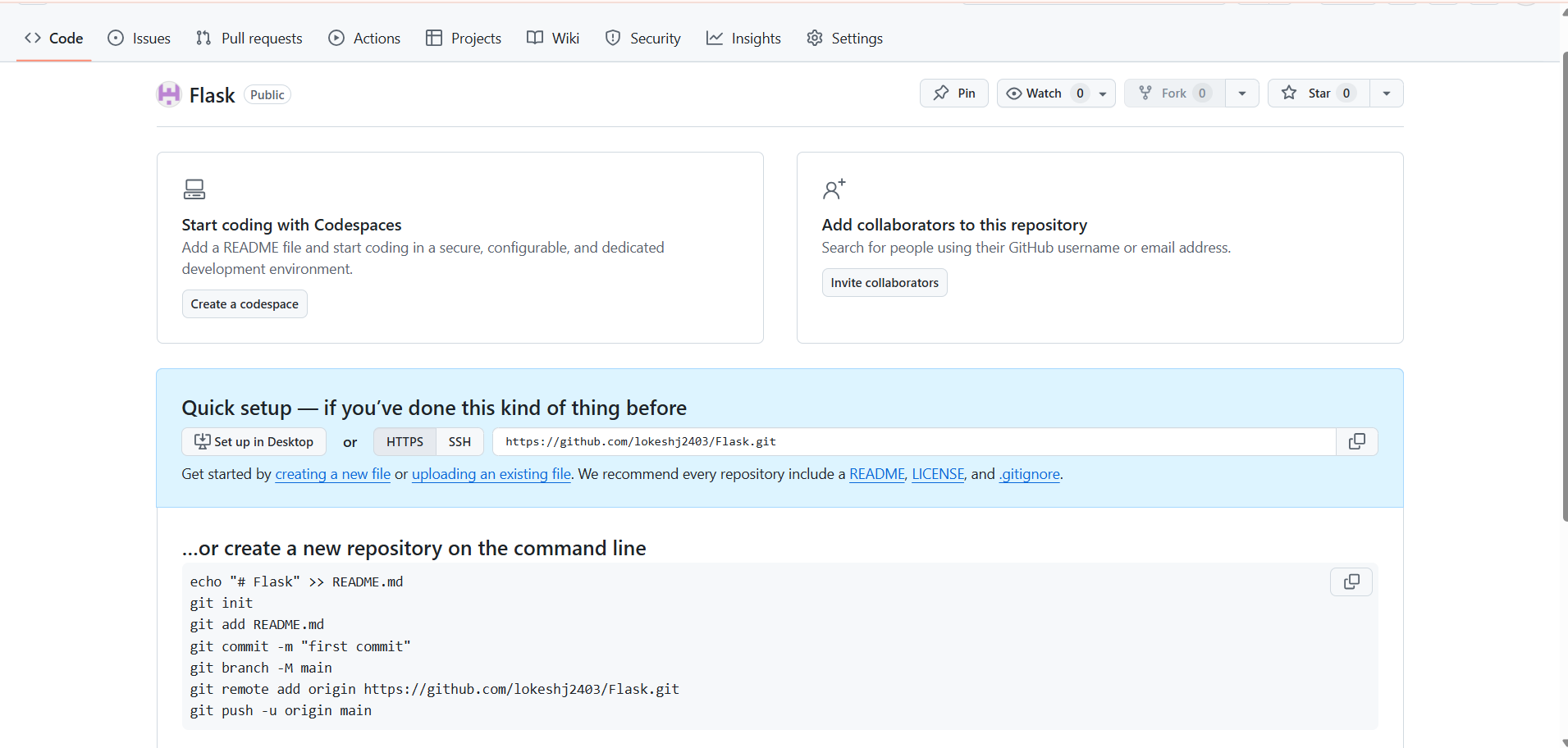
# GitHub Workflow Assignment Report

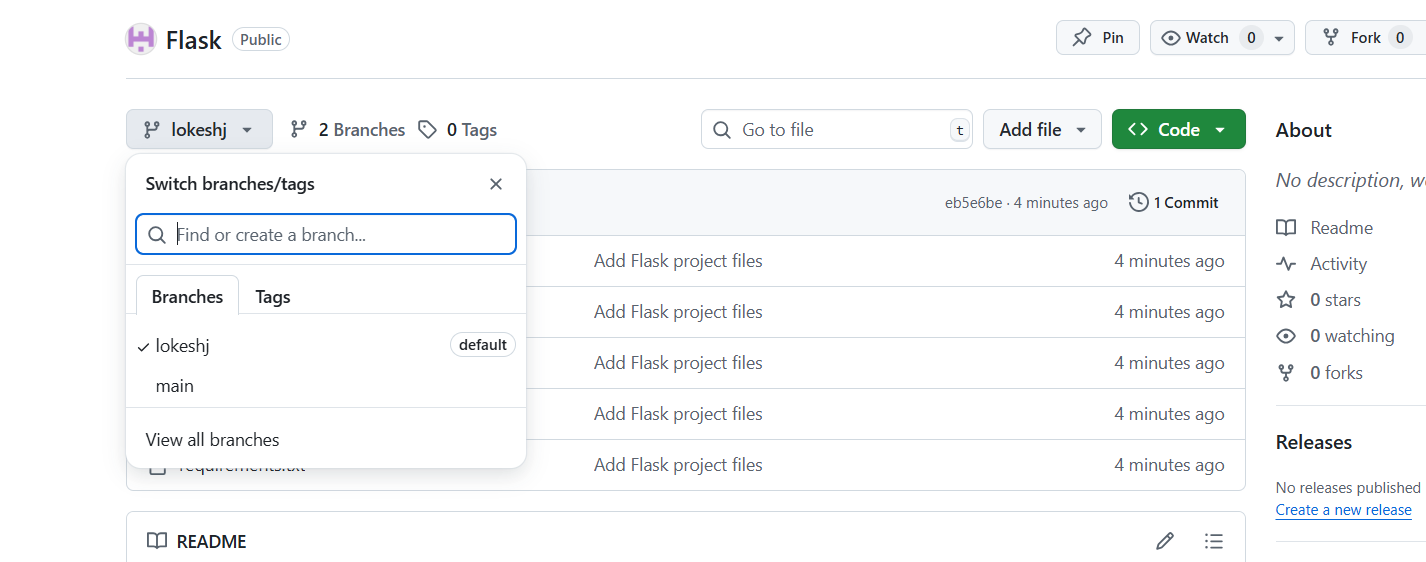
## 1. Introduction

This report documents the process and commands used to complete the GitHub workflow assignment. It covers repository creation, branching strategies, conflict resolution, merging, rebasing, and integration of both frontend and backend features into a Flask project.

