Github Project: Git Commands Documentation Template

Introduction to Version Control

You will use this template to copy and paste the git commands you used to complete all tasks on your local and remote git repository for this project. This file will serve as your submission for the GitHub project.

1. Set Up Your Repository

The following are the steps you will take to create your git repository, add your python code, and post your files on GitHub.

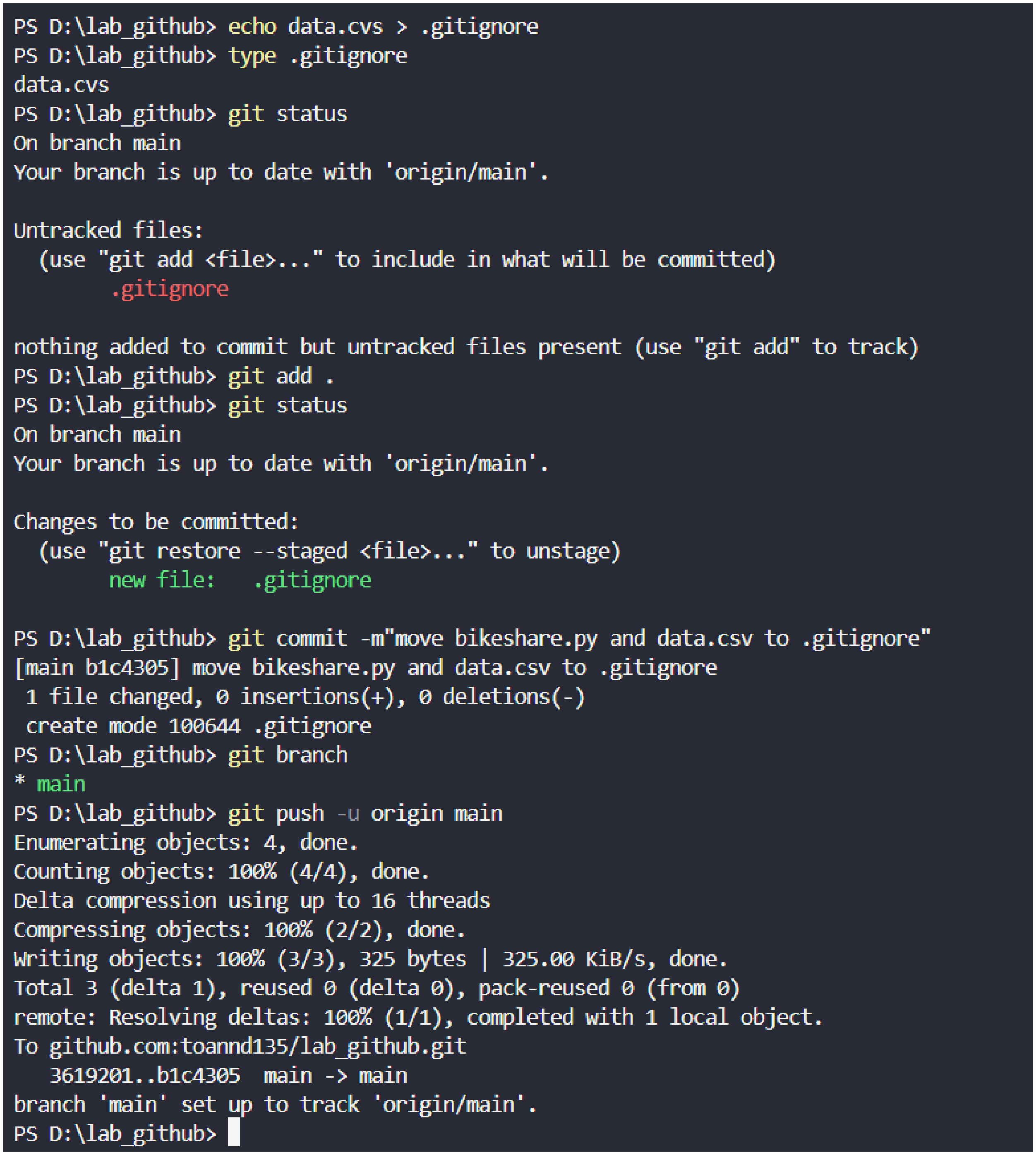
Step 1. Create a GitHub profile (if you don't already have one).
Step 2. Fork a repository from https://github.com/Maybetuandat/lab_github and provide a link to your forked GitHub repository here:

GitHub Repository Link
https://github.com/toannd135/lab_github

Step 3. Complete the tasks outlined in the table below and copy and paste your git commands into the "Git Commands" column. The first git command is partially filled out for you.

