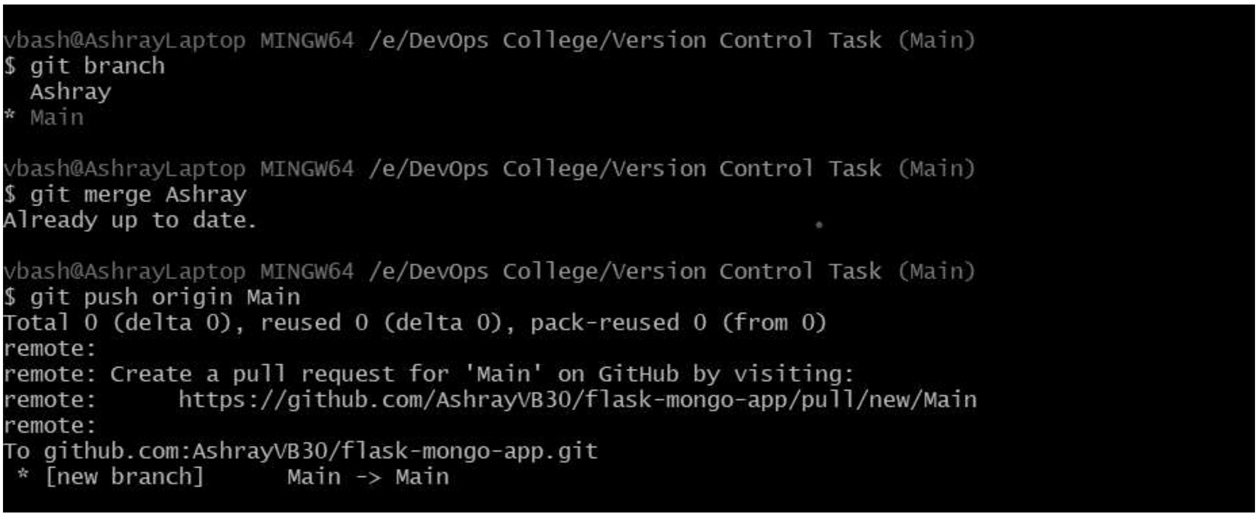
Assignement-4

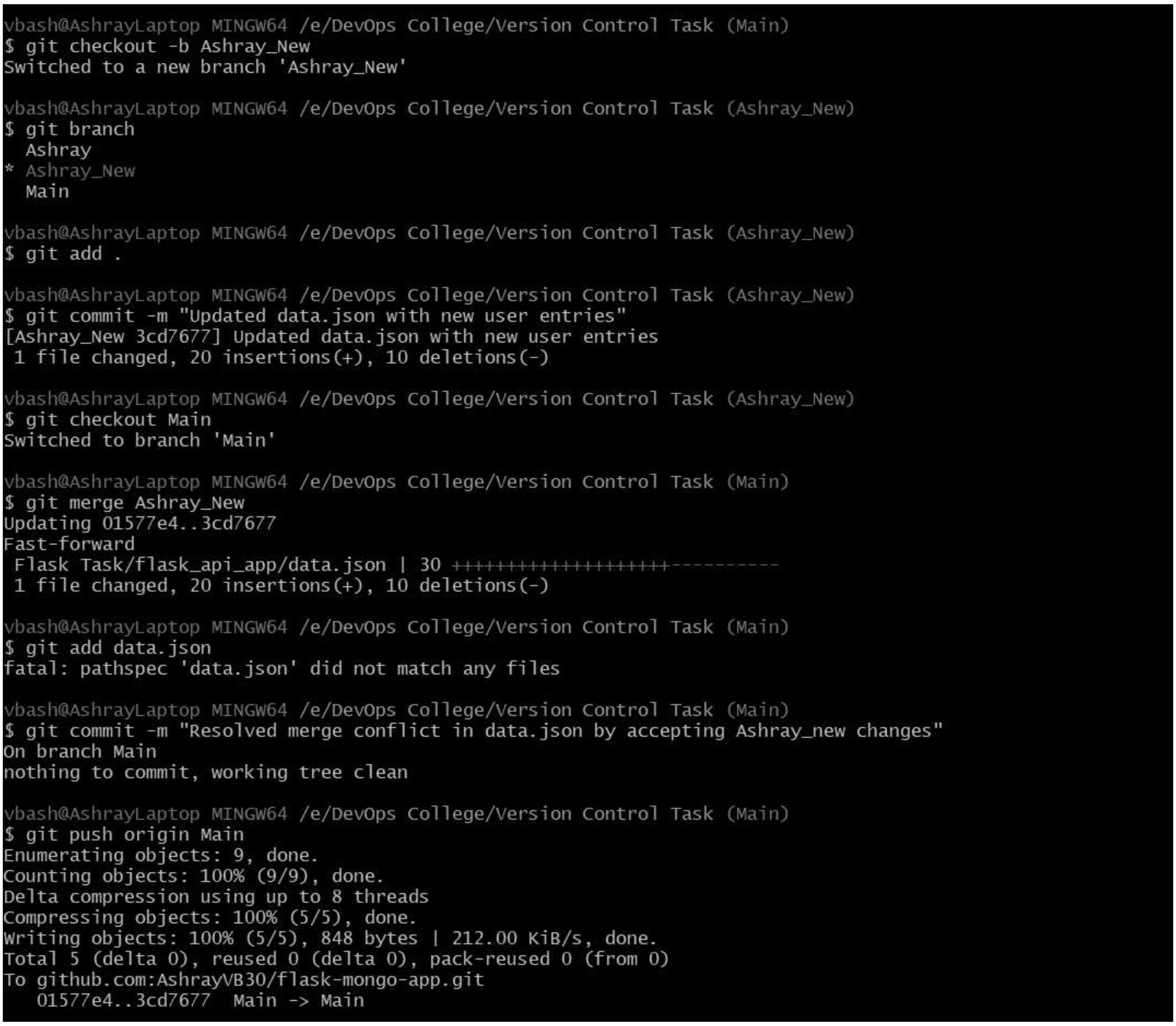
Version Control

1. Create a new GitHub repository.
   a. Clone the repository to your local machine using SSH (generate an SSH key if needed, add the public key to your GitHub account).
   b. Create a new branch named after your username (e.g., Tutedude).
   c. Add your Flask project files to this branch.
   d. Commit the changes and merge the branch into the main branch.



```
vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (Main)
$ git branch
  Ashray
* Main

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (Main)
$ git merge Ashray
Already up to date.

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (Main)
$ git push origin Main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'Main' on GitHub by visiting:
remote:      https://github.com/AshrayVB30/flask-mongo-app/pull/new/Main
remote:
To github.com:AshrayVB30/flask-mongo-app.git
 * [new branch]      Main -> Main
```

2. Create a new branch named <your_name>_new (e.g., Tutedude_new).
   a. Update the content of the JSON file used for the /api route in this branch.
   b. Merge the <your_name>_new branch into the main branch.
   c. If there are conflicts during the merge, resolve them by accepting the changes from the <your_name>_new branch.
   d. Add the resolved changes to the staging area, commit them, and push updates to the remote repository.



```
vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (Main)
$ git checkout -b Ashray_New
Switched to a new branch 'Ashray_New'

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (Ashray_New)
$ git branch
  Ashray
* Ashray_New
  Main

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (Ashray_New)
$ git add .

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (Ashray_New)
$ git commit -m "Updated data.json with new user entries"
[Ashray_New 3cd7677] Updated data.json with new user entries
 1 file changed, 20 insertions(+), 10 deletions(-)

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (Ashray_New)
$ git checkout Main
Switched to branch 'Main'

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (Main)
$ git merge Ashray_New
Updating 01577e4..3cd7677
Fast-forward
 Flask Task/flask_api_app/data.json | 30 ++++++++++++++++++++----------
 1 file changed, 20 insertions(+), 10 deletions(-)

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (Main)
$ git add data.json
fatal: pathspec 'data.json' did not match any files

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (Main)
$ git commit -m "Resolved merge conflict in data.json by accepting Ashray_new changes"
On branch Main
nothing to commit, working tree clean

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (Main)
$ git push origin Main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 848 bytes | 212.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:AshrayVB30/flask-mongo-app.git
   01577e4..3cd7677  Main -> Main
```

