NAME : UDDHAV BAPU GUND | DESIGNATION: JR.AWS DEVELOPER

DevOps TASK 1 :

Git & GitHub Basics

- Initialize a Git repo.
- Push/pull changes to GitHub.
- Create a feature branch → merge using Pull Request.

DevOps Task Report – Git & GitHub Basics

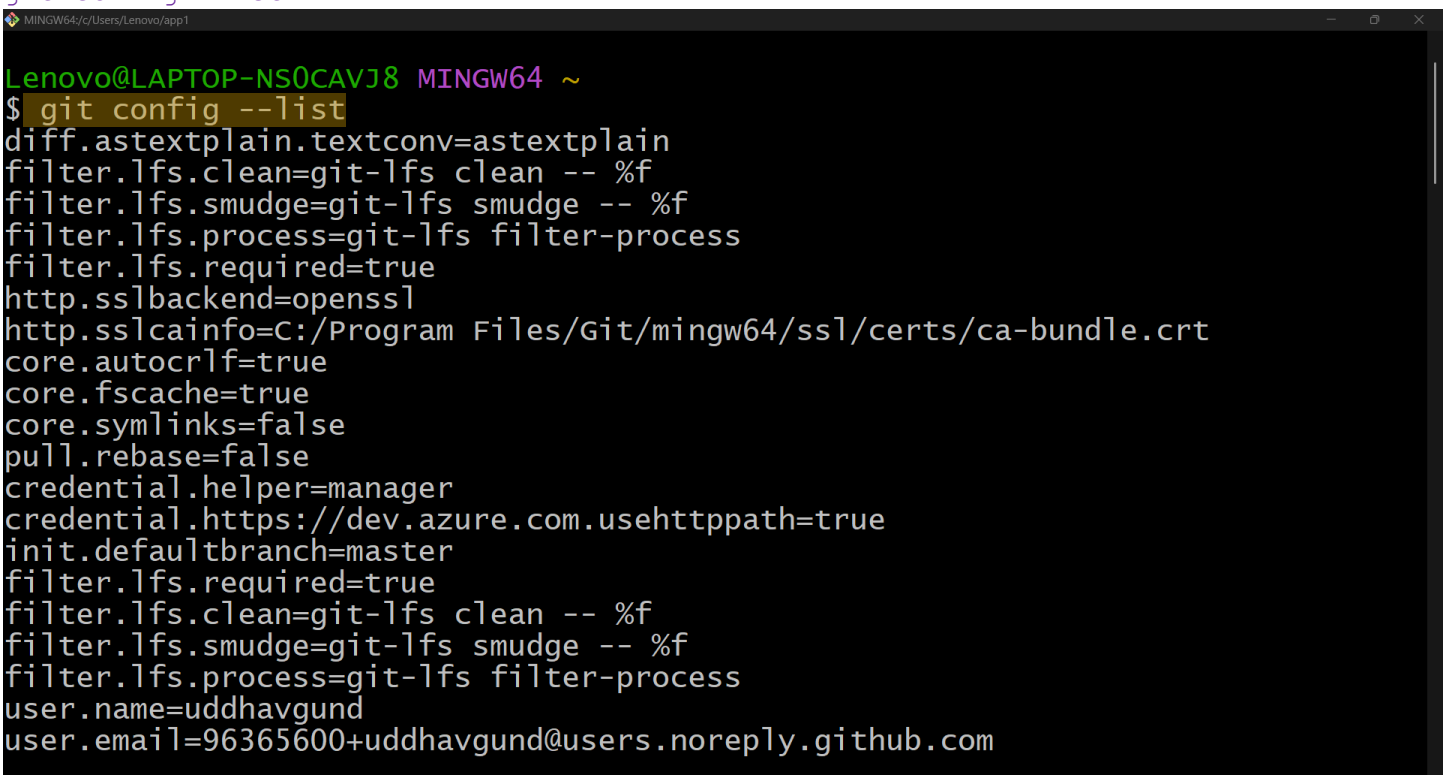
This report documents the step-by-step execution of the Git & GitHub Basics task. The exercise demonstrates repository initialization, remote linking, feature branching, pull requests, and merging workflow.

Step 1 — Configure Git (one-time)

```
git config --global user.name "Your Name"
git config --global user.email your-email@example.com
```

Check:

```
git config -list
```



Step 2: Create a new folder (local project)

Commands :

```
mkdir app2
cd app2
```

Step 3: Initialize Git repo

Commands:

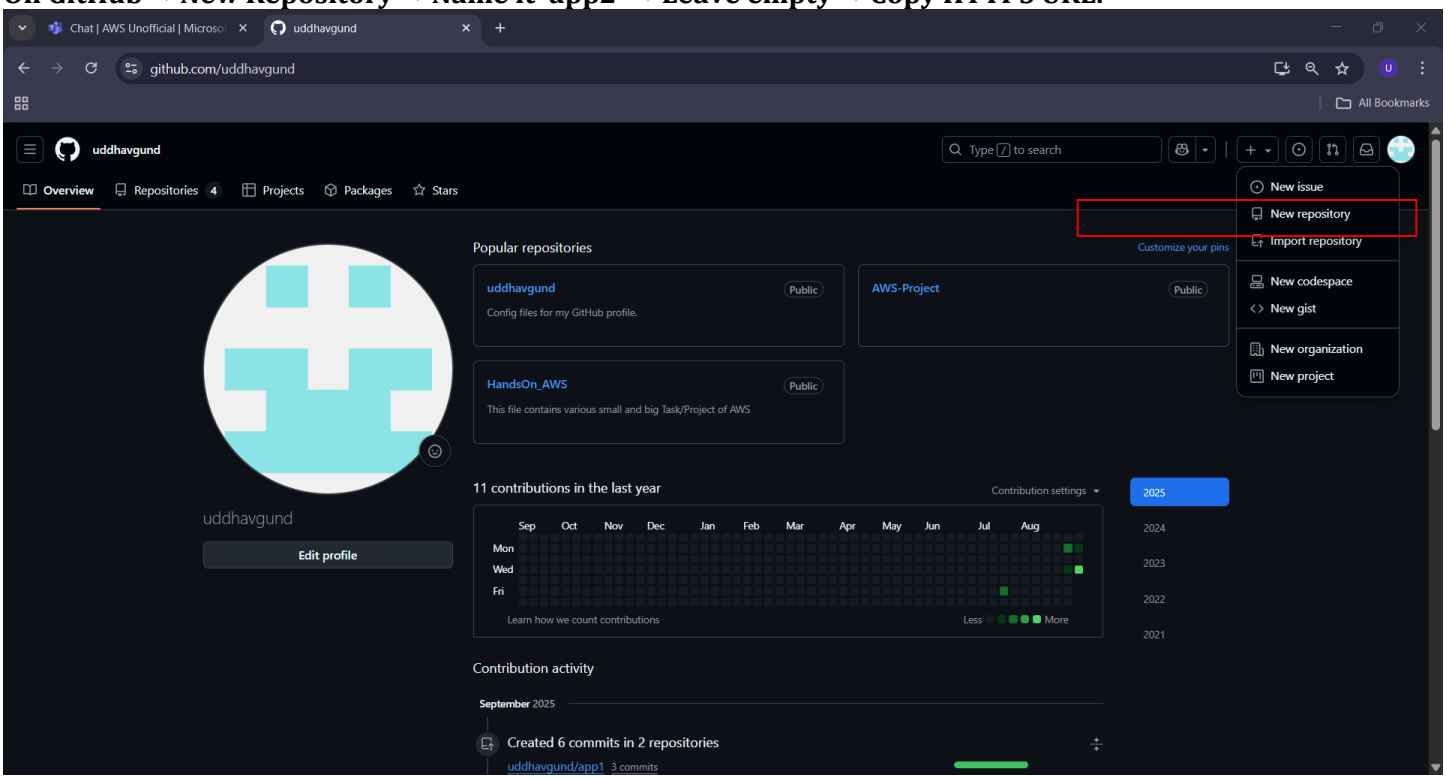
```
git init
echo "# app2 project" > README.md
git add README.md
git commit -m "chore: initial commit for app2"
```