	Tasks	Git Commands
A.	Clone the GitHub repository to your local repository.	Git clone git@github.com:toannd135/lab_github.git
B.	Move your bikeshare.py and data files into your local repository.	move D: \lab_github_clone\ bikeshare.py D:\git_lab D:\git_lab
C.	Create a .gitignore file containing the name of your data file.	echo data.csv > .gitignore
D.	List the file names associated with the data files you added to your .gitignore	type .gitignore
E.	Check the status of your files to make sure your	git status

	files are not being tracked	
F.	Stage your changes.	git add .
G.	Commit your changes with a descriptive message.	git commit -m”message”
H.	Push your commit to your remote repository.	git push -u origin main

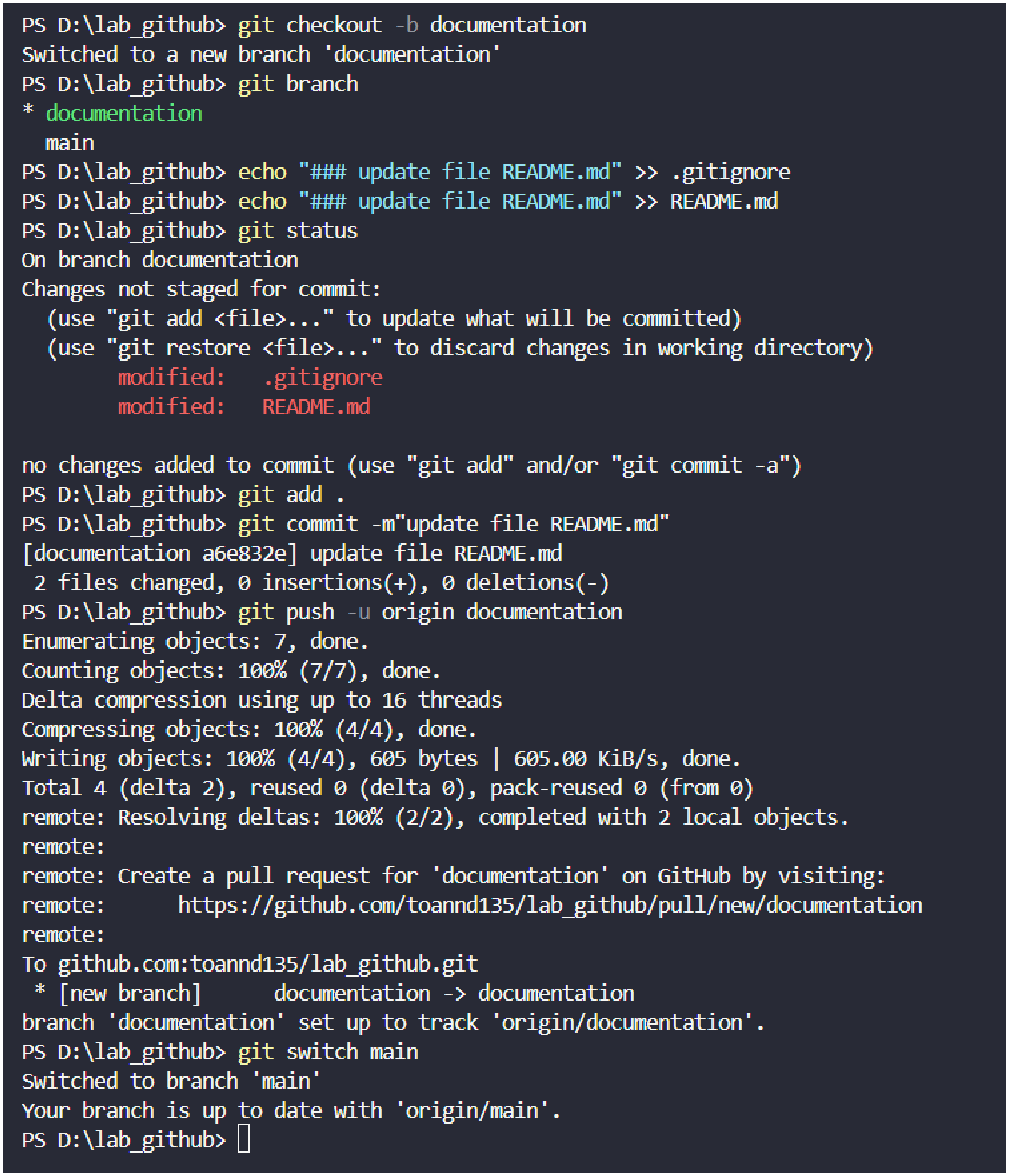


```
PS D:\lab_github> echo data.cvs > .gitignore
PS D:\lab_github> type .gitignore
data.cvs
PS D:\lab_github> git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

nothing added to commit but untracked files present (use "git add" to track)
PS D:\lab_github> git add .
PS D:\lab_github> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   .gitignore

PS D:\lab_github> git commit -m"move bikeshare.py and data.csv to .gitignore"
[main b1c4305] move bikeshare.py and data.csv to .gitignore
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 .gitignore
PS D:\lab_github> git branch
* main
PS D:\lab_github> git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 325 bytes | 325.00 KiB/s, done.
Total 3 (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:toannd135/lab_github.git
   3619201..b1c4305  main -> main
branch 'main' set up to track 'origin/main'.
PS D:\lab_github> 
```