3. **Branch Creation:**
    a. **Create two branches: master_1 and master_2 from the main branch.**
    b. **Feature Development in master_1:**
        i. **In the master_1 branch, create a To-Do Page in the front end.**
        ii. **The page should contain a form with the following fields:**
            1. **Item Name**
            2. **Item Description**
    c. **Backend API in master_2:**

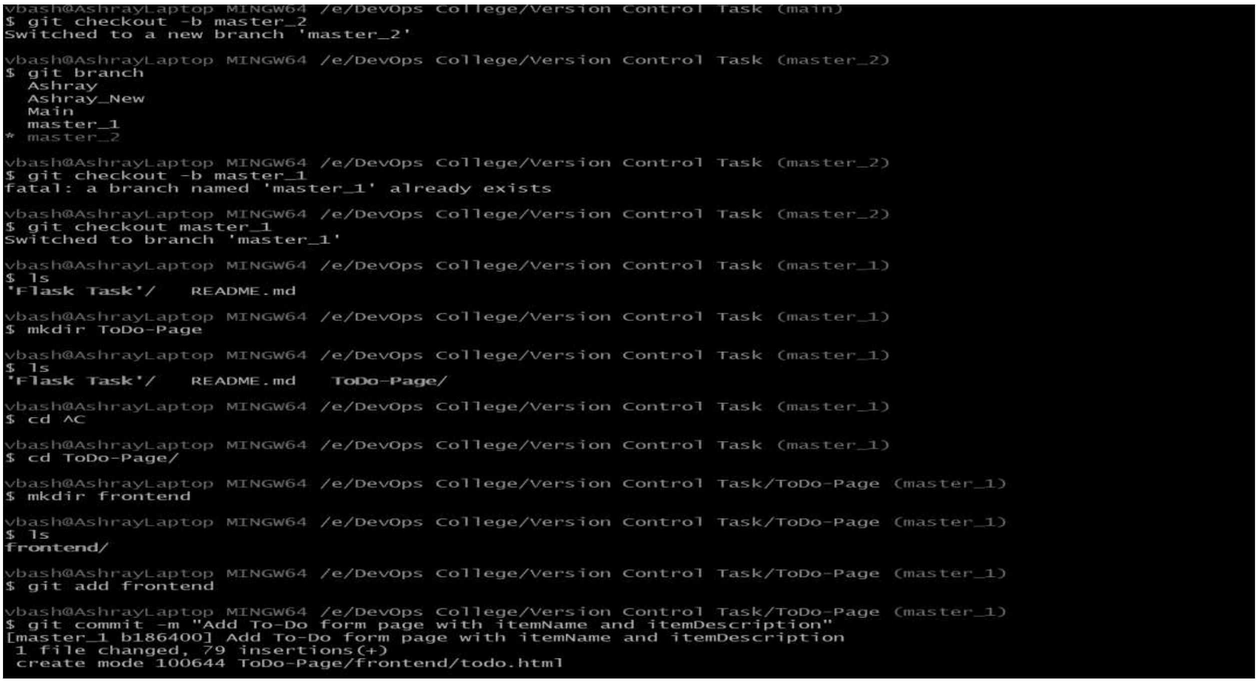
d. In the master_2 branch, create a backend route named /submittodoitem.

e. This route will:

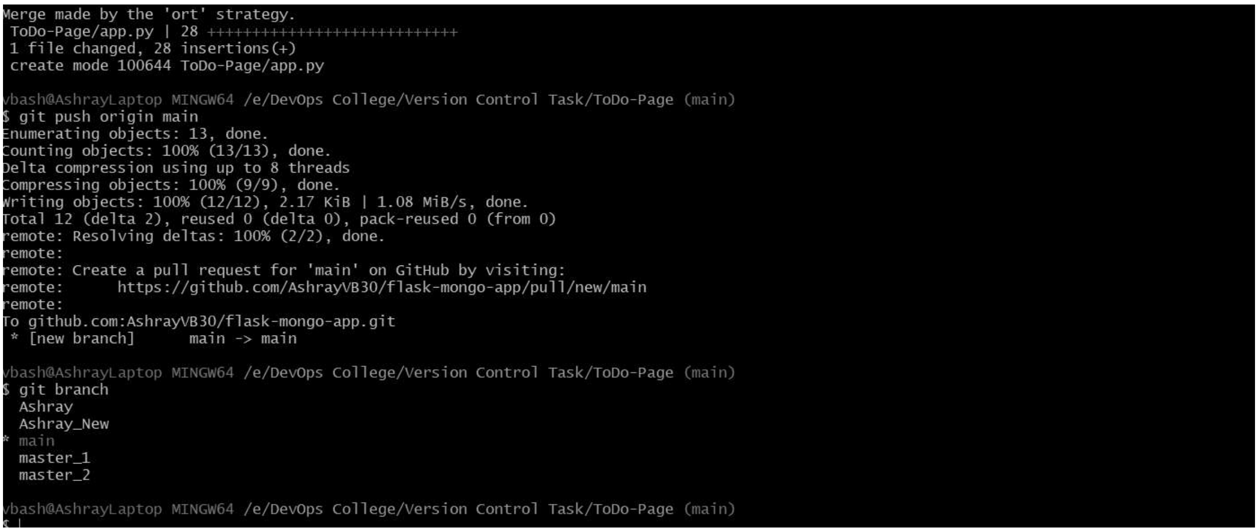
   i. Accept itemName and itemDescription via a POST request.

   ii. Store these details in a MongoDB database.

   iii. Merging Changes:Merge the changes from both master_1 and master_2 into the main branch.



```
vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (main)
$ git checkout -b master_2
Switched to a new branch 'master_2'

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (master_2)
$ git branch
  Ashray
  Ashray_New
  Main
  master_1
* master_2

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (master_2)
$ git checkout -b master_1
fatal: a branch named 'master_1' already exists

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (master_2)
$ git checkout master_1
Switched to branch 'master_1'

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (master_1)
$ ls
'Flask Task'/   README.md

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (master_1)
$ mkdir ToDo-Page

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (master_1)
$ ls
'Flask Task'/   README.md   ToDo-Page/

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (master_1)
$ cd ^C

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task (master_1)
$ cd ToDo-Page/

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task/ToDo-Page (master_1)
$ mkdir frontend

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task/ToDo-Page (master_1)
$ ls
frontend/

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task/ToDo-Page (master_1)
$ git add frontend

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task/ToDo-Page (master_1)
$ git commit -m "Add To-Do form page with itemName and itemDescription"
[master_1 b186400] Add To-Do form page with itemName and itemDescription
 1 file changed, 79 insertions(+)
 create mode 100644 ToDo-Page/frontend/todo.html
```



```
Merge made by the 'ort' strategy.
 ToDo-Page/app.py | 28 ++++++++++++++++++++++++++++
 1 file changed, 28 insertions(+)
 create mode 100644 ToDo-Page/app.py

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task/ToDo-Page (main)
$ git push origin main
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (12/12), 2.17 KiB | 1.08 MiB/s, done.
Total 12 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
remote:
remote: Create a pull request for 'main' on GitHub by visiting:
remote:      https://github.com/AshrayVB30/flask-mongo-app/pull/new/main
remote:
To github.com:AshrayVB30/flask-mongo-app.git
 * [new branch]      main -> main

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task/ToDo-Page (main)
$ git branch
  Ashray
  Ashray_New
* main
  master_1
  master_2

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task/ToDo-Page (main)
$ |
```

4. **Enhancing the To-Do Form in** master_1:

   In the master_1 branch, add the following fields to the To-Do form:

   a. **Item ID**
   b. **Item UUID**
   c. **Item Hash**

   Committing in Sequence:

   Add and commit each field separately in the following order:

   d. **First commit:** Add **Item ID** field.
   e. **Second commit:** Add **Item UUID** field.
   f. **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.

