDG
 (main)
$ git checkout feature-branch
Switched to branch 'feature-branch'

surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_GDG
 (feature-branch)
$ git add words_alpha.txt
warning: in the working copy of 'words_alpha.txt',
LF will be replaced by CRLF the next time Git touch
es it

surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_GDG
 (feature-branch)
$ git add words_alpha.txt

surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_GDG
 (feature-branch)
$ git commit -m "Updated this text fild in deature-
branch"
[feature-branch 0c6fd13] Updated this text fild in
deature-branch
 1 file changed, 1 insertion(+)
```



```
surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_GDG
 (feature-branch)
$ git checkout main
Switched to branch 'main'

surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_GDG
 (main)
$ git merge feature-branch
Updating 8af7d08..0c6fd13
Fast-forward
 words_alpha.txt | 1 +
 1 file changed, 1 insertion(+)

surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_GDG
 (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will
be committed)
        words_alpha.txt.bak

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

git branch feature-branch

git checkout feature-branch

git checkout main

git merge feature-branch

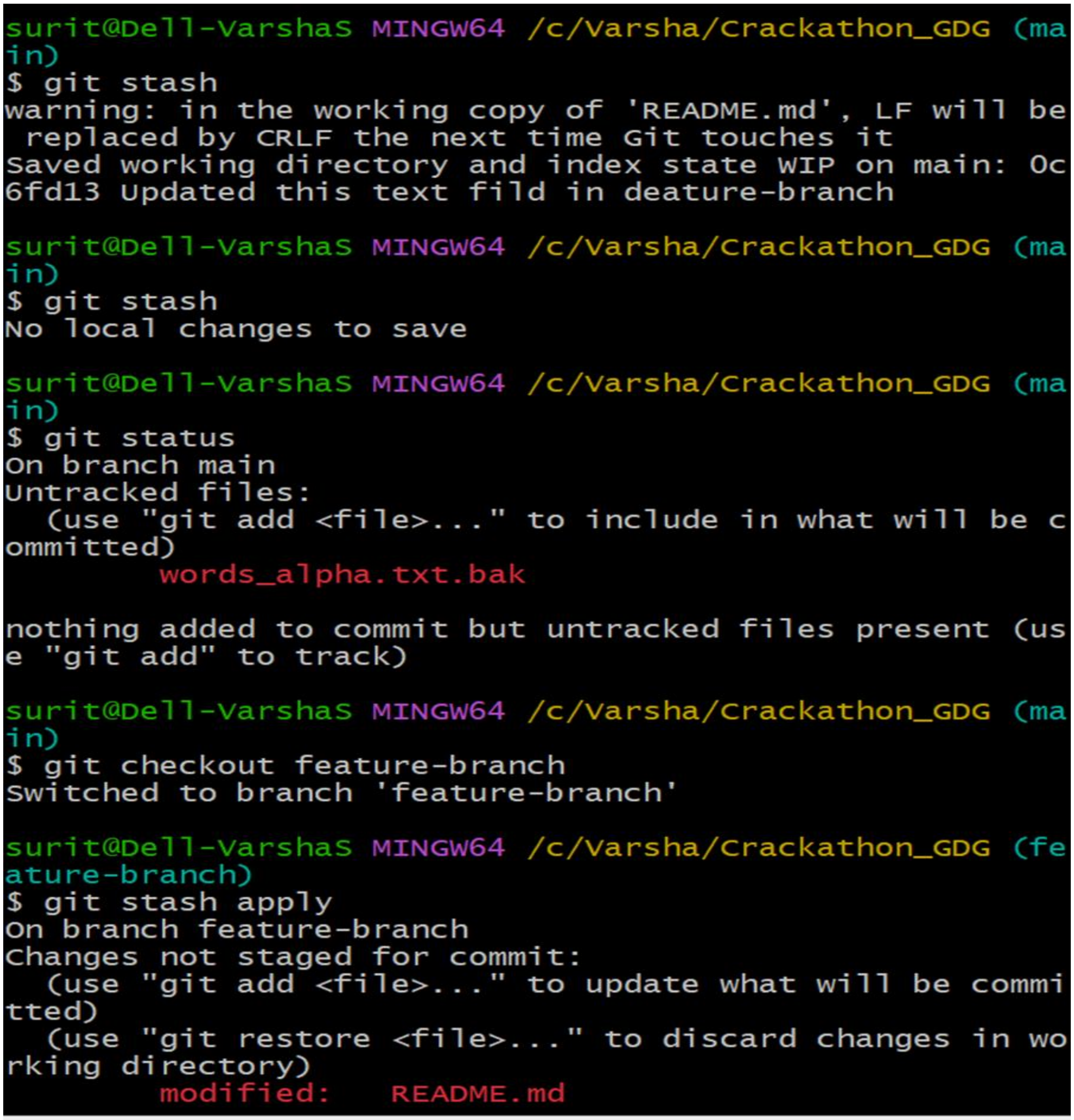
This experiment demonstrates how to create a new branch, switch between branches, and merge a feature branch into the main branch.

