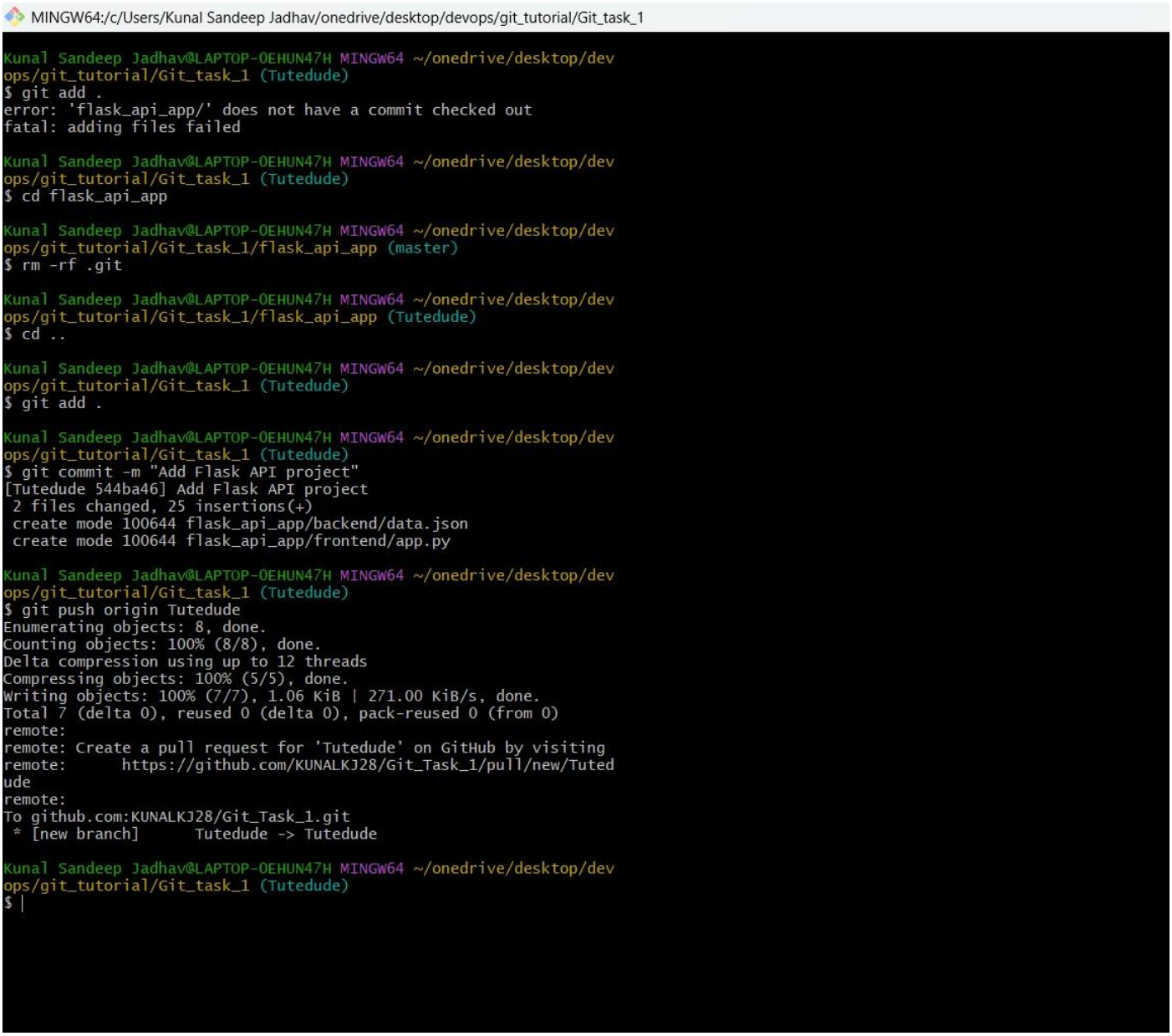
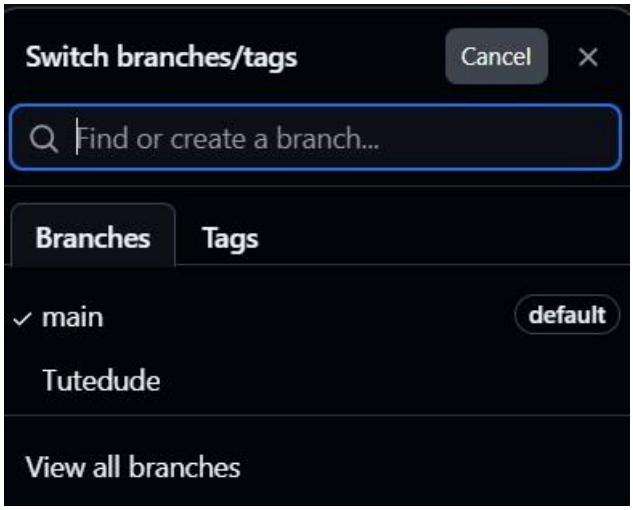
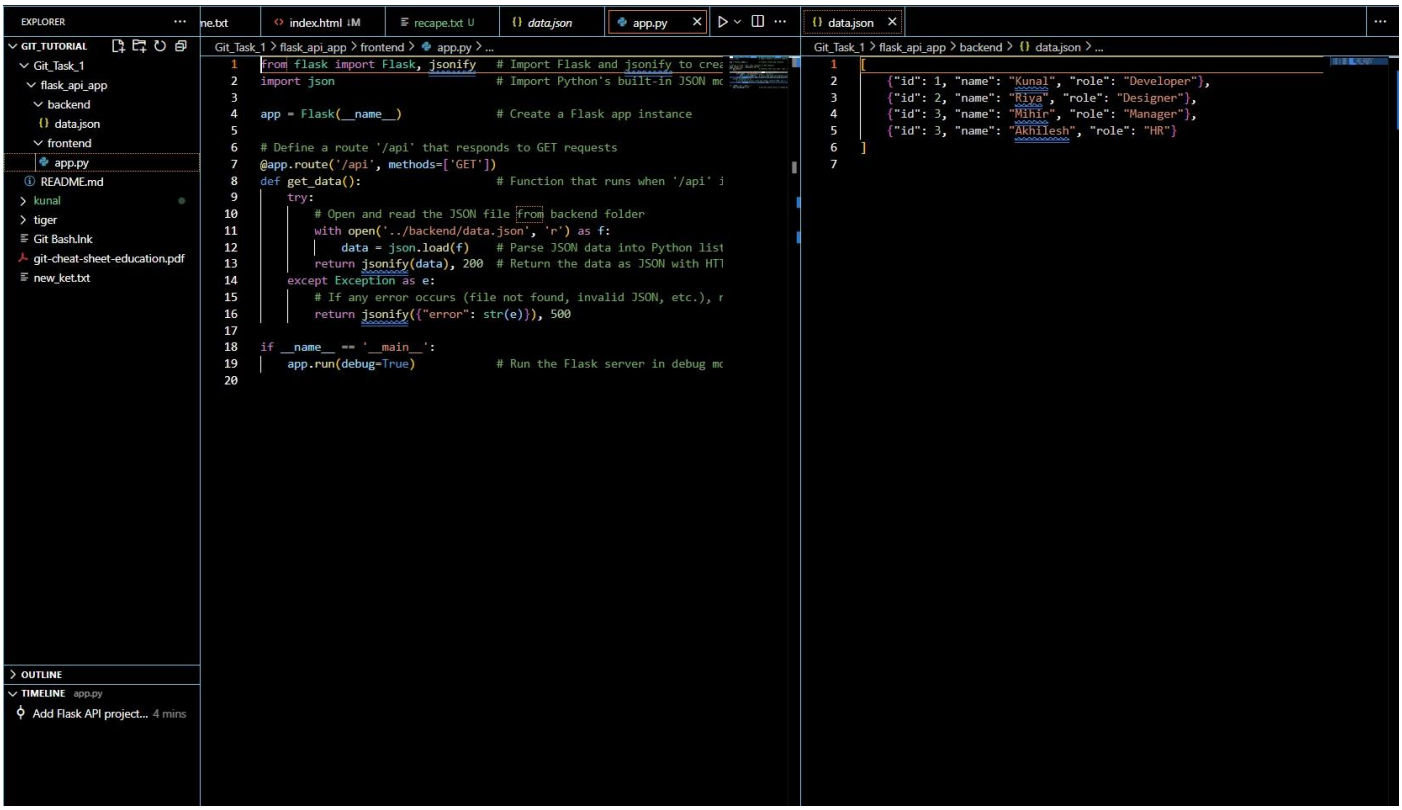
1. Create a new GitHub repository.

