Question 3

Step 1: Create a feature branch.

```
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment> git branch -a
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment> git branch
* main
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment> git branch feature
```

Step 2: Switch to the new branch.

```
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment> git branch
  feature
* main
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment> git checkout feature
Switched to branch 'feature'
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment>
```

GIT_assign > git_assignment > ≡ q1.txt

Step 3: open the file and make some changes to it.

```
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment> git status
On branch feature1
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:   f1.txt
        modified:   q2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ff1.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment>
```

Step 4: Add and commit the changes to the new branch.

```
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment> git add .
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment> git commit -m "q2.txt modified"
[feature 1227c8b] q2.txt modified
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Step 5: Push the changes to the new feature branch.

```
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment> git push origin feature
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 276 bytes | 276.00 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 https://github.com/kalai1217/git_assignment.git
   5ff37c7..1227c8b  feature -> feature
```

Step 6: Create a pull request.
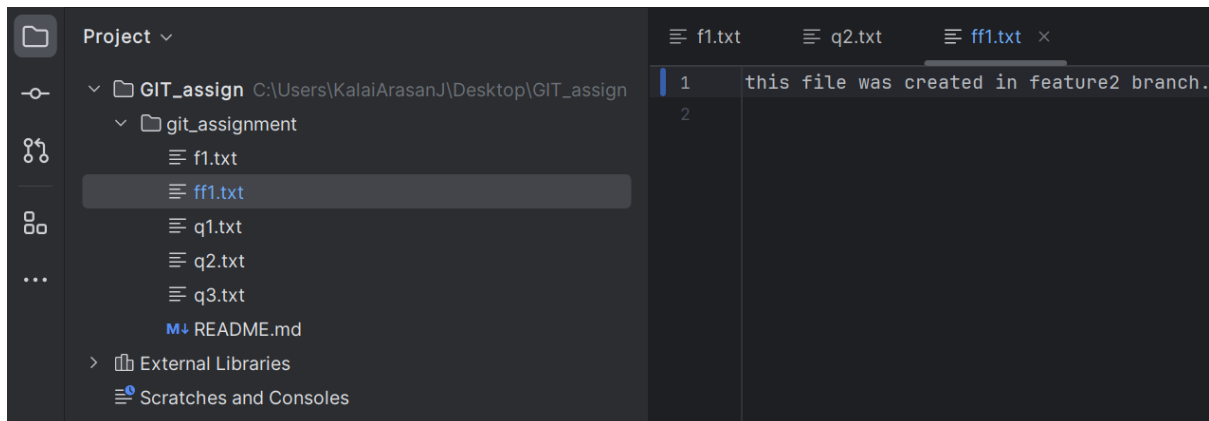
Step 6: As another user in the master branch make some changes to the same file.

Step 7: Add and commit the changes to the master branch.

