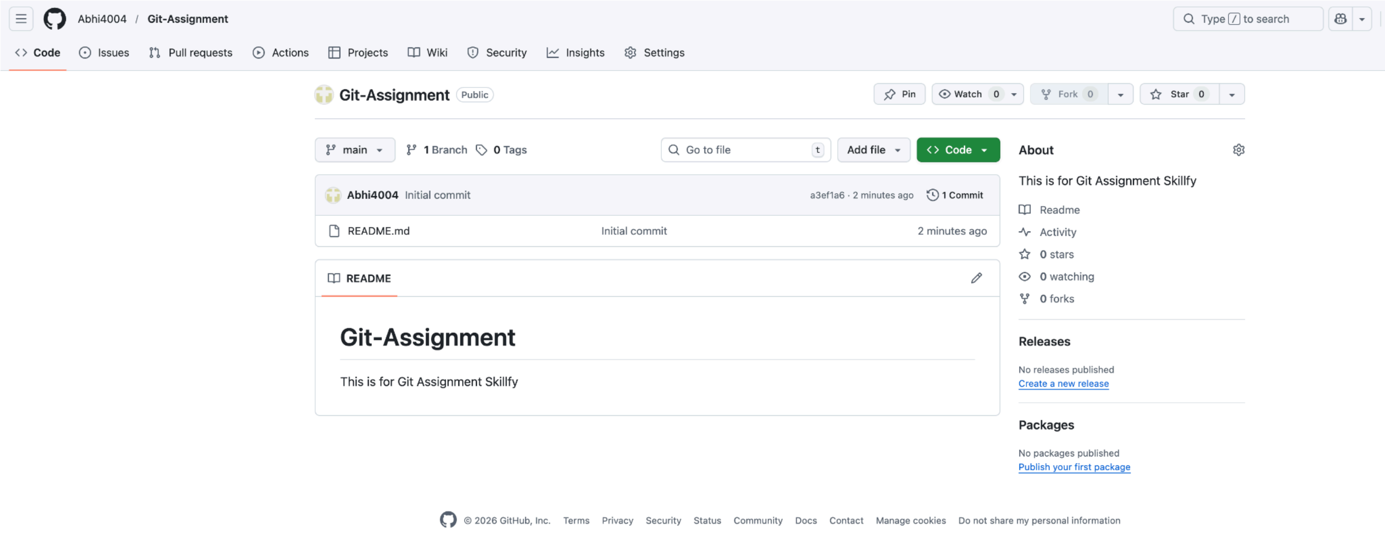
Task 1: Setting up a Git Repository

New repository created: https://github.com/Abhi4004/Git-Assignment



Git-Assignment Public

Abhi4004 Initial commit		a3ef1a6 · 2 minutes ago 1 Commit
README.md	Initial commit	2 minutes ago

README

Git-Assignment

This is for Git Assignment Skillfy

About

This is for Git Assignment Skillfy

Readme
Activity
0 stars
0 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

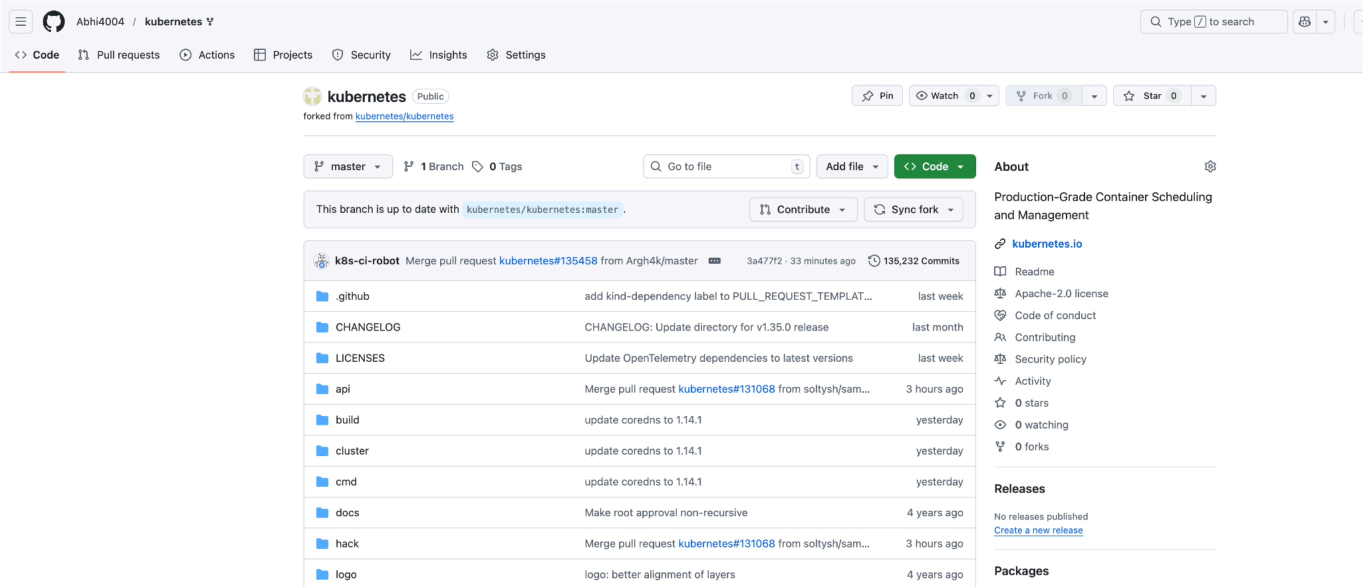
© 2026 GitHub, Inc. Terms Privacy Security Status Community Docs Contact Manage cookies Do not share my personal information

Generated SSH keys locally and added the public key to the repository, enabling seamless login.
The remote repository was then cloned to the local machine using the `git clone` command.

```
abhiram@ubuntuvm:~/git$ git clone git@github.com:Abhi4004/Git-Assignment.git
Cloning into 'Git-Assignment'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
abhiram@ubuntuvm:~/git$
abhiram@ubuntuvm:~/git$ ls -lrt
total 4
drwxrwxr-x 3 abhiram abhiram 4096 Jan 30 10:06 Git-Assignment
abhiram@ubuntuvm:~/git$
abhiram@ubuntuvm:~/git$ cd Git-Assignment/
abhiram@ubuntuvm:~/git/Git-Assignment$
abhiram@ubuntuvm:~/git/Git-Assignment$ ls -lrt
total 4
-rw-rw-r-- 1 abhiram abhiram 52 Jan 30 10:06 README.md
```

Task 2: Forking and cloning a project.

Forked the official Kubernetes repository (https://github.com/Abhi4004/kubernetes) to enable team contributions and engagement with the open-source community.



kubernetes Public

forked from kubernetes/kubernetes

This branch is up to date with `kubernetes/kubernetes:master`.

k8s-ci-robot Merge pull request kubernetes#135458 from Argh4k/master 3a477f2 · 33 minutes ago 135,232 Commits

.github	add kind-dependency label to PULL_REQUEST_TEMPLAT...	last week
CHANGELOG	CHANGELOG: Update directory for v1.35.0 release	last month
LICENSES	Update OpenTelemetry dependencies to latest versions	last week
api	Merge pull request kubernetes#131068 from soltysh/sam...	3 hours ago
build	update coredns to 1.14.1	yesterday
cluster	update coredns to 1.14.1	yesterday
cmd	update coredns to 1.14.1	yesterday
docs	Make root approval non-recursive	4 years ago
hack	Merge pull request kubernetes#131068 from soltysh/sam...	3 hours ago
logo	logo: better alignment of layers	4 years ago

About

Production-Grade Container Scheduling and Management

kubernetes.io

- Readme
- Apache-2.0 license
- Code of conduct
- Contributing
- Security policy
- Activity
- 0 stars
- 0 watching
- 0 forks