## 2. Step 1: Repository Creation and Initial Branch Setup

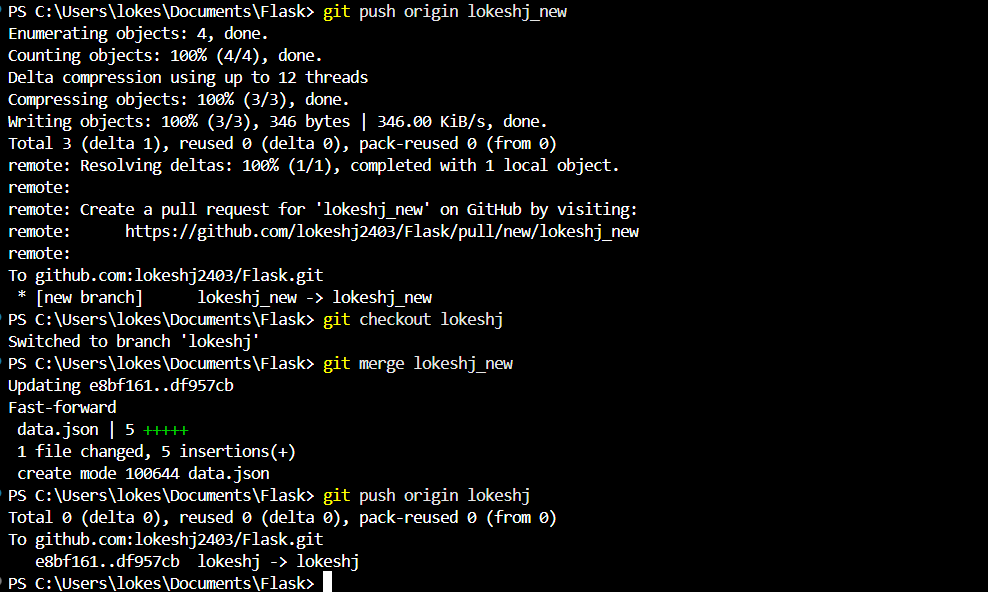
1. Created a new GitHub repository.(Flask)  
2. Cloned the repository to the local machine using SSH.  
3. Created a new branch named after the username (lokeshj).  
4. Added Flask project files to this branch.  
5. Committed the changes and merged the branch into the lokeshj branch.





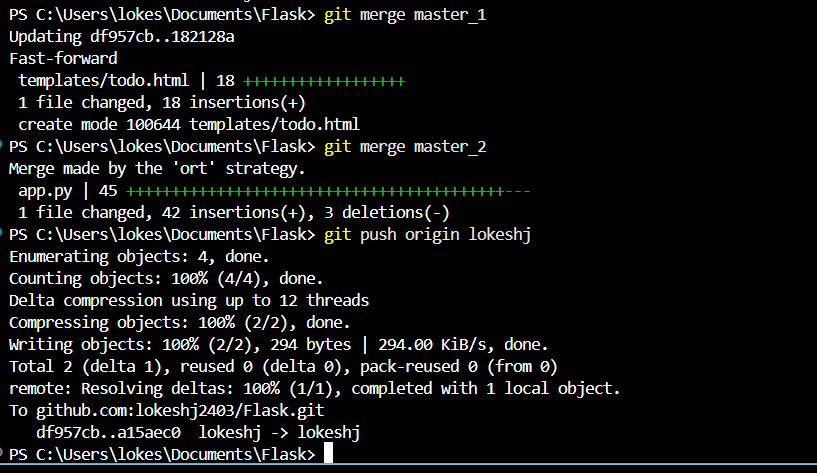
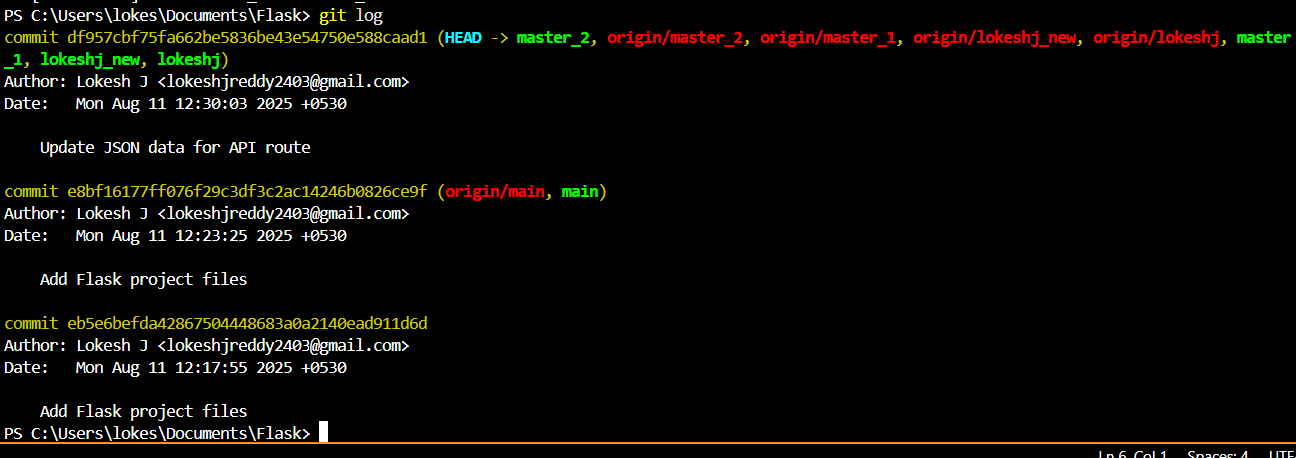
## 3. Step 2: Updating JSON File in New Branch

