

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

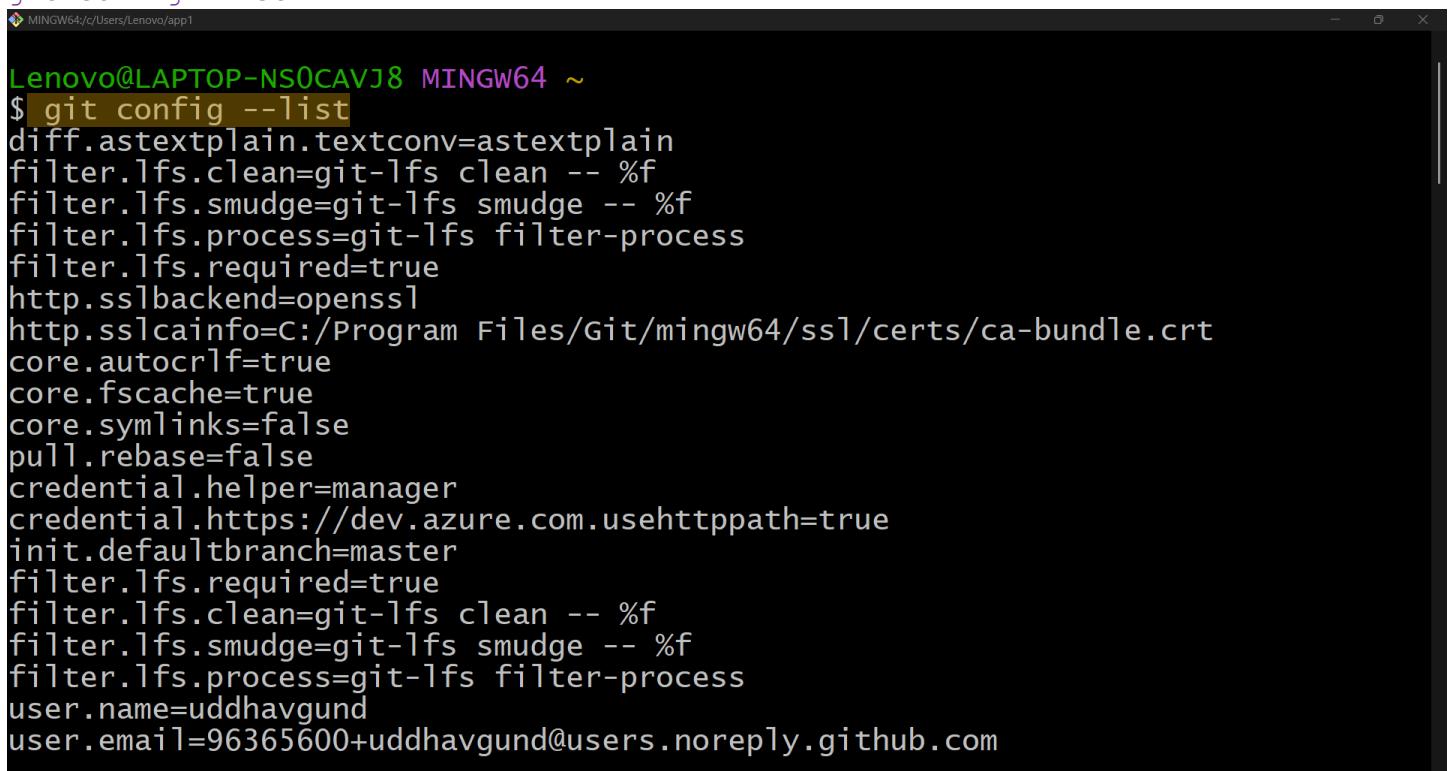
This report documents the step-by-step execution of the Git & GitHub Basics task. The exercise demonstrates repository initialization, remote linking, feature branching, pull requests, and merging workflow.