Releases

No releases published

Create a new release

Packages

Cloned the repo using **“Git clone”** to the local for fixing bugs in the open-source project

```
abhiram@ubuntuvm:~/git$ git clone git@github.com:Abhi4004/kubernetes.git
Cloning into 'kubernetes'...
remote: Enumerating objects: 1563331, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 1563331 (delta 3), reused 5 (delta 0), pack-reused 1563318
(from 2)
Receiving objects: 100% (1563331/1563331), 1.09 GiB | 5.25 MiB/s, done.
Resolving deltas: 100% (1135879/1135879), done.
abhiram@ubuntuvm:~/git$
abhiram@ubuntuvm:~/git$ cd kubernetes/
abhiram@ubuntuvm:~/git/kubernetes$ ls -lrt
total 192
lrwxrwxrwx  1 abhiram abhiram    19 Jan 30 10:20 CHANGELOG.md ->
CHANGELOG/README.md
-rw-rw-r--  1 abhiram abhiram 11358 Jan 30 10:20 LICENSE
-rw-rw-r--  1 abhiram abhiram   525 Jan 30 10:20 CONTRIBUTING.md
drwxrwxr-x  2 abhiram abhiram  4096 Jan 30 10:20 CHANGELOG
drwxrwxr-x  4 abhiram abhiram  4096 Jan 30 10:20 LICENSES
-rw-rw-r--  1 abhiram abhiram  1077 Jan 30 10:20 SUPPORT.md
-rw-rw-r--  1 abhiram abhiram   665 Jan 30 10:20 SECURITY_CONTACTS
```

Task 3: Branching strategy:

- Implemented efficient branch creation by using a shell script to generate three separate Git branches simultaneously.
- This approach facilitates seamless bug-fixing efforts.
- Branch naming conventions were established based on the specific fix to be addressed, ensuring clarity and preventing communication ambiguities within the team.
- Pushed new branches to the remote repository, establishing the upstream link using the `git push -u` command.

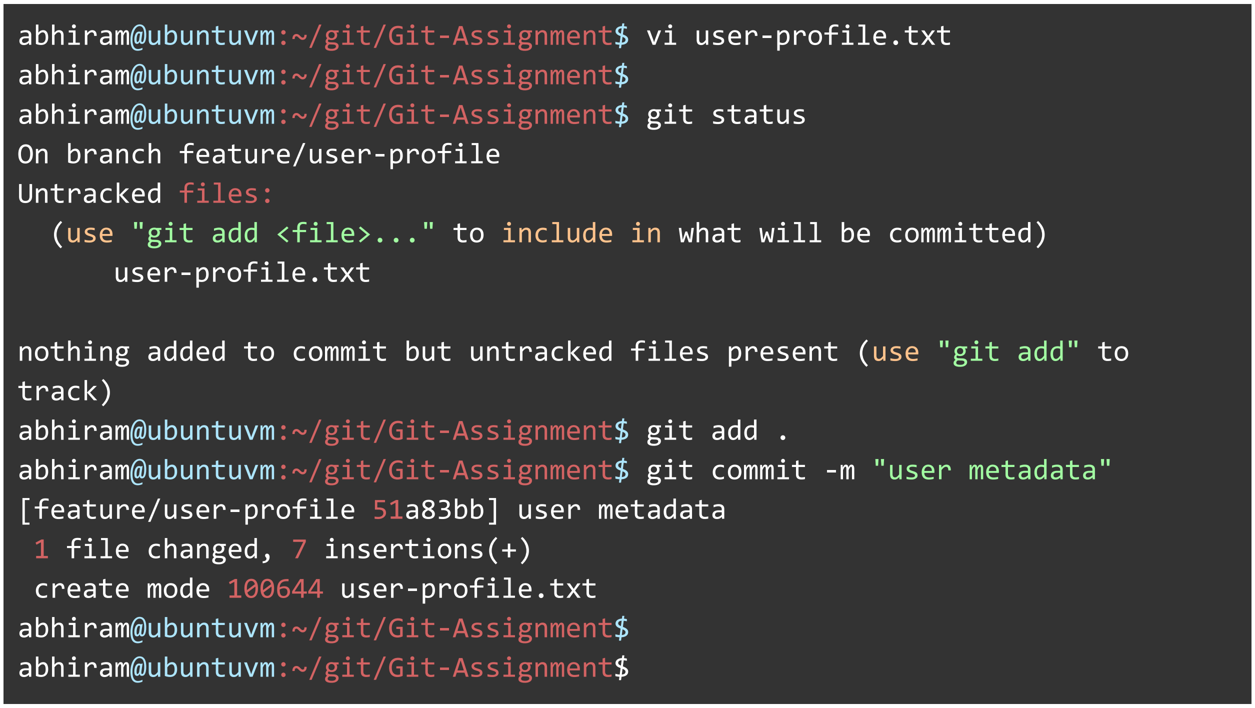
```
abhiram@ubuntuvm:~/git/Git-Assignment$ git branch
* main
abhiram@ubuntuvm:~/git/Git-Assignment$
abhiram@ubuntuvm:~/git/Git-Assignment$ git pull origin main
From github.com:Abhi4004/Git-Assignment
 * branch            main       -> FETCH_HEAD
Already up to date.
abhiram@ubuntuvm:~/git/Git-Assignment$
abhiram@ubuntuvm:~/git/Git-Assignment$ for b in feature/login-page
feature/user-profile feature/payment-integration; do git branch "$b"; done

abhiram@ubuntuvm:~/git/Git-Assignment$
abhiram@ubuntuvm:~/git/Git-Assignment$ git branch
  feature/login-page
  feature/payment-integration
  feature/user-profile
* main
abhiram@ubuntuvm:~/git/Git-Assignment$ for b in feature/login-page
feature/user-profile feature/payment-integration; do git push origin "$b";
done
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature/login-page' on GitHub by
visiting:
remote:
https://github.com/Abhi4004/Git-Assignment/pull/new/feature/login-page
remote:
To github.com:Abhi4004/Git-Assignment.git
 * [new branch]      feature/login-page -> feature/login-page
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature/user-profile' on GitHub by
```

```
visiting:
remote:
https://github.com/Abhi4004/Git-Assignment/pull/new/feature/user-profile
remote:
To github.com:Abhi4004/Git-Assignment.git
 * [new branch]      feature/user-profile -> feature/user-profile
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature/payment-integration' on GitHub
by visiting:
remote:
https://github.com/Abhi4004/Git-Assignment/pull/new/feature/payment-integra
tion
remote:
To github.com:Abhi4004/Git-Assignment.git
 * [new branch]      feature/payment-integration ->
feature/payment-integration
abhiram@ubuntuvm:~/git/Git-Assignment$ git branch
  feature/login-page
  feature/payment-integration
  feature/user-profile
* main
```

Task 4: collaboration and pull requests.

