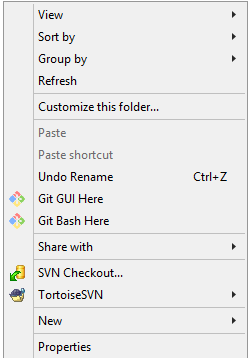
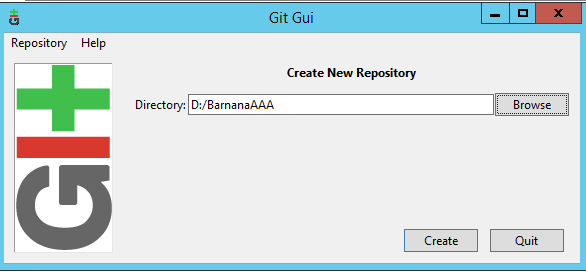
***Topics covered in Git:***

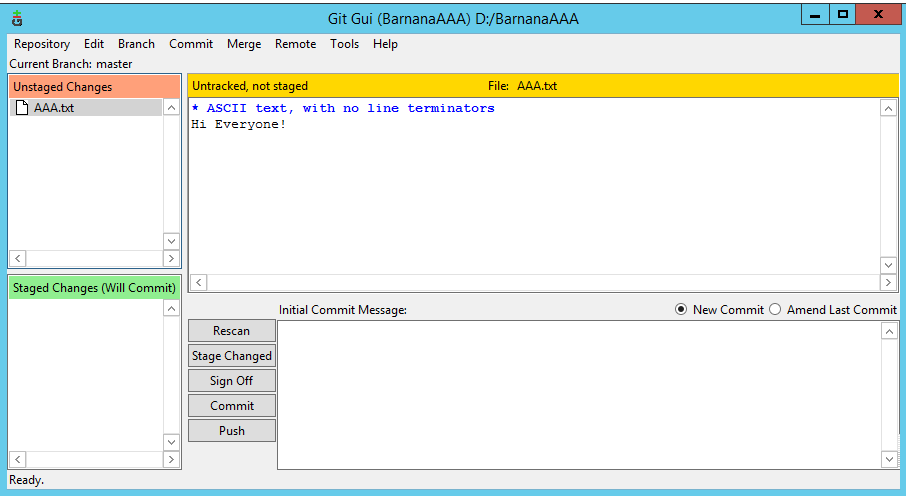
1. Git is downloaded for windows from the URL: <https://git-scm.com/download/win>
2. The commits can be made through two different ways. They are:
3. Git GUI.
4. Git Bash.
5. In order to start with Git GUI we have created a folder and named as BarnanaAAA. As of now, we didn’t add any of the documents in the folder, then on the right click we get the options as *Git GUI Here* or *Git Bash Here* as shown in the screenshot:



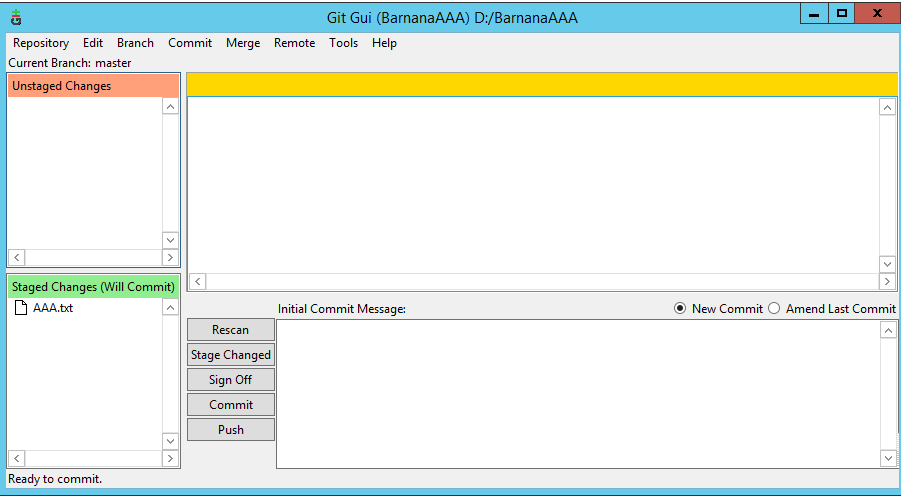
1. On the click of *Git GUI Here* we get navigated to Git GUI where we need to create a repository for our folder. On click of Create New Repository we need to give the address and click Create as shown in the screenshot:



1. Initially there will not be any changes available in the Git Gui, then we created a text file AAA.txt. On click of Rescan in Git Gui we are able to see our modified changes and it will be available in the Unstaged area.