```
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment> git add .
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment> git commit -m "q2.txt modified"
[feature 1227c8b] q2.txt modified
 1 file changed, 1 insertion(+), 1 deletion(-)
```
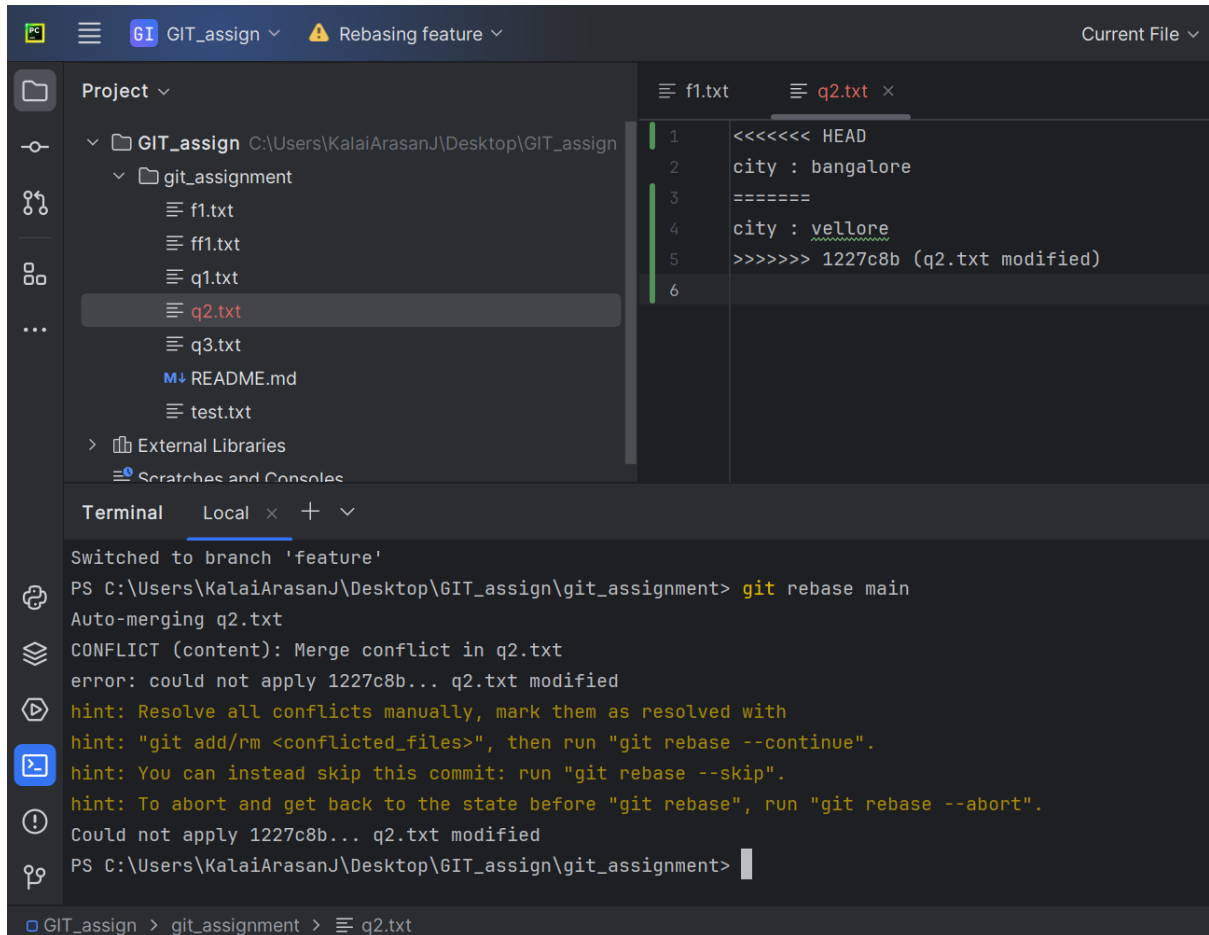
Step 8: Push the changes to the master branch.

