Git Workflow Tasks Documentation

# 1. Initial Repository Setup

- Create a GitHub repository.  
- Clone it using SSH: `git clone git@github.com:<username>/<repo>.git`  
- Create SSH key (if needed): `ssh-keygen -t rsa -b 4096 -C "your\_email@example.com"` and add public key to GitHub.  
- Create and checkout a branch: `git checkout -b <username>`  
- Add Flask project files: `git add . && git commit -m "Initial commit"`  
- Merge branch to main: `git checkout main && git merge <username> && git push`

# 2. JSON Update Branch

- Create branch: `git checkout -b <your\_name>\_new`  
- Update the JSON file (`data.json`) used by `/api` route.  
- Commit changes: `git add data.json && git commit -m "Updated API data"`  
- Merge to main: `git checkout main && git merge <your\_name>\_new`  
- If conflicts occur, resolve in favor of `<your\_name>\_new` branch, then:  
 ```  
 git add .  
 git commit -m "Resolved merge conflict with <your\_name>\_new"  
 git push  
 ```

# 3. Feature Development Using Branches

- Create branches:  
 ```  
 git checkout main  
 git checkout -b master\_1  
 git checkout main  
 git checkout -b master\_2  
 ```  
  
- In `master\_1`, create frontend To-Do page (with Item Name, Item Description).  
- In `master\_2`, create `/submittodoitem` backend POST route to accept and store the fields into MongoDB.  
  
- Merge both branches into main:  
 ```  
 git checkout main  
 git merge master\_1  
 git merge master\_2  
 git push  
 ```

# 4. To-Do Form Enhancements and Git Reset

- In `master\_1`, add these fields one by one and commit separately:  
 1. `Item ID` – `git add . && git commit -m "Add Item ID"`  
 2. `Item UUID` – `git add . && git commit -m "Add Item UUID"`  
 3. `Item Hash` – `git add . && git commit -m "Add Item Hash"`  
  
- Merge `master\_1` into main:  
 ```  
 git checkout main  
 git merge master\_1  
 ```  
  
- Git Reset:  
 ```  
 git log # Get commit hash of "Add Item ID"  
 git reset --soft <commit\_hash>  
 git commit -m "Revert to only Item ID commit"  
 git push --force  
 ```  
  
- Rebase back to master\_1:  
 ```  
 git checkout master\_1  
 git rebase main  
 ```  
  
- During rebase, preserve commit history (do not squash).