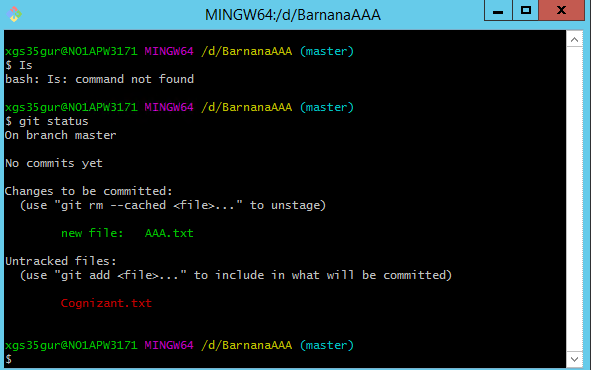
1. In order to stage the text document we need to navigate to commit 🡪 Stage To Commit then the changes will be moved to Staged changes from the Unstaged changes.



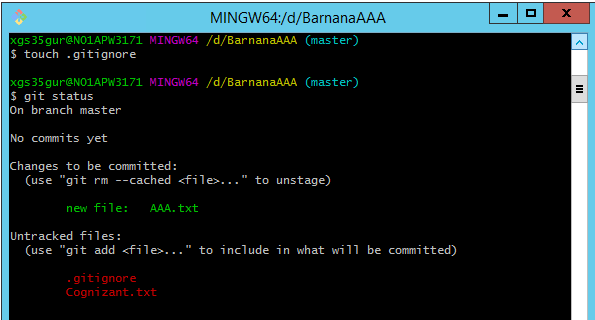
1. In order to work with *Git Bash Here* we need to left click in the folder where we are making changes and then click Hit Bash Here.
2. Basically there are three areas in Git. They are as follows:

* Working Directory.
* Staging Area.
* Committed File.

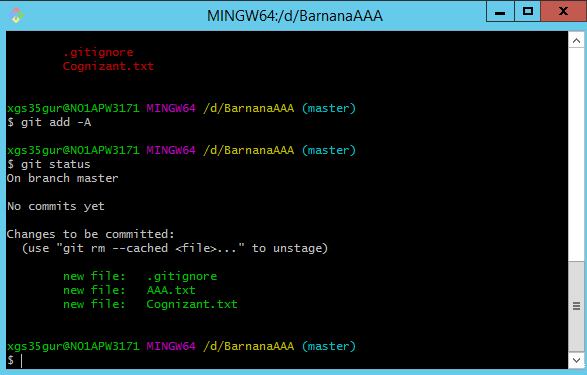
1. The place where we make all are changes is termed as Working Directory, all the untracked and the modification of files will be done here, whereas staging area is the one where we want to organize and want to commit to the repository. In case we have lots of files to be committed then in staging area we can have chunks of the files and then commit it which will lead to the easier way of committing.
2. In order to know, what all files are available in our working directory we can give the command as git status the screenshot for the same is attached below, which tells that two files are available out of which AAA.txt file is already in the staging area whereas the Cognizant.txt file is not available in the staging area, its available in the working directory itself:



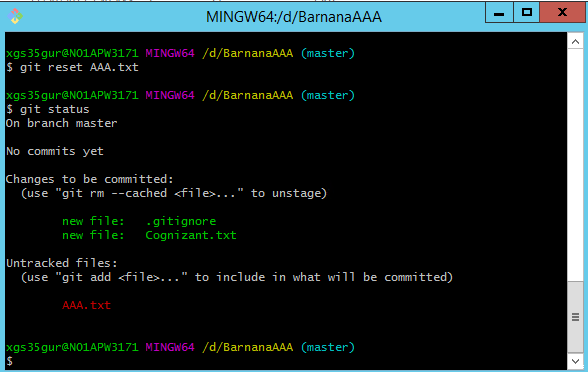
1. In some scenarios, we would make some code but we won’t be sharing it to the other users so we can hide those files by the command touch .gitignore, and the .gitignore file will be created automatically in our folder path. For example:- we have created another file Barnana.zip and in the .gitignore file we have written \*.zip, which indicates that the files having an extension as zip will not be shown in the staging area after giving the command git status. The screenshot for the same is shown below:



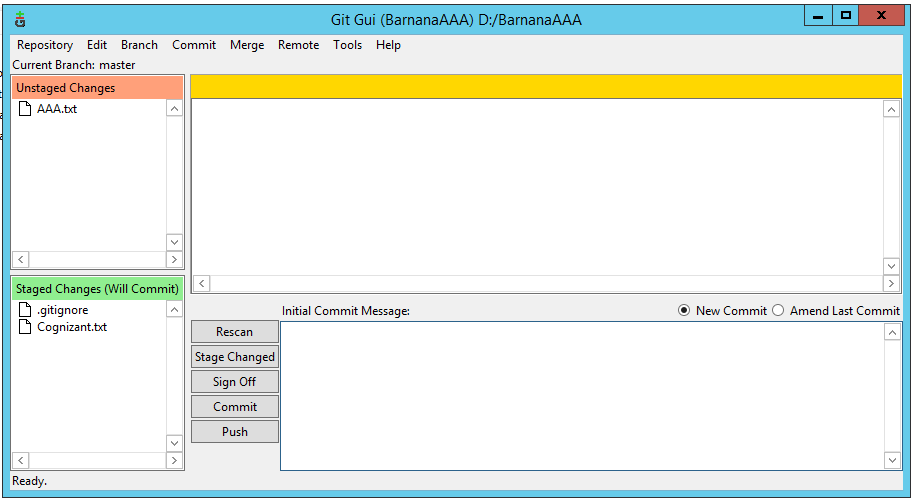
1. From the above screenshot it’s visible that Barnana.zip file is not available in the working directory or staging area. Now, in order to add the files .gitignore and Cognizant.txt file in the staging directory we will use the command git Add –A. These will make the files available in the staging area. The screenshot for the same is shown below, which shows that all the three files are available in the staging area:



1. In order to remove files from the staging area we need to give the command as git reset <filename> the example for the same is shown below:

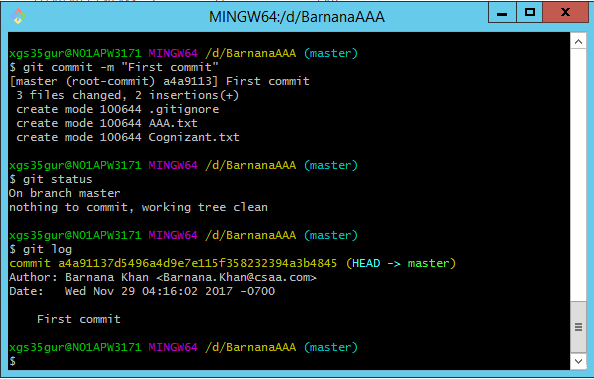


git reset AAA.txt command brings back the file into the staging area, and the status is viewed by giving git status.



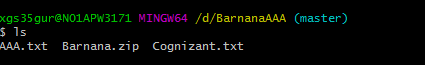
Above screenshot shows the UI for GIT GUI Here which get updated symmetrically.

1. Once again after putting AAA.txt file to staging we are now ready to commit the files to the local repository. The command to commit the files is: git commit –m “comments” the example for the same is shown below:

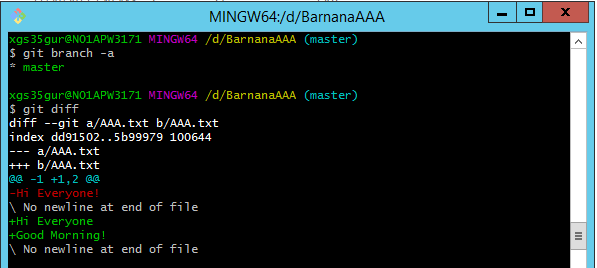


In the above screenshot, we can see that the git commit is made and all files are available in the local repository, then on giving the git status command we can see that there is nothing to commit. In order to check who the details of the user who committed we use the command git log, we also get the unique number for the particular commit.

1. In order to check all the files available in the local repository we can give ls and all the files will be listed as shown in below screenshot.