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
```



```
Lenovo@LAPTOP-NS0CAVJ8 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  
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt  
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.required=true  
filter.lfs.clean=git-lfs clean -- %f  
filter.lfs.smudge=git-lfs smudge -- %f  
filter.lfs.process=git-lfs filter-process  
user.name=uddhavgund  
user.email=96365600+uddhavgund@users.noreply.github.com
```

Step 2: Create a new folder (local project)

Commands :

```
mkdir app2  
cd app2
```

```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 ~  
$ mkdir app2
```

```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 ~  
$ 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-NS0CAVJ8 MINGW64 ~/app2
$ git init
Initialized empty Git repository in C:/Users/Lenovo/app2/.git/
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 ~/app2 (master)
$ echo "#app2 project" > README.md

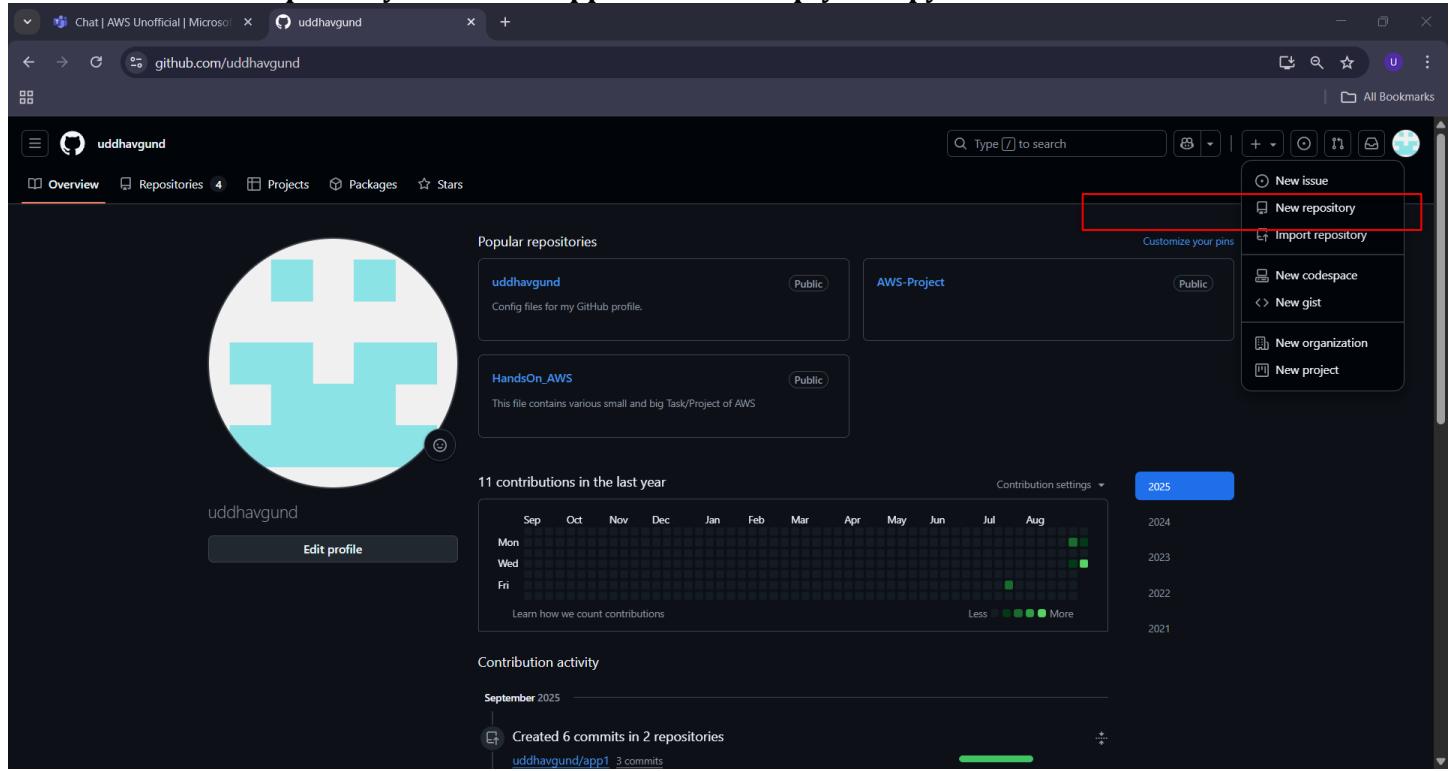
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 ~/app2 (master)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

Lenovo@LAPTOP-NS0CAVJ8 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

New repository

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

Great repository names are short and memorable. How about crispy-waffle?