2. Improve Documentation

Now you will be working in your local repository, on the BikeShare python file and the README.md file. You should repeat steps C through E three times to make at least three commits as you work on your documentation improvements.

	Tasks	**Git Commands**
A.	Create a branch named *documentation* on your local repository.	Git checkout -b documentation
B.	Switch to the *documentation* branch.	
C.	Update your README.md file.	echo “###update file README.md” >> README.md
D.	Stage your changes.	git add .
E.	Commit your work with a descriptive message.	git commit -m”update file README.md”
F.	Push your commit to your remote repository branch.	git push -u origin documentation
G.	Switch back to the master (main) branch.	git switch main

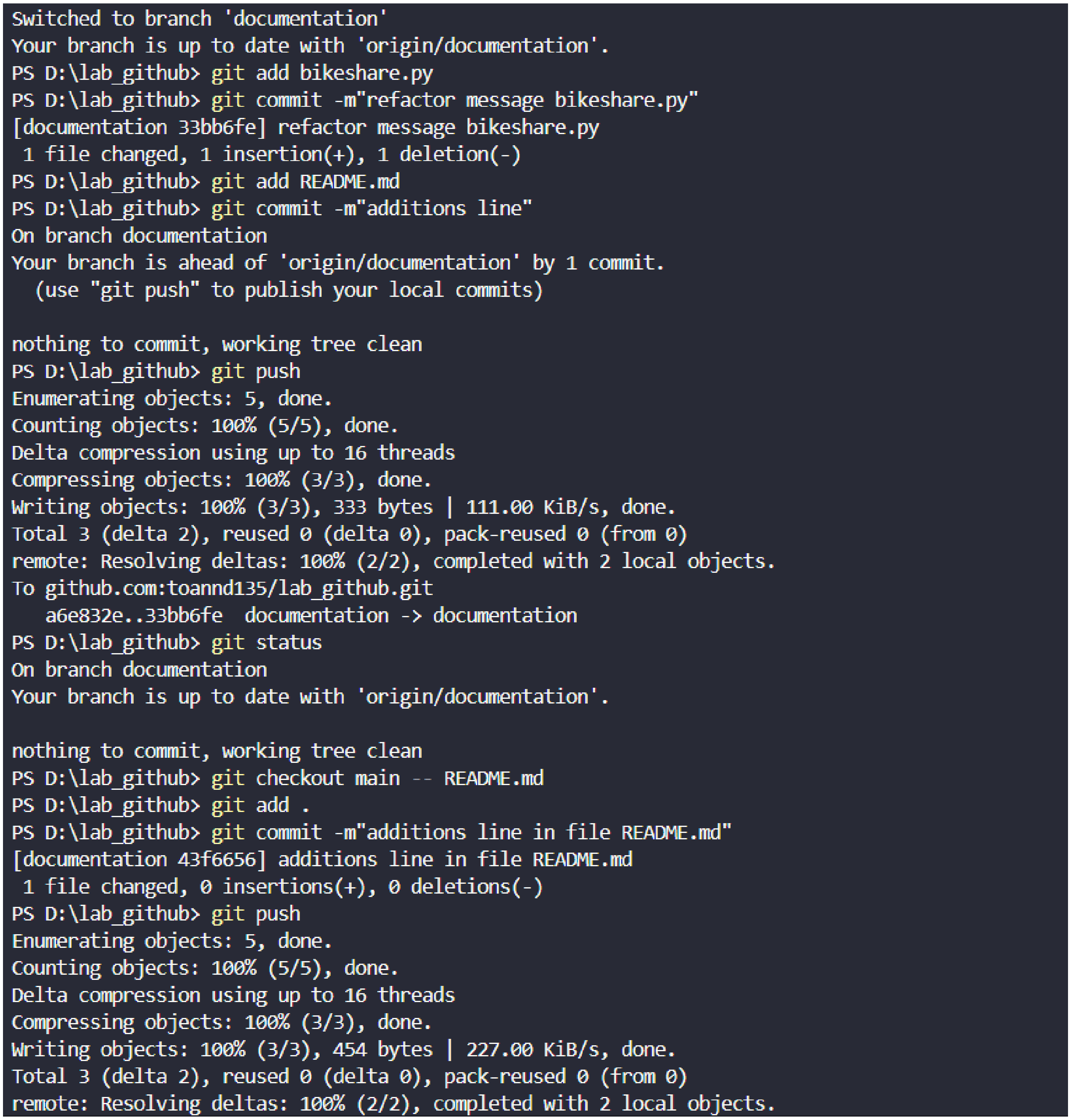


```
PS D:\lab_github> git checkout -b documentation
Switched to a new branch 'documentation'
PS D:\lab_github> git branch
* documentation
  main
PS D:\lab_github> echo "### update file README.md" >> .gitignore
PS D:\lab_github> echo "### update file README.md" >> README.md
PS D:\lab_github> git status
On branch documentation
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .gitignore
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
PS D:\lab_github> git add .
PS D:\lab_github> git commit -m"update file README.md"
[documentation a6e832e] update file README.md
 2 files changed, 0 insertions(+), 0 deletions(-)
PS D:\lab_github> git push -u origin documentation
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 605 bytes | 605.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'documentation' on GitHub by visiting:
remote:      https://github.com/toannd135/lab_github/pull/new/documentation
remote:
To github.com:toannd135/lab_github.git
 * [new branch]      documentation -> documentation
branch 'documentation' set up to track 'origin/documentation'.
PS D:\lab_github> git switch main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS D:\lab_github> 
```