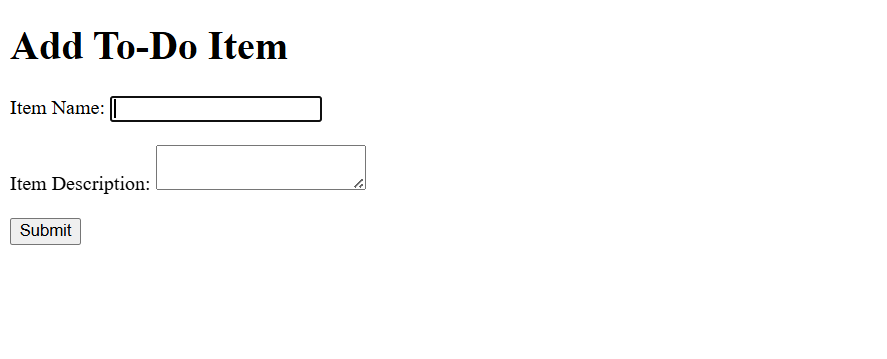
1. Created a new branch named lokeshj\_new .  
2. Updated the content of the JSON file used for the /api route.  
3. Merged the lokeshj\_new branch into the main branch.  
4. If conflicts occurred, resolved them by accepting changes from the lokeshj\_new branch.  
5. Added resolved changes to the staging area, committed, and pushed updates to the remote repository.

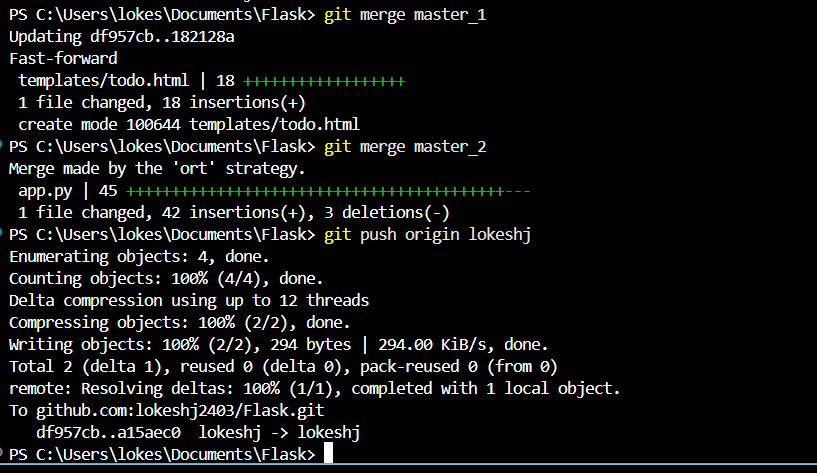


## 4. Step 3: Feature Development in Separate Branches

1. Created two branches from main: master\_1 and master\_2.  
2. In master\_1 branch:  
 - Created a To-Do Page in the frontend with fields for Item Name and Item Description.  
3. In master\_2 branch:  
 - Created a backend route named /submittodoitem.  
 - Configured it to accept itemName and itemDescription via a POST request.  
 - Stored these details in a MongoDB database.  
4. Merged changes from both master\_1 and master\_2 into the main branch.

