Complete Git and GitHub Masterclass : Beginner to Git Expert Whole practice Assignment

Consider you are working in a development project in a team and want to collaborate among the team. You are creating some new features for the application and want to share your work with the team. Use Git and Github as version control tools for the project to keep track of all the modifications that you make to your code.

Create a new repository for the project using Git to store all the project files. Use git add command to stage a new file. Use “git commit” command with commit message to commit changes in the git repository. Check the status using “git status” command. Check commit history in Git Repository using git log.

Modify the file in the git repository and check the status using –s option. Add the file to staging area and check the status. Use git commit command with message. Use git log command to check commit history.

Create a branch using “git branch branch\_name” command. Make the branch active. Use “git log –oneline” command”.

Update the file multiple times in the git repository and use “git commit –am” command to add and commit. Check the commit history and perform git checkout. Use git checkout command on file name and perform git commit. Use git revert command to revert the changes. Modify a file and use git reset command to undo the changes.

Create an account in GitHub and create a new repository. Use git pull command to pull changes from GitHub and git push command to

push local changes into GitHub repository. Push all the changes on your local repository to the remote repository.

Edit the file in remote repository in Github. Commit the changes. Use git pull command in order to sync changes to the local repository.

Edit a file in the local git repository. Add changes to staging area and commit the changes. Use git push command to push the changes to the remote repository.

Create a feature branch in the git repository using git checkout –b command and modify the file. Commit the changes. Switch to the master branch. Use git diff command to view differences between master branch and feature branch. Use git difftool command to view differences in the p4merge tool.

Create a branch and modify a file. Add and commit changes and check out to the master branch. Modify the same file at the same places. Use p4merge tool to resolve the merge conflicts. Perform commit and git log to view the commit history.

Create annotated tags in the git repository and compare the tags using git diff command. Update a tag and delete a tag. Push annotated tags only to remote repository in GitHub.

Use git stash – u command to stash the changes for tracked and untracked files. Create a branch from the stash using git stash branch command to move the changes to new branch. Add changes to staging area and commit the changes and merge the branch to master.