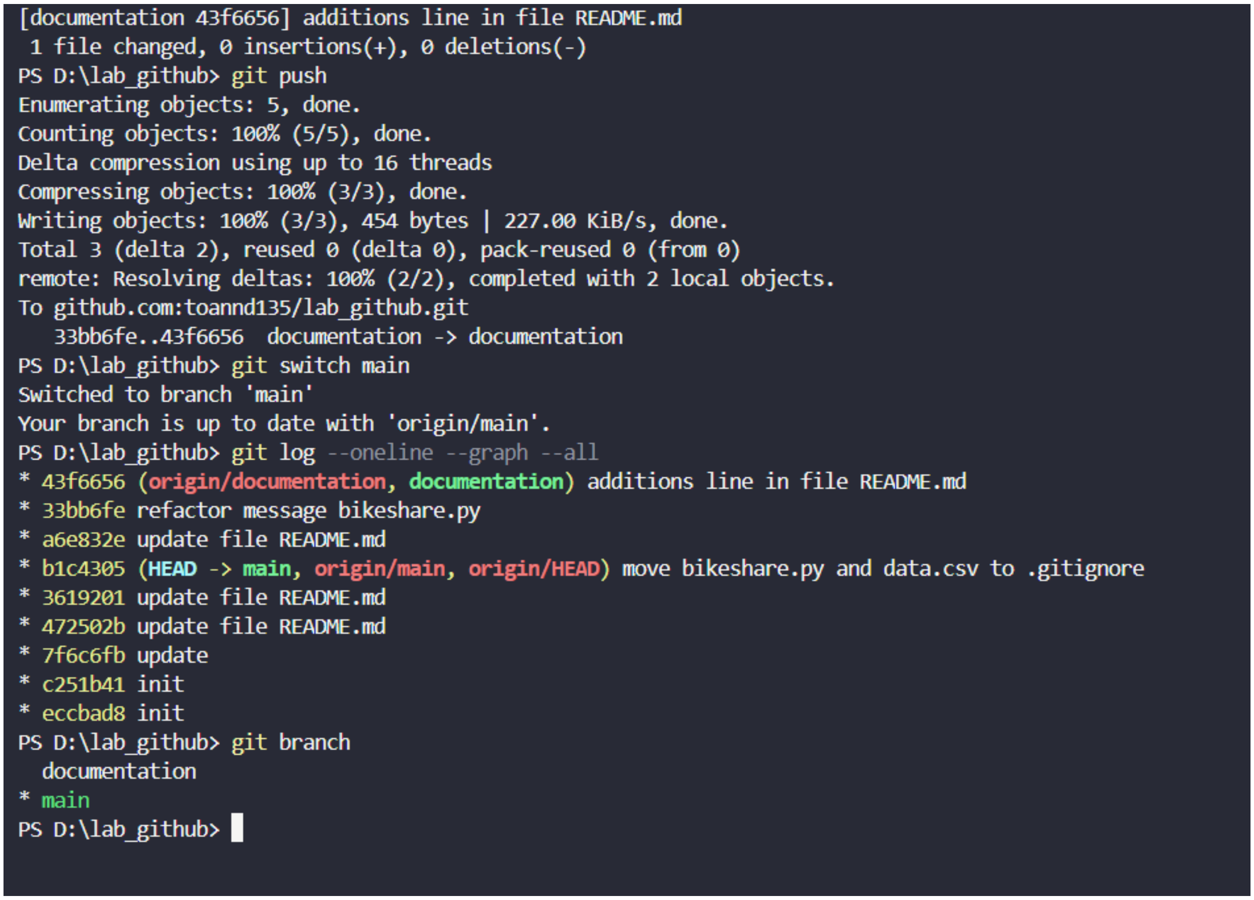
3. Additional Changes to Documentation

In a real world situation, you or other members of your team would likely be making other changes to documentation on the documentation branch. To simulate this follow the tasks below.

	Tasks	**Git Commands**
A.	Switch to the *documentation* branch.	git switch documentation
B.	Make at least 2 additional changes to the documentation - this might be additional changes to the README or changes to the document strings and line comments of the bikeshare file.	
C.	After each change, stage and commit your changes. When you commit your work, you should use a descriptive message of the changes made. Your changes should be small and aligned with your commit message.	git add README.md && git commit -m "refactor message bikeshare.py"
D.	Push your changes to the remote repository branch.	git push origin documentation
E.	Switch back to the *master* branch.	git switch main
F.	Check the local repository log to see how *all the branches* have changed.	git log --oneline --graph --all
G.	Go to Github. Notice that you now have two branches available for your project, and when you change branches the README changes.	



```
Switched to branch 'documentation'
Your branch is up to date with 'origin/documentation'.
PS D:\lab_github> git add bikeshare.py
PS D:\lab_github> git commit -m"refactor message bikeshare.py"
[documentation 33bb6fe] refactor message bikeshare.py
 1 file changed, 1 insertion(+), 1 deletion(-)
PS D:\lab_github> git add README.md
PS D:\lab_github> git commit -m"additions line"
On branch documentation
Your branch is ahead of 'origin/documentation' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
PS D:\lab_github> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 333 bytes | 111.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:toannd135/lab_github.git
   a6e832e..33bb6fe  documentation -> documentation
PS D:\lab_github> git status
On branch documentation
Your branch is up to date with 'origin/documentation'.

nothing to commit, working tree clean
PS D:\lab_github> git checkout main -- README.md
PS D:\lab_github> git add .
PS D:\lab_github> git commit -m"additions line in file README.md"
[documentation 43f6656] additions line in file README.md
 1 file changed, 0 insertions(+), 0 deletions(-)
PS D:\lab_github> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 454 bytes | 227.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
```



```
[documentation 43f6656] additions line in file README.md
 1 file changed, 0 insertions(+), 0 deletions(-)
PS D:\lab_github> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 454 bytes | 227.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:toannd135/lab_github.git
   33bb6fe..43f6656  documentation -> documentation
PS D:\lab_github> git switch main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS D:\lab_github> git log --oneline --graph --all
* 43f6656 (origin/documentation, documentation) additions line in file README.md
* 33bb6fe refactor message bikeshare.py
* a6e832e update file README.md
* b1c4305 (HEAD -> main, origin/main, origin/HEAD) move bikeshare.py and data.csv to .gitignore
* 3619201 update file README.md
* 472502b update file README.md
* 7f6c6fb update
* c251b41 init
* eccbad8 init
PS D:\lab_github> git branch
  documentation
* main
PS D:\lab_github> 
```

4. Refactor Code

Now you will be working in your local repository, on the code in your BikeShare python file to make improvements to its efficiency and readability. You should repeat steps C through E three times to make at least three commits as you refactor.