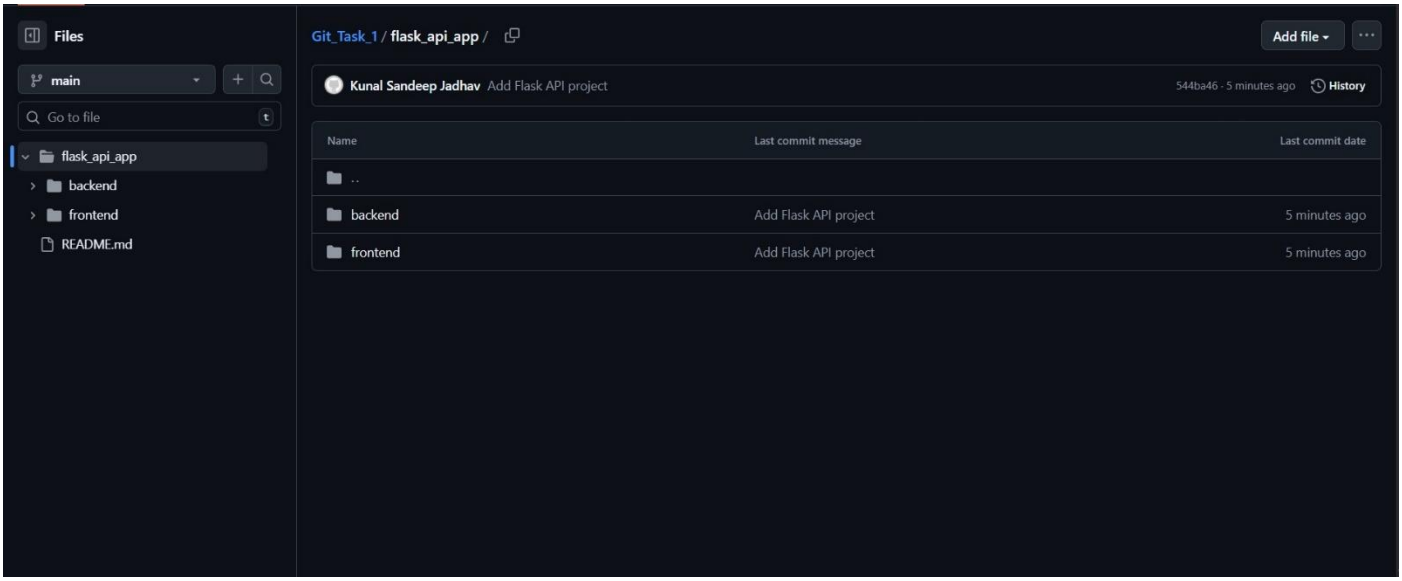
- Clone the repository to your local machine using SSH (generate an SSH key if needed, add the public key to your GitHub account).
- Create a new branch named after your username (e.g., Tutedude).
- Add your Flask project files to this branch.

Commit the changes and merge the branch into the main branch.