Program 3: Stashing Changes

git stash

git checkout feature-branch

git stash apply

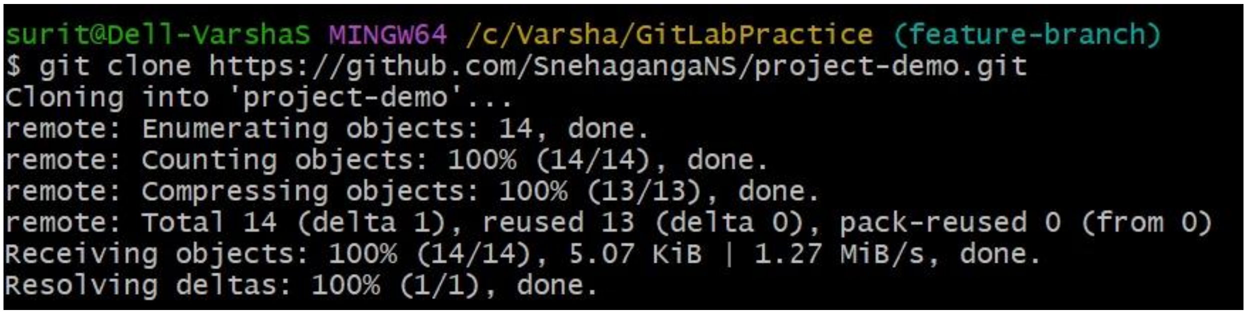


```
surit@Dell-Varshas MINGW64 /c/Varsha/Crackathon_GDG (ma
in)
$ git stash
warning: in the working copy of 'README.md', LF will be
 replaced by CRLF the next time Git touches it
Saved working directory and index state WIP on main: 0c
6fd13 Updated this text fild in deature-branch

surit@Dell-Varshas MINGW64 /c/Varsha/Crackathon_GDG (ma
in)
$ git stash
No local changes to save

surit@Dell-Varshas MINGW64 /c/Varsha/Crackathon_GDG (ma
in)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be c
ommitted)
        words_alpha.txt.bak

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

surit@Dell-Varshas MINGW64 /c/Varsha/Crackathon_GDG (ma
in)
$ git checkout feature-branch
Switched to branch 'feature-branch'

surit@Dell-Varshas MINGW64 /c/Varsha/Crackathon_GDG (fe
ature-branch)
$ git stash apply
On branch feature-branch
Changes not staged for commit:
  (use "git add <file>..." to update what will be commi
tted)
  (use "git restore <file>..." to discard changes in wo
rking directory)
        modified:   README.md
```

This experiment temporarily stores uncommitted changes using Git stash, switches branches safely, and restores the stashed changes.