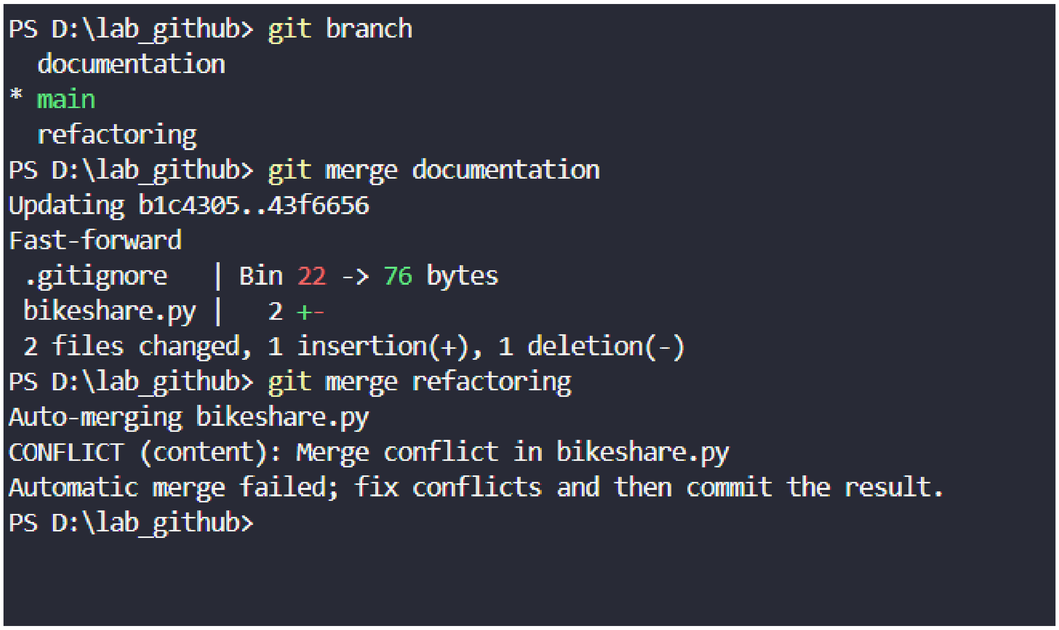
	Tasks	**Git Commands**
A.	Create a branch named *refactoring* on your local repository.	git switch -c refactoring
B.	Switch to the *refactoring* branch.	
C.	Similar to the process you used in making the documentation changes, make 2 or more changes in refactoring your code.	
D.	*For each change,* stage and commit your work	git add .

	with a descriptive message of the changes made.	git commit -m”refactor file bikeshare.py”
E.	Push your commits to your remote repository branch.	git push -u origin main
F.	Switch back to the *master* branch.	git switch main
G.	Check the local repository log to see how *all the branches* have changed.	git log --oneline --graph --all
H.	Go to GitHub. Notice that you now have 3 branches. Notice how the files change as you move through the branches.	