Switch branches/tags
Cancel
Find or create a branch...
Branches
Tags
main
default
Tutedude
View all branches


EXPLORER
GIT_TUTORIAL
Git_Task_1
flask_api_app
backend
data.json
frontend
app.py
README.md
kunal
tiger
Git Bash.lnk
git-cheat-sheet-education.pdf
new_ket.txt
OUTLINE
TIMELINE app.py
Add Flask API project... 4 mins
index.html M
recape.txt U
data.json
app.py
data.json
Git_Task_1 > flask_api_app > frontend > app.py > ...
Git_Task_1 > flask_api_app > backend > {} data.json > ...
from flask import Flask, jsonify # Import Flask and jsonify to crea
import json # Import Python's built-in JSON mo
app = Flask(__name__) # Create a Flask app instance
# Define a route '/api' that responds to GET requests
@app.route('/api', methods=['GET'])
def get_data(): # Function that runs when '/api' i
try:
# Open and read the JSON file from backend folder
with open('../backend/data.json', 'r') as f:
data = json.load(f) # Parse JSON data into Python list
return jsonify(data), 200 # Return the data as JSON with HTT
except Exception as e:
If any error occurs (file not found, invalid JSON, etc.), r
return jsonify({"error": str(e)}), 500
if __name__ == '__main__':
app.run(debug=True) # Run the Flask server in debug mo
[
{"id": 1, "name": "Kunal", "role": "Developer"},
{"id": 2, "name": "Riya", "role": "Designer"},
{"id": 3, "name": "Mihir", "role": "Manager"},
{"id": 3, "name": "Akhilesh", "role": "HR"}
]


Files
main
Go to file
flask_api_app
backend
frontend
README.md
Git_Task_1 / flask_api_app /
Add file
Kunal Sandeep Jadhav Add Flask API project
544ba46 · 5 minutes ago
History
Name
Last commit message
Last commit date
..
backend
Add Flask API project
5 minutes ago
frontend
Add Flask API project
5 minutes ago

2. Create a new branch named <your_name>_new (e.g., Tutedude_new).

. Update the content of the JSON file used for the / api route in this branch.

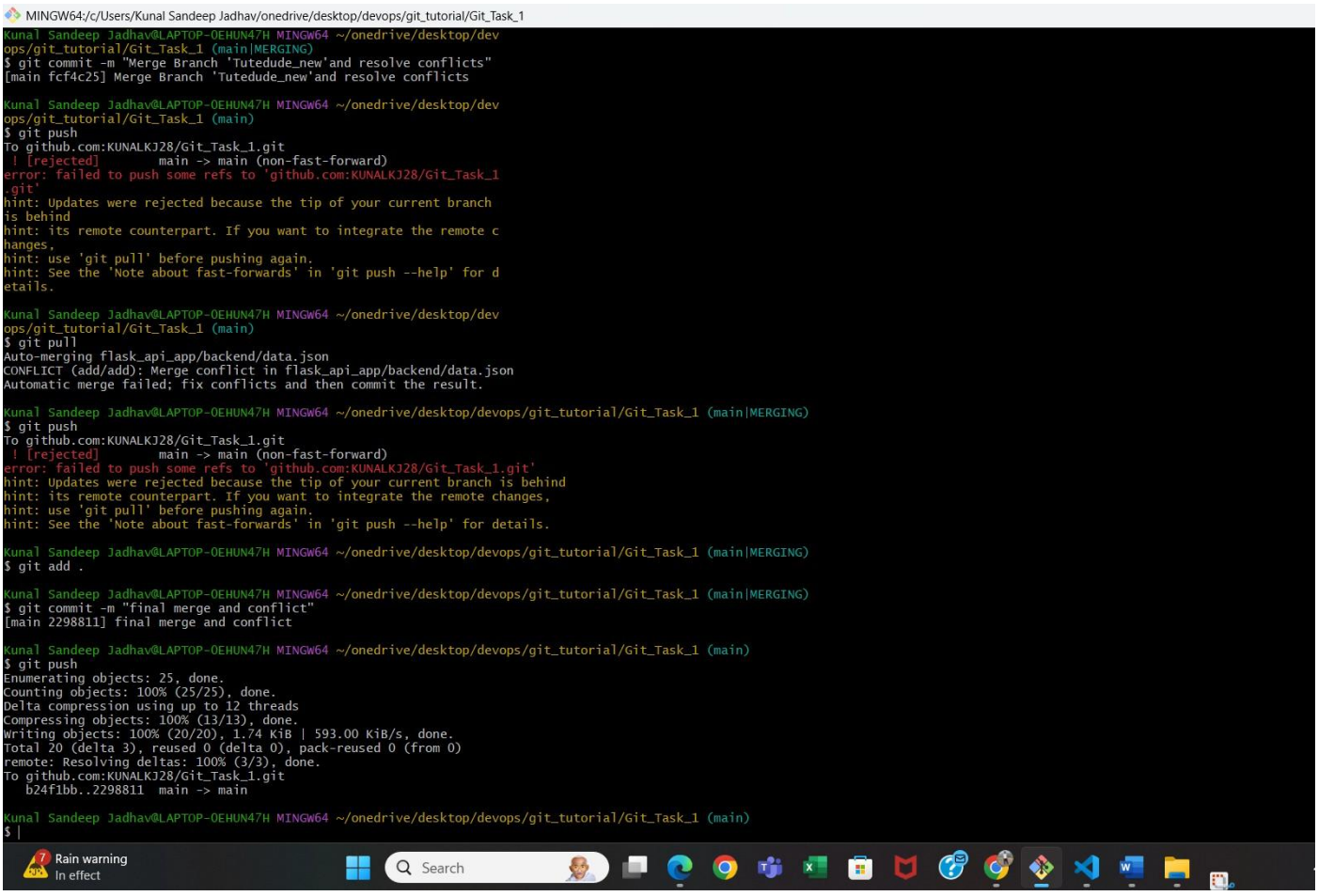
. Merge the <your_name>_new branch into the main branch.