Description: This file contains various small and big Task/Project of DevOps

2 Configuration

Choose visibility: Public

Add README: Off

Add .gitignore: No .gitignore

Add license: No license

Create repository

Chat | AWS Unofficial | Microsoft | uddhavgund/app2

github.com/uddhavgund/app2

uddhavgund / app2

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

app2 (Public)

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 | SH | 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 "#!/bin/bash" >> 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
```

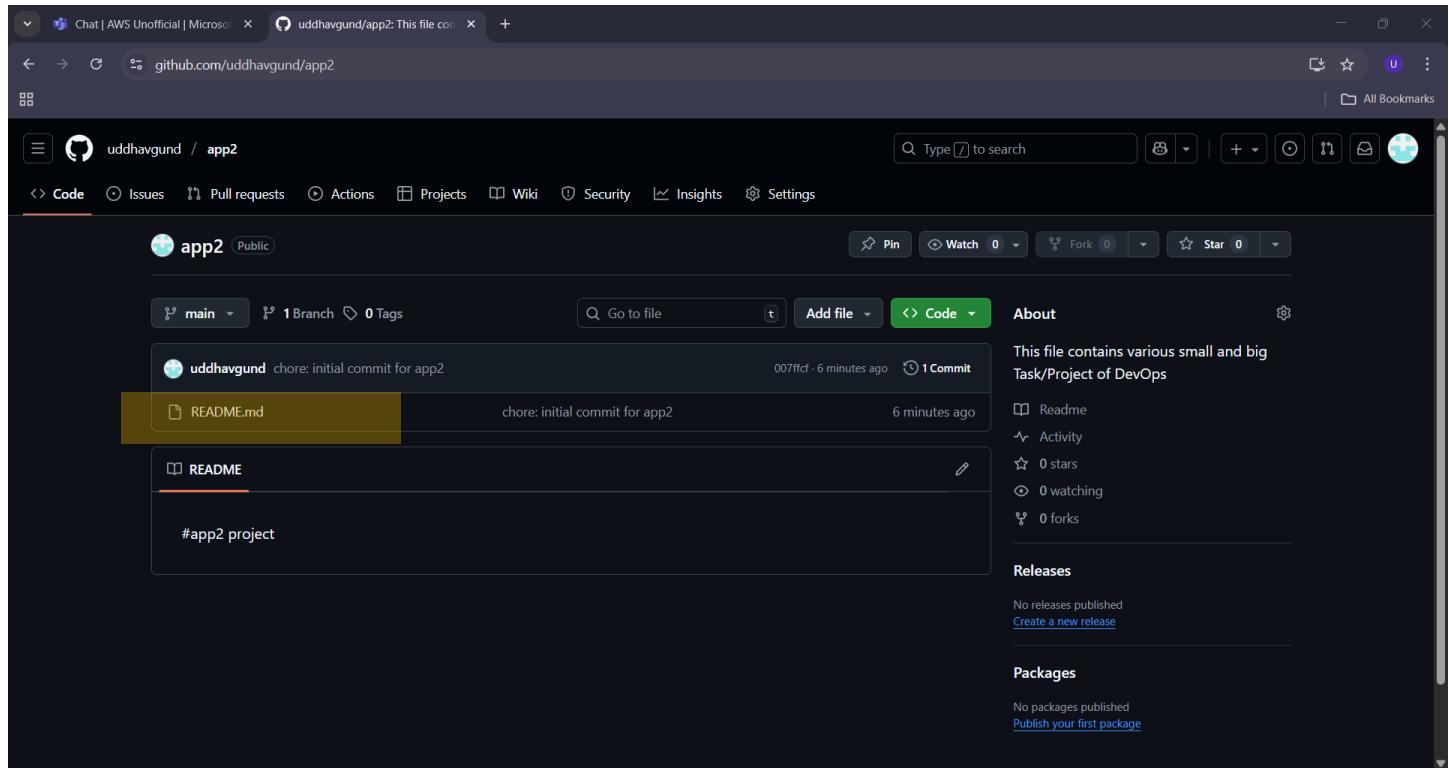
Step 4: Link local repo to GitHub & push

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.

The screenshot shows the GitHub repository page for 'app2'. The repository is public and has one branch ('main') and one commit from 'uddhavgund'. The README file contains '#app2 project'. The 'About' section notes recent pushes and provides links for Readme, Activity, Stars, Forks, and Releases. The 'Packages' section shows no packages published.

Created pull request!

The screenshot shows the 'Open a pull request' dialog on GitHub. It shows the base branch as 'main' and the compare branch as 'feature/add-dashboard'. The title is 'feat: add dashboard feature' and the description is '//////This PR adds dashboard.txt file to app2 project////////'. The dialog includes sections for Reviewers, Assignees, Labels, Projects, Milestone, Development, and Helpful resources. A 'Create pull request' button is at the bottom.

Merged pull request

The screenshot shows a GitHub pull request merge dialog. At the top, it says "feat: add dashboard feature #1" and "Open". It indicates "uddhavgund wants to merge 1 commit into main from feature/add-dashboard". The commit message is "//////This PR adds dasboard.txt file to app2 project//////". Below the commit message, there is a green box stating "No conflicts with base branch" and "Merging can be performed automatically." A large green button says "Merge pull request". To the right, there are sections for "Reviewers" (No reviews), "Assignees" (None yet—assign yourself), "Labels" (None yet), "Projects" (None yet), "Milestone" (No milestone), and "Development" (Successfully merging this pull request may close these issues). There is also a "Customize" link at the bottom right.