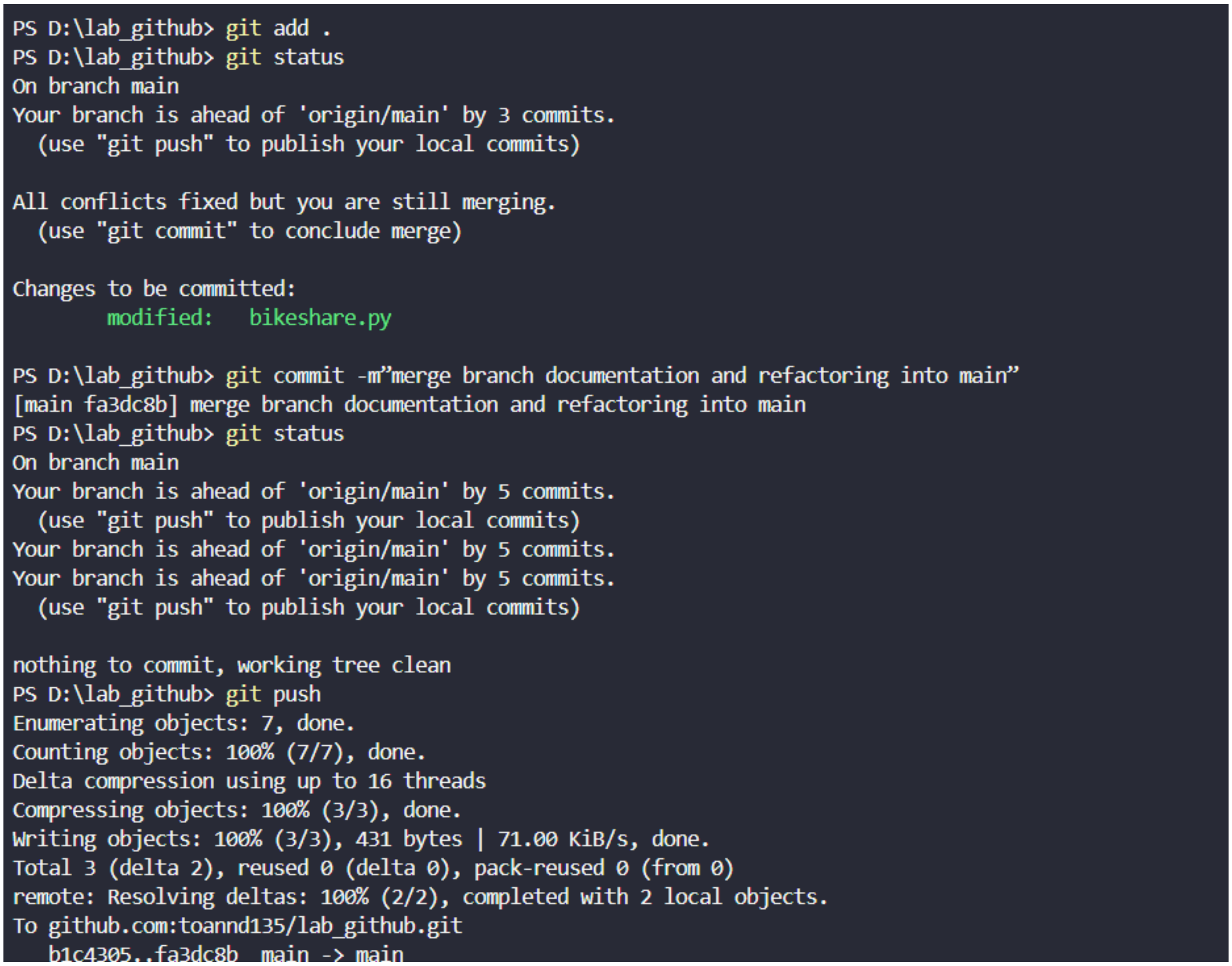
5. Merge Branches

	Tasks	**Git Commands**
A.	Switch to the *master* branch.	git switch main
B.	Pull the changes you and your coworkers might have made in the passing days (in this case, you won't have any updates, but pulling changes is often the first thing you do each day).	git pull origin main
C.	Since your changes are all ready to go, merge all the branches into the master. Address any merge conflicts. If you split up your work among your branches correctly, you should have no merge conflicts.	git branch git merge documentation git merge refactoring
D.	You should see a message that shows the changes to the files, insertions, and deletions.	
E.	Push the repository to your remote repository.	git add . git commit -m”merge branch documentation and refactoring into

		main”
F.	Go to GitHub. Notice that your master branch has all of the changes.	



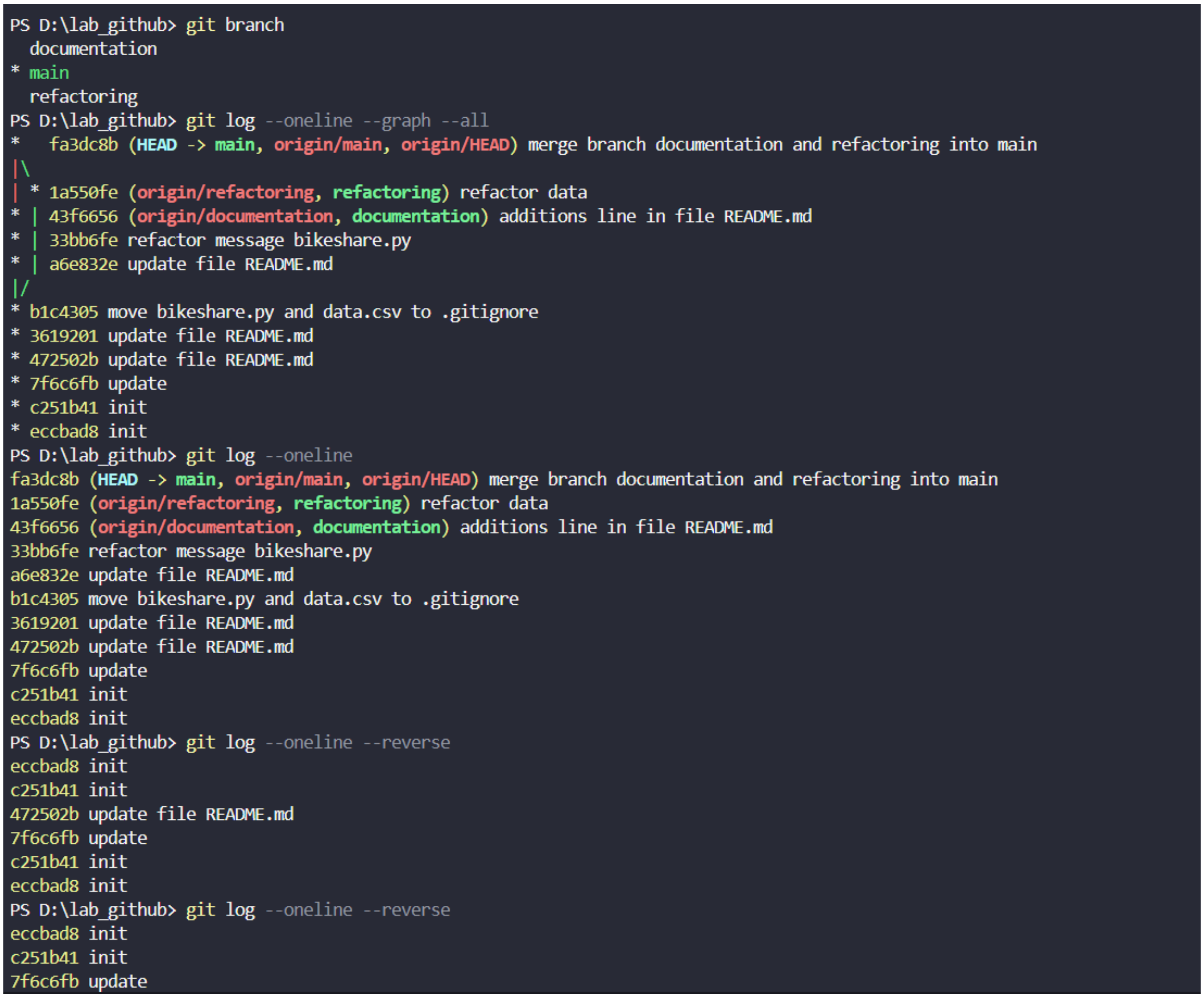
```
PS D:\lab_github> git branch
  documentation
* main
  refactoring
PS D:\lab_github> git merge documentation
Updating b1c4305..43f6656
Fast-forward
 .gitignore    | Bin 22 -> 76 bytes
 bikeshare.py |   2 +-
 2 files changed, 1 insertion(+), 1 deletion(-)
PS D:\lab_github> git merge refactoring
Auto-merging bikeshare.py
CONFLICT (content): Merge conflict in bikeshare.py
Automatic merge failed; fix conflicts and then commit the result.
PS D:\lab_github>
```



```
PS D:\lab_github> git add .
PS D:\lab_github> git status
On branch main
Your branch is ahead of 'origin/main' by 3 commits.
  (use "git push" to publish your local commits)

All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
        modified:   bikeshare.py

PS D:\lab_github> git commit -m”merge branch documentation and refactoring into main”
[main fa3dc8b] merge branch documentation and refactoring into main
PS D:\lab_github> git status
On branch main
Your branch is ahead of 'origin/main' by 5 commits.
  (use "git push" to publish your local commits)
Your branch is ahead of 'origin/main' by 5 commits.
Your branch is ahead of 'origin/main' by 5 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
PS D:\lab_github> git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 431 bytes | 71.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:toannd135/lab_github.git
   b1c4305..fa3dc8b  main -> main
```