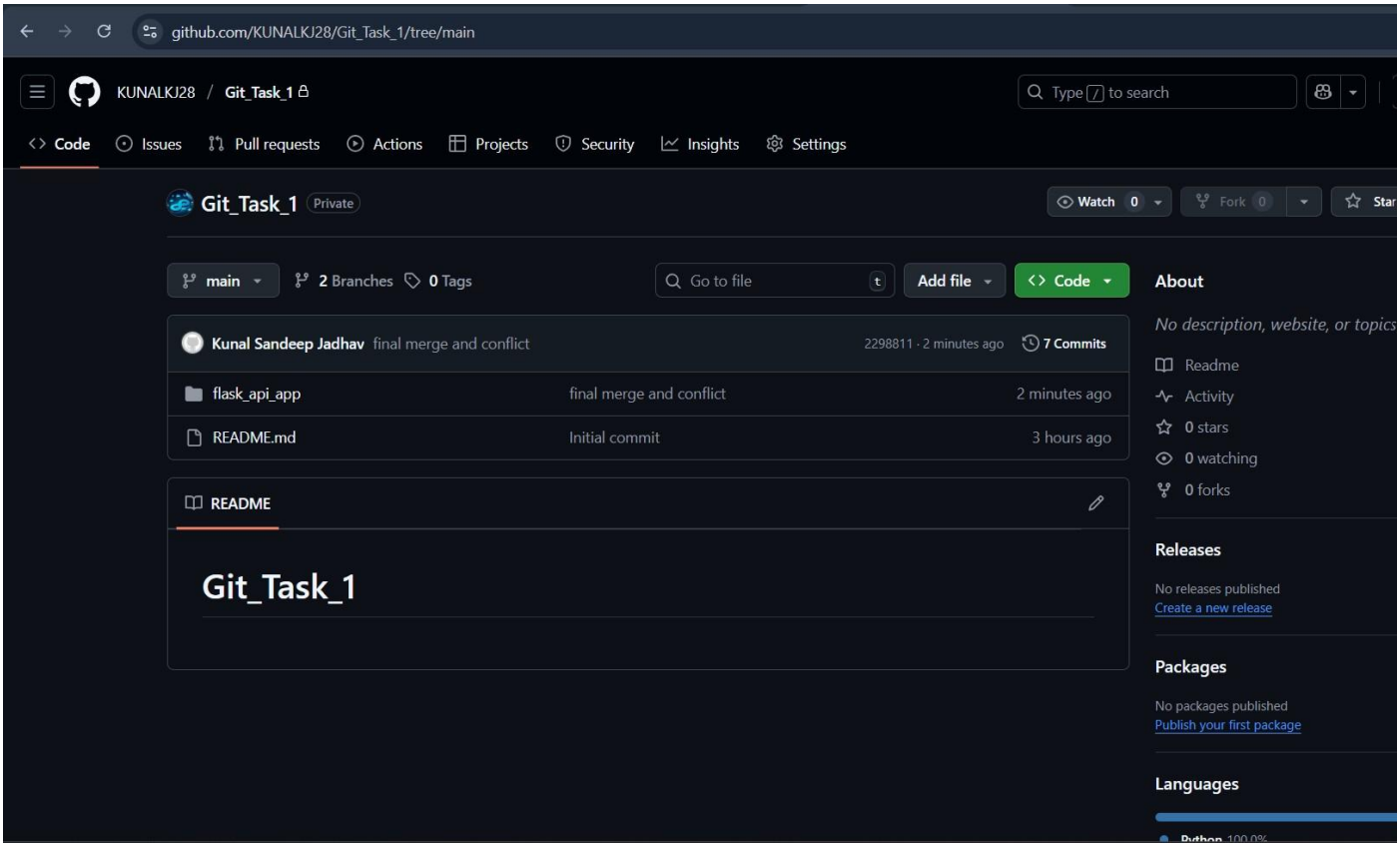
. If there are conflicts during the merge, resolve them by accepting the changes from the

<your_name> _new branch.

. Add the resolved changes to the staging area, commit them, and push the updates to

the remote repository.





3. Branch Creation:

. Create two branches: master_1 and master_2 from the main branch.

. Feature Development in master 1:

. In the master_1 branch, create a To-Do Page in the frontend.

o The page should contain a form with the following fields:

Item Name

. Item Description

. Backend API in master 2:

. In the master 2 branch, create a backend route named / submittodoitem.

. This route will:

o Accept itemName and itemDescription via a POST request.

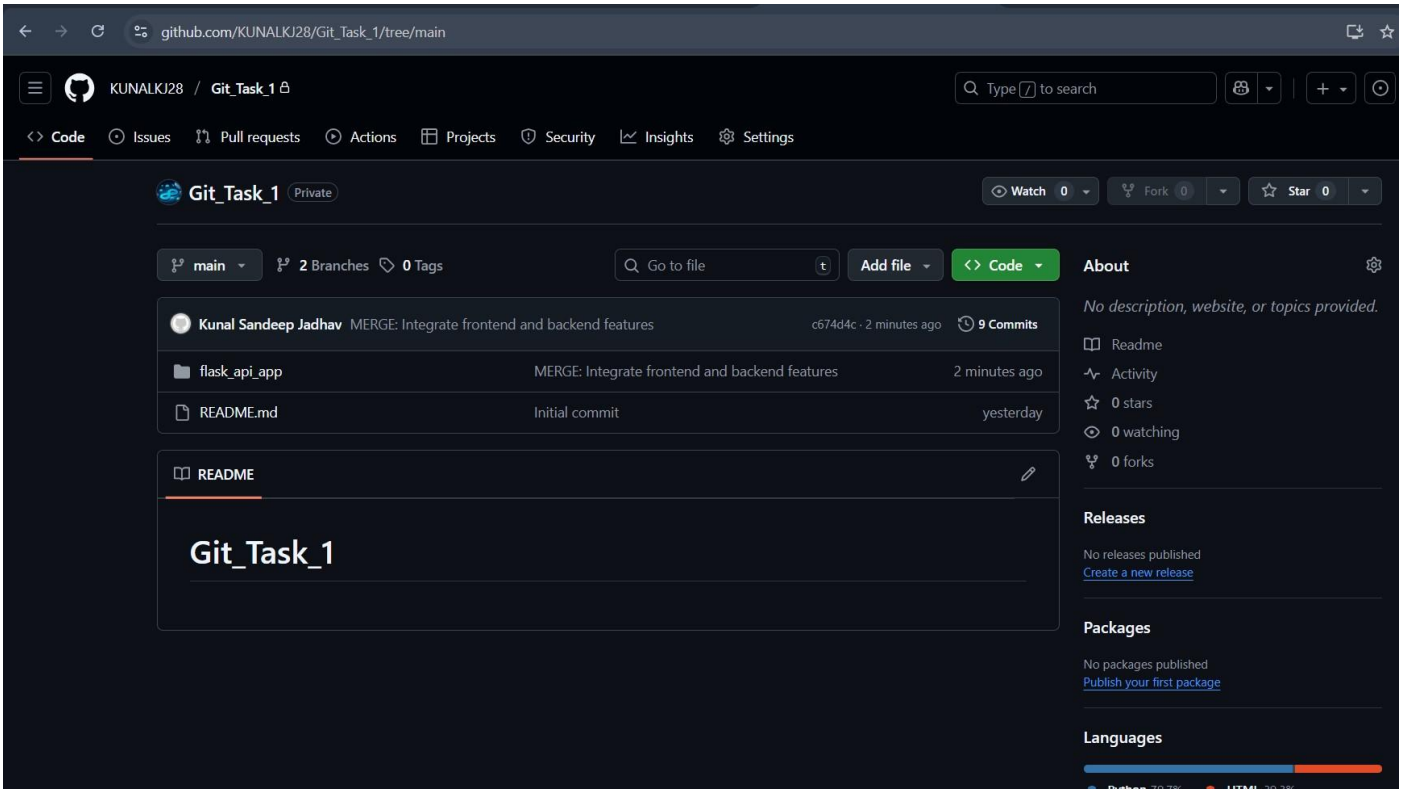
Store these details in a MongoDB database.

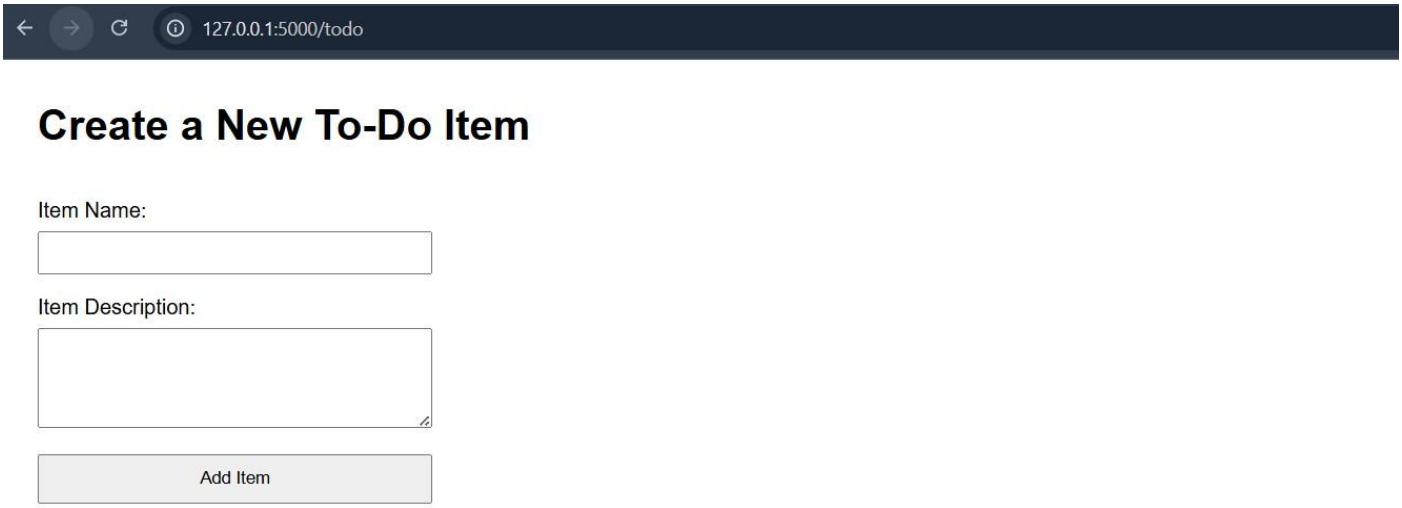
. Merging Changes:

. Merge the changes from both master_1 and master_2 into the main branch.

MINGW64:/c/Users/Kunal Sandeep Jadhav/onedrive/desktop/devops/git_tutorial/Git_Task_1

```
Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop
$ cd devops

Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops
$ cd git_tutorial

Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutor
ial
$ cd Git_Task_1

Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutor
ial/Git_Task_1 (master_2)
$ git switch main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutor
ial/Git_Task_1 (main)
$ git merge master_1
Updating 2298811..36f16f3
Fast-forward
 flask_api_app/frontend/app.py             |  6 +++++-
 flask_api_app/frontend/template/todo.html | 26 ++++++++++++++++++++++++++
 2 files changed, 31 insertions(+), 1 deletion(-)
 create mode 100644 flask_api_app/frontend/template/todo.html

Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutor
ial/Git_Task_1 (main)
$ git merge master_2
Already up to date.

Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutor
ial/Git_Task_1 (main)
$ git add .

Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutor
ial/Git_Task_1 (main)
$ git commit -m "MERGE: Integrate frontend and backend features"
[main c674d4c] MERGE: Integrate frontend and backend features
 1 file changed, 32 insertions(+)
 create mode 100644 flask_api_app/backend/app.py

Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutor
ial/Git_Task_1 (main)
$ git push
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 12 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (12/12), 1.95 KiB | 284.00 KiB/s, done.
Total 12 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:KUNALKJ28/Git_Task_1.git
   2298811..c674d4c  main -> main

Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutor
ial/Git_Task_1 (main)
$
```