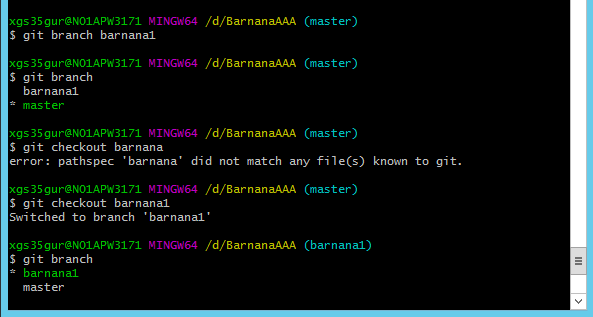


1. In case we want to create branch and do our changes there then we use the command git branch <branchname> in order to see the branches available and the branch in which we are working we give the command as: git branch –a and make changes there. In AAA.txt file we have removed the ! sign and have added another line Good Morning in the text file. In order to see the changes we give the command git diff then on click of git status we could see the AAA.txt file has been modified and is available in the staging area screenshot for the same is given below:



The \* sign available before master tells that we are currently working on the branch master.

1. We have created a branch with the name git branch barnana1 in order to work in the branch barnana1 we will give the command git checkout barnana1, which is shown in the screenshot:

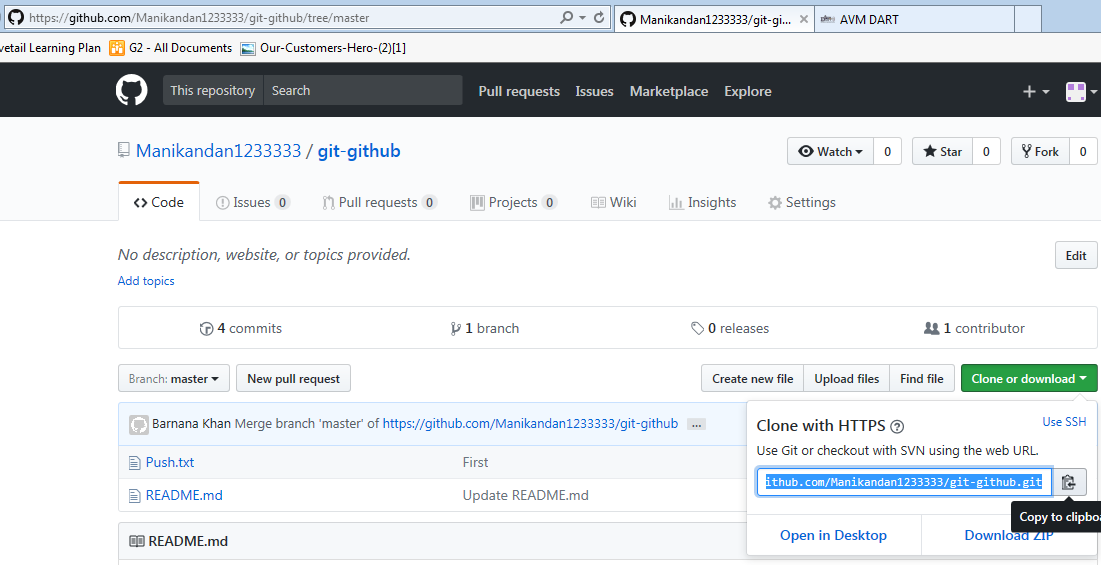


We can see in the above screenshot that the \* sign is in front of the barnana1 branch which indicates further changes will be made in the branch barnana1.

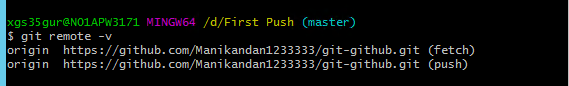
***Steps to migrate code to Remote repository:***

In order to migrate the code from local repository to remote repository we need to have an account in the GitHub. Hence, we have created an account in GitHub.

