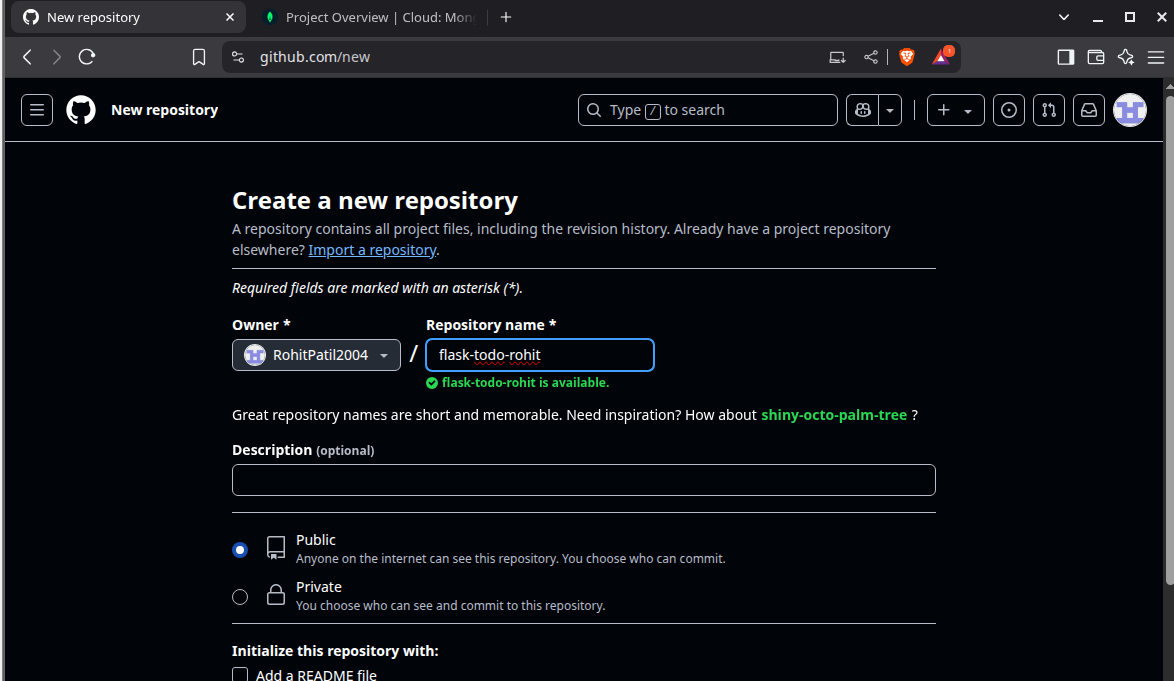
1. Initial GitHub Setup
   1. Create a new GitHub Repository

Go to GitHub -> New Repository -> Name : flask-todo-rohit -> Create



* 1. Clone Repository via SSH

# Generate SSH key (if not already generated)

ssh-keygen -t rsa -b 4096 -C "rohitvpatil04@gmail.com "

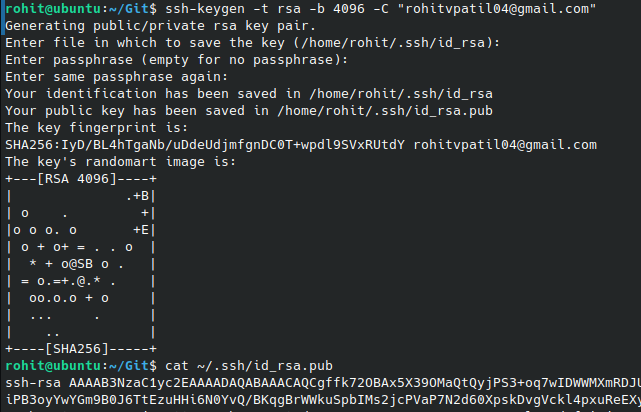
# Add public key to GitHub (in SSH settings)

cat ~/.ssh/id\_rsa.pub

# Clone the repo

git clone git@github.com:RohitPatil2004/flask-todo-rohit.git

cd flask-todo-rohit





* 1. Create and switch to a new branch:

git checkout -b RohitPatil



* 1. Add Flask Project Files

# Assuming you have your Flask app ready in a directory

cp -r /path/to/flask\_project/\* .

git add .

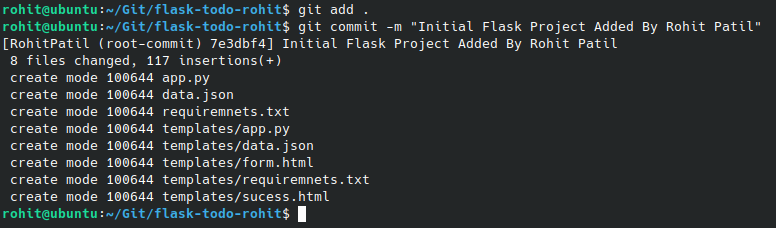
git commit -m "Initial Flask project added by Rohit Patil"

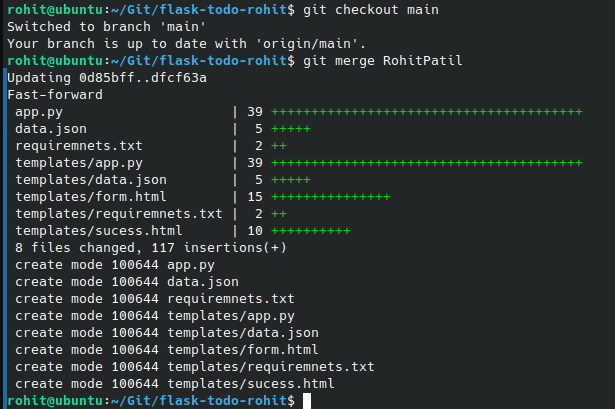
git checkout main

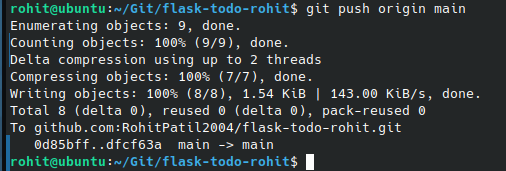
git merge RohitPatil

git push origin main









1. JSON Update Flow
   1. Create a new branch

git checkout -b RohitPatil\_new

* 1. Update the JSON used by /api route

# Example JSON update in data.json

{

"todo": [

{

"id": 1,

"task": "Write backend",

"status": "pending"

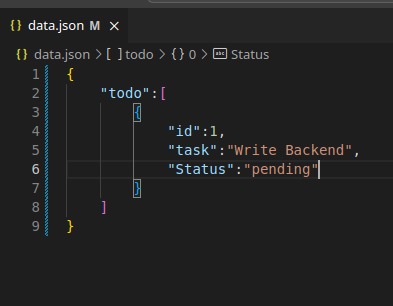
}

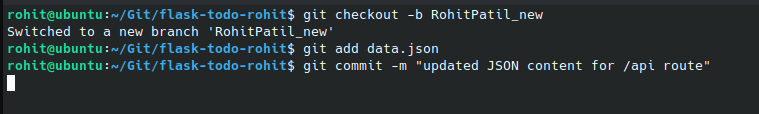
]

}

git add data.json

git commit -m "Updated JSON content for /api route"





* 1. Merge and resolve conflicts if needed

git checkout main

git merge RohitPatil\_new

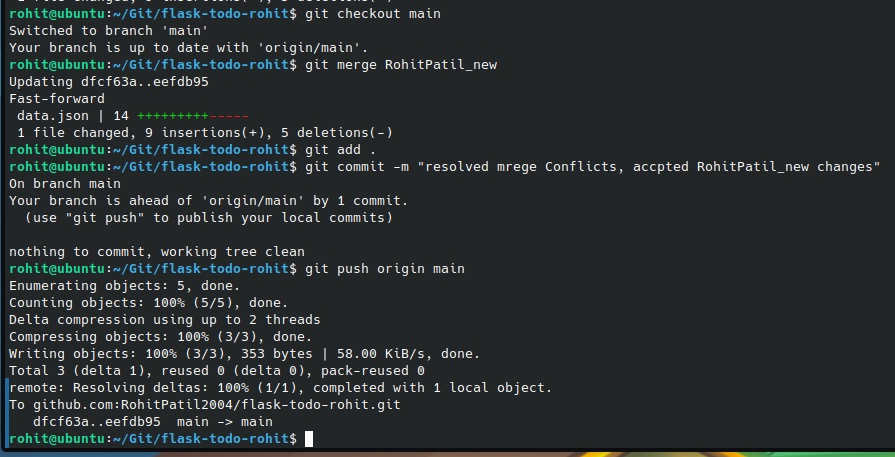
# If conflict:

# Edit file manually to accept RohitPatil\_new changes

git add .

git commit -m "Resolved merge conflicts, accepted RohitPatil\_new changes"

git push origin main



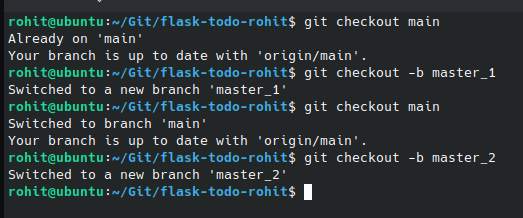
1. Branching And Feature Development
   1. Create two branches

git checkout main

git checkout -b master\_1

git checkout main

git checkout -b master\_2



* 1. Feature : To – Do Form in Master\_1

HTML Form in templates/todo.html

<form method="POST" action="/submit">

<input type="text" name="itemName" placeholder="Item Name"><br>

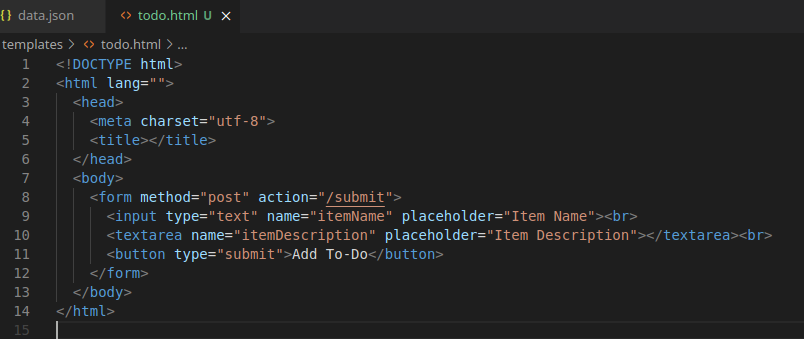
<textarea name="itemDescription" placeholder="Item Description"></textarea><br>

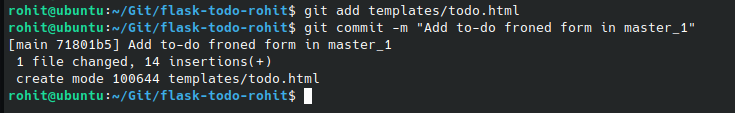
<button type="submit">Add To-Do</button>

</form>

git add templates/todo.html

git commit -m "Add To-Do frontend form in master\_1"





* 1. Backend route in master\_2

@app.route('/submittodoitem', methods=['POST'])

def submit\_item():

data = request.get\_json()

item\_name = data.get('itemName')

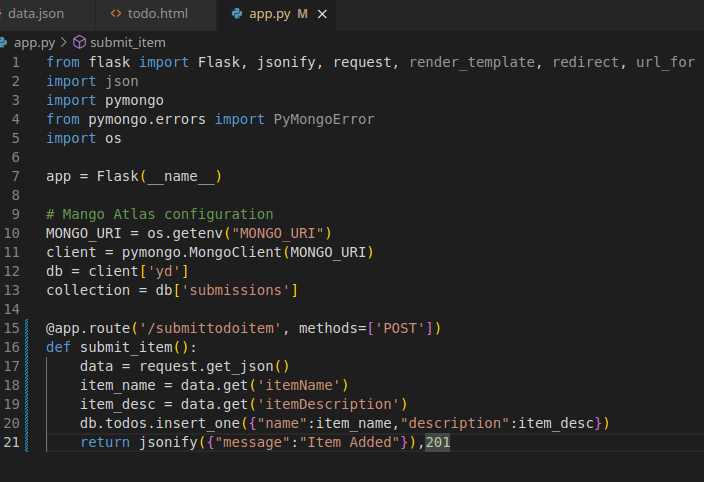
item\_desc = data.get('itemDescription')

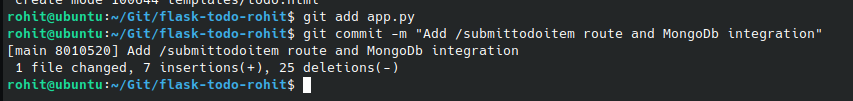
db.todos.insert\_one({"name": item\_name, "description": item\_desc})

return jsonify({"message": "Item added"}), 201

git add app.py

git commit -m "Add /submittodoitem route and MongoDB integration"





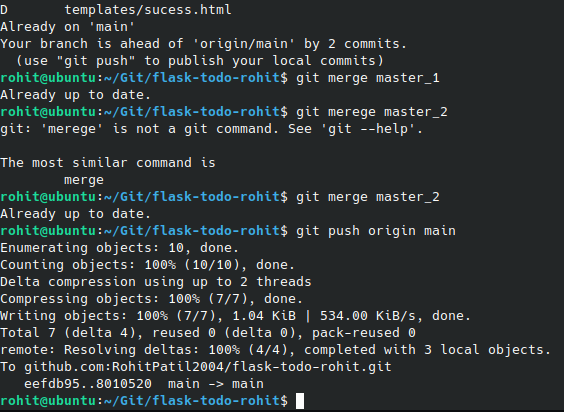
* 1. Merge both branches into main

git checkout main

git merge master\_1

git merge master\_2

git push origin main

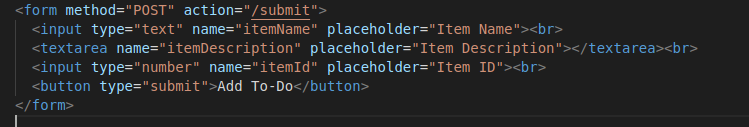


1. Enhance To-Do Form
   1. In master\_1, add Item ID

<input type="number" name="itemId" placeholder="Item ID"><br>

git add templates/todo.html

git commit -m "Add Item ID field"



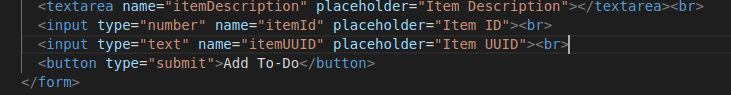


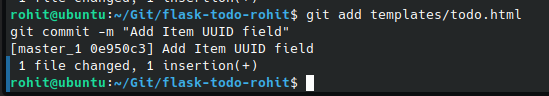
* 1. Add Item UUID

<input type="text" name="itemUUID" placeholder="Item UUID"><br>

git add templates/todo.html

git commit -m "Add Item UUID field"



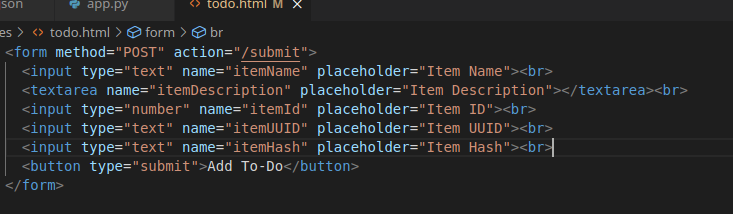


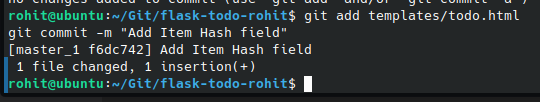
* 1. Add Item Hash

<input type="text" name="itemHash" placeholder="Item Hash"><br>

git add templates/todo.html

git commit -m "Add Item Hash field"



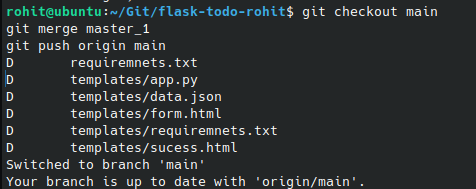


* 1. Merge to main

git checkout main

git merge master\_1

git push origin main



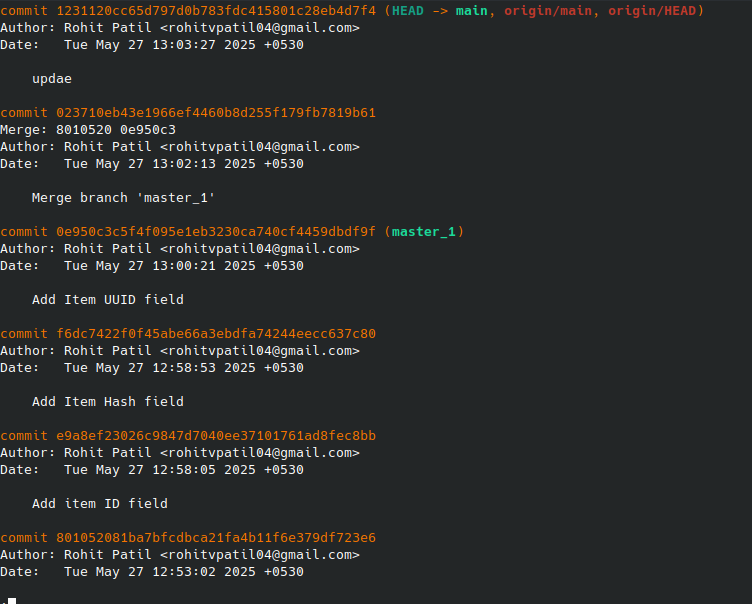
1. Git Reset and Re-Commit
   1. Reset to Commit with only Item ID:

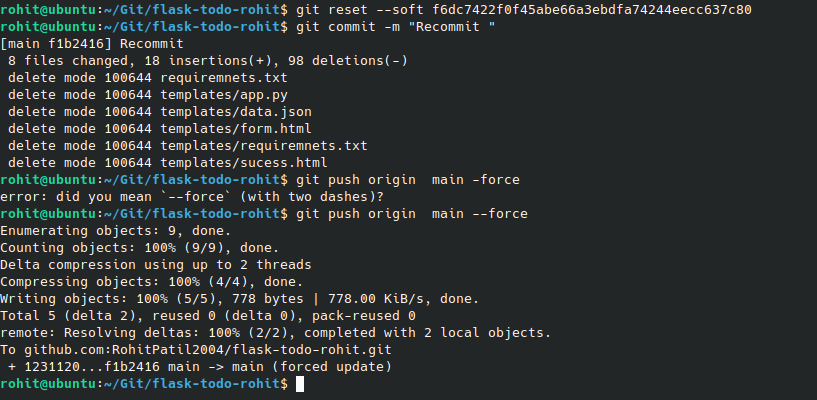
git log # Find commit hash with "Add Item ID field"

git reset --soft f6dc7422f0f45abe66a3ebdfa74244eecc637c80

git commit -m "Recommit "

git push origin main –force





1. Rebasing to master\_1

git checkout master\_1

git rebase main