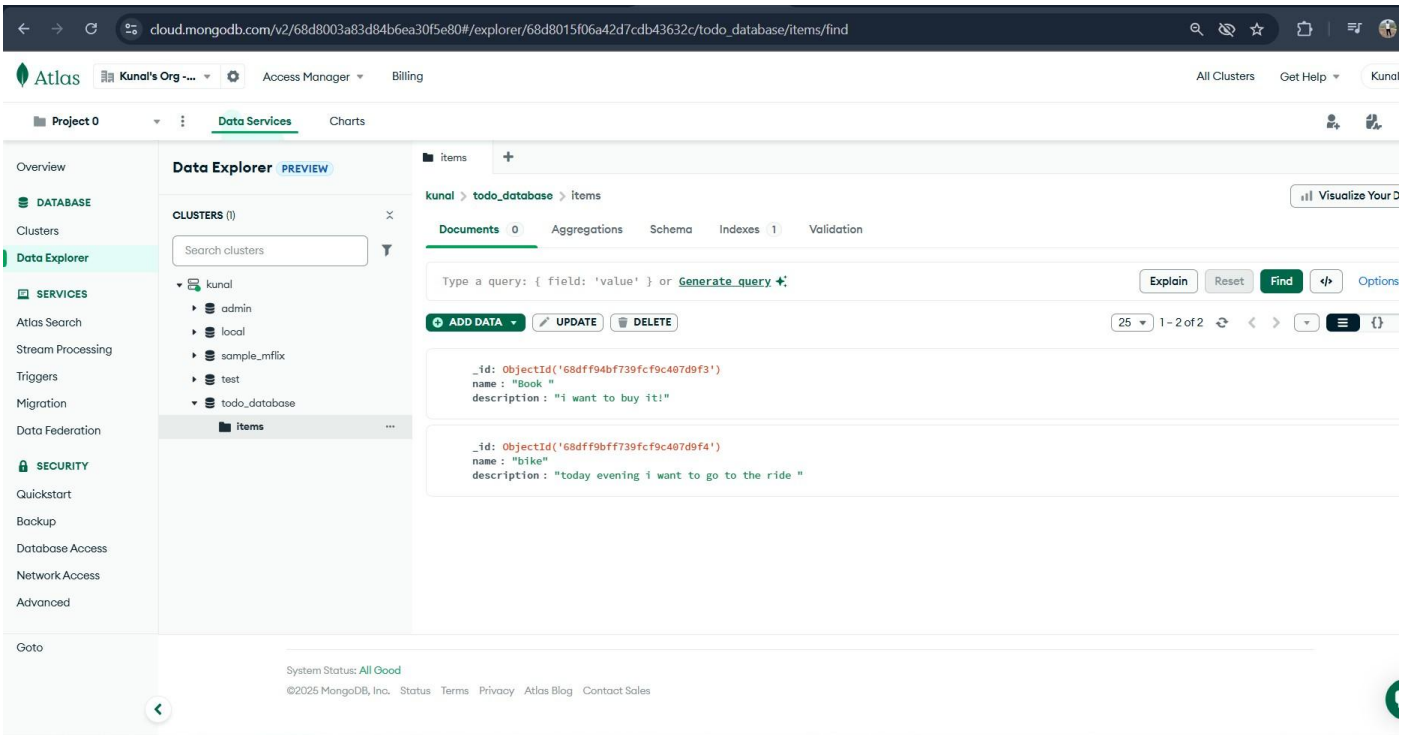


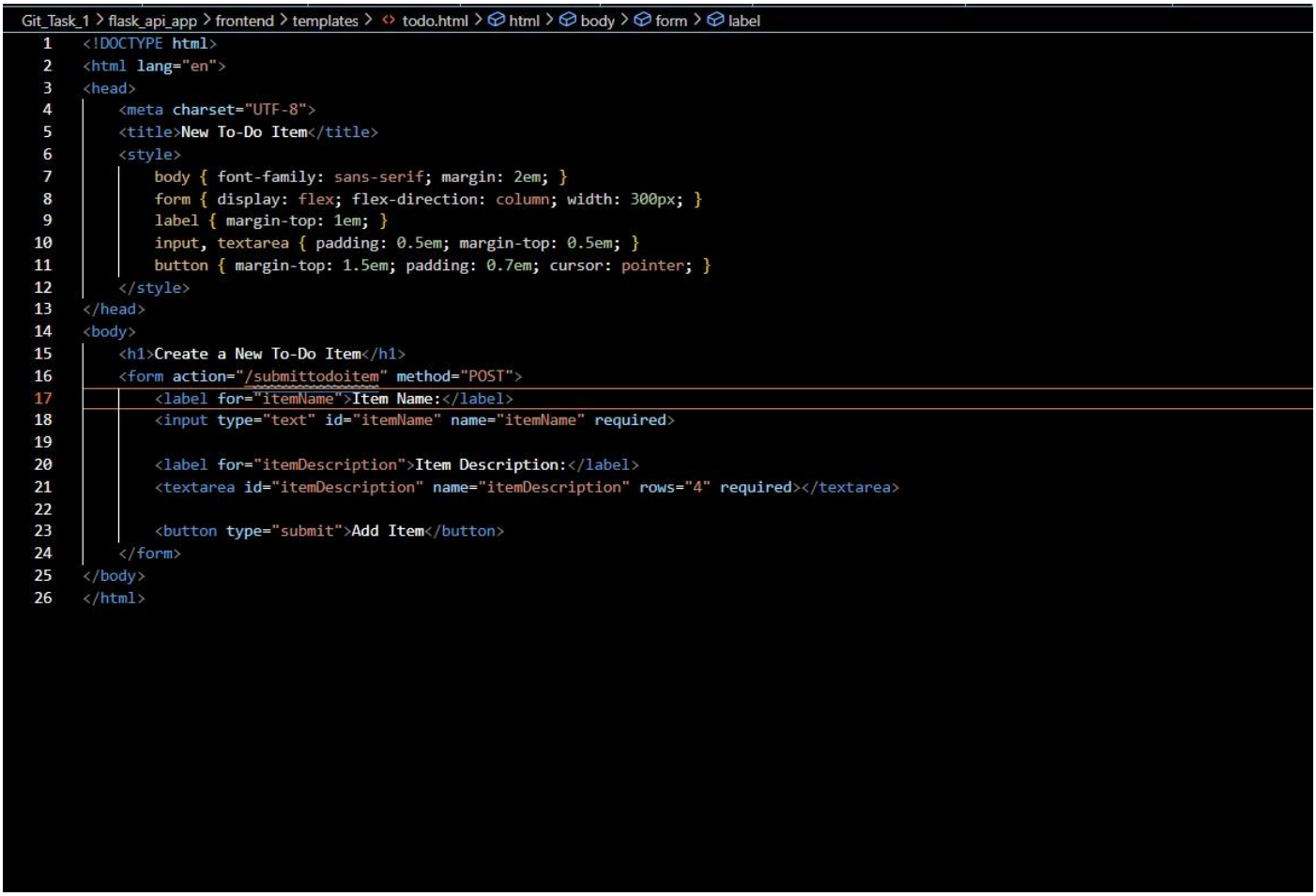


Git_Task_1 > flask_api_app > backend > app.py > submit_todo_item

```python
# ==================================
#         1. IMPORTS
# ==================================
from flask import Flask, jsonify, render_template, request, redirect, url_for
from pymongo import MongoClient
from dotenv import load_dotenv
import os
import json

# ==================================
#     2. APP & DATABASE SETUP
# ==================================
app = Flask(__name__, template_folder='../frontend/templates')

# Load environment variables
load_dotenv()
MONGO_URI = os.getenv("MONGO_URI")

# MongoDB Connection
client = MongoClient(MONGO_URI)
db = client['todo_database']
collection = db['items']

# ==================================
#           3. ROUTES
# ==================================
@app.route('/')
def home():
    return redirect(url_for('todo_page'))

@app.route('/todo')
def todo_page():
    return render_template('todo.html')

@app.route('/submittodoitem', methods=['POST'])
def submit_todo_item():
    try:
        itemName = request.form['itemName']
        itemDescription = request.form['itemDescription']

        todo_item = {
            "name": itemName,
            "description": itemDescription
        }
```

```python
        collection.insert_one(todo_item)