The screenshot shows a GitHub pull request merge dialog. At the top, it says "feat: add dashboard feature #1" and "Open". It indicates "uddhavgund wants to merge 1 commit into main from feature/add-dashboard". The commit message is "//////This PR adds dasboard.txt file to app2 project//////". Below the commit message, there is a green box with "Commit message" containing "Merge pull request #1 from uddhavgund/feature/add-dashboard" and "Extended description" containing "feat: add dashboard feature". A note at the bottom says "This commit will be authored by 96365600+uddhavgund@users.noreply.github.com.". At the bottom, there are two buttons: "Confirm merge" and "Cancel". To the right, there are sections for "Reviewers" (No reviews), "Assignees" (None yet—assign yourself), "Labels" (None yet), "Projects" (None yet), "Milestone" (No milestone), "Development" (Successfully merging this pull request may close these issues), and "Notifications" (Customize).

Merged!

The screenshot shows a GitHub pull request page for a repository named 'app2'. The pull request is titled 'feat: add dashboard feature #1'. A prominent yellow banner at the top indicates the PR is 'Merged' by 'uddhavgund' into the 'main' branch. The main content area displays the commit history, which includes a comment from 'uddhavgund' stating '//////This PR adds dasboard.txt file to app2 project////////'. Below this is another commit from 'uddhavgund' merging commit '827f859' into 'main'. A message box at the bottom confirms the 'Pull request successfully merged and closed'. On the right side, there are sections for 'Reviewers', 'Assignees', 'Labels', 'Projects', and 'Milestone', all of which are currently empty or set to 'None yet'. The interface is in dark mode.

Step 9: Update local main after merge

Commands / Actions:

```
git checkout main  
git pull origin main
```

```
MINGW64:/c/Users/Lenovo/app2  
board  
branch 'feature/add-dashboard' set up to track 'origin/feature/  
/add-dashboard'.  
  
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 ~/app2 (feature/add-dashboard)  
$ git checkout main  
Switched to branch 'main'  
Your branch is up to date with 'origin/main'.  
  
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 ~/app2 (main)  
$ git pull origin main  
remote: Enumerating objects: 1, done.  
remote: Counting objects: 100% (1/1), done.  
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reus  
ed 0 (from 0)  
Unpacking objects: 100% (1/1), 905 bytes | 301.00 KiB/s,  
done.  
From https://github.com/uddhavgund/app2  
 * branch            main      -> FETCH_HEAD  
   007fffcf..827f859  main      -> origin/main  
Updating 007fffcf..827f859  
Fast-forward  
 dasboard.txt | 1 +  
 1 file changed, 1 insertion(+)  
 create mode 100644 dasboard.txt  
  
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 ~/app2 (main)  
$ |
```

Step 10: Cleanup branches

Commands / Actions:

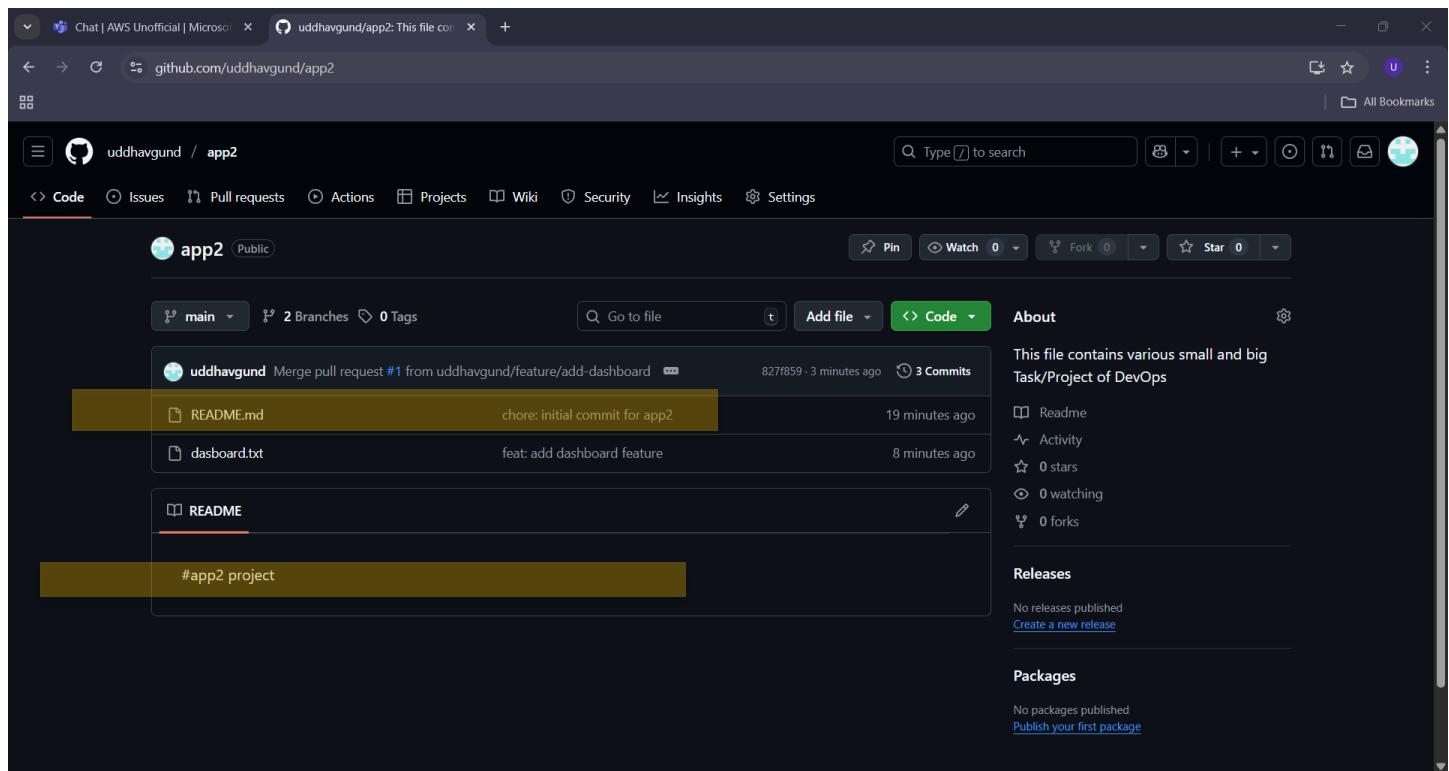
```
git branch -d feature/add-dashboard  
git push origin --delete feature/add-dashboard
```

```
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 dasboard feature for app2

The screenshot shows a GitHub repository interface for a project named 'app2'. On the left, there's a sidebar with 'Files' and a list containing 'main' (selected), 'README.md', and 'dashboard.txt' (highlighted with a yellow background). The main area displays the contents of 'dashboard.txt': 'This is dasboard feature for app2'. Above this, a commit card is visible: 'uddhavgund feat: add dashboard feature' (commit hash: 9d3fb97, 8 minutes ago). The top navigation bar includes links for Chat, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings.