Assignment 4 - Git and GitHub

# 1. Creating a New GitHub Repository

Explanation:  
 This step sets up a new GitHub repository and a working branch for development.  
  
 Commands:  
 1. Clone the repository using SSH:  
 git clone git@github.com:yourusername/your-repo.git  
  
 2. Navigate to the repo:  
 cd your-repo  
  
 3. Create a new branch named after your username (e.g., tutedude):  
 git checkout -b tutedude  
  
 4. Add Flask project files:  
 (Place your project files into the cloned folder)  
  
 5. Stage and commit the changes:  
 git add .  
 git commit -m "Added Flask project files"  
  
 6. Merge your branch into main:  
 git checkout main  
 git merge tutedude  
  
 7. Push changes to GitHub:  
 git push origin main

# 2. Creating and Merging a New Branch

Explanation:  
 You create a new branch to modify the JSON data used in the `/api` route, then merge those changes into the main branch.  
  
 Commands:  
 1. Create a new branch:  
 git checkout -b yourname\_new  
  
 2. Update the JSON file used for the /api route  
  
 3. Stage and commit the changes:  
 git add data.json  
 git commit -m "Updated JSON data for /api route"  
  
 4. Switch to main and merge:  
 git checkout main  
 git merge yourname\_new  
  
 5. If there are conflicts:  
 - Resolve conflicts in your text editor  
     - Then run:  
   git add .  
   git commit -m "Resolved merge conflicts"  
  
 6. Push the updated main branch:  
 git push origin main

# 3. Branch Creation for To-Do Feature

Explanation:  
 You are creating a feature split into two branches — frontend in master\_1 and backend API in master\_2.  
  
 Commands:  
 1. Create two branches from main:  
 git checkout main  
 git checkout -b master\_1  
 git checkout main  
 git checkout -b master\_2  
  
 Frontend (master\_1):  
 - Create a form in the frontend with:  
 - Item Name  
 - Item Description  
  
 Backend (master\_2):  
 - Create a Flask route /submittodoitem that accepts POST requests and stores the data in MongoDB.  
  
 Merging to main:  
 1. git checkout main  
 2. git merge master\_1  
 3. git merge master\_2  
  
 4. Push to remote:  
 git push origin main

# 4. Enhancing the To-Do Form in master\_1

Explanation:

Enhance the To-Do form by adding three fields: Item ID, UUID, and Hash. Each should be added and committed separately.

Commands:

1. Switch to master\_1:

git checkout master\_1

2. Add Item ID:

# Add field in code

git add .

git commit -m "Added Item ID field"

3. Add Item UUID:

# Add field in code

git add .

git commit -m "Added Item UUID field"

4. Add Item Hash:

# Add field in code

git add .

git commit -m "Added Item Hash field"

5. Merge into main:

git checkout main

git merge master\_1

git push origin main

6. Git Reset (if needed):

# To reset to commit with only Item ID:

git log  # copy the commit ID of desired commit

git reset --soft <commit\_id>

git commit -m "Recommitting from reset state"

git push origin main --force

7. Rebasing (optional cleanup):

git checkout master\_1

git rebase main

Repo Link : <https://github.com/BabluuMemes20005/Tutude-Devops-Assignment>