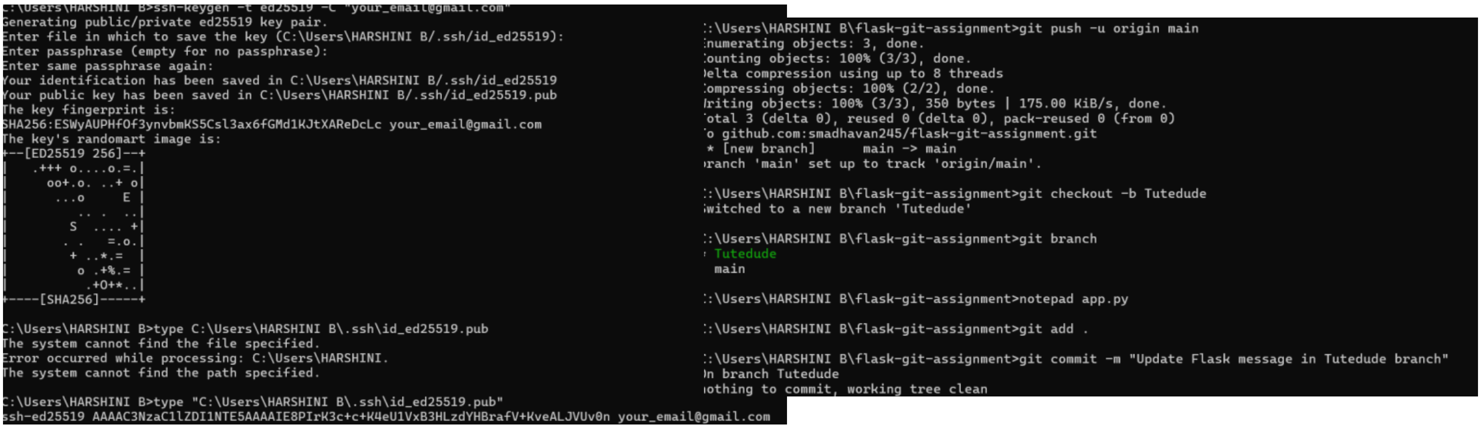
1. Create a new GitHub repository.

- 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

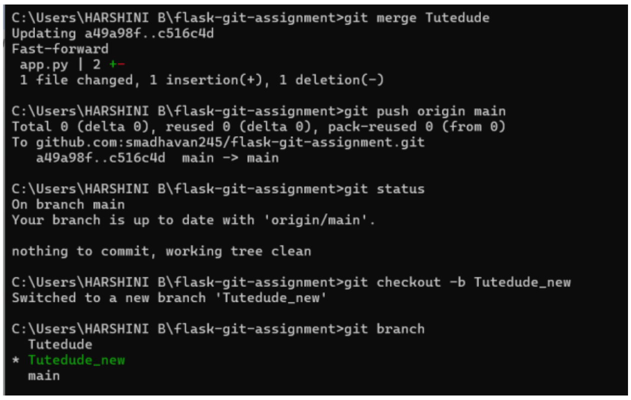


```
C:\Users\HARSHINI B>ssh-keygen -t ed25519 -C "your_email@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\HARSHINI B/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\HARSHINI B/.ssh/id_ed25519
Your public key has been saved in C:\Users\HARSHINI B/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:ESWyAUPHfOf3ynvbmKS5Csl3ax6fGMd1KJtXAReDcLc your_email@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|    .+++ o....o.=.|
|      oo+.o. ..+ o|
|       ...o     E |
|          .. .  ..|
|         S  .... +|
|        . .   =.o.|
|         + ..*.=  |
|          o .+%.= |
|            .+O+*..|
+----[SHA256]-----+

C:\Users\HARSHINI B>type C:\Users\HARSHINI B\.ssh\id_ed25519.pub
The system cannot find the file specified.
Error occurred while processing: C:\Users\HARSHINI.
The system cannot find the path specified.

C:\Users\HARSHINI B>type "C:\Users\HARSHINI B\.ssh\id_ed25519.pub"
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIE8PIrK3c+c+K4eU1VxB3HLzdYHBrafV+KveALJVUv0n your_email@gmail.com
```

```
:\Users\HARSHINI B\flask-git-assignment>git push -u origin main
numerating objects: 3, done.
ounting objects: 100% (3/3), done.
elta compression using up to 8 threads
ompressing objects: 100% (2/2), done.
riting objects: 100% (3/3), 350 bytes | 175.00 KiB/s, done.
otal 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
o github.com:smadhavan245/flask-git-assignment.git
 * [new branch]      main -> main
ranch 'main' set up to track 'origin/main'.

:\Users\HARSHINI B\flask-git-assignment>git checkout -b Tutedude
witched to a new branch 'Tutedude'

:\Users\HARSHINI B\flask-git-assignment>git branch
 Tutedude
 main

:\Users\HARSHINI B\flask-git-assignment>notepad app.py

:\Users\HARSHINI B\flask-git-assignment>git add .

:\Users\HARSHINI B\flask-git-assignment>git commit -m "Update Flask message in Tutedude branch"
n branch Tutedude
othing to commit, working tree clean
```