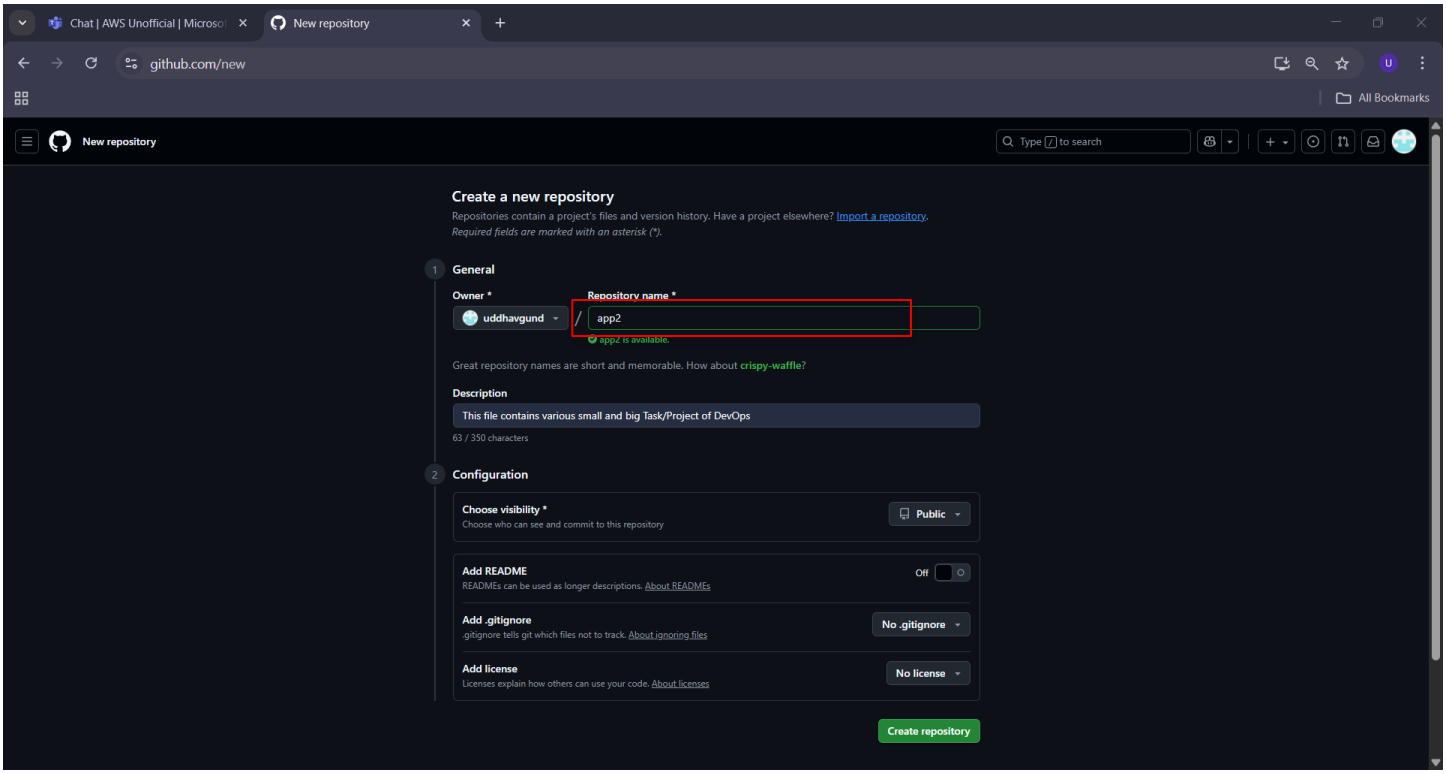
```
Lenovo@LAPTOP-NSOCAVJ8 MINGW64 ~/app2
$ git init
Initialized empty Git repository in C:/Users/Lenovo/app2/.git/

Lenovo@LAPTOP-NSOCAVJ8 MINGW64 ~/app2 (master)
$ echo "#app2 project" > README.md

Lenovo@LAPTOP-NSOCAVJ8 MINGW64 ~/app2 (master)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the nex
t time Git touches it

Lenovo@LAPTOP-NSOCAVJ8 MINGW64 ~/app2 (master)
$ git commit -m "chore: initial commit for app2"
[master (root-commit) 007ffcf] chore: initial commit for app2
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

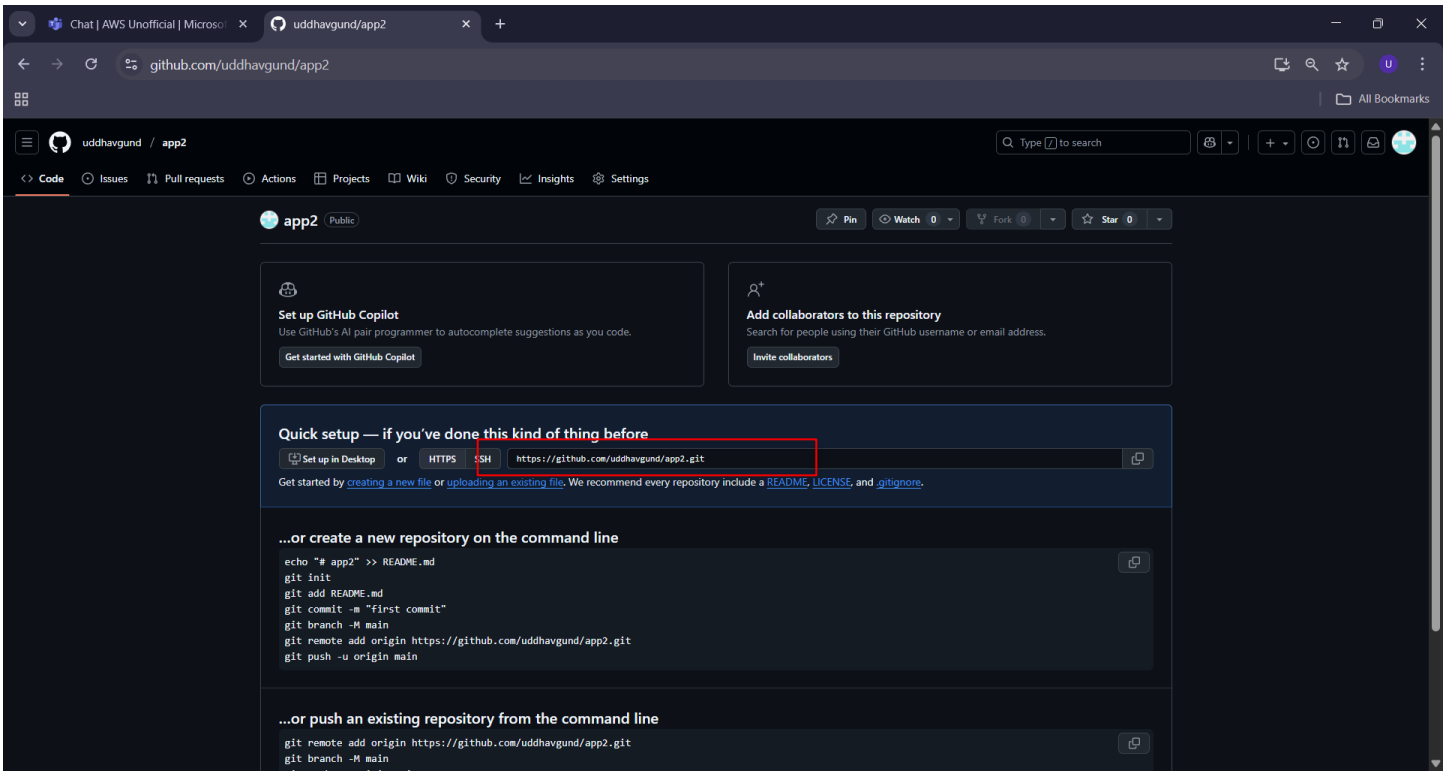
Step 3: Create a new GitHub repository

Commands / Actions:

On GitHub → New Repository → Name it 'app2' → Leave empty → Copy HTTPS URL.




Chat | AWS Unofficial | Microsoft
New repository
github.com/new
All Bookmarks
New repository
Type / to search
Create a new repository
Repositories contain a project's files and version history. Have a project elsewhere? Import a repository.
Required fields are marked with an asterisk (*).
1 General
Owner *
uddhavgund
Repository name *
app2
app2 is available.
Great repository names are short and memorable. How about crispy-waffle?
Description
This file contains various small and big Task/Project of DevOps
63 / 350 characters
2 Configuration
Choose visibility *