        return redirect(url_for('todo_page'))
    except Exception as e:
        return jsonify({"error": str(e)}), 500

@app.route('/api', methods=['GET'])
def get_data():
    try:
        with open('data.json', 'r') as f:
            data = json.load(f)
        return jsonify(data), 200
    except Exception as e:
        return jsonify({"error": str(e)}), 500

# ==================================
#        4. RUN THE SERVER
# ==================================
if __name__ == '__main__':
    app.run(debug=True, use_reloader=False)
```

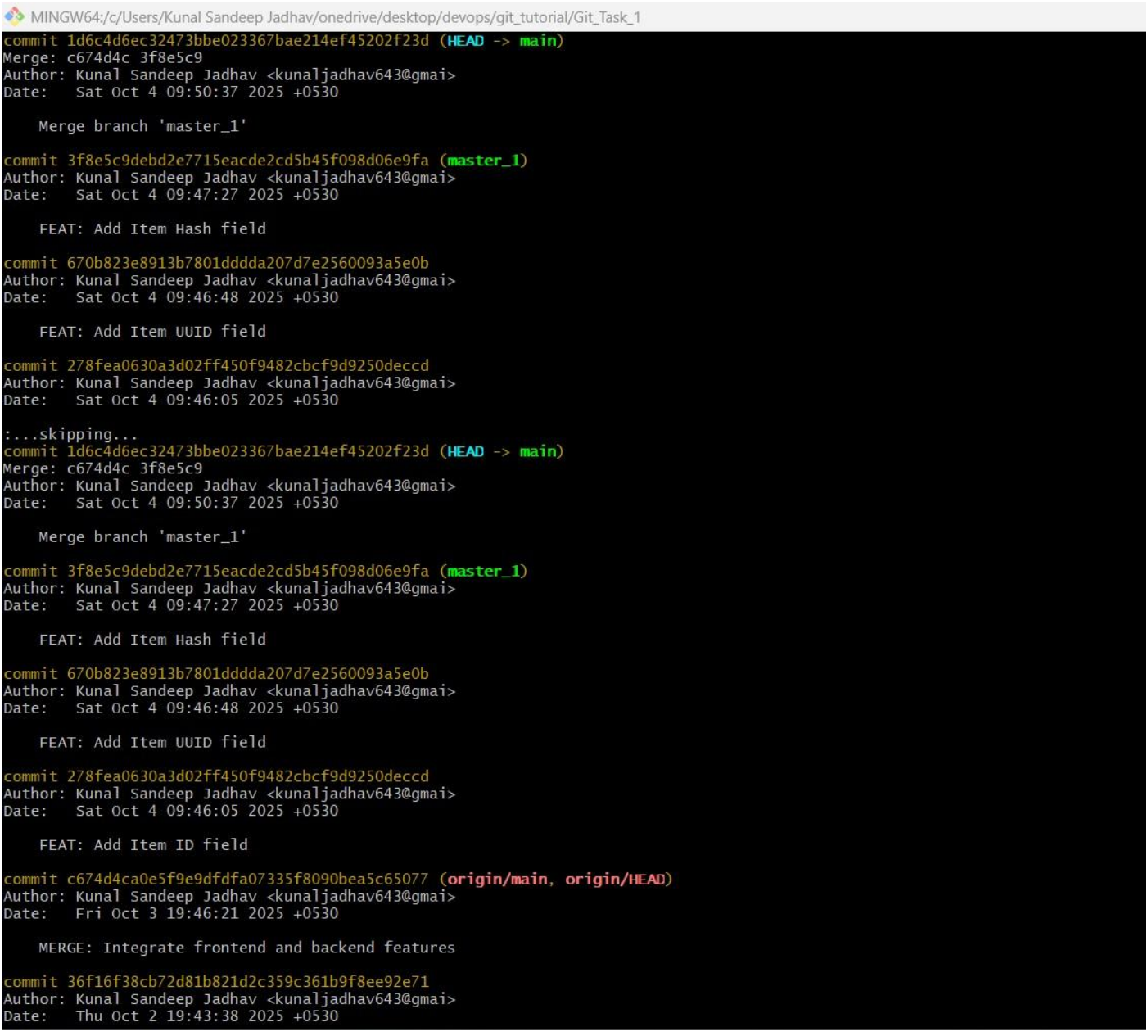




4. **Enhancing the To-Do Form in** master_1:

- In the master_1 branch, add the following fields to the To-Do form:
    - **Item ID**
    - **Item UUID**
    - **Item Hash**
- **Committing in Sequence**:
- Add and commit each field separately in the following order:
    - **First commit**: Add **Item ID** field.
    - **Second commit**: Add **Item UUID** field.
    - **Third commit**: Add **Item Hash** field.
- **Merging to** main:
- Merge the master_1 branch into the main branch.
- **Git Reset and Commit Deletion**:
- In the main branch, use **Git Reset** to roll back to the commit where only the **Item ID** field was added.
- Use git reset --soft to ensure changes remain staged.
- Re-commit this state to the main branch.
- Merge this updated state to the main branch.