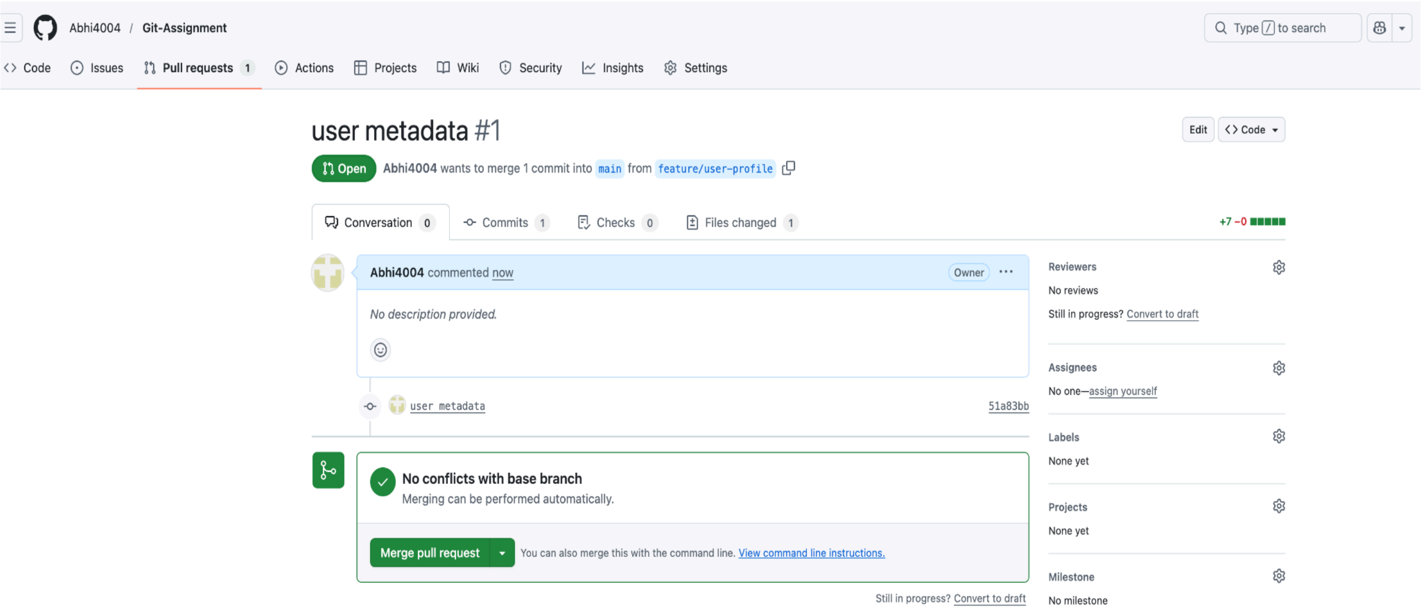
I checked out the `feature/user-profile` branch, created the user's metadata, and committed the file.
The command `git push --set-upstream origin feature/user-profile` was used to establish the upstream tracking branch for the local branch.

```
abhiram@ubuntuvm:~/git/Git-Assignment$ git branch
  feature/login-page
  feature/payment-integration
  feature/user-profile
* main
abhiram@ubuntuvm:~/git/Git-Assignment$ git checkout feature/user-profile
Switched to branch 'feature/user-profile'
abhiram@ubuntuvm:~/git/Git-Assignment$
```



```
abhiram@ubuntuvm:~/git/Git-Assignment$ vi user-profile.txt
abhiram@ubuntuvm:~/git/Git-Assignment$
abhiram@ubuntuvm:~/git/Git-Assignment$ git status
On branch feature/user-profile
Untracked files:
  (use "git add <file>..." to include in what will be committed)
	user-profile.txt

nothing added to commit but untracked files present (use "git add" to
track)
abhiram@ubuntuvm:~/git/Git-Assignment$ git add .
abhiram@ubuntuvm:~/git/Git-Assignment$ git commit -m "user metadata"
[feature/user-profile 51a83bb] user metadata
 1 file changed, 7 insertions(+)
 create mode 100644 user-profile.txt
abhiram@ubuntuvm:~/git/Git-Assignment$
abhiram@ubuntuvm:~/git/Git-Assignment$
```

The Pull Request is now available for the team's review.



user metadata #1

Open Abhi4004 wants to merge 1 commit into main from feature/user-profile

Abhi4004 commented now

No description provided.

user metadata 51a83bb

No conflicts with base branch

Merging can be performed automatically.

You can also merge this with the command line. View command line instructions.

Still in progress? Convert to draft

Reviewers

No reviews

Still in progress? Convert to draft

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

After creation, the Pull Request (PR) undergoes a review process to incorporate feedback. It is then closed upon successful validation.
https://github.com/Abhi4004/Git-Assignment/pull/1

Task 5: Handling Merge conflicts

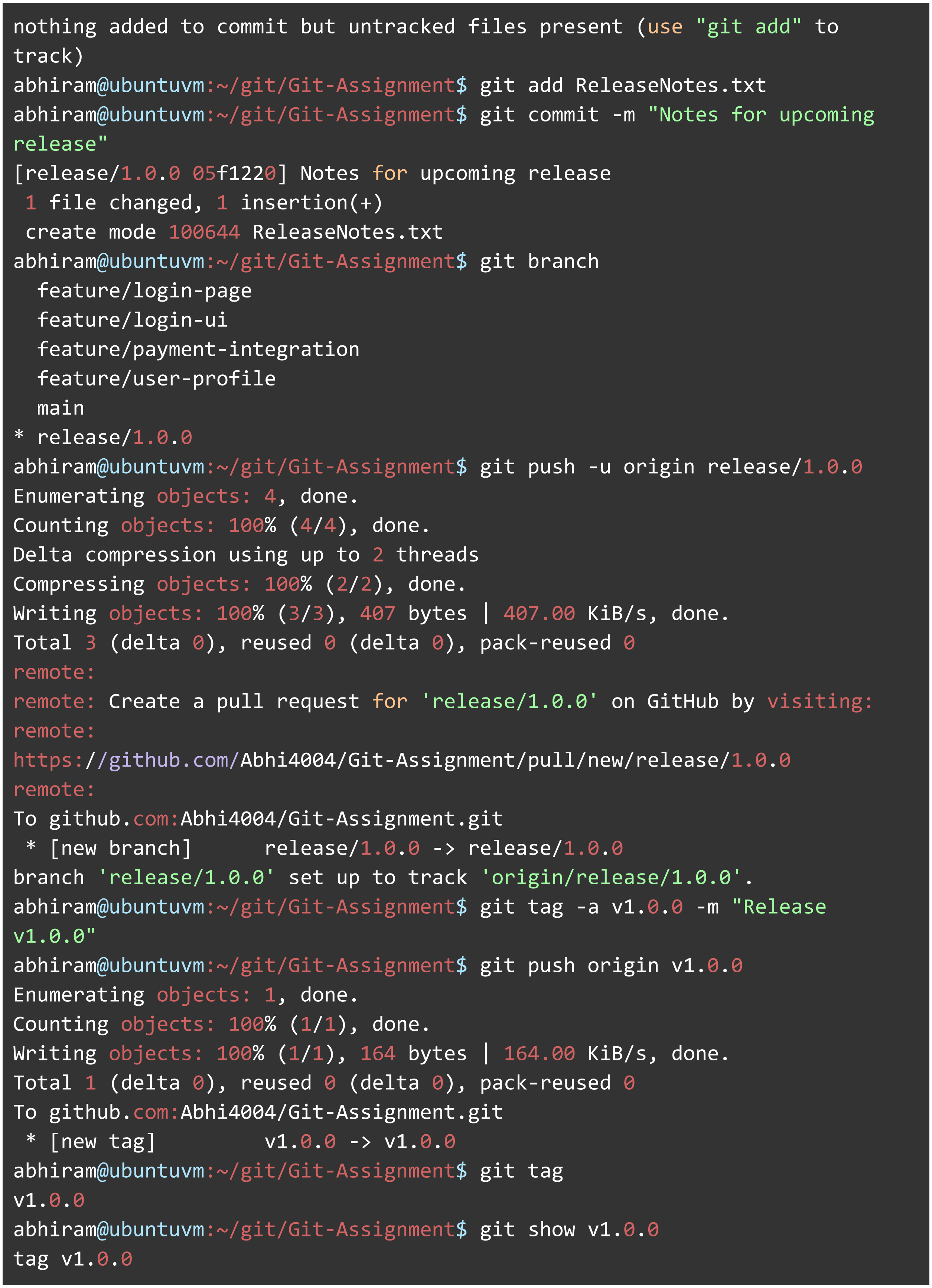
While I was working on a feature branch, another teammate made changes directly in the `main` branch. Both changes were done in the **same file and same lines**, which created a merge conflict when I attempted to merge my feature branch into `main`.

```
git status
git diff

# open the conflicted file and fix manually
vi app.txt

# after fixing, mark as resolved
git add app.txt

# complete the merge with a commit
git commit -m "Resolved merge conflict"
```