Choose who can see and commit to this repository
Public
Add README
READMEs can be used as longer descriptions. About READMEs
Off
Add .gitignore
.gitignore tells git which files not to track. About ignoring files
No .gitignore
Add license
Licenses explain how others can use your code. About licenses
No license
Create repository


Chat | AWS Unofficial | Microsoft
uddhavgund/app2
github.com/uddhavgund/app2
All Bookmarks
uddhavgund / app2
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
app2
Public
Pin
Watch 0
Fork 0
Star 0
Set up GitHub Copilot
Use GitHub's AI pair programmer to autocomplete suggestions as you code.
Get started with GitHub Copilot
Add collaborators to this repository
Search for people using their GitHub username or email address.
Invite collaborators
Quick setup — if you've done this kind of thing before
Set up in Desktop
or
HTTPS
SSH
https://github.com/uddhavgund/app2.git
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.
...or create a new repository on the command line
echo "# app2" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/uddhavgund/app2.git
git push -u origin main
...or push an existing repository from the command line
git remote add origin https://github.com/uddhavgund/app2.git
git branch -M main

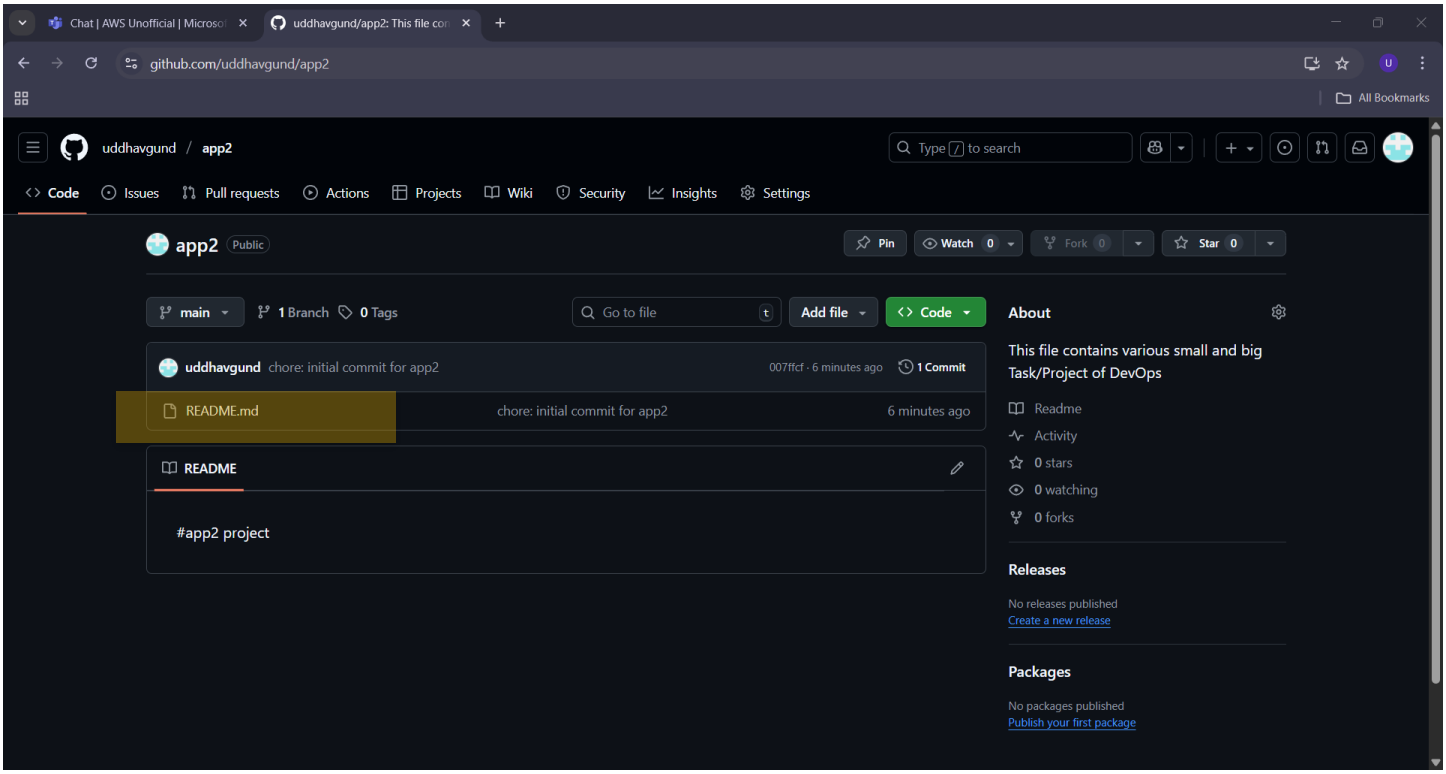
Step 4: Link local repo to GitHub & push

Commands / Actions:

```
git branch -M main
git remote add origin https://github.com/uddhavgund/app2.git
git push -u origin main
```

```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 ~/app2 (master)
$ git branch -M main

Lenovo@LAPTOP-NS0CAVJ8 MINGW64 ~/app2 (main)
$ git remote add origin https://github.com/uddhavgund/app2.git

Lenovo@LAPTOP-NS0CAVJ8 MINGW64 ~/app2 (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 251 bytes | 251.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/uddhavgund/app2.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

✅ Now GitHub will show your **README.md**.



Step 5: Create a feature branch

Commands / Actions:

```
git checkout -b feature/add-dashboard
```

```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 ~/app2 (main)
$ git checkout -b feature/add-dashboard
Switched to a new branch 'feature/add-dashboard'
```

Step 6: Add new data (file)

Commands / Actions:

```
echo "This is dashboard feature for app2" > dasboard.txt
git add dasboard.txt
git commit -m "feat: add dashboard feature"
```

```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 ~/app2 (feature/add-dashboard)
$ echo "This is dasboard feature for app2" > dasboard.txt

Lenovo@LAPTOP-NS0CAVJ8 MINGW64 ~/app2 (feature/add-dashboard)
$ git add dashboard.txt
fatal: pathspec 'dashboard.txt' did not match any files

Lenovo@LAPTOP-NS0CAVJ8 MINGW64 ~/app2 (feature/add-dashboard)
$ git add dasboard.txt
warning: in the working copy of 'dasboard.txt', LF will be rep
laced by CRLF the next time Git touches it

Lenovo@LAPTOP-NS0CAVJ8 MINGW64 ~/app2 (feature/add-dashboard)
$ git commit -m "feat: add dashboard feature"
[feature/add-dashboard 9d3fb97] feat: add dashboard feature
 1 file changed, 1 insertion(+)
 create mode 100644 dasboard.txt
```

Step 7: Push branch to GitHub

Commands / Actions:

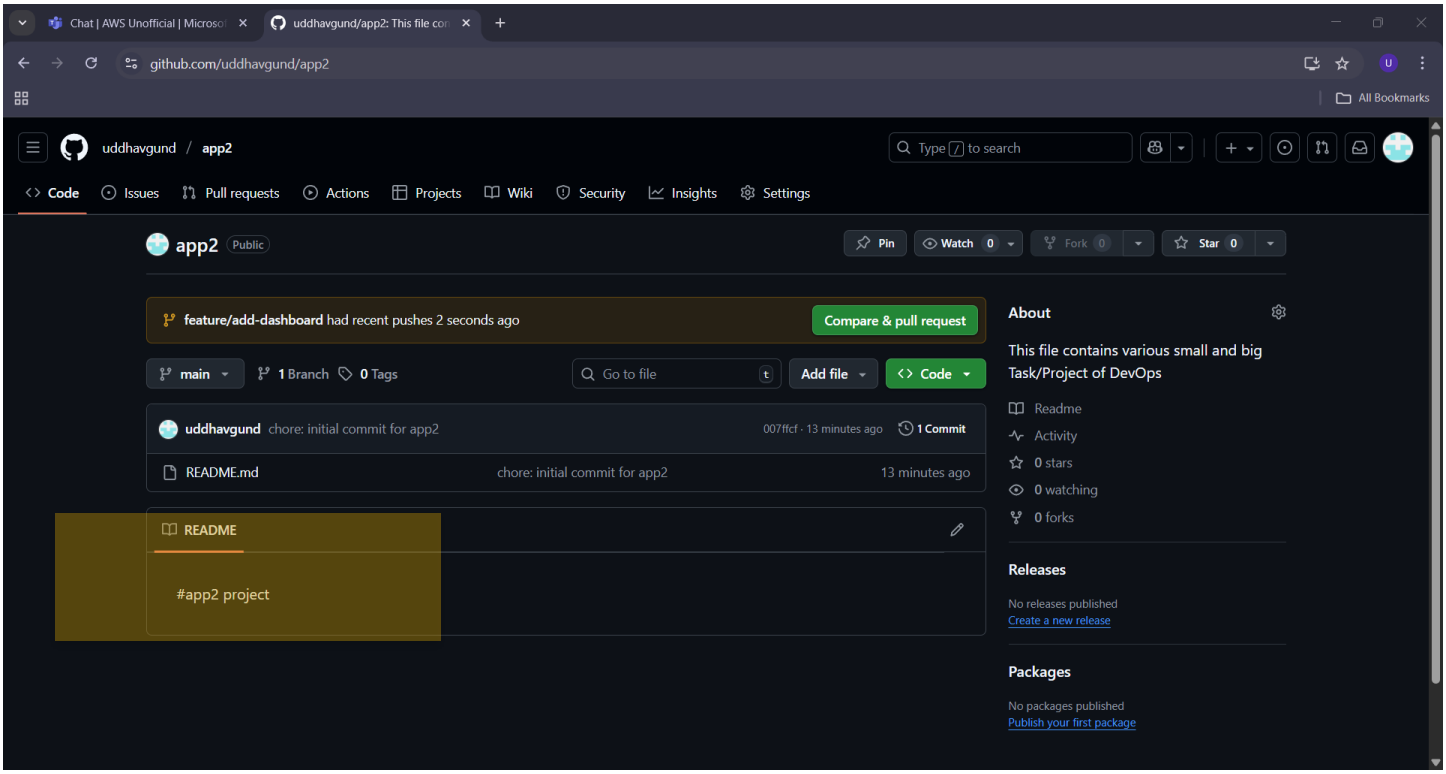
```
git push -u origin feature/add-dashboard
```

```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 ~/app2 (feature/add-dashboard)
$ git push -u origin feature/add-dashboard
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 334 bytes | 334.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature/add-dashboard' on G
itHub by visiting:
remote:      https://github.com/uddhavgund/app2/pull/new/featu
re/add-dashboard
remote:
To https://github.com/uddhavgund/app2.git
 * [new branch]      feature/add-dashboard -> feature/add-dash
board
branch 'feature/add-dashboard' set up to track 'origin/feature
/add-dashboard'.

Lenovo@LAPTOP-NS0CAVJ8 MINGW64 ~/app2 (feature/add-dashboard)
$
```

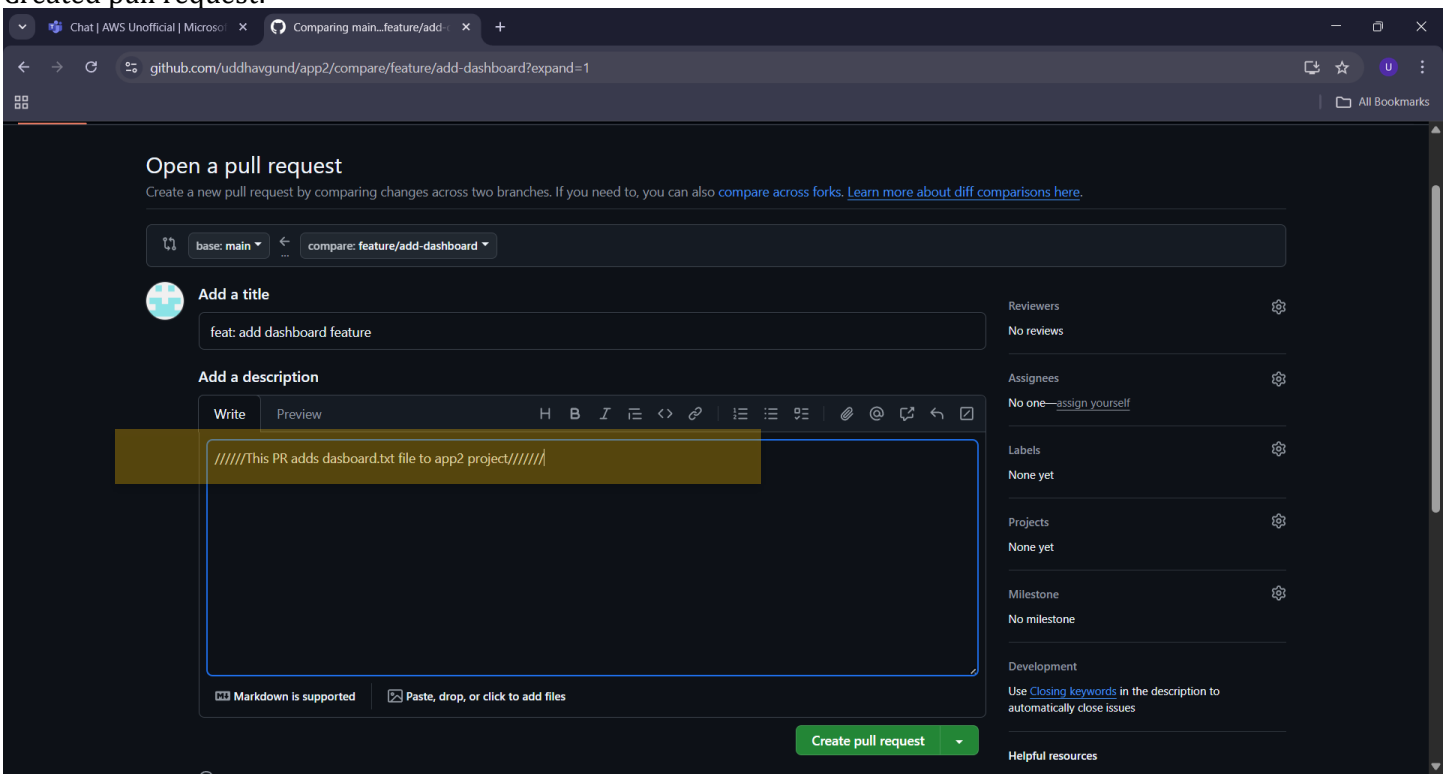
Step 8: Create Pull Request on GitHub

Commands / Actions:

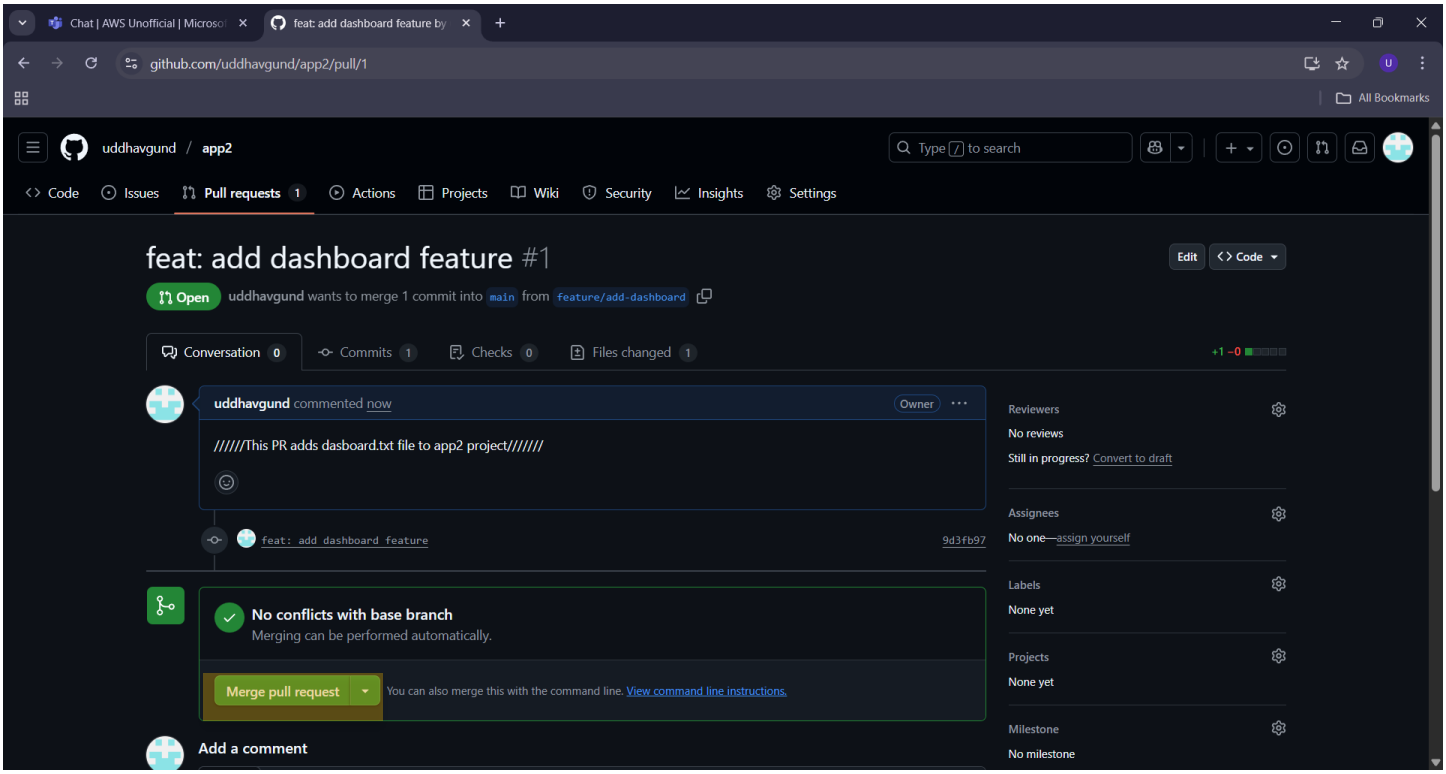
Go to repo → 'Compare & pull request' → Title: feat: add dashboard feature → Merge PR.