- **Rebasing Changes**:
- Rebase the updated changes in the main branch to the master_1 branch.
  Clarification:
    - During rebasing, **preserve individual commits** to maintain the commit history for each change (i.e., do not squash commits).
    - Use git rebase main master_1 to integrate changes from the main branch back into the master_1 branch.



MINGW64:/c/Users/Kunal Sandeep Jadhav/onedrive/desktop/devops/git_tutorial/Git_Task_1

```
commit 36f16f38cb72d81b821d2c359c361b9f8ee92e71
Author: Kunal Sandeep Jadhav <kunaljadhav643@gmai>
Date:   Thu Oct 2 19:43:38 2025 +0530

    FEAT:Add frontend To-DO page

commit 2298811af330e4f290a98a478bad036f3c09e11c (master_2)
Merge: fcf4c25 b24f1bb
Author: Kunal Sandeep Jadhav <kunaljadhav643@gmai>
Date:   Thu Oct 2 19:11:59 2025 +0530

    final merge and conflict

commit fcf4c253f1955da6a6b0e5517f1300c70baae4e1
Merge: 2c14ecc 4be6851
Author: Kunal Sandeep Jadhav <kunaljadhav643@gmai>
Date:   Thu Oct 2 19:06:33 2025 +0530

    Merge Branch 'Tutedude_new'and resolve conflicts

commit 4be6851abb81620ac14254b917ddd920ea72f55e (Tutedude_new)
Author: Kunal Sandeep Jadhav <kunaljadhav643@gmai>
Date:   Thu Oct 2 18:59:40 2025 +0530

    Update Json


Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutorial/Git_Task_1 (main)
$ git reset --soft 273f6c0
fatal: ambiguous argument '273f6c0': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutorial/Git_Task_1 (main)
$ git reset --soft master_1~2

Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutorial/Git_Task_1 (main)
$ git status
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 1 different commits each, respectively.
  (use "git pull" if you want to integrate the remote branch with yours)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   flask_api_app/backend/app.py
        modified:   flask_api_app/frontend/templates/todo.html


Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutorial/Git_Task_1 (main)
$ git commit -m "FEAT: Add Item UUID and Hash fields"
[main a5f481a] FEAT: Add Item UUID and Hash fields
 2 files changed, 36 insertions(+)
 create mode 100644 flask_api_app/backend/app.py

Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutorial/Git_Task_1 (main)
$ git push
```

MINGW64:/c/Users/Kunal Sandeep Jadhav/onedrive/desktop/devops/git_tutorial/Git_Task_1

```
Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutorial/Git_Task_1 (main)
$ git commit -m "FEAT: Add Item UUID and Hash fields"
[main a5f481a] FEAT: Add Item UUID and Hash fields
 2 files changed, 36 insertions(+)
 create mode 100644 flask_api_app/backend/app.py

Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutorial/Git_Task_1 (main)
$ git push
To github.com:KUNALKJ28/Git_Task_1.git
 ! [rejected]        main -> main (non-fast-forward)
error: failed to push some refs to 'github.com:KUNALKJ28/Git_Task_1.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. If you want to integrate the remote changes,
hint: use 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutorial/Git_Task_1 (main)
$ git push origin main --force
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 12 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (16/16), 2.22 KiB | 98.00 KiB/s, done.
Total 16 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:KUNALKJ28/Git_Task_1.git
 + c674d4c...a5f481a main -> main (forced update)

Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutorial/Git_Task_1 (main)
$ git checkout master_1
Switched to branch 'master_1'

Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutorial/Git_Task_1 (master_1)
$ git rebase main
Auto-merging flask_api_app/frontend/templates/todo.html
CONFLICT (content): Merge conflict in flask_api_app/frontend/templates/todo.html
error: could not apply 670b823... FEAT: Add Item UUID field
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
hint: Disable this message with "git config set advice.mergeConflict false"
Could not apply 670b823... # FEAT: Add Item UUID field

Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutorial/Git_Task_1 (master_1|REBASE 1/2)
$ git add .

Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutorial/Git_Task_1 (master
_1|REBASE 1/2)
$ git rebase --continue
dropping 3f8e5c9debd2e7715eacde2cd5b45f098d06e9fa FEAT: Add Item Hash field -- patch contents already
upstream
Successfully rebased and updated refs/heads/master_1.

Kunal Sandeep Jadhav@LAPTOP-0EHUN47H MINGW64 ~/onedrive/desktop/devops/git_tutorial/Git_Task_1 (master
_1)
$
```

27°C
Mostly sunny
Search

name.txt | index.html M | recape.txt U | data.json | todo.html | app.py ...\templates | app.py ...\backend | .env

Git_Task_1 > flask_api_app > frontend > templates > todo.html > html > body > form

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>New To-Do Item</title>
    <style>
        body { font-family: sans-serif; margin: 2em; }
        form { display: flex; flex-direction: column; width: 300px; }
        label { margin-top: 1em; }
        input, textarea { padding: 0.5em; margin-top: 0.5em; }
        button { margin-top: 1.5em; padding: 0.7em; cursor: pointer; }
    </style>
</head>
<body>
    <h1>Create a New To-Do Item</h1>
    <form action="/submittodoitem" method="POST">
        <label for="itemId">Item ID:</label>
        <input type="text" id="itemId" name="itemId" required>
        <label for="itemUUID">Item UUID:</label>
        <input type="text" id="itemUUID" name="itemUUID" required>
        <label for="itemHash">Item Hash:</label>
        <input type="text" id="itemHash" name="itemHash" required>
        <label for="itemName">Item Name:</label>
        <input type="text" id="itemName" name="itemName" required>

        <label for="itemDescription">Item Description:</label>
        <textarea id="itemDescription" name="itemDescription" rows="4" required></textarea>

        <button type="submit">Add Item</button>
    </form>
</body>
</html>
```

KUNALKJ28 / Git_Task_1

Code | Issues | Pull requests | Actions | Projects | Security | Insights | Settings

Git_Task_1 Private

Watch 0

main | 2 Branches | 0 Tags | Go to file t | Add file | Code

Kunal Sandeep Jadhav FEAT: Add Item UUID and Hash fields a5f481a · 10 minutes ago 10 Commits

flask_api_app	FEAT: Add Item UUID and Hash fields	10 minutes ago
README.md	Initial commit	2 days ago

README

Git_Task_1