Task 6: Creating a release and tag

- I established a stable version of the project by creating a release.
- This process involved using **Git tagging** to label the codebase with a specific version number, such as v1.0.0.
- Tags are crucial for easily identifying designated release points, which aids the team in deploying a particular version or rolling back if necessary.
- After local creation, I ensured the tag was available to the entire team by pushing it to the remote repository.

```
abhiram@ubuntuvm:~/git/Git-Assignment$ git pull origin
Already up to date.
abhiram@ubuntuvm:~/git/Git-Assignment$ git checkout -b release/1.0.0
Switched to a new branch 'release/1.0.0'
abhiram@ubuntuvm:~/git/Git-Assignment$ vi ReleaseNotes.txt
abhiram@ubuntuvm:~/git/Git-Assignment$
abhiram@ubuntuvm:~/git/Git-Assignment$ git status
On branch release/1.0.0
Untracked files:
  (use "git add <file>..." to include in what will be committed)
      ReleaseNotes.txt
```



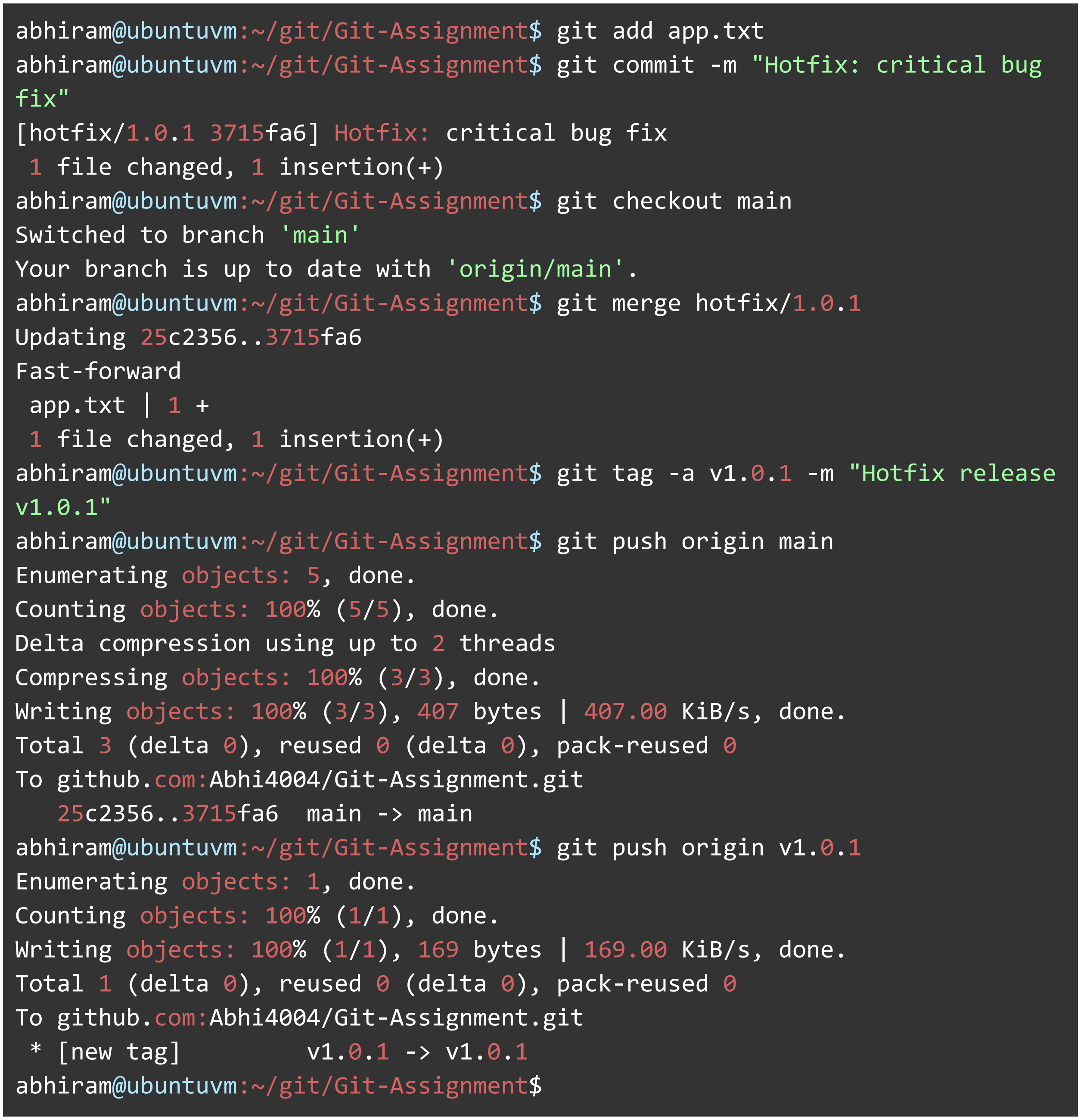
```
nothing added to commit but untracked files present (use "git add" to
track)
abhiram@ubuntuvm:~/git/Git-Assignment$ git add ReleaseNotes.txt
abhiram@ubuntuvm:~/git/Git-Assignment$ git commit -m "Notes for upcoming
release"
[release/1.0.0 05f1220] Notes for upcoming release
 1 file changed, 1 insertion(+)
 create mode 100644 ReleaseNotes.txt
abhiram@ubuntuvm:~/git/Git-Assignment$ git branch
  feature/login-page
  feature/login-ui
  feature/payment-integration
  feature/user-profile
  main
* release/1.0.0
abhiram@ubuntuvm:~/git/Git-Assignment$ git push -u origin release/1.0.0
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 407 bytes | 407.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'release/1.0.0' on GitHub by visiting:
remote:
https://github.com/Abhi4004/Git-Assignment/pull/new/release/1.0.0
remote:
To github.com:Abhi4004/Git-Assignment.git
 * [new branch]      release/1.0.0 -> release/1.0.0
branch 'release/1.0.0' set up to track 'origin/release/1.0.0'.
abhiram@ubuntuvm:~/git/Git-Assignment$ git tag -a v1.0.0 -m "Release
v1.0.0"
abhiram@ubuntuvm:~/git/Git-Assignment$ git push origin v1.0.0
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 164 bytes | 164.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Abhi4004/Git-Assignment.git
 * [new tag]         v1.0.0 -> v1.0.0
abhiram@ubuntuvm:~/git/Git-Assignment$ git tag
v1.0.0
abhiram@ubuntuvm:~/git/Git-Assignment$ git show v1.0.0
tag v1.0.0
```

```
Tagger: Abhiram <268.reddabhiram@gmail.com>
Date:   Fri Jan 30 12:17:30 2026 +0000

Release v1.0.0

commit 05f12209ab2a531c2f01c8783906a746060df425 (HEAD -> release/1.0.0,
tag: v1.0.0, origin/release/1.0.0)
Author: Abhiram <268.reddabhiram@gmail.com>
Date:   Fri Jan 30 12:16:20 2026 +0000
    Notes for upcoming release

diff --git a/ReleaseNotes.txt b/ReleaseNotes.txt
new file mode 100644
index 0000000..f7e7c71
--- /dev/null
+++ b/ReleaseNotes.txt
@@ -0,0 +1 @@
+This is notes for upcoming relase of Version 1.0
abhiram@ubuntuvm:~/git/Git-Assignment$
```

