# Git assignment

Through this assignment we will be learning following concepts:

1. Creating an account with Github and configuring SSH key for working with SSH protocol
2. Creating a new git repository from scratch
3. Linking local repo with the remote repo
4. Pushing local branches to the remote repo and setting upstream branch
5. Working with multiple branches by creating separate feature branch for each of the application features
6. Creating pull requests, doing code reviews and one merging branches on remote github repo
7. Resolving merge conflicts and updating pull requests
8. Tagging source code for release

# Problem statement:

Build a simple calculator application using HTML5, CSS3, Javascript and Bootstrap4 framework. Your project code must be hosted on a private Github repository. Add “SreeharshaJois” as the collaborator for your project.

Steps to be followed for completing the assignment:

1. Create a new repository with name “SimpleCalculator” in your Github account. (Name of the repository must be “SimpleCalculator”)
2. Create an directory on your local machine with name SimpleCalculator (you can’t use any other name for your repository)
3. Add 3 empty files with names calculator.html, calculator.js and calculator.css into the SimpleCalculator directory
4. Convert SimpleCalculator into a git repository
5. Stage all the newly added files and commit with message “Initial commit containing empty files for html, css and js” (Note: Use the exact message)
6. Setup Github SimpleCalculator repository as the remote repository for the local SimpleCalculator repo
7. Push local “master” branch to the remote repository by making remote “master” branch as the upstream branch
8. Create a new branch with name “develop”
9. Update calculator.html file with 3 input elements and commit with message “Added card layout with 3 input elements, 2 for operands and 1 for result”

Note: Get the contents from the second commit of the template repo

1. Push the local develop branch to remote repo with remote “develop” branch as the upstream branch
2. Create 6 separate branches from the develop branch for header, footer, add, subtract, multiply and divide features. All feature branches must have “feature/” as prefix i.e, feature/header, feature/footer etc
3. Switch to header branch to add the header related code
4. Stage all the changes and commit with message “Adding application header”
5. Push the feature/header branch to remote repo by setting up the upstream branch
6. Raise pull request with feature/header as source branch and develop as the target/destination branch
7. Checkout to footer branch and add footer feature related code
8. Stage all the changes and commit with message “Adding application footer”
9. Push the feature/footer branch to remote repo by setting up the upstream branch
10. Raise pull request with feature/footer as source branch and develop as the target/destination branch
11. Checkout to add branch and add the addition related code
12. Stage all the changes and commit with message “Adding addition feature to the app”
13. Push the feature/add branch to remote repo by setting up the upstream branch
14. Raise pull request with feature/add as source branch and develop as the target/destination branch
15. Checkout to subtract branch and add the subtraction related code
16. Stage all the changes and commit with message “Adding subtract feature to the app”
17. Push the feature/subtract branch to remote repo by setting up the upstream branch
18. Raise pull request with feature/subtract as source branch and develop as the target/destination branch
19. Checkout to multiply branch and add the addition related code
20. Stage all the changes and commit with message “Adding multiply feature to the app”
21. Push the feature/multiply branch to remote repo by setting up the upstream branch
22. Raise pull request with feature/multiply as source branch and develop as the target/destination branch
23. Checkout to divide branch and add the addition related code
24. Stage all the changes and commit with message “Adding divide feature to the app”
25. Push the feature/divide branch to remote repo by setting up the upstream branch
26. Raise pull request with feature/divide as source branch and develop as the target/destination branch
27. Merge feature/header branch into develop branch after code review
28. Resolve merge conflicts in feature/footer branch and push the code to the upstream feature/footer branch
29. Merge remote feature/footer branch to remote develop branch
30. Resolve merge conflicts in feature/add branch and push the code to the upstream feature/add branch
31. Merge remote feature/add branch to remote develop branch
32. Resolve merge conflicts in feature/subtract branch and push the code to the upstream feature/subtract branch
33. Merge remote feature/subtract branch to remote develop branch
34. Resolve merge conflicts in feature/multiply branch and push the code to the upstream feature/footer branch
35. Merge remote feature/multiply branch to remote develop branch
36. Resolve merge conflicts in feature/divide branch and push the code to the upstream feature/footer branch
37. Merge remote feature/divide branch to remote develop branch
38. Raise a pull request with develop as source branch and master as destination branch
39. Merge the pull request
40. Update local develop and master branches
41. Checkout to master branch
42. Create an annotated tag with name “v1.0” and message “First version of SimpleCalulator app with add, subtract, multiply and divide features”
43. Push the tag to the remote repo

* Repo exists and name is proper or not
* All branches are created as per the guidelines or not
* Code in each branch exactly matches the template repo’s respective branch. Also the number of commits in each branch matches the template branch or not
* Commit with merge conflict resolution message
* Tag been created or not and whether it is on master branch or not
* Total number of commits in develop branch matches the template develop branch commit count or not

Create github account