# Task 2

- Consider that your want to start an open-source project in your organization. Perform all the standard operation to create a repository with minimal permission for all the users. It should contain.

1. Proper open source structure
2. Proper Readme
3. Add 2 collaborator
4. Host GitHub Pages using settings (Designed to host your personal, organization, or project pages from a GitHub repository)

Step 1: Creating an open source project structure using prebuilt project template.

https://github.com/akash-soni/DVC-Mkdocs-project-template
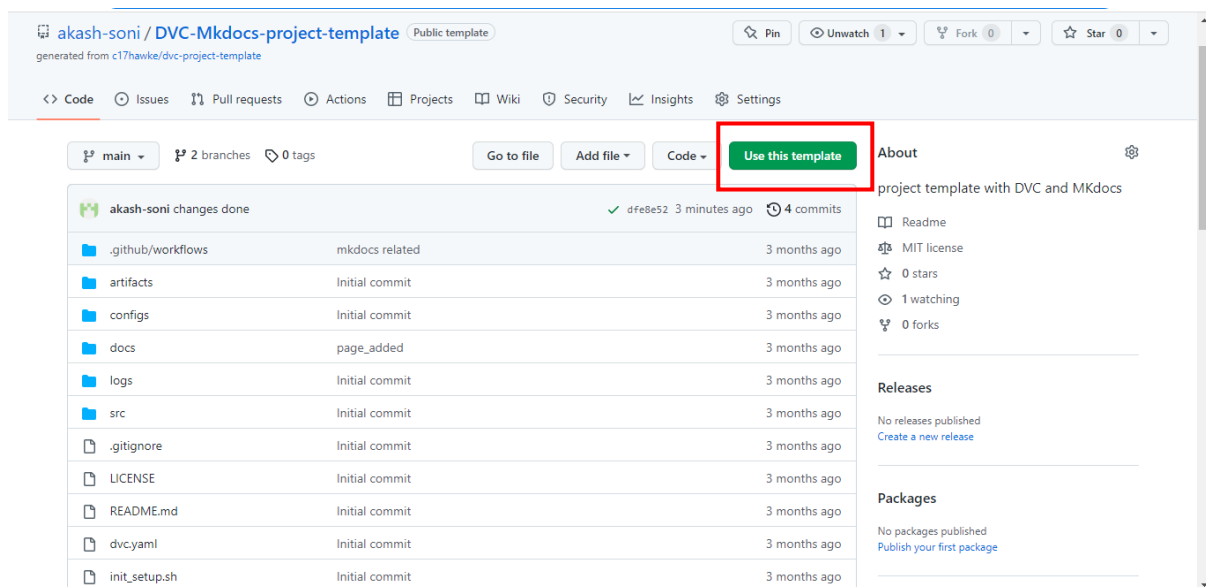
Step 2: Click on "***Use this template***" and fill all the further details

Create a new repository from DVC-Mkdocs-project-template

The new repository will start with the same files and folders as akash-soni/DVC-Mkdocs-project-template.

Owner *   **Repository name** *

akash-soni ▾   /   collaboration Project-Repo   ✔

Great repository names are   Your new repository will be created as collaboration-Project-Repo.   potato?

Description (optional)

This Repo is a demonstration of collaboration with developers

◉ 📖 **Public**
Anyone on the internet can see this repository. You choose who can commit.

○ 🔒 **Private**
You choose who can see and commit to this repository.

☑ **Include all branches**
Copy all branches from akash-soni/DVC-Mkdocs-project-template and not just main.

ⓘ You are creating a public repository in your personal account.

Create repository from template

Step 3: Cloning the repository into the local Repo

<> Code   ⊙ Issues   Pull requests   ▷ Actions   ⊞ Projects   📖 Wiki   🛡 Security   ⊵ Insights   ⚙ Settings

🔀 main ▾   🔀 2 branches   ⬡ 0 tags   Go to file   Add file ▾   Code ▾

akash-soni Initial commit		
📁 .github/workflows	Initial commit	
📁 artifacts	Initial commit	
📁 configs	Initial commit	
📁 docs	Initial commit	
📁 logs	Initial commit	
📁 src	Initial commit	
📄 .gitignore	Initial commit	1 minute ago
📄 LICENSE	Initial commit	1 minute ago
📄 README.md	Initial commit	1 minute ago
📄 dvc.yaml	Initial commit	1 minute ago
📄 init_setup.sh	Initial commit	1 minute ago

Clone ⊡   ⑦

HTTPS   SSH   GitHub CLI

https://github.com/akash-soni/collaboratic   ⧉

Use Git or checkout with SVN using the web URL.

💻 Open with GitHub Desktop

📄 Download ZIP

---

https://github.com/akash-soni/collaboration-Project-Repo.git

- Copy the link and git clone into local

Step 4: Adding collaborators
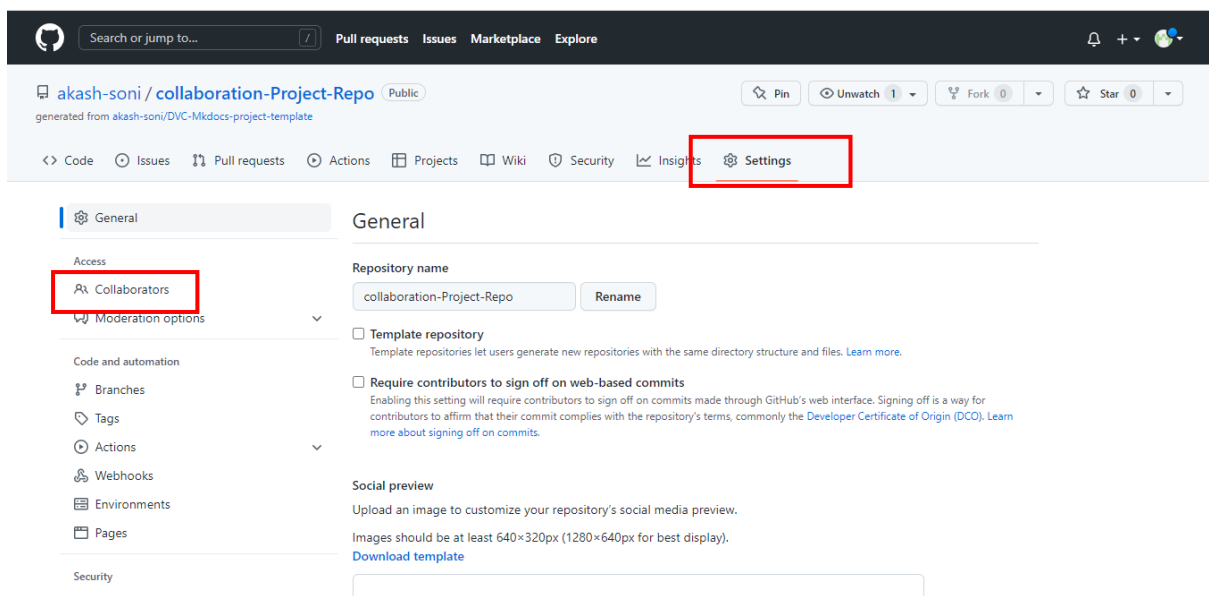
- Invite sent to 2 collaborator

Manage access

Add people

☐ Select all		Type ▾

🔍 Find a collaborator...

☐ ✉ **aakash.soni@live.in** Awaiting response	Pending Invite ⧉	Remove
☐ ✉ **toakashsoni@gmail.com** Awaiting response	Pending Invite ⧉	Remove

🖥 **akash-soni** / **collaboration-Project-Repo** (Public)
generated from akash-soni/DVC-Mkdocs-project-template

👁 Watch 1 ▾

<> Code   ⊙ Issues   ⇵ Pull requests   ⊙ Actions   ⊞ Projects   🛡 Security   📈 Insights

akash-soni invited you to collaborate

Accept invitation   **Decline**

🔒 **Owners** of collaboration-Project-Repo will be able to see:

- Your public profile information
- Certain activity within this repository
- Country of request origin
- Your access level for this repository
- Your IP address

Is this user sending spam or malicious content?
Block akash-soni

- Once the collaborators accepts the invite we can proceed further

Step 5: Now each collaborator will fork the repository so that they will get exact copy of the Repository on their github.

- Once collaborator fork is done collaborators will get their own copy which they can then clone and get it into their local
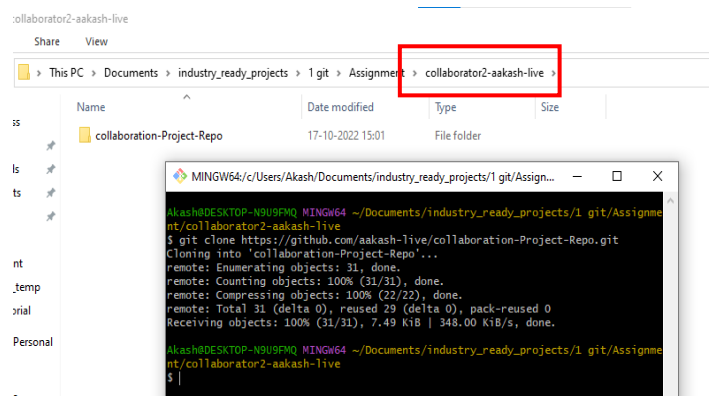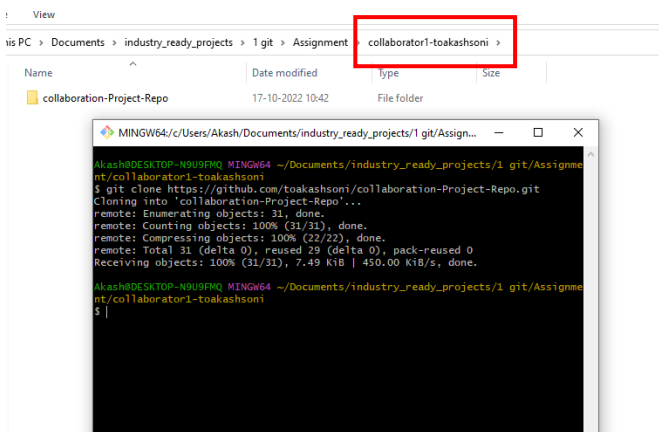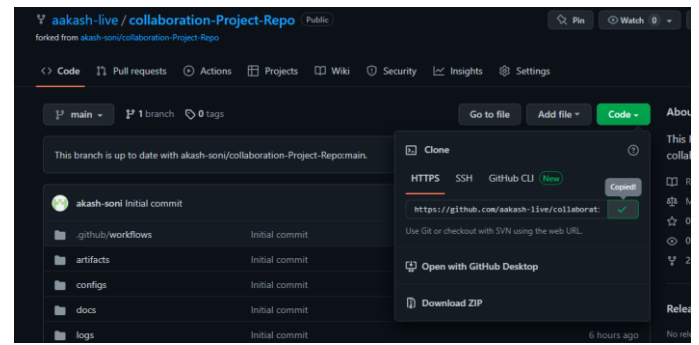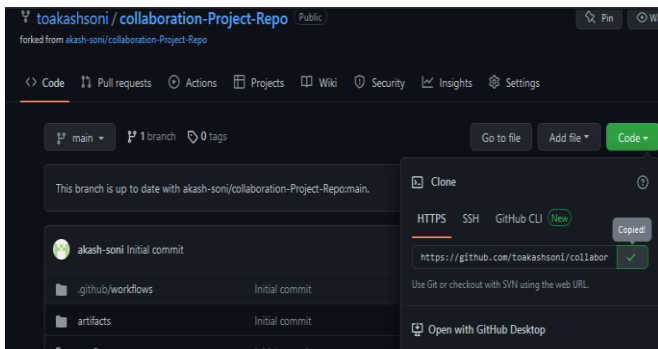
Step 6: Lets Make some changes for collaborator 1.

1. We have added data_ingestion.py and have updated mkdocs.md
2. Then we add the changes
3. Commit the changes
4. Push them to the remote repository of collaborator 1

5. Now observe on Remote Repo
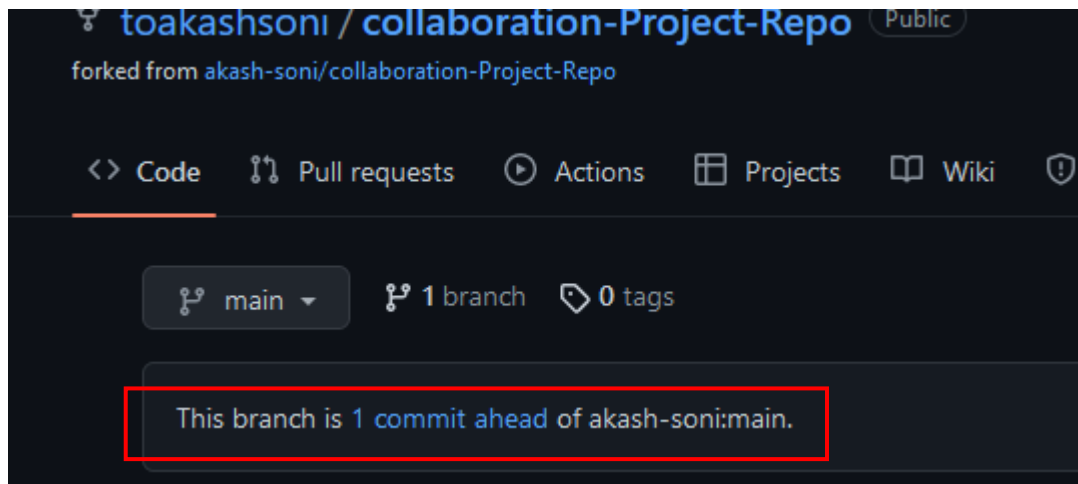
It means that currently this particular collaborator has made a commit and is now ahead of Base project assigner branch

6. Now collaborator 1 will generate a Pull request from his end so that Base project assigner can verify and merge the changes in the base repo

Handwritten annotations on the image:
- base Repo where changes will be merged once verified
- Repo of Collaborator1 who has initiated a Pull Request
- Changes done by Collaborators1
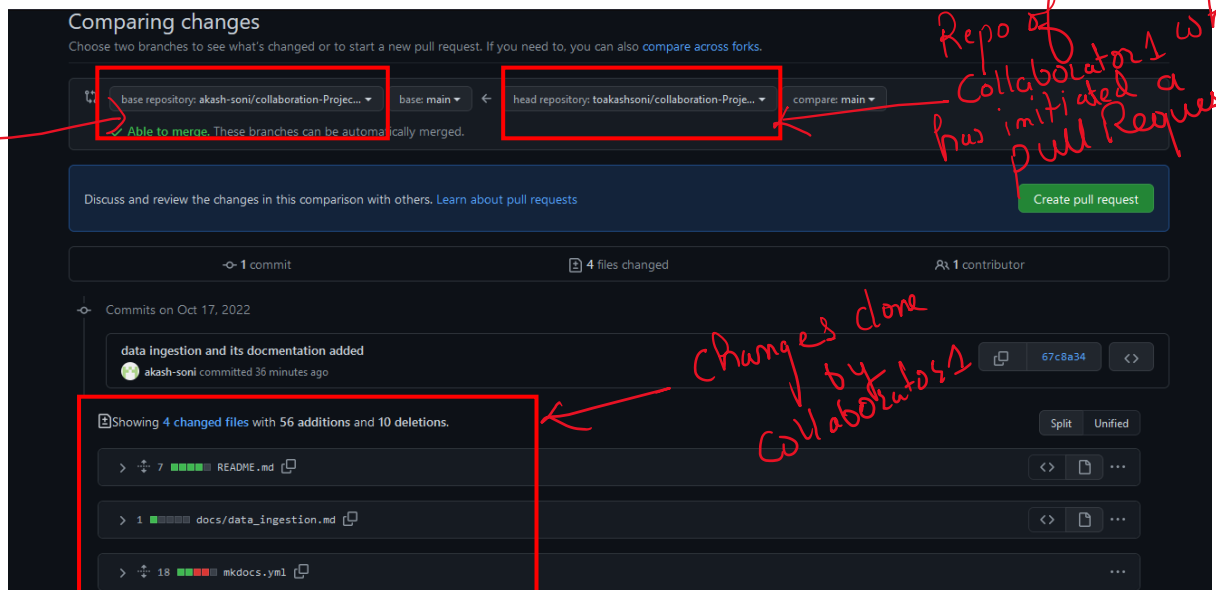
7. Now Collabrator 1 will "*create pull Request*"
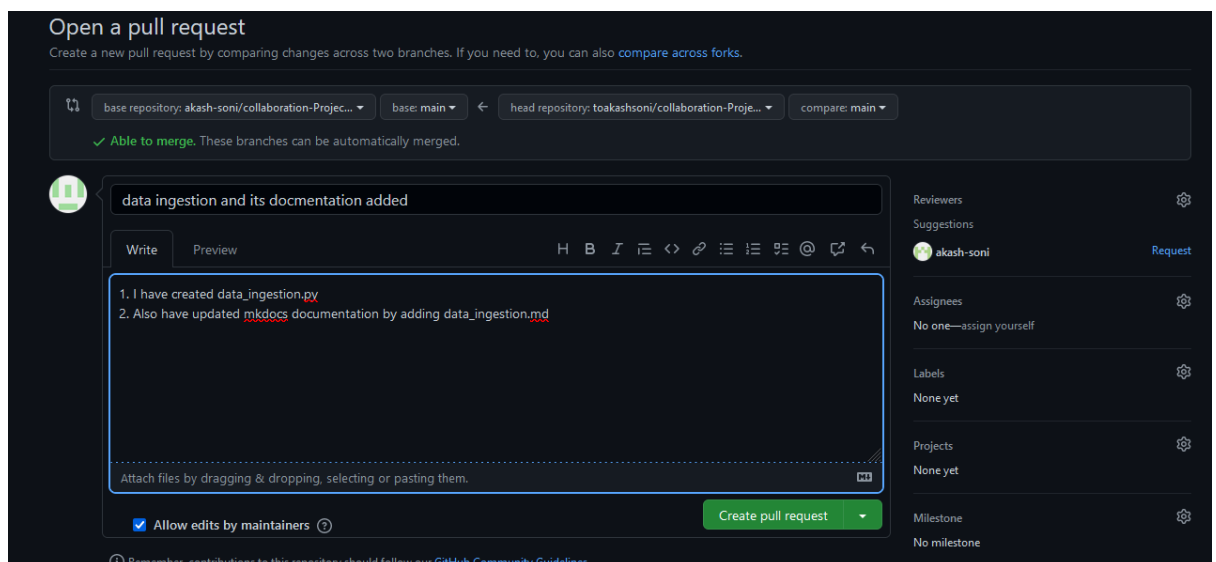
8. This will now finally generate pull request from collaborator 1
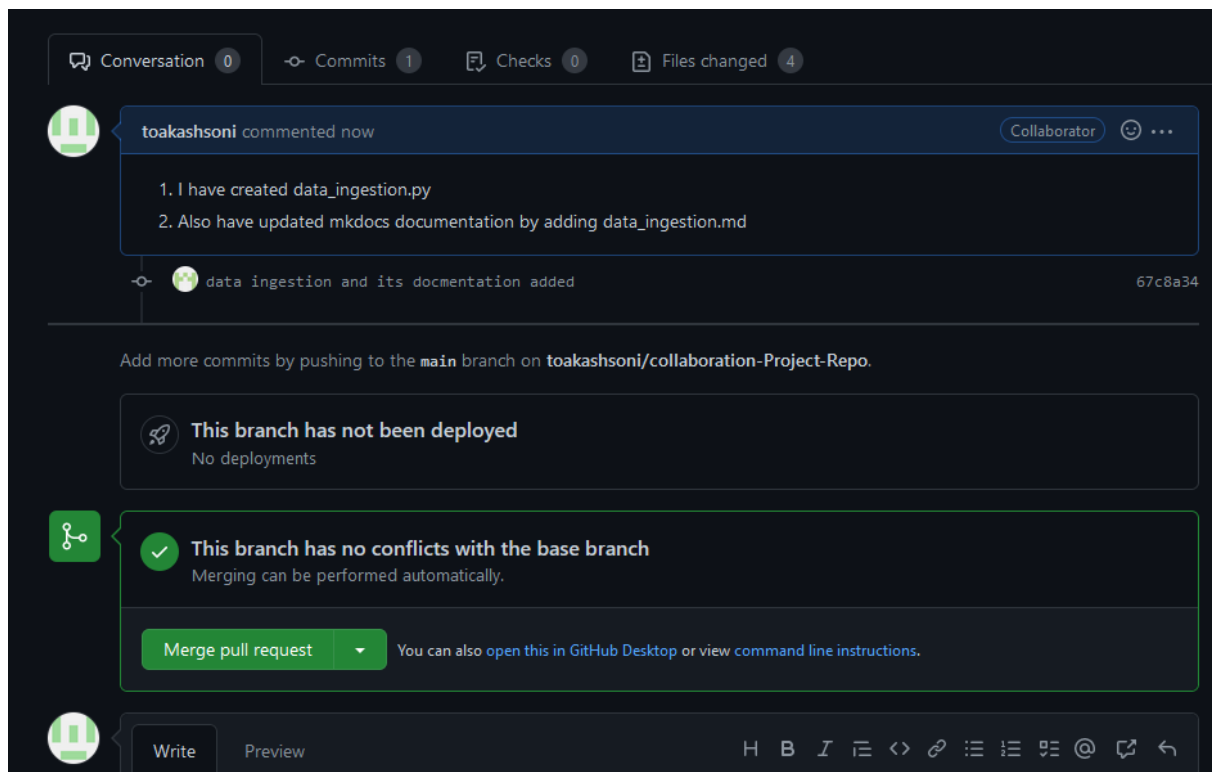
9. Now base project manager will receive one pull request

10. Here Base Project Manager will verify what commits, changes Collaborator 1 has done, if satisfied he will "*merge pull request*"

data ingestion and its docmentation added #1

Open · toakashsoni wants to merge 1 commit into `akash-soni:main` from `toakashsoni:main`

Conversation 0	Commits 1	Checks 0	Files changed 4	+56 −10

toakashsoni commented 7 minutes ago   Collaborator

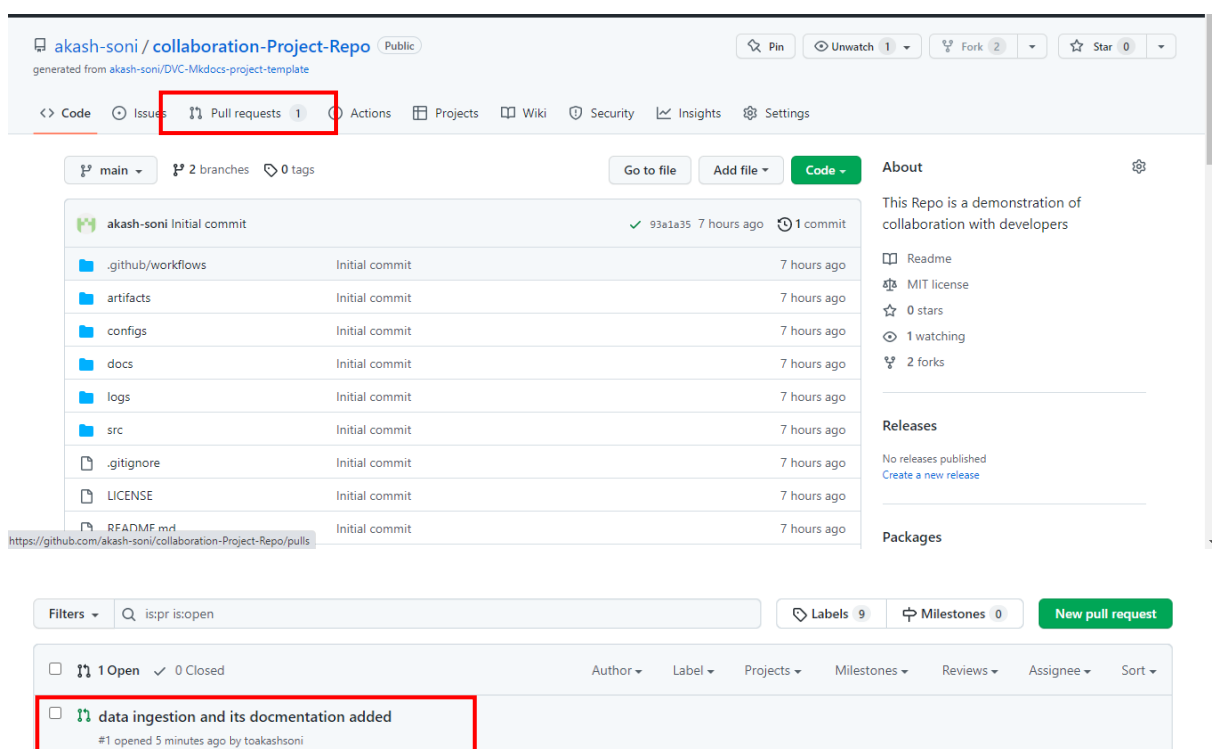1. I have created data_ingestion.py
2. Also have updated mkdocs documentation by adding data_ingestion.md

data ingestion and its docmentation added   67c8a34

Add more commits by pushing to the **main** branch on **toakashsoni/collaboration-Project-Repo**.

🚀 **This branch has not been deployed**
No deployments

✓ **This branch has no conflicts with the base branch**
Merging can be performed automatically.

Merge pull request ▾   You can also open this in GitHub Desktop or view command line instructions.

Reviewers ⚙
Suggestions
akash-soni   Request
Still in progress? Convert to draft

Assignees ⚙
No one—assign yourself

Labels ⚙
None yet

Projects ⚙
None yet

Milestone ⚙
No milestone

11. On "**_Confirm Merge_**" whatever changes collaborator 1 has done will be reflected into base Repository.