Task 7: Hotfix in Production

- After releasing version **v1.0.0**, a **critical bug** was identified in the production environment.
- Since the bug impacted production, an immediate fix was required without waiting for the next planned release.
- The team created a **hotfix branch** (hotfix/1.0.1) from the released version (main branch or release tag).
- The bug fix was implemented and committed in the hotfix branch.
  The hotfix branch was then merged back into the **main branch** to ensure production contains the fix.
- A new tag **v1.0.1** was created to mark the patched release version.
- Finally, the updated main branch and new tag were pushed to the remote repository for deployment.

```
abhiram@ubuntuvm:~/git/Git-Assignment$ git checkout -b hotfix/1.0.1
Switched to a new branch 'hotfix/1.0.1'
abhiram@ubuntuvm:~/git/Git-Assignment$ vi app.txt
abhiram@ubuntuvm:~/git/Git-Assignment$
```



```
abhiram@ubuntuvm:~/git/Git-Assignment$ git add app.txt
abhiram@ubuntuvm:~/git/Git-Assignment$ git commit -m "Hotfix: critical bug
fix"
[hotfix/1.0.1 3715fa6] Hotfix: critical bug fix
 1 file changed, 1 insertion(+)
abhiram@ubuntuvm:~/git/Git-Assignment$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
abhiram@ubuntuvm:~/git/Git-Assignment$ git merge hotfix/1.0.1
Updating 25c2356..3715fa6
Fast-forward
 app.txt | 1 +
 1 file changed, 1 insertion(+)
abhiram@ubuntuvm:~/git/Git-Assignment$ git tag -a v1.0.1 -m "Hotfix release
v1.0.1"
abhiram@ubuntuvm:~/git/Git-Assignment$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 407 bytes | 407.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Abhi4004/Git-Assignment.git
   25c2356..3715fa6  main -> main
abhiram@ubuntuvm:~/git/Git-Assignment$ git push origin v1.0.1
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 169 bytes | 169.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Abhi4004/Git-Assignment.git
 * [new tag]         v1.0.1 -> v1.0.1
abhiram@ubuntuvm:~/git/Git-Assignment$
```

Task 8: Stashing changes

An **urgent issue** was reported requiring immediate action, interrupting my current feature development.

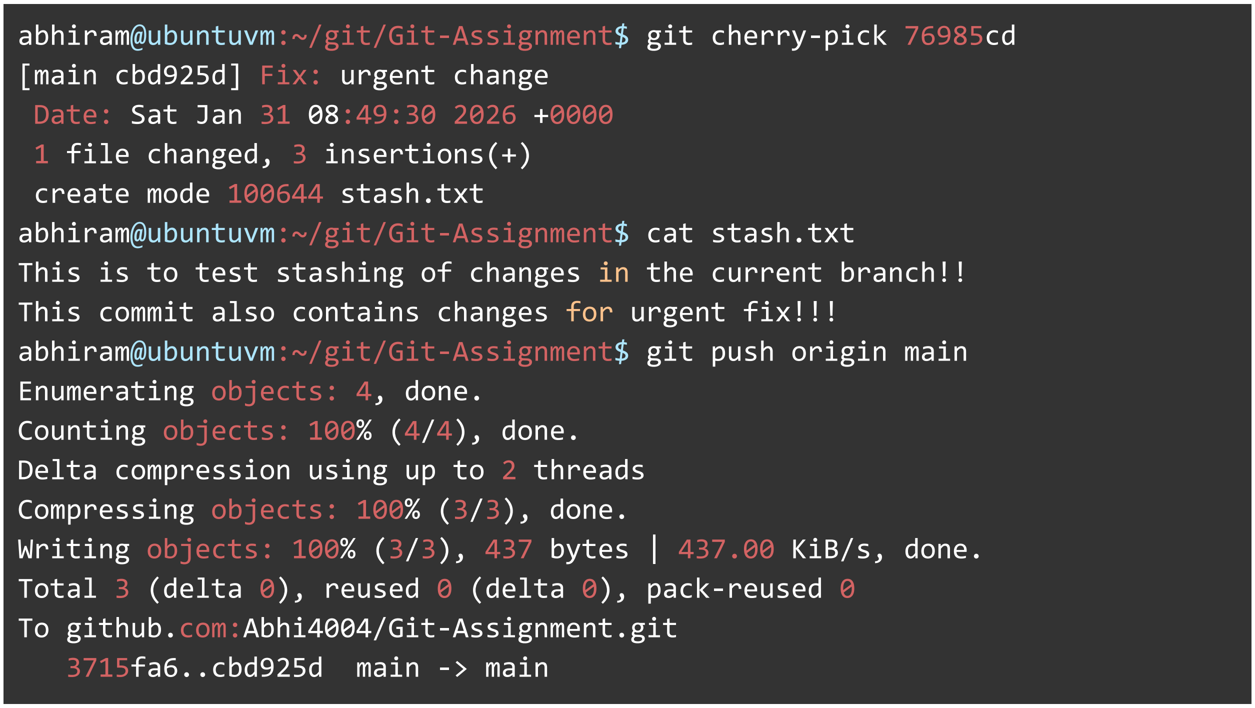
To address this without losing my incomplete work, I utilized **Git stash** to temporarily save my uncommitted changes. This allowed me to safely switch to another branch (e.g., `main`) to handle the urgent matter.

```
abhiram@ubuntuvm:~/git/Git-Assignment$ git status
On branch feature/login-ui
Your branch is up to date with 'origin/feature/login-ui'.

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

nothing added to commit but untracked files present (use "git add" to
track)
abhiram@ubuntuvm:~/git/Git-Assignment$ git add stash.txt
abhiram@ubuntuvm:~/git/Git-Assignment$ git stash push -m "WIP: feature
work"
Saved working directory and index state On feature/login-ui: WIP: feature
work
abhiram@ubuntuvm:~/git/Git-Assignment$
abhiram@ubuntuvm:~/git/Git-Assignment$ git status
On branch feature/login-ui
Your branch is up to date with 'origin/feature/login-ui'.

nothing to commit, working tree clean
abhiram@ubuntuvm:~/git/Git-Assignment$ git stash list
stash@{0}: On feature/login-ui: WIP: feature work
abhiram@ubuntuvm:~/git/Git-Assignment$
abhiram@ubuntuvm:~/git/Git-Assignment$
abhiram@ubuntuvm:~/git/Git-Assignment$
abhiram@ubuntuvm:~/git/Git-Assignment$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
abhiram@ubuntuvm:~/git/Git-Assignment$ git branch
  feature/login-page
  feature/login-ui
  feature/payment-integration
  feature/user-profile
  hotfix/1.0.1
* main
  release/1.0.0
```