```
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment> git push origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 550 bytes | 550.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/kalai1217/git_assignment.git
   137e38f..f4129f8  main -> main
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment> git checkout feature
Switched to branch 'feature'
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment> git rebase main
Auto-merging q2.txt
CONFLICT (content): Merge conflict in q2.txt
error: could not apply 1227c8b... q2.txt modified
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
Could not apply 1227c8b... q2.txt modified
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment> git status
interactive rebase in progress; onto f4129f8
Last command done (1 command done):
```

Note : There will be a conflict in the pull request, how do we resolve it??
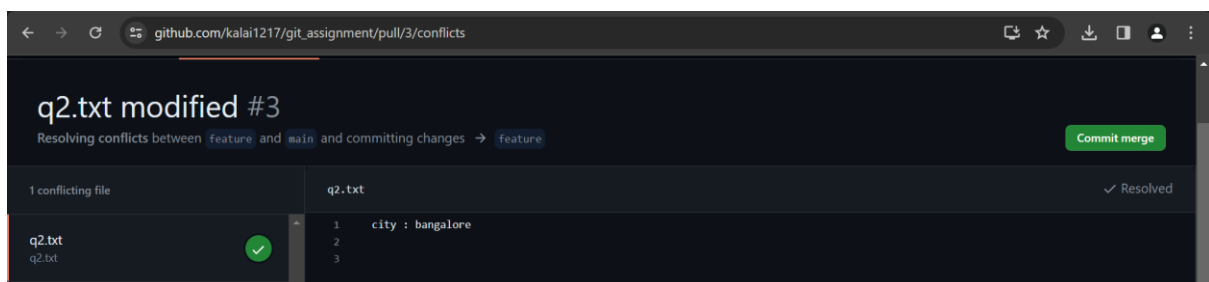
Hint: **git rebase**

We can see here that q2.txt has been **over written with different values** for the same attribute city.
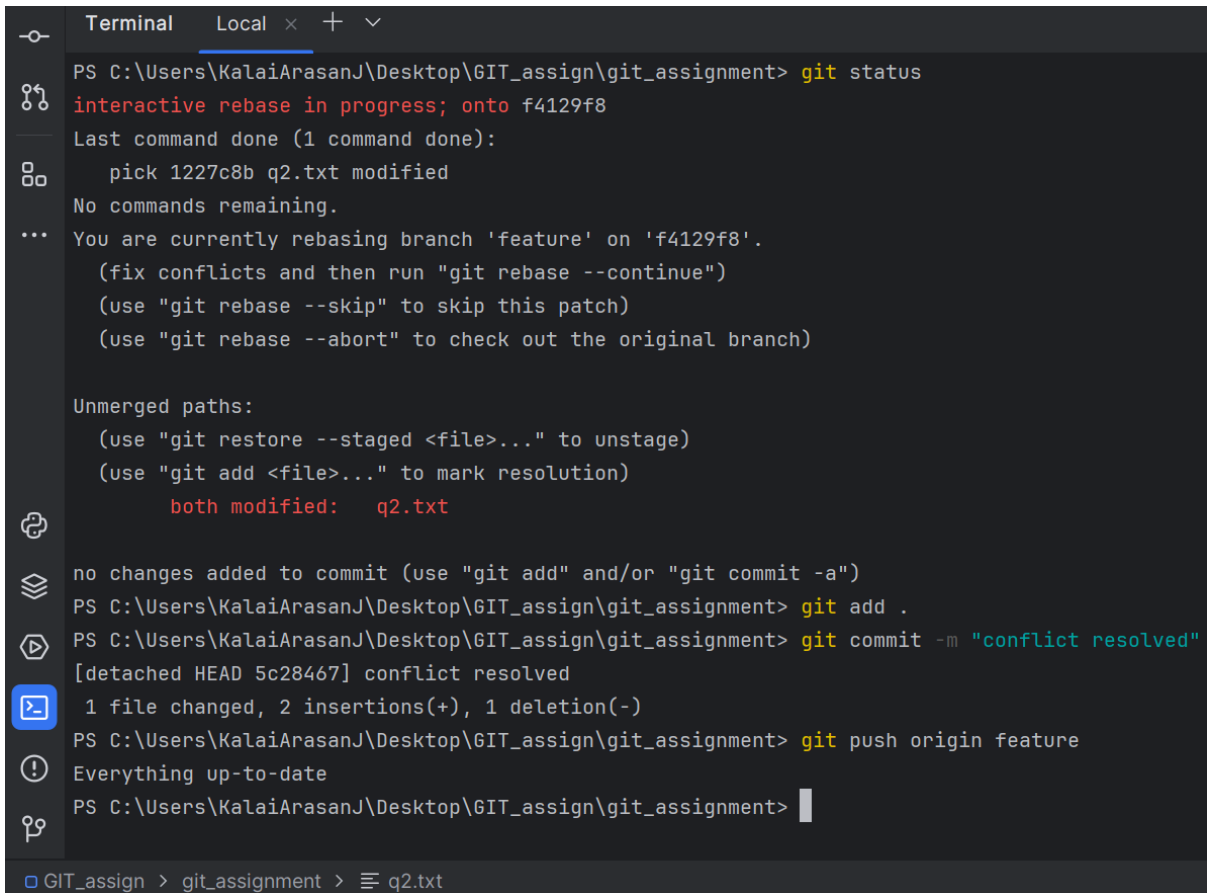

In order to **resolve conflict,** we have to justify the value for the attribute city, In this case the value for city is set to Bangalore.

After resolving conflicts, we can stage the file in the staging area and push into the git repo.

```
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment> git status
interactive rebase in progress; onto f4129f8
Last command done (1 command done):
    pick 1227c8b q2.txt modified
No commands remaining.
You are currently rebasing branch 'feature' on 'f4129f8'.
  (fix conflicts and then run "git rebase --continue")
  (use "git rebase --skip" to skip this patch)
  (use "git rebase --abort" to check out the original branch)

Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
        both modified:   q2.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment> git add .
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment> git commit -m "conflict resolved"
[detached HEAD 5c28467] conflict resolved
 1 file changed, 2 insertions(+), 1 deletion(-)
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment> git push origin feature
Everything up-to-date
PS C:\Users\KalaiArasanJ\Desktop\GIT_assign\git_assignment>
```

GIT_assign > git_assignment > q2.txt