**Instruction**

Joe, the proud owner of the Git-Café is now happy that he is able to track his recipe files using git as a version control.

With the increased efficiency that git provided him, he is able to focus more on his recipes and managing his customers better at his Café. He is even considering opening new branches of his Café with some of his connoisseur friends joining him as partners.

Joe and his friends could together work on new interesting recipes to continue impressing their customers. They could all pick and choose their respective branch menus from a central collection of recipes that they would all create together. Additionally, each branch would also have one signature recipe that would be exclusively served at that branch alone.

Now all of this needs a lot of collaboration. And they also need accessibility of the complete work from any of their locations.

However, Joe is currently struggling with following problems

* Joe is having to operate from only his Office desktop where his local git repository is present. He wants to move his repository to a central server from where all his friends could collaborate.
* Joe wants a structured way to help his friends collaborate to his collection of recipes while still having their own private place to manage their menus and signature recipes.

Create a new GitHub Repository for Joe’s Café recipes. Push the local repository from Joe into the Online GitHub repository.

Assume there are 2 new café - branches that Joe opened. Create a new file for each of the café – branch menus using GitHub interface.

Assume you are one his partner- friends. Create a new topic branch for yourself. Submit a new\_cool\_recipe.txt to his collection through a GitHub pull request and trigger a code review process.

Finally ensure there are no merge conflicts and merge your new topic branch to the main branch – everything on GitHub.

**Key Skills**

The intent of this assignment is to give you a thorough practice on the following Git features:

1. Creating a new GitHub Repository.
2. Setting up a remote and Pushing a local git repository to GitHub.
3. Creating, editing files using GitHub interface.
4. Cloning a remote repository and making changes.
5. Keeping Local and remote repositories in sync.
6. Working with GitHub branches.
7. Raising a pull request.
8. Viewing commit history and commit diffs between files & branches using GitHub.
9. Merging branches on GitHub.

**Task Checklist**

* Create a GitHub Account.
* Create a new GitHub Repository with name “Joe\_gitcafe”.
* Configure remote for the local git repository gitCafe and push the changes to “Joe\_gitcafe” on GitHub.
* Commit a new file café\_branch1\_menu.md file at the root of the local git repository with title “Joe’s café - branch1 menu” and Push the file to GitHub.
* Using GitHub interface, commit another new file café\_branch2\_menu.md file at the root of the local git repository with title “Joe’s café – branch2 menu”.
* Create a new branch called “signature\_recipes” from GitHub interface.
* Pull the latest changes to local repository using git pull.
* Add a new file signature\_latte.txt into the branch by adding and committing it under recipes folder.
* Push the new branch and the changes into GitHub repo.
* Raise a pull request from the branch “signature\_recipes” to “master”. Include one of your friends as a Code reviewer adding their GH handle (s) as reviewers.
* Have a code review comment provided to modify the Menu file by adding the reference to the new signature\_latte into the menu.
* Address the code review comment by including the suggested change and resolving the comment.
* After all the comments are resolved, have the reviewers approve the pull request and finally merge the changes.

# Navigate to local repository

cd /path/to/gitCafe

# Add GitHub remote and push

git remote add origin https://github.com/YOUR\_USERNAME/Joe\_gitcafe.git

git push -u origin master

# Create and commit branch1 menu file

echo "# Joe’s café - branch1 menu" > café\_branch1\_menu.md

git add café\_branch1\_menu.md

git commit -m "Add café\_branch1\_menu.md"

git push origin master

# Create branch2 menu file on GitHub

# Create and switch to a new branch

# This is done on GitHub interface, no command required here

# Pull latest changes

git pull origin master

# Add signature recipe file

mkdir -p recipes

echo "Signature Latte Recipe" > recipes/signature\_latte.txt

git add recipes/signature\_latte.txt

git commit -m "Add signature\_latte.txt"

git push -u origin signature\_recipes

# Create a pull request on GitHub

# This is done on GitHub interface, no command required here

# Modify branch1 menu file (if required)

nano café\_branch1\_menu.md

git add café\_branch1\_menu.md

git commit -m "Update café\_branch1\_menu.md to include signature\_latte"

git push origin signature\_recipes

# Merge pull request on GitHub

# This is done on GitHub interface, no command required here