Program 4: Cloning a Remote Repository

git clone <repository-url>



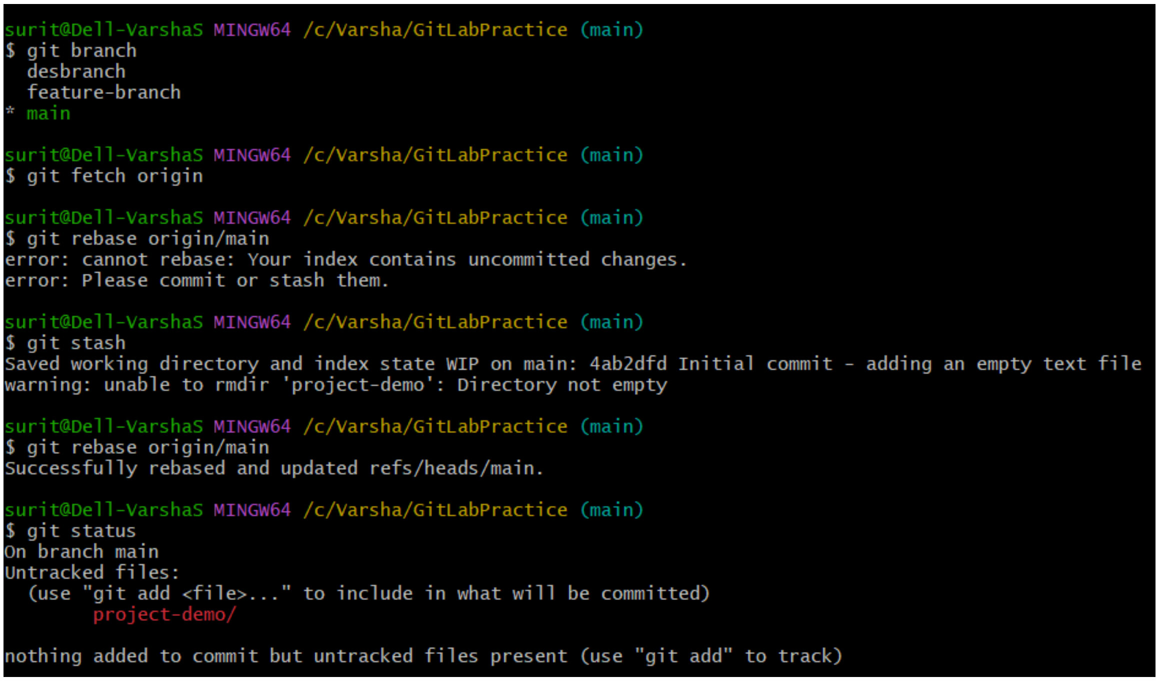
```
surit@Dell-VarshaS MINGW64 /c/Varsha/GitLabPractice (feature-branch)
$ git clone https://github.com/SnehagangaNS/project-demo.git
Cloning into 'project-demo'...
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 14 (delta 1), reused 13 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (14/14), 5.07 KiB | 1.27 MiB/s, done.
Resolving deltas: 100% (1/1), done.
```

This experiment clones a remote Git repository to the local machine, including all files, branches, and commit history.