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

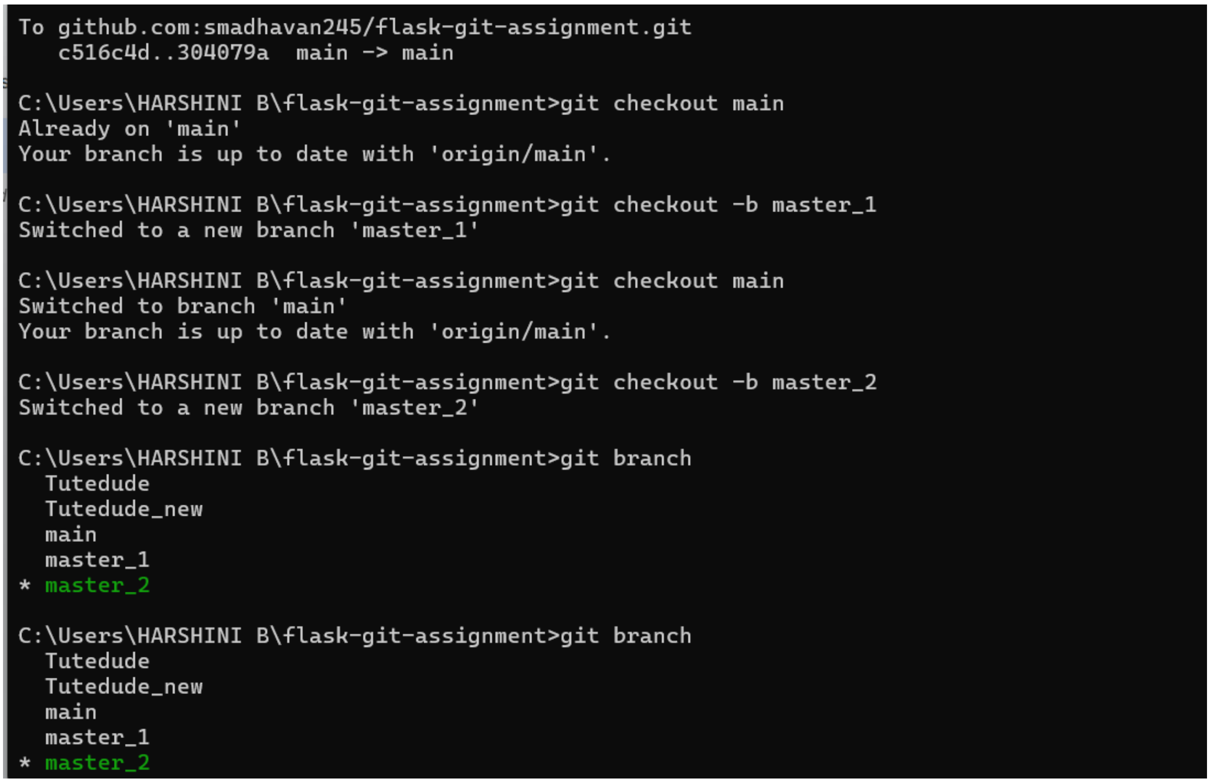
- Update the content of the JSON file used for the /api route in this branch.
- Merge the <your_name>_new branch into the main branch.
- If there are conflicts during the merge, resolve them by accepting the changes from the <your_name>_new branch.
- Add the resolved changes to the staging area, commit them, and push the updates to the remote repository.



```
C:\Users\HARSHINI B\flask-git-assignment>git merge Tutedude
Updating a49a98f..c516c4d
Fast-forward
 app.py | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\HARSHINI B\flask-git-assignment>git push origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:smadhavan245/flask-git-assignment.git
   a49a98f..c516c4d  main -> main

C:\Users\HARSHINI B\flask-git-assignment>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

C:\Users\HARSHINI B\flask-git-assignment>git checkout -b Tutedude_new
Switched to a new branch 'Tutedude_new'

C:\Users\HARSHINI B\flask-git-assignment>git branch
  Tutedude
* Tutedude_new
  main
```