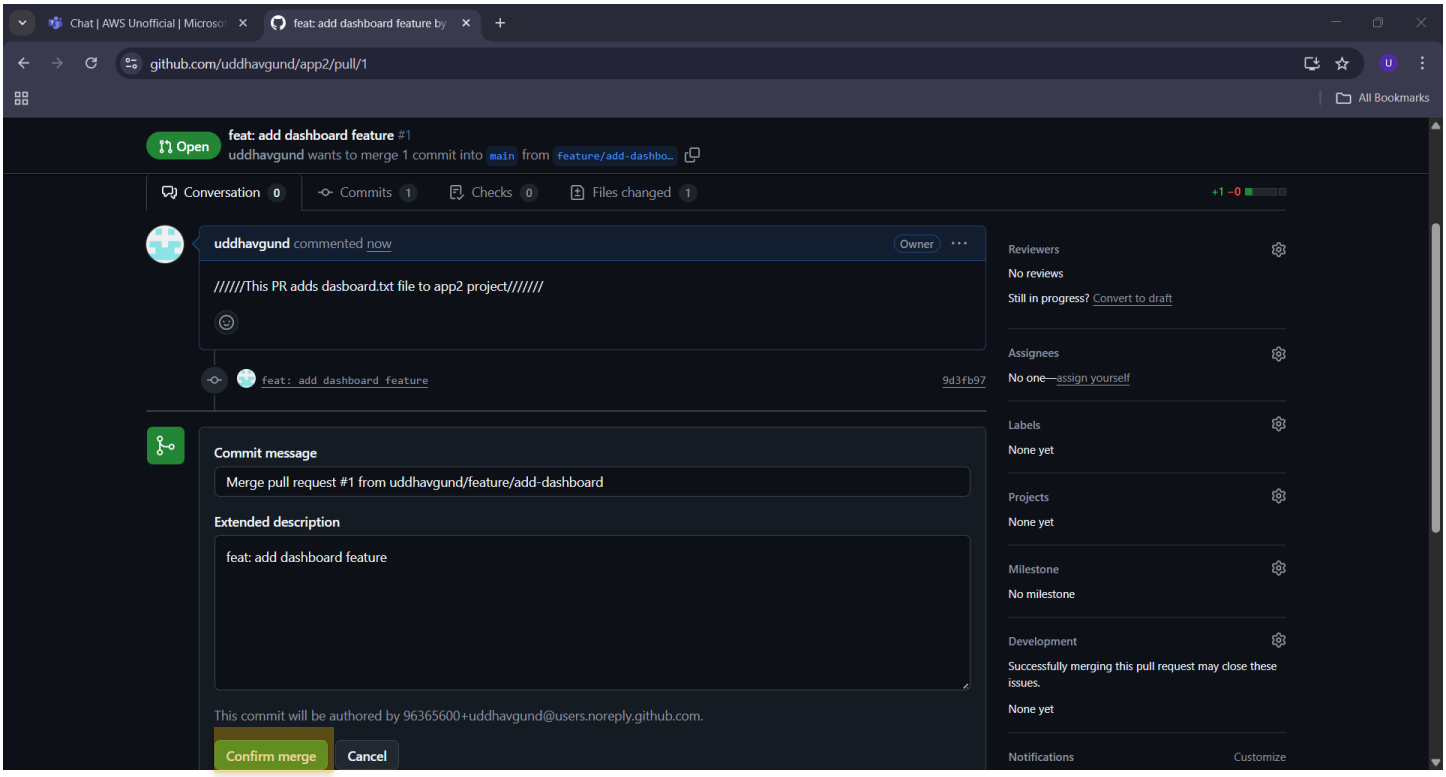


Created pull request!