Program 5: Fetch and Rebase

git fetch origin

git rebase origin/main



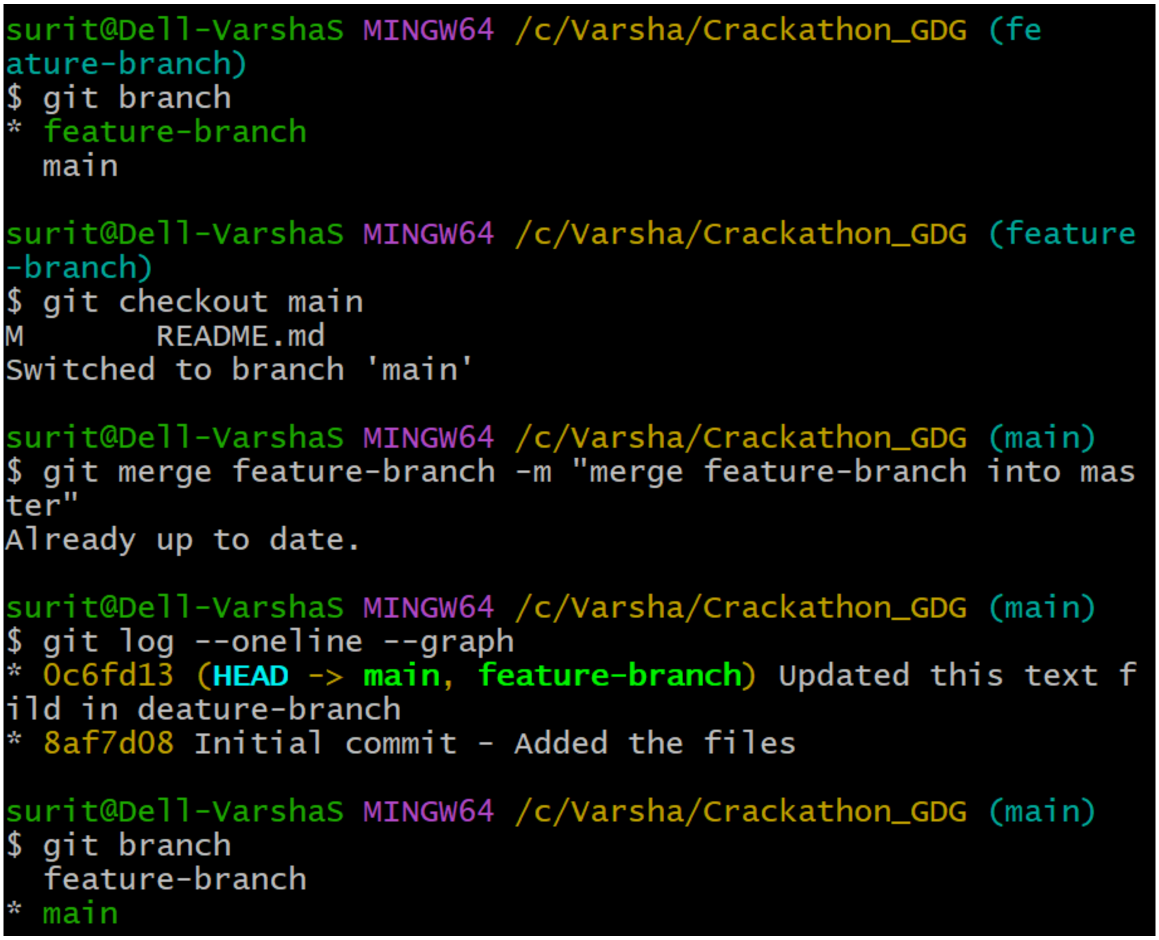
```
surit@Dell-VarshaS MINGW64 /c/Varsha/GitLabPractice (main)
$ git branch
  desbranch
  feature-branch
* main

surit@Dell-VarshaS MINGW64 /c/Varsha/GitLabPractice (main)
$ git fetch origin

surit@Dell-VarshaS MINGW64 /c/Varsha/GitLabPractice (main)
$ git rebase origin/main
error: cannot rebase: Your index contains uncommitted changes.
error: Please commit or stash them.

surit@Dell-VarshaS MINGW64 /c/Varsha/GitLabPractice (main)
$ git stash
Saved working directory and index state WIP on main: 4ab2dfd Initial commit - adding an empty text file
warning: unable to rmdir 'project-demo': Directory not empty

surit@Dell-VarshaS MINGW64 /c/Varsha/GitLabPractice (main)
$ git rebase origin/main
Successfully rebased and updated refs/heads/main.

surit@Dell-VarshaS MINGW64 /c/Varsha/GitLabPractice (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        project-demo/

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

This experiment fetches the latest changes from the remote repository and rebases the local branch onto the updated remote branch to maintain a clean history.

Program 6: Merge with Custom Message

git checkout main

git merge feature-branch -m "Merge feature-branch into main"



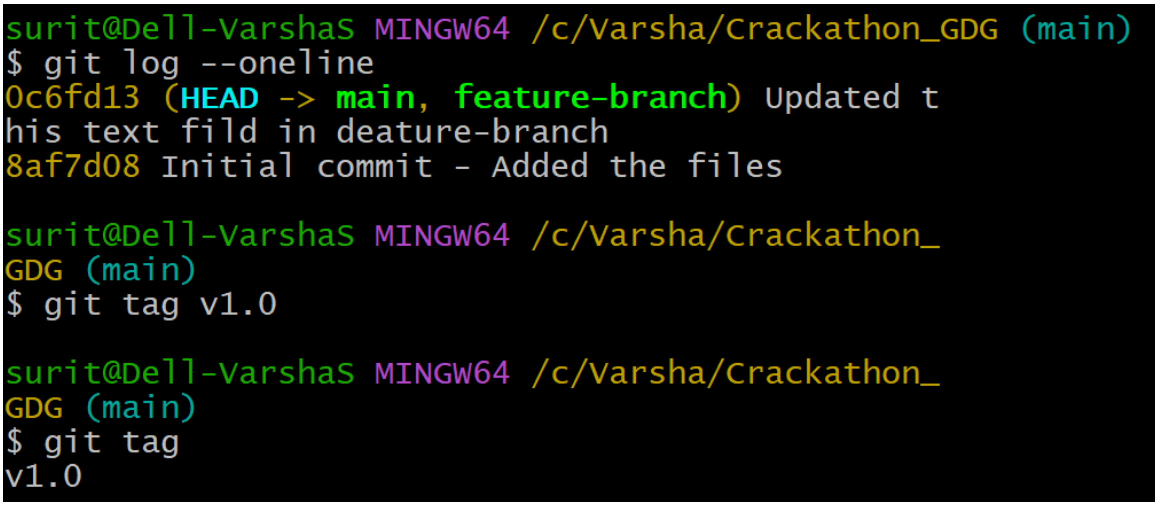
```
surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_GDG (fe
ature-branch)
$ git branch
* feature-branch
  main

surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_GDG (feature
-branch)
$ git checkout main
M       README.md
Switched to branch 'main'

surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_GDG (main)
$ git merge feature-branch -m "merge feature-branch into mas
ter"
Already up to date.

surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_GDG (main)
$ git log --oneline --graph
* 0c6fd13 (HEAD -> main, feature-branch) Updated this text f
ild in deature-branch
* 8af7d08 Initial commit - Added the files

surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_GDG (main)
$ git branch
  feature-branch
* main
```

This experiment merges a feature branch into the main branch while providing a custom merge commit message.

Program 7: Git Tags

Command:

git tag v1.0

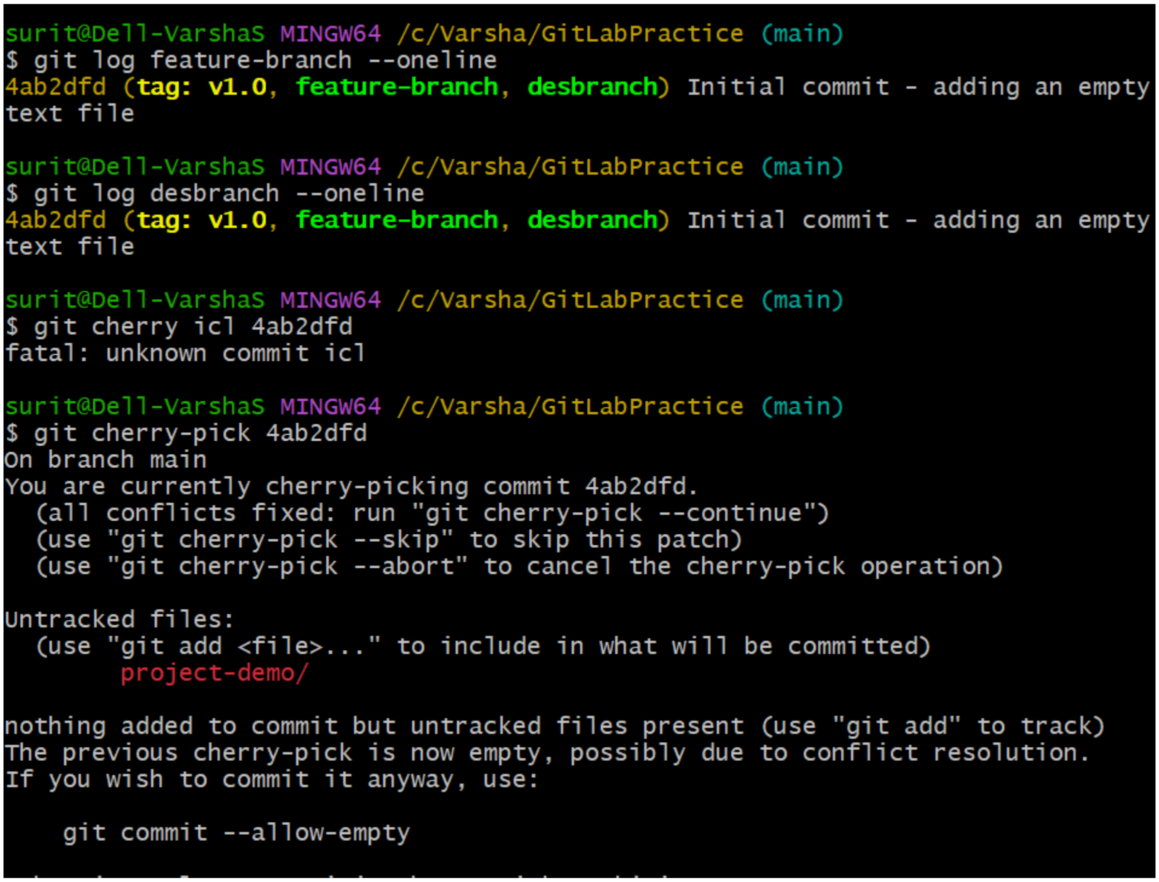