Task 9: cherry-picking a commit

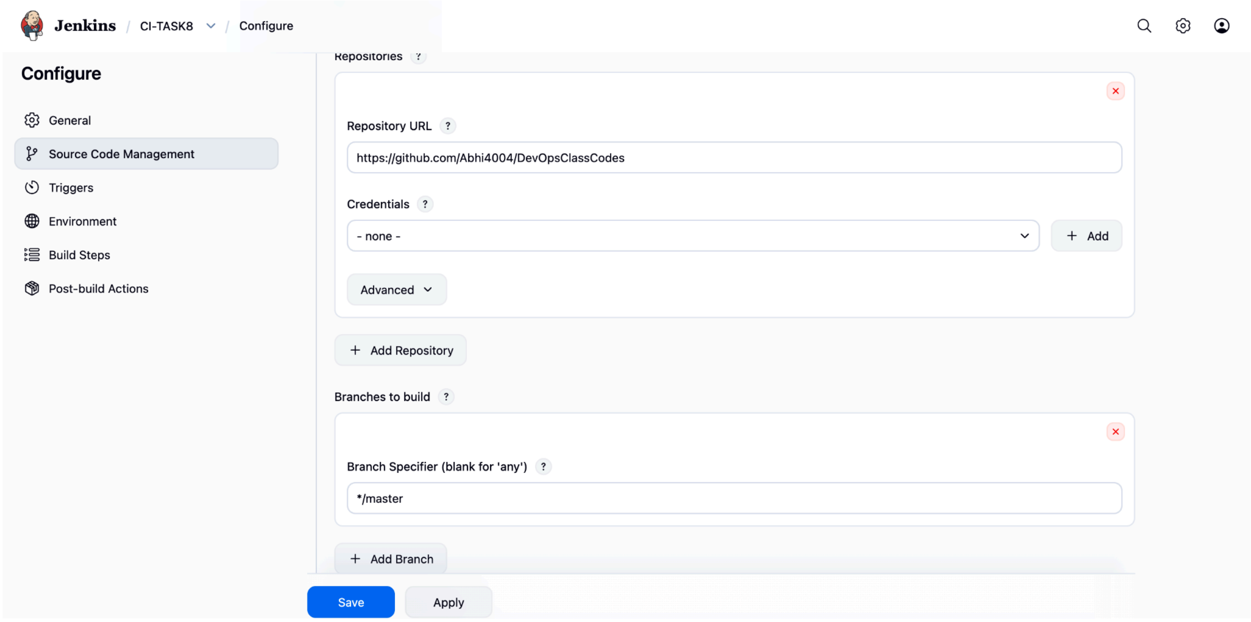
I made an **urgent fix** in the `feature/login-ui` branch and committed the changes (`Fix: urgent change`).
I verified the commit hash using `git log --oneline` and noted the commit ID `76985cd`.
I switched to the `main` branch and pulled the latest updates to ensure `main` was up to date.
I used **git cherry-pick** to apply only the urgent fix commit (`76985cd`) from the feature branch into `main` without merging the entire branch.
After cherry-picking, the commit was successfully applied in `main` as a new commit (`cbd925d`).
I validated the changes using `cat stash.txt` to confirm the urgent fix content was present in `main`.
Finally, I pushed the updated `main` branch to the remote repository using `git push origin main`.

```
abhiram@ubuntuvm:~/git/Git-Assignment$ git commit -m "Fix: urgent change"
[feature/login-ui 76985cd] Fix: urgent change
 1 file changed, 3 insertions(+)
 create mode 100644 stash.txt
abhiram@ubuntuvm:~/git/Git-Assignment$ git log --oneline -5
76985cd (HEAD -> feature/login-ui) Fix: urgent change
71dd801 (origin/feature/login-ui) Updated login page UI
f09924e Merge pull request #1 from Abhi4004/feature/user-profile
51a83bb (origin/feature/user-profile, feature/user-profile) user metadata
a3ef1a6 (origin/feature/payment-integration, origin/feature/login-page,
feature/payment-integration, feature/login-page) Initial commit

abhiram@ubuntuvm:~/git/Git-Assignment$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
abhiram@ubuntuvm:~/git/Git-Assignment$ git pull origin main
From github.com:Abhi4004/Git-Assignment
 * branch            main       -> FETCH_HEAD
Already up to date.
```



```
abhiram@ubuntuvm:~/git/Git-Assignment$ git cherry-pick 76985cd
[main cbd925d] Fix: urgent change
 Date: Sat Jan 31 08:49:30 2026 +0000
 1 file changed, 3 insertions(+)
 create mode 100644 stash.txt
abhiram@ubuntuvm:~/git/Git-Assignment$ cat stash.txt
This is to test stashing of changes in the current branch!!
This commit also contains changes for urgent fix!!!
abhiram@ubuntuvm:~/git/Git-Assignment$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 437 bytes | 437.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Abhi4004/Git-Assignment.git
   3715fa6..cbd925d  main -> main
```

Task 10: Integrating Continuous Integration:

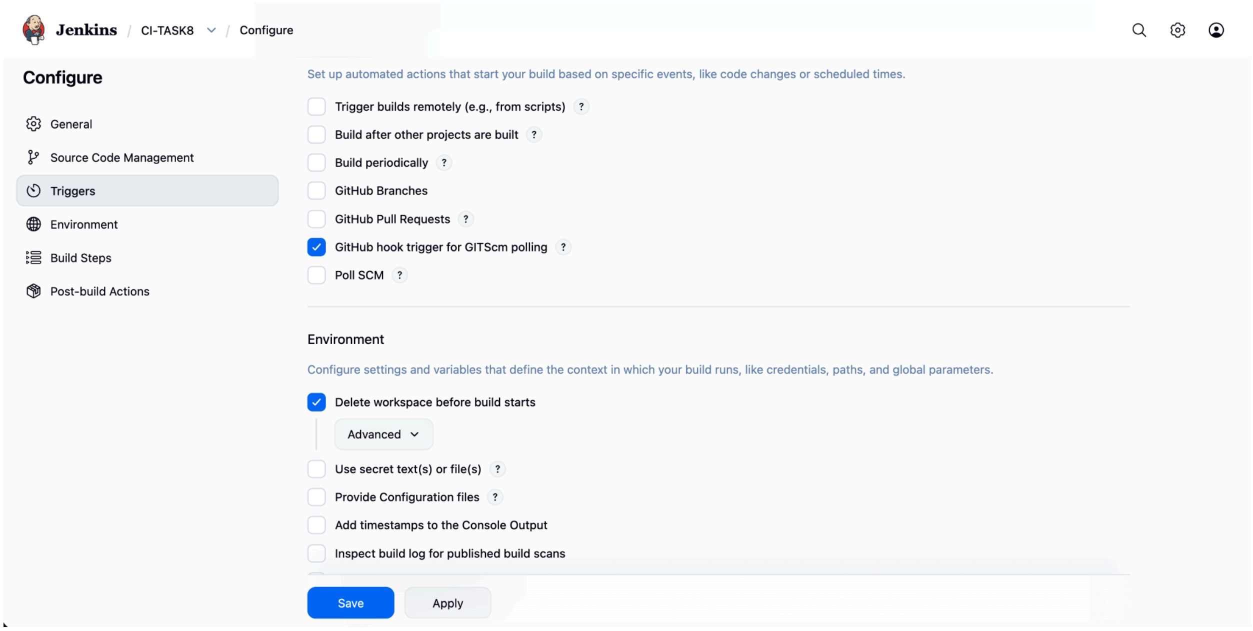


Repository URL: https://github.com/Abhi4004/DevOpsClassCodes

Credentials: - none -

Branch Specifier (blank for 'any'): */master

A new Jenkins Freestyle project was set up, utilizing a public Github repository for the SCM. No credentials were required for this public repository.



Set up automated actions that start your build based on specific events, like code changes or scheduled times.

☐ Trigger builds remotely (e.g., from scripts) ?
☐ Build after other projects are built ?
☐ Build periodically ?
☐ GitHub Branches
☐ GitHub Pull Requests ?
☑ GitHub hook trigger for GITScm polling ?
☐ Poll SCM ?

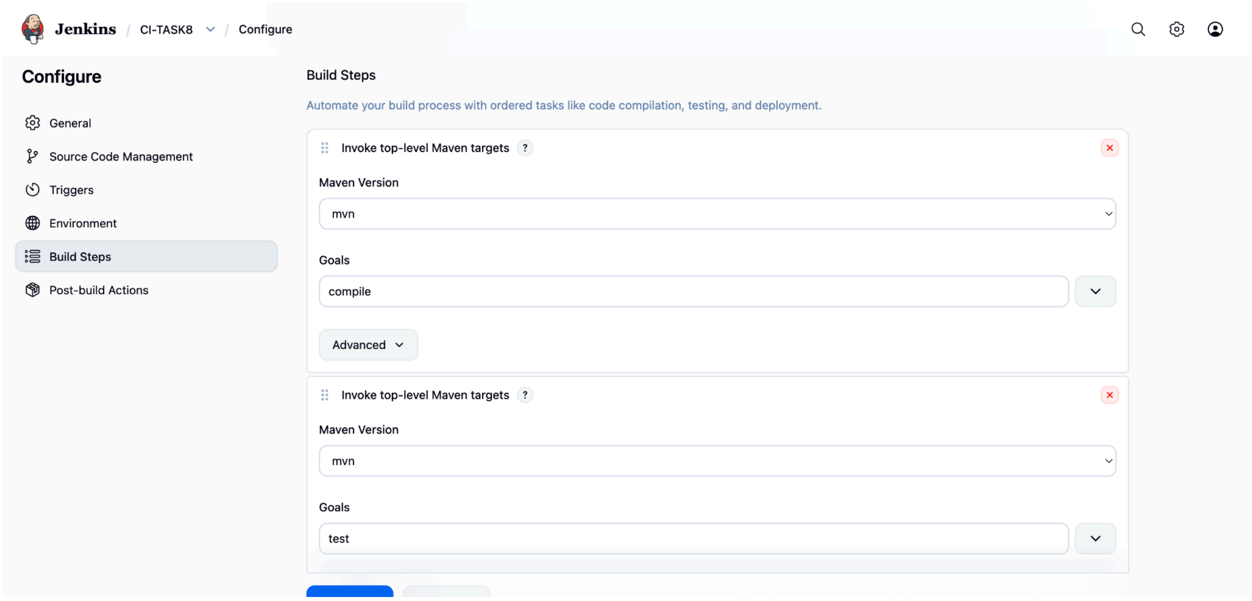
Environment

Configure settings and variables that define the context in which your build runs, like credentials, paths, and global parameters.

☑ Delete workspace before build starts
Advanced
☐ Use secret text(s) or file(s) ?
☐ Provide Configuration files ?
☐ Add timestamps to the Console Output
☐ Inspect build log for published build scans

To ensure the pipeline is triggered upon code changes, the **GitHub webhook trigger** was selected.

Additionally, the **Delete workspace** option was chosen to maintain a clean environment for subsequent builds.



Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Invoke top-level Maven targets ?

Maven Version: mvn

Goals: compile

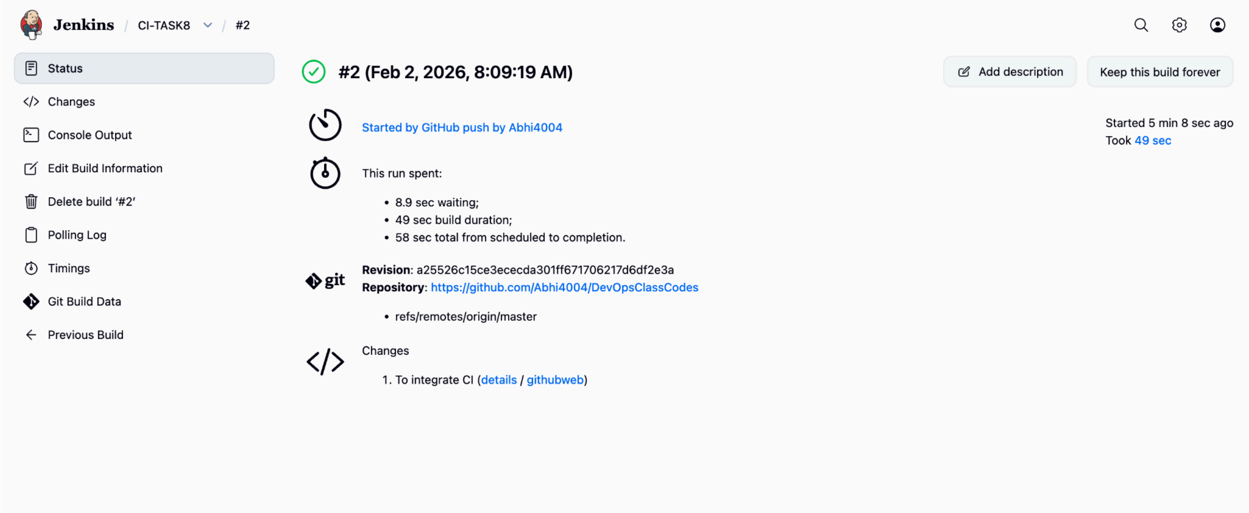
Advanced

Invoke top-level Maven targets ?

Maven Version: mvn

Goals: test

The process was enhanced by adding three steps—for compiling, testing, and packaging—which are responsible for code checks, testing, and ultimately packaging the code.



#2 (Feb 2, 2026, 8:09:19 AM)

Started by GitHub push by Abhi4004

Started 5 min 8 sec ago
Took 49 sec

This run spent:

- 8.9 sec waiting;
- 49 sec build duration;
- 58 sec total from scheduled to completion.

Revision: a25526c15ce3ececda301ff671706217d6df2e3a
Repository: https://github.com/Abhi4004/DevOpsClassCodes

- refs/remotes/origin/master

Changes

1. To integrate CI (details / githubweb)

The trigger was successfully verified: the pipeline executed upon the commit titled "To Integrate CI."

Task 11 (Rebase vs Merge):

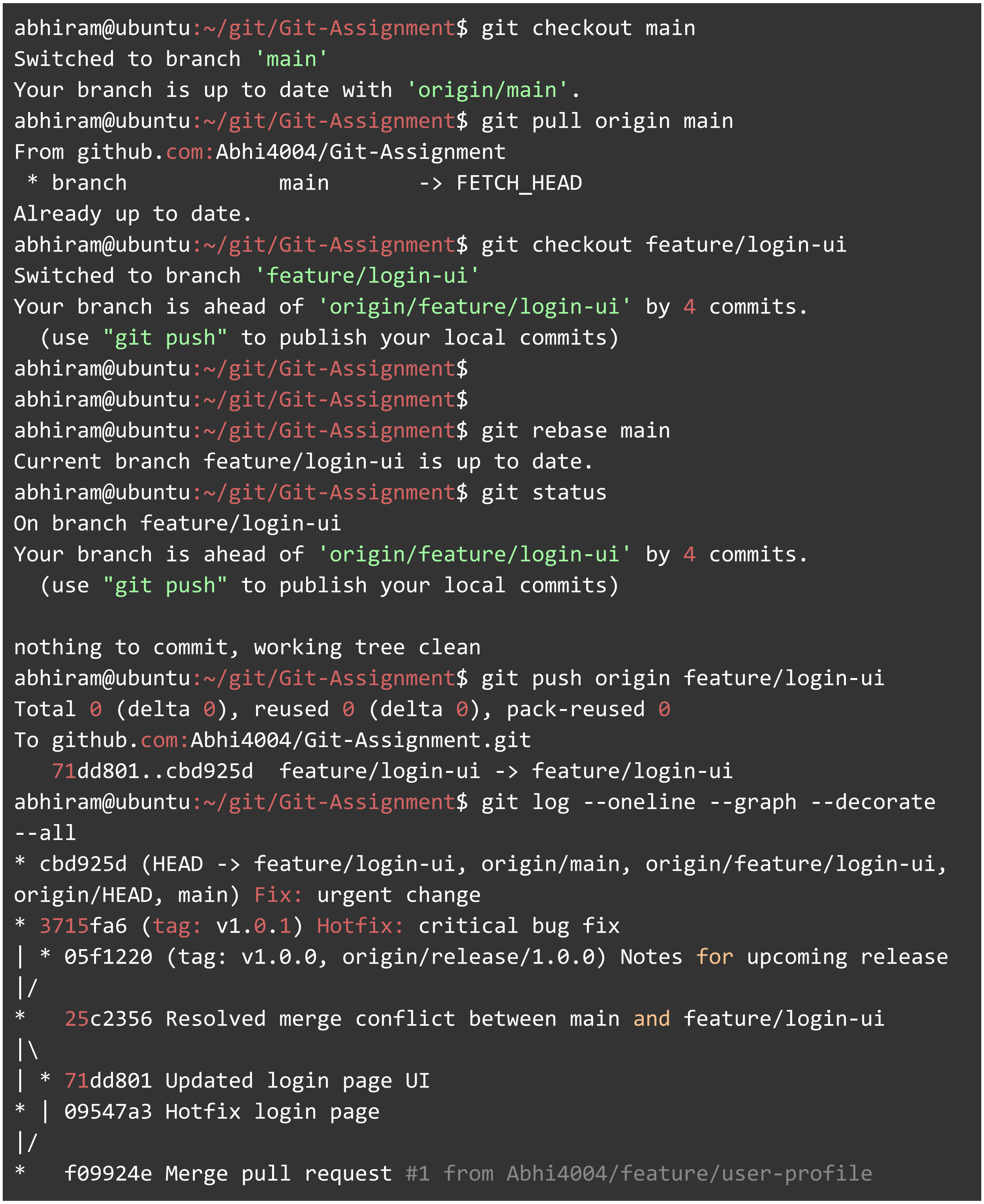
- Our team prefers **git rebase** to keep a **clean, linear commit history** instead of using merge commits.
- `git rebase main` was executed on `feature/login-ui` branch to bring the latest changes from `main`.
- Output showed: **"Current branch feature/login-ui is up to date."**
  - This means the feature branch already contained the latest commits from `main`, so no rebase actions were required.
- `git status` confirmed the branch had **4 local commits ahead of origin**, meaning commits existed only in the local repository.

Those commits were pushed to GitHub using:

git push origin feature/login-ui

-
- `git log --oneline --graph --decorate --all` was used to verify commit history.
- The log displayed an existing **merge commit**:
  - `Resolved merge conflict between main and feature/login-ui`
  - This proves merge creates a branched history with a merge commit.
- After rebase and push, both `main` and `feature/login-ui` pointed to the same latest commit (`cbd925d`), showing the branch was synchronized.

- Rebase is preferred because it avoids extra merge commits and makes history easier to review and track.



```
abhiram@ubuntu:~/git/Git-Assignment$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
abhiram@ubuntu:~/git/Git-Assignment$ git pull origin main
From github.com:Abhi4004/Git-Assignment
 * branch            main       -> FETCH_HEAD
Already up to date.
abhiram@ubuntu:~/git/Git-Assignment$ git checkout feature/login-ui
Switched to branch 'feature/login-ui'
Your branch is ahead of 'origin/feature/login-ui' by 4 commits.
  (use "git push" to publish your local commits)
abhiram@ubuntu:~/git/Git-Assignment$
abhiram@ubuntu:~/git/Git-Assignment$
abhiram@ubuntu:~/git/Git-Assignment$ git rebase main
Current branch feature/login-ui is up to date.
abhiram@ubuntu:~/git/Git-Assignment$ git status
On branch feature/login-ui
Your branch is ahead of 'origin/feature/login-ui' by 4 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
abhiram@ubuntu:~/git/Git-Assignment$ git push origin feature/login-ui
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Abhi4004/Git-Assignment.git
   71dd801..cbd925d  feature/login-ui -> feature/login-ui
abhiram@ubuntu:~/git/Git-Assignment$ git log --oneline --graph --decorate
--all
* cbd925d (HEAD -> feature/login-ui, origin/main, origin/feature/login-ui,
origin/HEAD, main) Fix: urgent change
* 3715fa6 (tag: v1.0.1) Hotfix: critical bug fix
| * 05f1220 (tag: v1.0.0, origin/release/1.0.0) Notes for upcoming release
|/
*   25c2356 Resolved merge conflict between main and feature/login-ui
|\
| * 71dd801 Updated login page UI
* | 09547a3 Hotfix login page
|/
*   f09924e Merge pull request #1 from Abhi4004/feature/user-profile
```

```
|\
| * 51a83bb (origin/feature/user-profile) user metadata
|/
* a3ef1a6 (origin/feature/payment-integration, origin/feature/login-page)
Initial commit
abhiram@ubuntu:~/git/Git-Assignment$
```

Task 12 Squashing Commits:

The team wanted a **clean commit history**, so we squashed multiple commits in feature branch into **one commit** before merging to `main`.

Used interactive rebase to squash last 4 commits:

```
git checkout feature/login-ui
git rebase -i HEAD~4
```

In rebase editor, changed commits to:

```
pick 09547a3 Hotfix login page
squash 71dd801 Updated login page UI
squash 3715fa6 Hotfix: critical bug fix
squash cbd925d Fix: urgent change
```

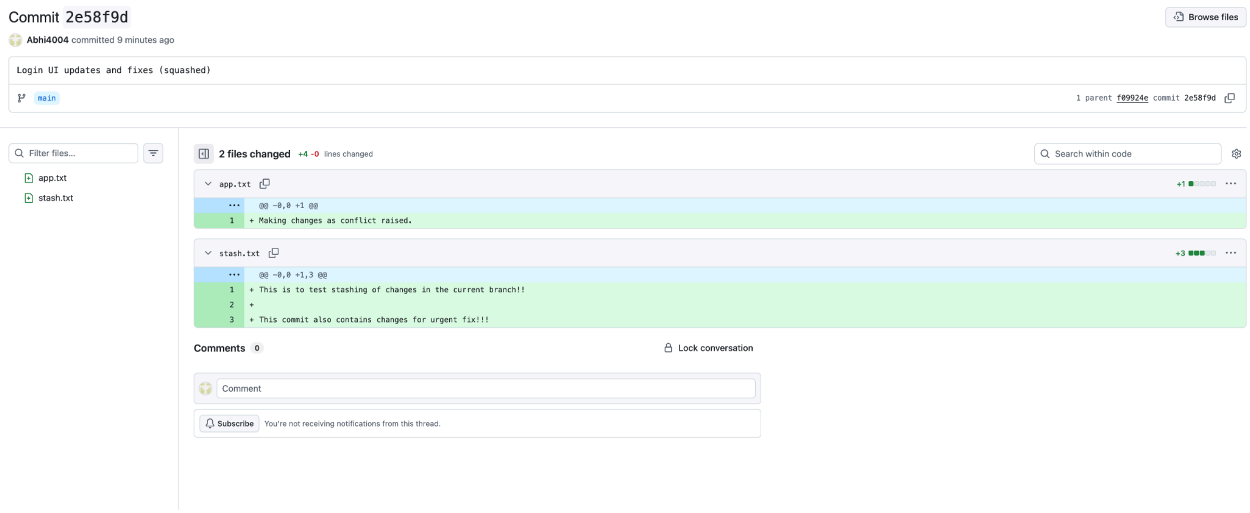
- Provided a single final commit message:
  ✅ `Login UI updates and fixes (squashed)`

Resolved conflict in `app.txt`, continued rebase, then pushed squashed branch:

```
git push origin feature/login-ui --force-with-lease
```

Finally merged feature branch into main and pushed:

```
git checkout main
git merge feature/login-ui
git push origin main
```



Commit 2e58f9d

Abhi4004 committed 9 minutes ago

```
Login UI updates and fixes (squashed)
```

main — 1 parent f09924e commit 2e58f9d

2 files changed +4 -0 lines changed

app.txt +1

```
@@ -0,0 +1 @@
1 + Making changes as conflict raised.
```

stash.txt +3

```
@@ -0,0 +1,3 @@
1 + This is to test stashing of changes in the current branch!!
2 +
3 + This commit also contains changes for urgent fix!!!
```

Comments 0

Task 13 Git Blame for Debugging:

Used `git blame` to identify who introduced the bug line in `app.txt`.

Ran: `git blame app.txt` to get commit ID + author for each line.

Found commits: `3715fa6b` and `2e58f9df` (Author: Abhiram).

Used `git show 2e58f9df` to view full commit details and confirm the change.

This helped trace the issue to the exact commit and developer.

```
abhiram@ubuntu:~/git/Git-Assignment$ git branch
  feature/login-ui
* main
abhiram@ubuntu:~/git/Git-Assignment$ git blame app.txt
3715fa6b (Abhiram 2026-01-30 13:12:40 +0000 1) Hot fix has been added and
notes has been updated !!
2e58f9df (Abhiram 2026-01-30 12:06:32 +0000 2) Making changes as conflict
raised.
abhiram@ubuntu:~/git/Git-Assignment$
abhiram@ubuntu:~/git/Git-Assignment$ git show 2e58f9df
commit 2e58f9df0d405dbe4b1739a5b486df3cd480b858 (origin/feature/login-ui,
feature/login-ui)
Author: Abhiram <268.reddabhiram@gmail.com>
```

```
Date:   Fri Jan 30 12:06:32 2026 +0000

    Login UI updates and fixes (squashed)

diff --git a/app.txt b/app.txt
new file mode 100644
index 0000000..150fd73
--- /dev/null
+++ b/app.txt
@@ -0,0 +1 @@
+Making changes as conflict raised.
diff --git a/stash.txt b/stash.txt
new file mode 100644
index 0000000..6859aef
--- /dev/null
+++ b/stash.txt
@@ -0,0 +1,3 @@
+This is to test stashing of changes in the current branch!!
+
+This commit also contains changes for urgent fix!!!
```