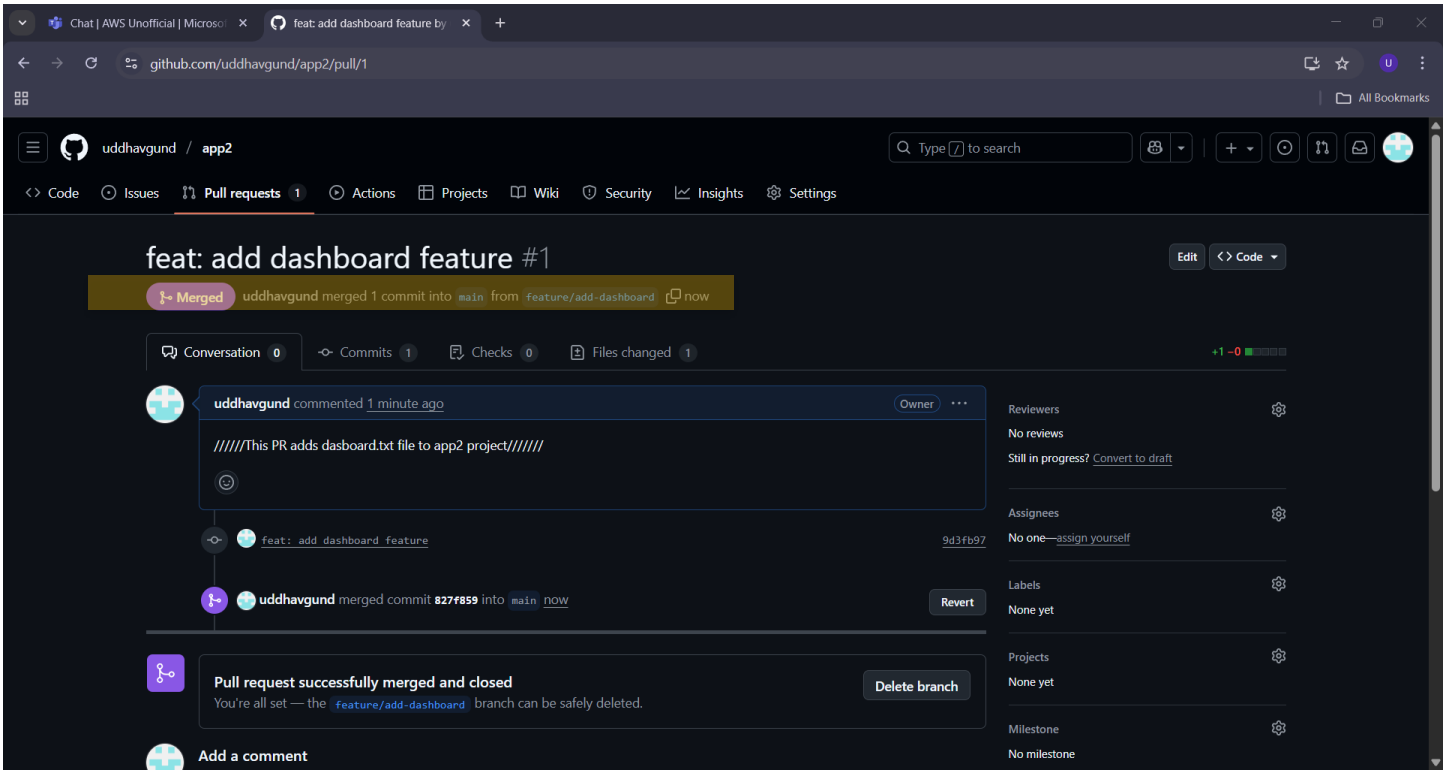


Merged pull request





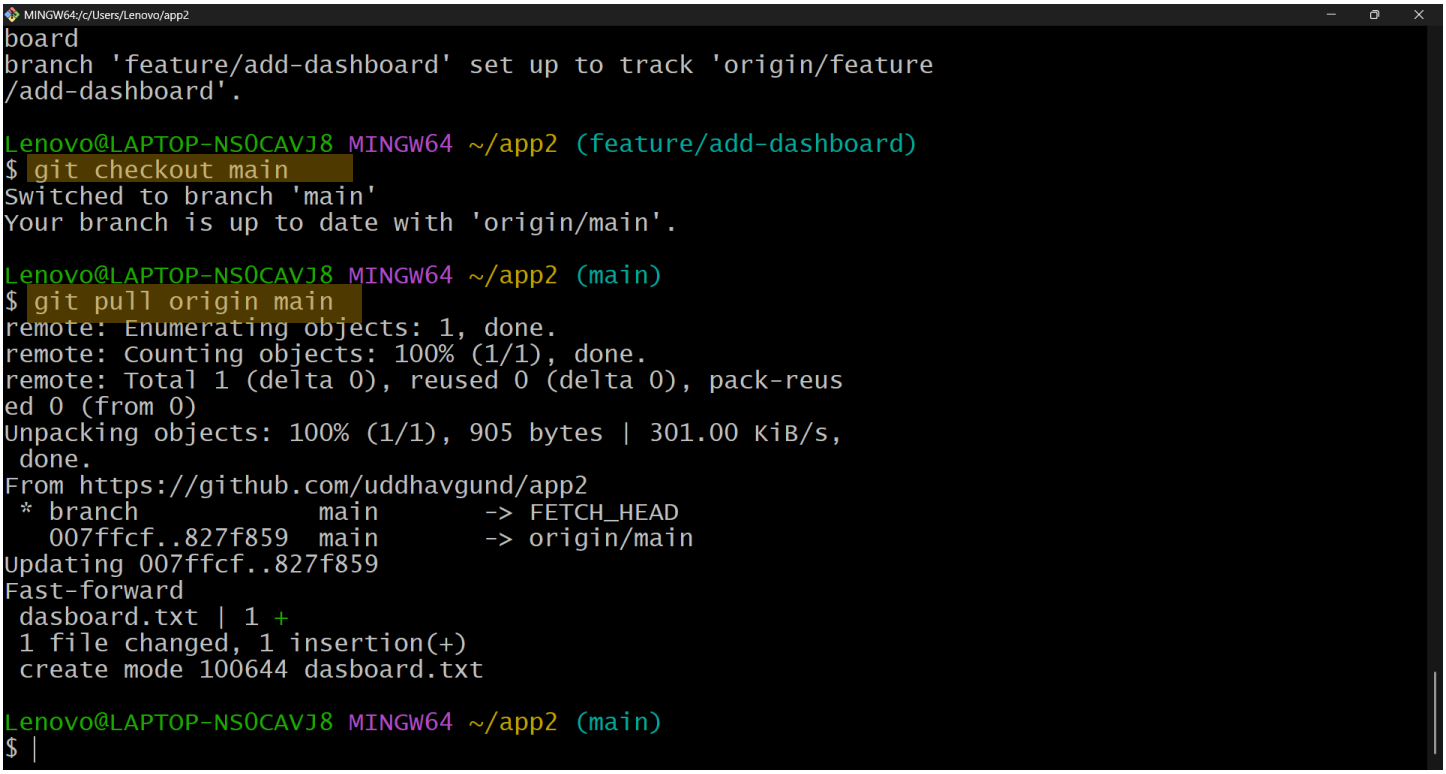
Merged!



Step 9: Update local main after merge

Commands / Actions:

git checkout main
git pull origin main



Step 10: Cleanup branches

Commands / Actions:

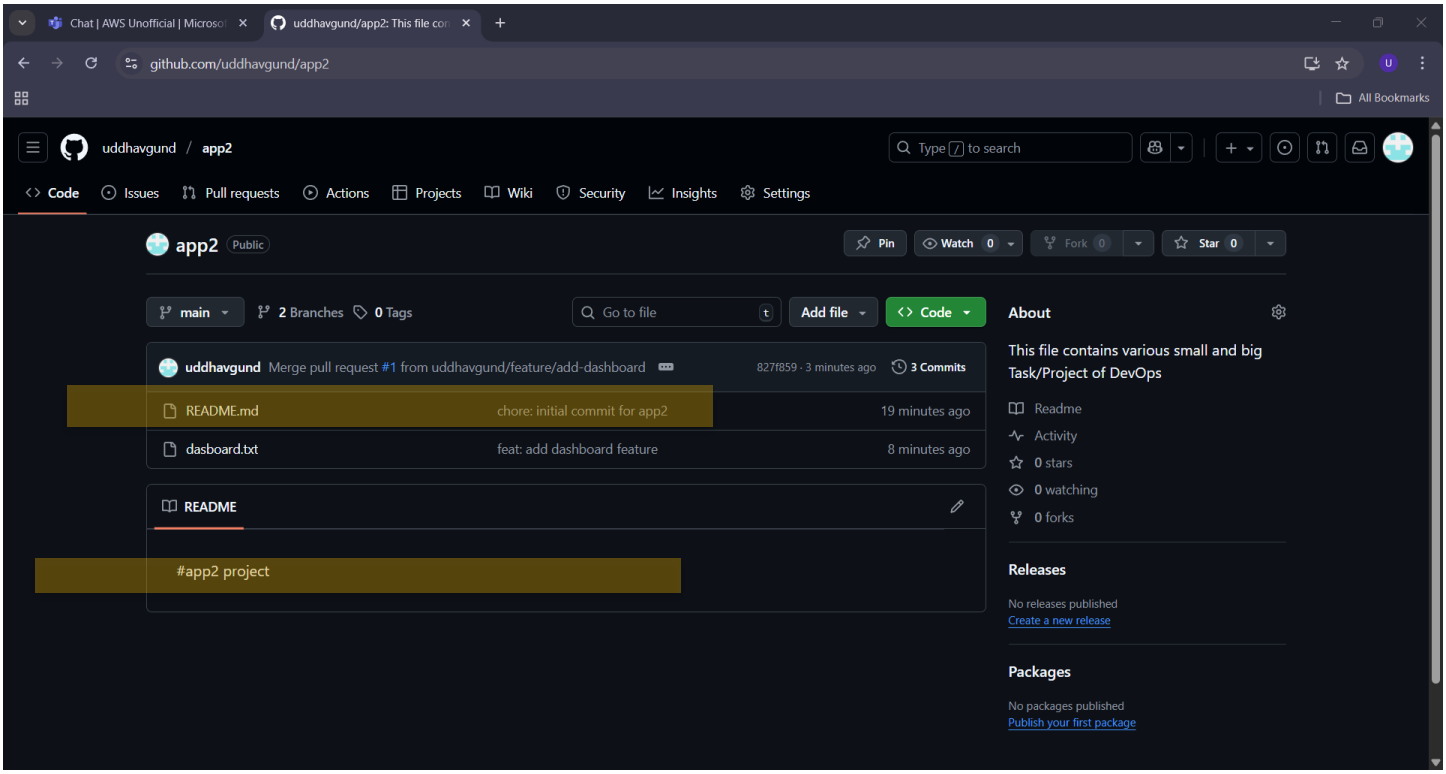
```
git branch -d feature/add-dashboard
git push origin --delete feature/add-dashboard
```

```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 ~/app2 (main)
$ git branch -d feature/add-dashboard
Deleted branch feature/add-dashboard (was 9d3fb97).

Lenovo@LAPTOP-NS0CAVJ8 MINGW64 ~/app2 (main)
$ git push origin --delete feature/add-dashboard
To https://github.com/uddhavgund/app2.git
 - [deleted]         feature/add-dashboard
```

Final Repository Files

README.md content:

```
# app2 project
```



dasboard.txt content:

```
This is dashboard feature for app2
```