3. Branch Creation:

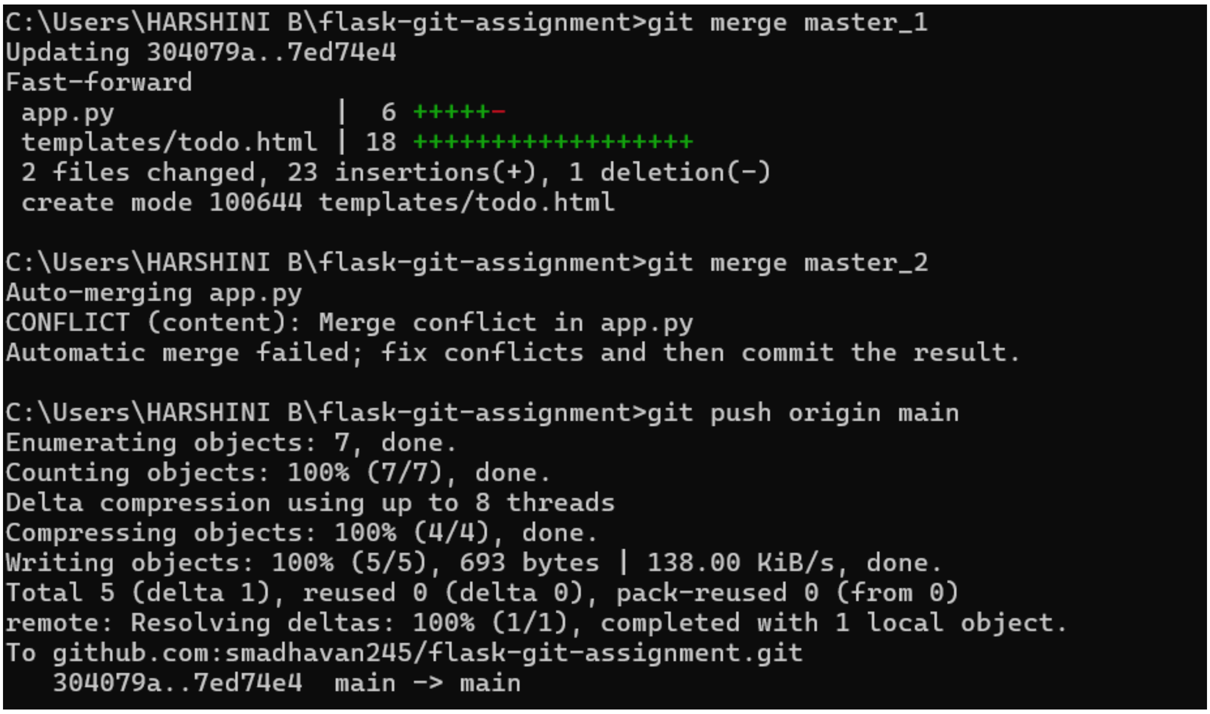
- **Create two branches: master_1 and master_2 from the main branch.**
- **Feature Development in master_1:**
- **In the master_1 branch, create a To-Do Page in the frontend.**
    - **The page should contain a form with the following fields:**
        - **Item Name**
        - **Item Description**
- **Backend API in master_2:**
- **In the master_2 branch, create a backend route named /submittodoitem.**
- **This route will:**
    - **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.**



```
To github.com:smadhavan245/flask-git-assignment.git
   c516c4d..304079a  main -> main

C:\Users\HARSHINI B\flask-git-assignment>git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.

C:\Users\HARSHINI B\flask-git-assignment>git checkout -b master_1
Switched to a new branch 'master_1'

C:\Users\HARSHINI B\flask-git-assignment>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\Users\HARSHINI B\flask-git-assignment>git checkout -b master_2
Switched to a new branch 'master_2'

C:\Users\HARSHINI B\flask-git-assignment>git branch
  Tutedude
  Tutedude_new
  main
  master_1
* master_2

C:\Users\HARSHINI B\flask-git-assignment>git branch
  Tutedude
  Tutedude_new
  main
  master_1
* master_2
```