6. Changing older commit messages

	Tasks	**Git Commands**
A.	Switch to the *master* branch.	git switch main
B.	Check the local repository log to see how *all the branches* have changed.	git log --oneline --graph --all
C.	Display a list of the commits that need edit messages.	git log --oneline hoặc git log --oneline --reverse
D.	Adding types to the titles commit into older commit messages	git rebase -i HEAD~n
E.	Push the repository to your remote repository.	git push origin main
F.	Go to GitHub. Notice that your master branch has all of the changes.	



```
PS D:\lab_github> git branch
  documentation
* main
  refactoring
PS D:\lab_github> git log --oneline --graph --all
*   fa3dc8b (HEAD -> main, origin/main, origin/HEAD) merge branch documentation and refactoring into main
|\
| * 1a550fe (origin/refactoring, refactoring) refactor data
* | 43f6656 (origin/documentation, documentation) additions line in file README.md
* | 33bb6fe refactor message bikeshare.py
* | a6e832e update file README.md
|/
* b1c4305 move bikeshare.py and data.csv to .gitignore
* 3619201 update file README.md
* 472502b update file README.md
* 7f6c6fb update
* c251b41 init
* eccbad8 init
PS D:\lab_github> git log --oneline
fa3dc8b (HEAD -> main, origin/main, origin/HEAD) merge branch documentation and refactoring into main
1a550fe (origin/refactoring, refactoring) refactor data
43f6656 (origin/documentation, documentation) additions line in file README.md
33bb6fe refactor message bikeshare.py
a6e832e update file README.md
b1c4305 move bikeshare.py and data.csv to .gitignore
3619201 update file README.md
472502b update file README.md
7f6c6fb update
c251b41 init
eccbad8 init
PS D:\lab_github> git log --oneline --reverse
eccbad8 init
c251b41 init
472502b update file README.md
7f6c6fb update
c251b41 init
eccbad8 init
PS D:\lab_github> git log --oneline --reverse
eccbad8 init
c251b41 init
7f6c6fb update
```

Submission:

This concludes the project.

- Please review this document to make sure you entered all the required response fields in all four sections.
- Download this document as a PDF file.
- Push this pdf file into your fork repository