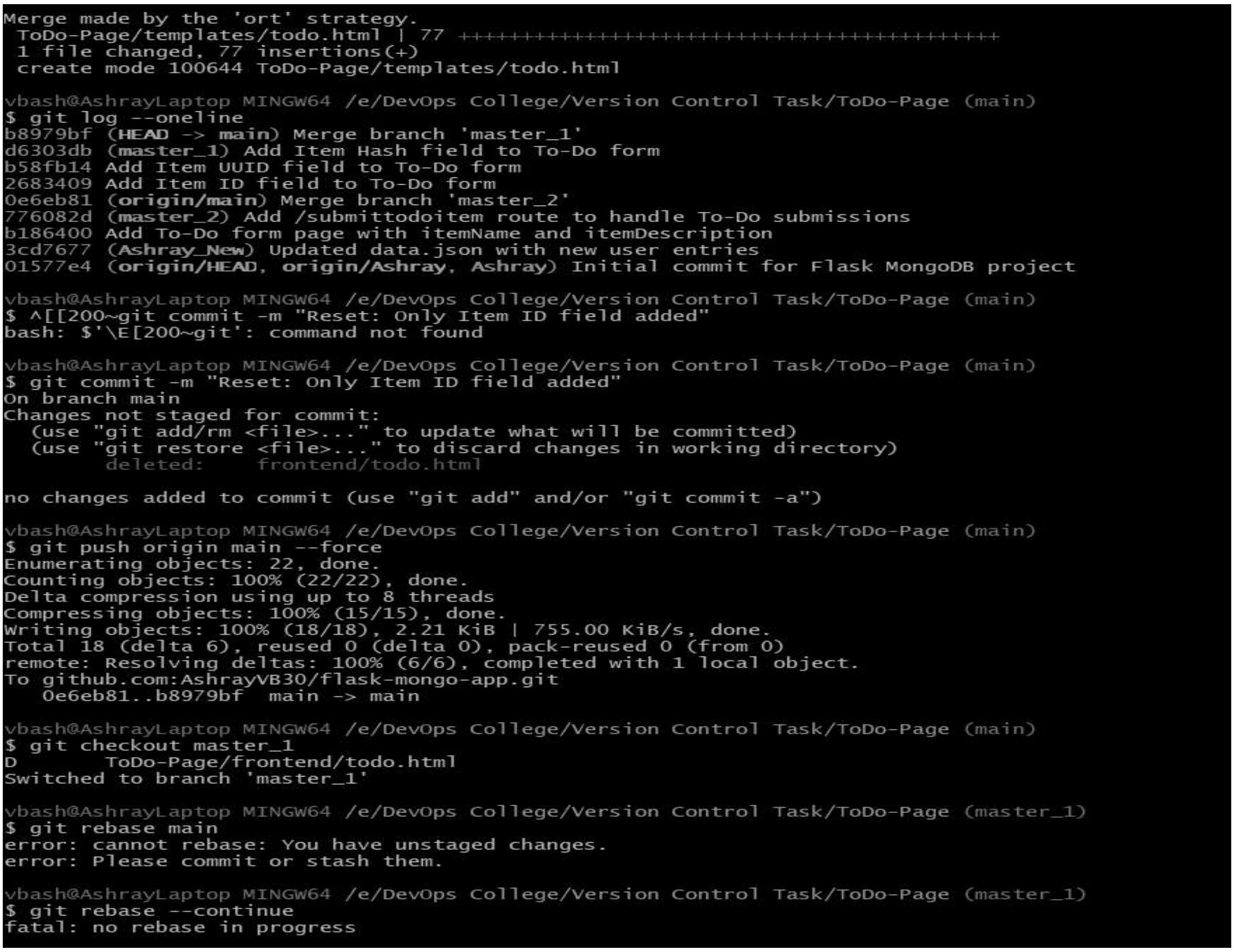
   Rebasing Changes:

   Rebase the updated changes in the main branch to the master_1 branch.