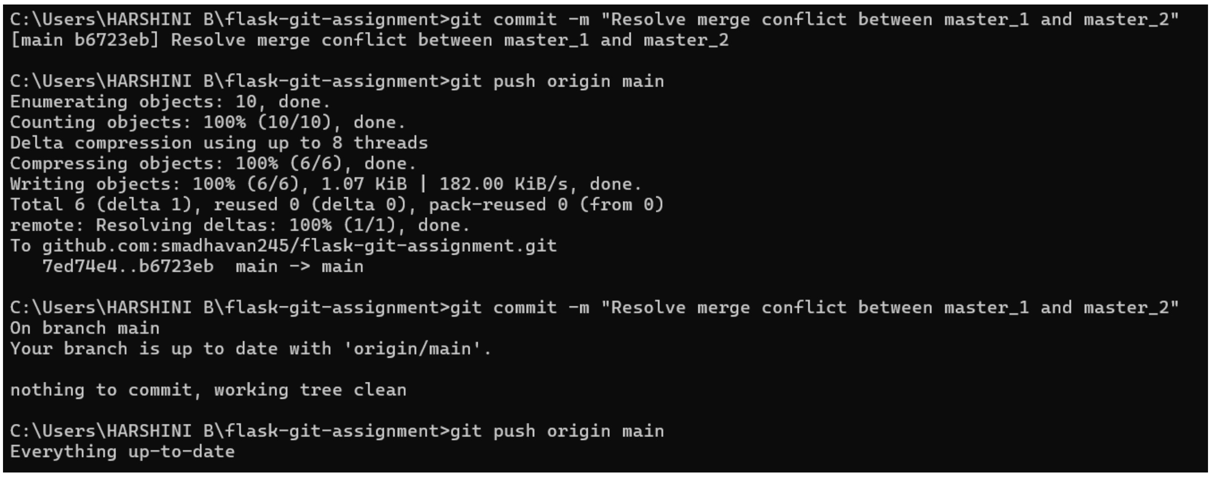
4. Enhancing the To-Do Form in master_1:

- **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.**



```
C:\Users\HARSHINI B\flask-git-assignment>git merge master_1
Updating 304079a..7ed74e4
Fast-forward
 app.py              |  6 +++++-
 templates/todo.html | 18 ++++++++++++++++++
 2 files changed, 23 insertions(+), 1 deletion(-)
 create mode 100644 templates/todo.html

C:\Users\HARSHINI B\flask-git-assignment>git merge master_2
Auto-merging app.py
CONFLICT (content): Merge conflict in app.py
Automatic merge failed; fix conflicts and then commit the result.

C:\Users\HARSHINI B\flask-git-assignment>git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 693 bytes | 138.00 KiB/s, done.
Total 5 (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:smadhavan245/flask-git-assignment.git
   304079a..7ed74e4  main -> main
```

○



```
C:\Users\HARSHINI B\flask-git-assignment>git commit -m "Resolve merge conflict between master_1 and master_2"
[main b6723eb] Resolve merge conflict between master_1 and master_2

C:\Users\HARSHINI B\flask-git-assignment>git push origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.07 KiB | 182.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:smadhavan245/flask-git-assignment.git
   7ed74e4..b6723eb  main -> main

C:\Users\HARSHINI B\flask-git-assignment>git commit -m "Resolve merge conflict between master_1 and master_2"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

C:\Users\HARSHINI B\flask-git-assignment>git push origin main
Everything up-to-date
```

https://github.com/smadhavan245/flask-git-assignment.git