```
surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_GDG (main)
$ git log --oneline
0c6fd13 (HEAD -> main, feature-branch) Updated t
his text fild in deature-branch
8af7d08 Initial commit - Added the files

surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_
GDG (main)
$ git tag v1.0

surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_
GDG (main)
$ git tag
v1.0
```

This experiment creates a lightweight Git tag named v1.0 that points to a specific commit in the repository.

Program 8: Cherry-pick Commits

git cherry-pick <start-commit>^..<end-commit>

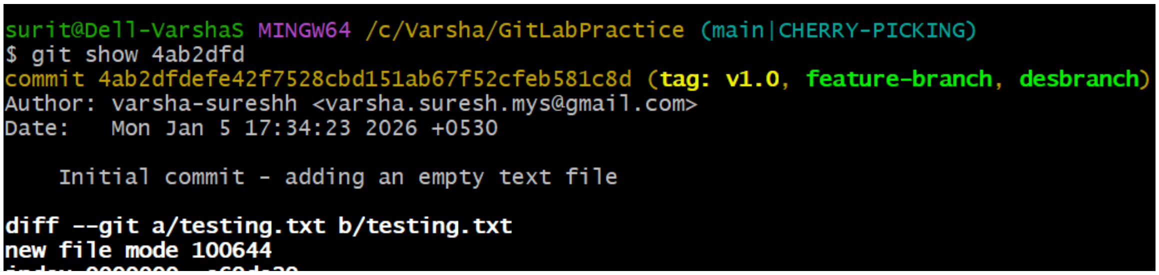


```
surit@Dell-VarshaS MINGW64 /c/Varsha/GitLabPractice (main)
$ git log feature-branch --oneline
4ab2dfd (tag: v1.0, feature-branch, desbranch) Initial commit - adding an empty
text file

surit@Dell-VarshaS MINGW64 /c/Varsha/GitLabPractice (main)
$ git log desbranch --oneline
4ab2dfd (tag: v1.0, feature-branch, desbranch) Initial commit - adding an empty
text file

surit@Dell-VarshaS MINGW64 /c/Varsha/GitLabPractice (main)
$ git cherry icl 4ab2dfd
fatal: unknown commit icl

surit@Dell-VarshaS MINGW64 /c/Varsha/GitLabPractice (main)
$ git cherry-pick 4ab2dfd
On branch main
You are currently cherry-picking commit 4ab2dfd.
  (all conflicts fixed: run "git cherry-pick --continue")
  (use "git cherry-pick --skip" to skip this patch)
  (use "git cherry-pick --abort" to cancel the cherry-pick operation)