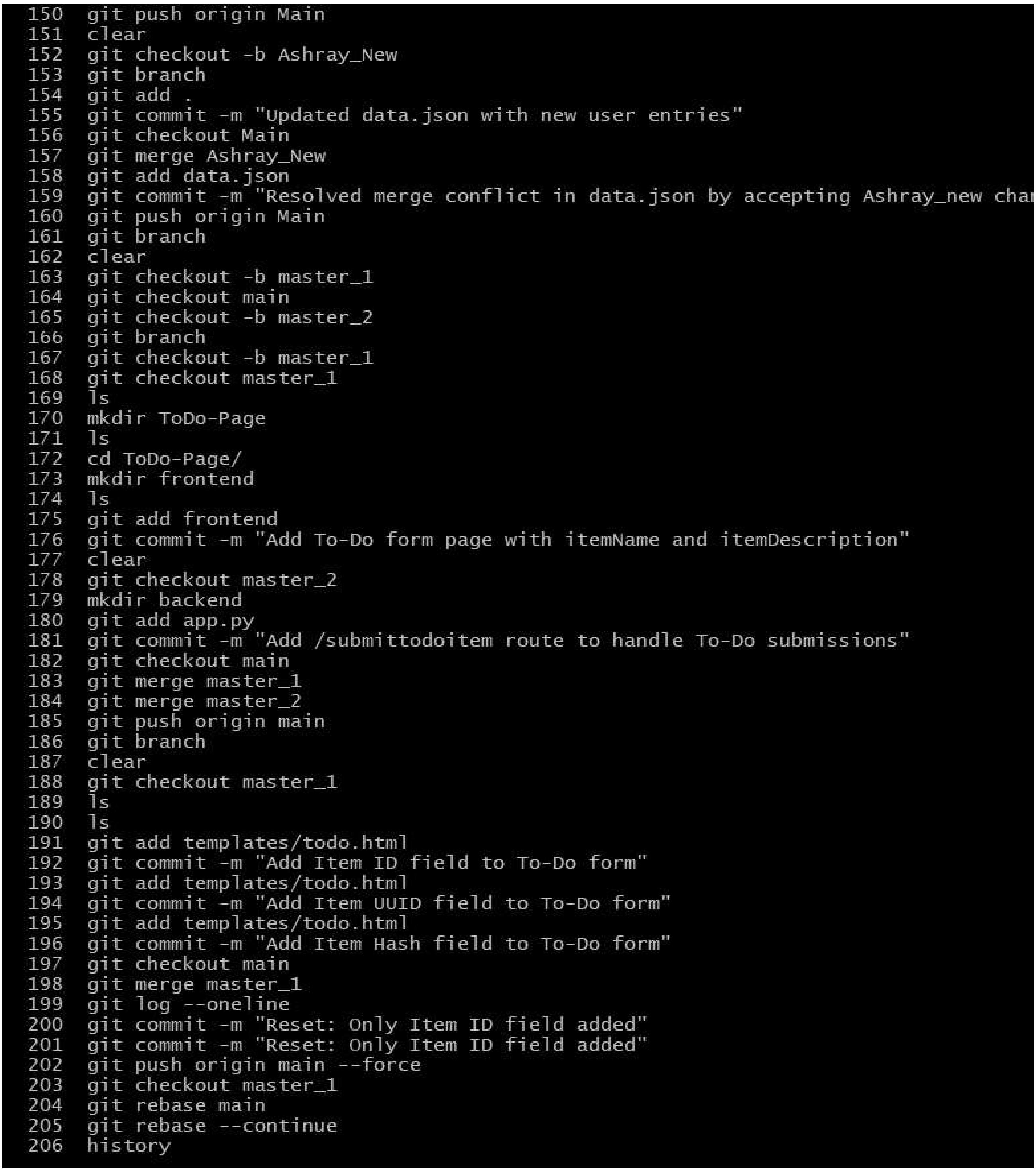
   Clarification:

   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.



```
Merge made by the 'ort' strategy.
 ToDo-Page/templates/todo.html | 77 ++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 77 insertions(+)
 create mode 100644 ToDo-Page/templates/todo.html

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task/ToDo-Page (main)
$ git log --oneline
b8979bf (HEAD -> main) Merge branch 'master_1'
d6303db (master_1) Add Item Hash field to To-Do form
b58fb14 Add Item UUID field to To-Do form
2683409 Add Item ID field to To-Do form
0e6eb81 (origin/main) Merge branch 'master_2'
776082d (master_2) Add /submittodoitem route to handle To-Do submissions
b186400 Add To-Do form page with itemName and itemDescription
3cd7677 (Ashray_New) Updated data.json with new user entries
01577e4 (origin/HEAD, origin/Ashray, Ashray) Initial commit for Flask MongoDB project

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task/ToDo-Page (main)
$ ^[[200~git commit -m "Reset: Only Item ID field added"
bash: $'\E[200~git': command not found

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task/ToDo-Page (main)
$ git commit -m "Reset: Only Item ID field added"
On branch main
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    frontend/todo.html

no changes added to commit (use "git add" and/or "git commit -a")

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task/ToDo-Page (main)
$ git push origin main --force
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 8 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (18/18), 2.21 KiB | 755.00 KiB/s, done.
Total 18 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (6/6), completed with 1 local object.
To github.com:AshrayVB30/flask-mongo-app.git
   0e6eb81..b8979bf  main -> main

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task/ToDo-Page (main)
$ git checkout master_1
D       ToDo-Page/frontend/todo.html
Switched to branch 'master_1'

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task/ToDo-Page (master_1)
$ git rebase main
error: cannot rebase: You have unstaged changes.
error: Please commit or stash them.

vbash@AshrayLaptop MINGW64 /e/DevOps College/Version Control Task/ToDo-Page (master_1)
$ git rebase --continue
fatal: no rebase in progress
```



```
  150  git push origin Main
  151  clear
  152  git checkout -b Ashray_New
  153  git branch
  154  git add .
  155  git commit -m "Updated data.json with new user entries"
  156  git checkout Main
  157  git merge Ashray_New
  158  git add data.json
  159  git commit -m "Resolved merge conflict in data.json by accepting Ashray_new cha
  160  git push origin Main
  161  git branch
  162  clear
  163  git checkout -b master_1
  164  git checkout main
  165  git checkout -b master_2
  166  git branch
  167  git checkout -b master_1
  168  git checkout master_1
  169  ls
  170  mkdir ToDo-Page
  171  ls
  172  cd ToDo-Page/
  173  mkdir frontend
  174  ls
  175  git add frontend
  176  git commit -m "Add To-Do form page with itemName and itemDescription"
  177  clear
  178  git checkout master_2
  179  mkdir backend
  180  git add app.py
  181  git commit -m "Add /submittodoitem route to handle To-Do submissions"
  182  git checkout main
  183  git merge master_1
  184  git merge master_2
  185  git push origin main
  186  git branch
  187  clear
  188  git checkout master_1
  189  ls
  190  ls
  191  git add templates/todo.html
  192  git commit -m "Add Item ID field to To-Do form"
  193  git add templates/todo.html
  194  git commit -m "Add Item UUID field to To-Do form"
  195  git add templates/todo.html
  196  git commit -m "Add Item Hash field to To-Do form"
  197  git checkout main
  198  git merge master_1
  199  git log --oneline
  200  git commit -m "Reset: Only Item ID field added"
  201  git commit -m "Reset: Only Item ID field added"
  202  git push origin main --force
  203  git checkout master_1
  204  git rebase main
  205  git rebase --continue
  206  history
```