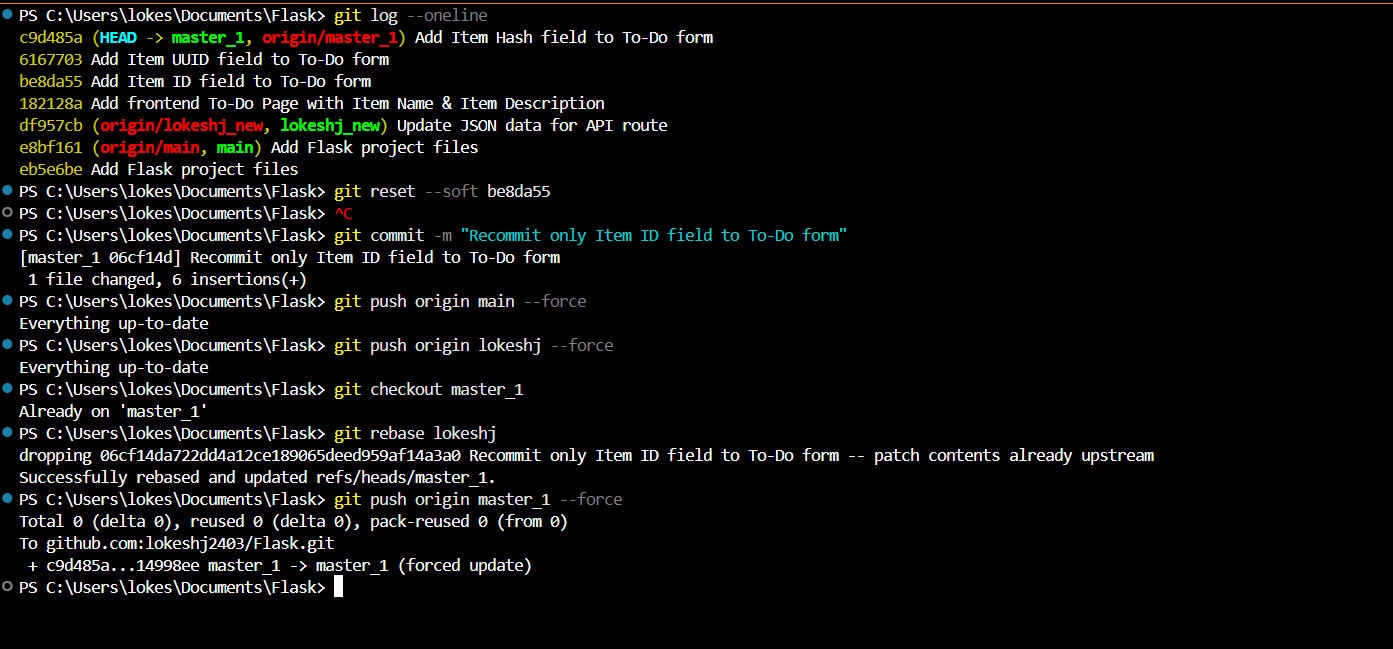






## 5. Step 4: Enhancing the To-Do Form and Git Reset/Rebase

1. In master\_1 branch:  
 - Added Item ID field (first commit).  
 - Added Item UUID field (second commit).  
 - Added Item Hash field (third commit).  
2. Merged master\_1 branch into main.  
3. In main branch:  
 - Used 'git reset --soft' to roll back to the commit where only the Item ID field was added.  
 - Re-committed this state and merged into main.  
4. Rebased updated changes from main into master\_1 using 'git rebase main master\_1', preserving individual commits.



## 6. Commands Used

git clone <repo\_ssh\_url>  
git branch <branch\_name>  
git checkout <branch\_name>  
git add .  
git commit -m 'Commit message'  
git push origin <branch\_name>  
git merge <branch\_name>  
git reset --soft <commit\_hash>  
git rebase main master\_1

## 7. Conclusion

This assignment demonstrated the effective use of Git branching, merging, conflict resolution, rebasing, and version history management in a real project workflow. The experience strengthened skills in collaborative development and efficient version control practices.