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

nothing added to commit but untracked files present (use "git add" to track)
The previous cherry-pick is now empty, possibly due to conflict resolution.
If you wish to commit it anyway, use:

    git commit --allow-empty
```

This experiment applies a selected range of commits from a source branch onto the current branch using cherry-pick.

Program 9: View Commit Details

git show <commit id>

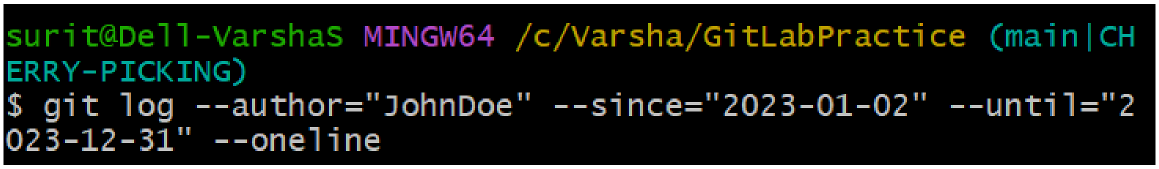


```
surit@Dell-VarshaS MINGW64 /c/Varsha/GitLabPractice (main|CHERRY-PICKING)
$ git show 4ab2dfd
commit 4ab2dfdefe42f7528cbd151ab67f52cfeb581c8d (tag: v1.0, feature-branch, desbranch)
Author: varsha-sureshh <varsha.suresh.mys@gmail.com>
Date:   Mon Jan 5 17:34:23 2026 +0530

    Initial commit - adding an empty text file

diff --git a/testing.txt b/testing.txt
new file mode 100644
index 0000000..e69de29
```

This shows the details made by that commit

Program 10: List Commits by Author

git log --author="JohnDoe" --since="2023-01-01" --until="2023-12-31"

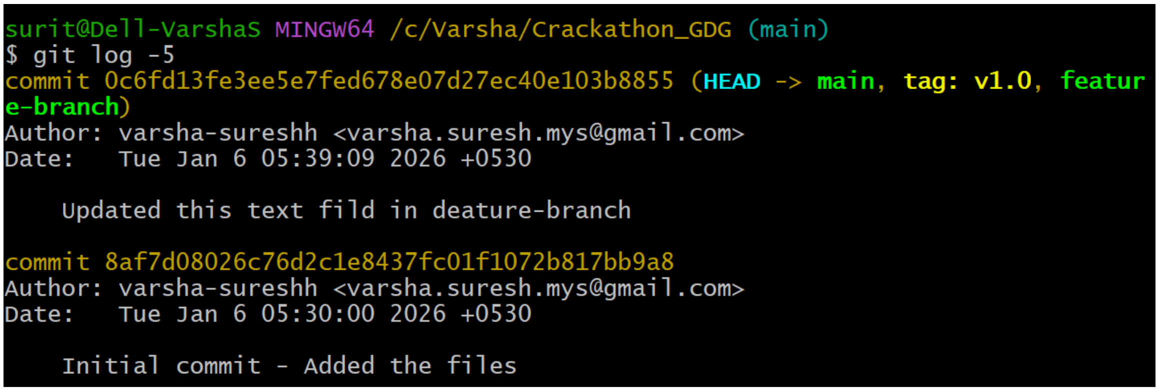


```
surit@Dell-VarshaS MINGW64 /c/Varsha/GitLabPractice (main|CH
ERRY-PICKING)
$ git log --author="JohnDoe" --since="2023-01-02" --until="2
023-12-31" --oneline
```

This experiment displays all commits made by a specific author within a given date range.

Program 11: Last Five Commits

git log -5



```
surit@Dell-VarshaS MINGW64 /c/Varsha/Crackathon_GDG (main)
$ git log -5
commit 0c6fd13fe3ee5e7fed678e07d27ec40e103b8855 (HEAD -> main, tag: v1.0, featur
e-branch)
Author: varsha-sureshh <varsha.suresh.mys@gmail.com>
Date:   Tue Jan 6 05:39:09 2026 +0530

    Updated this text fild in deature-branch

commit 8af7d08026c76d2c1e8437fc01f1072b817bb9a8
Author: varsha-sureshh <varsha.suresh.mys@gmail.com>
Date:   Tue Jan 6 05:30:00 2026 +0530

    Initial commit - Added the files
```

This experiment displays the most recent five commits in the repository’s history

Program 12: Undo Commit

git revert abc123

This experiment undoes the changes introduced by a specific commit by creating a new commit that reverses those changes.