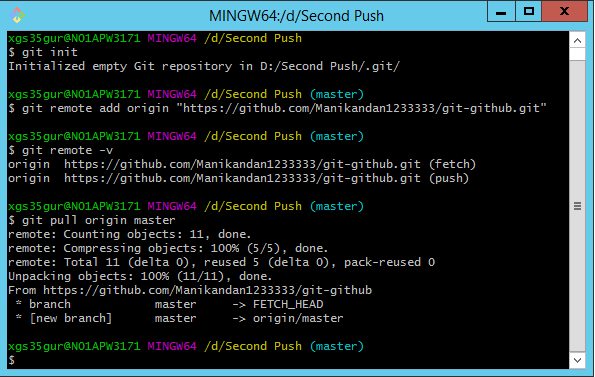
1. Navigate to the URL: <https://github.com> , which will let us to the page where we can create our own account.
2. We have created our account with the User Name: Manikandan1233333 and the email address [Manikandan.Guruparan@csaa.com](mailto:Manikandan.Guruparan@csaa.com) and the password as \*\*\*\*\*\*\*\* (ent password).
3. Now in order to transfer files from local repository to remote repository we need to establish a connection between the two repositories. The connection is made on the basis of the command git remote add origin <remote URL>. We will get the remote URL by navigating to the GitHub windows and clicking on Clone or Download we need to copy the URL and paste it in command window. The GitHub UI where the link is available is shown in the screenshot:



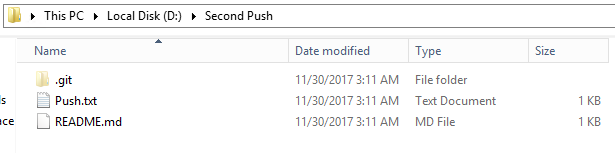
1. To check the remote repository of the particular solution we give the command as: git remote –v as shown in the below screenshot.



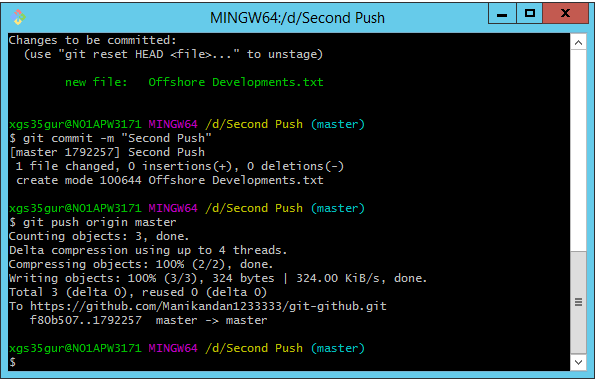
1. Now, we can either pull the documents from the repository and can also push to the repository. To pull the documents from the repository we give the command as git pull <origin name> <destination name>. In our case we have given the command as git pull origin master. This command will get all the documents that are available in the remote repository. The example for the same is shown in the below screenshot:



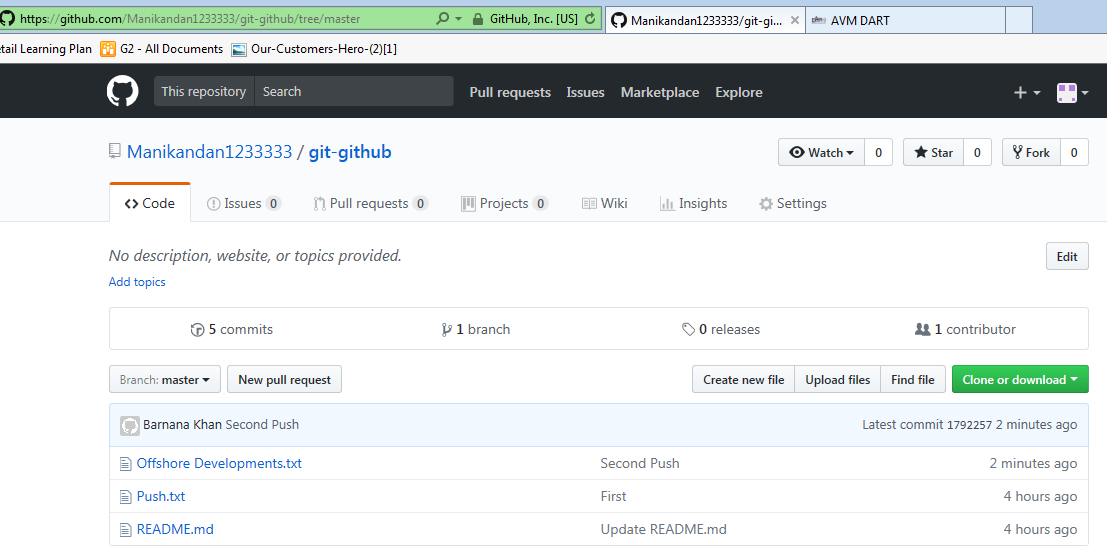
After giving the pull commands we can find that the documents available in the remote repository will be available in our folder path as shown in the screenshot:



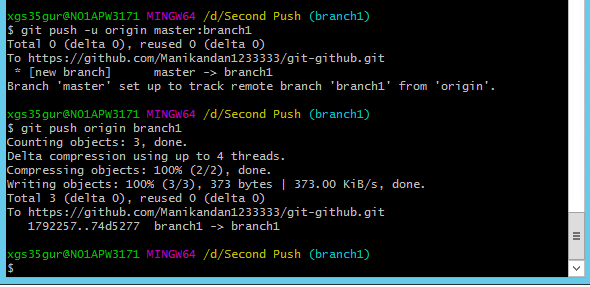
1. Moreover we can also push the documents to the remote repository, we use the command git push <origin name> <destination name>. For an example we have created a txt document in our path with the name “Offshore Developments”. Now, first we need to add the file in staging area then we need to commit it and then we will push it to the remote repository. The procedure is shown in the below screenshot:



1. Now, on refreshing the GitHub we can see our file in the GitHub repository. The UI for GitHub is shown in the below screenshot:

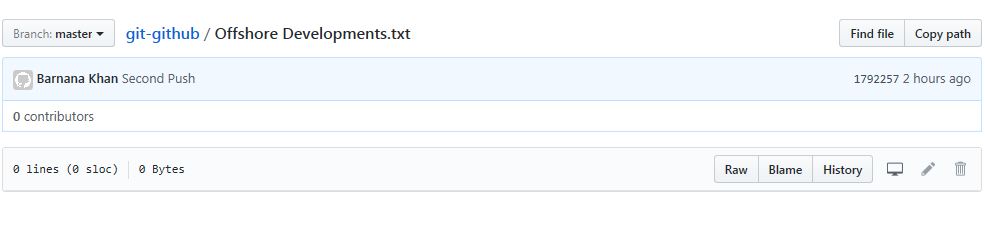


1. If we want to create branch and do our changes in the branch and then update it in remote repository then we need to first create a branch in local repository and then in that branch we should have our changes. In order to update the branch in GitHub We need to run the command as: git push –u <origin name> <destination name>:<new branch name> . The example for the same is shown in the below screenshot:

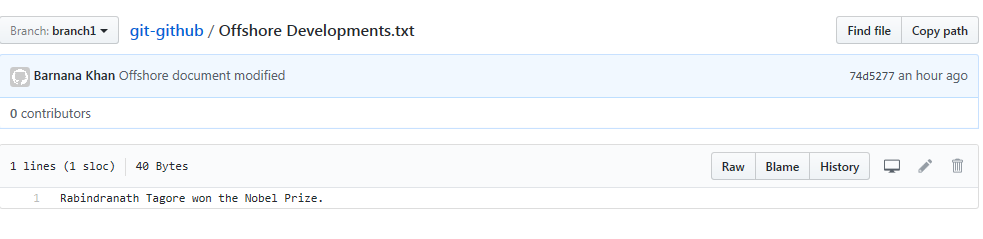


Some changes were made in the folder Offshore Developments.txt under the branch name branch1 and it was committed to the local repository. Now when we give the command git push origin branch1 then it will create the branch in the remote repository.

In the GitHub, under the branch name master if we open the file Offshore Developments.txt then we can find that the file is empty as shown in the screenshot:

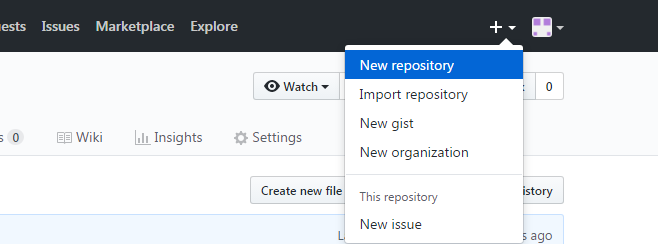


Whereas, under the branch name branch1 if we open the file Offshore Developments.txt then we can find that the file is having the modified version that was created under the branch branch1. The screenshot for the same is given below:

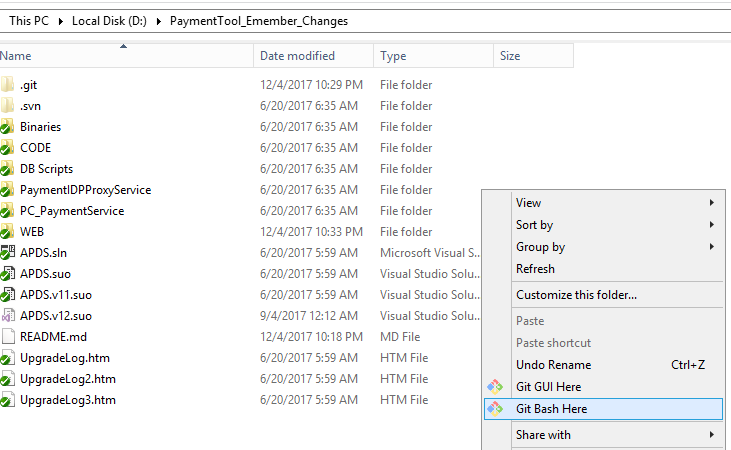


***Working with Visual Studio:***

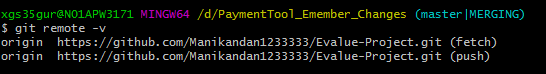
In order to commit the changes made in the code in Visual Studio in the GitHub, we have created a new repository as shown in the screenshot in GitHub and named as Evalue\_Project.



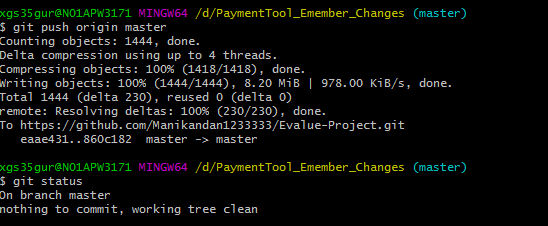
Then on navigating to the folder where the Solution is present, we opened the Git Bash Here as shown in the below screenshot.



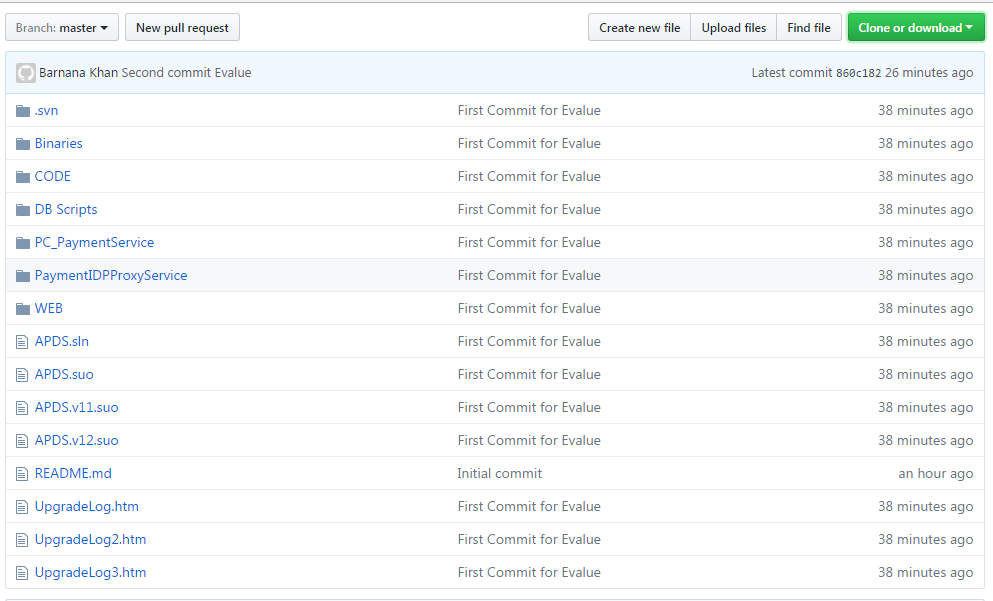
Then we have created the connection between the visual studio and the GitHub on the basis of the command git remote add origin “<https://github.com/Manikandan1233333/Evalue-Project.git>” .In order to check whether the branch got established or not we give the command as git remote –v the output for the same is shown in the below screenshot:



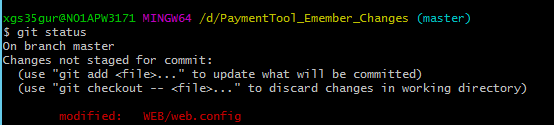
After establishing the connection, we committed the solution in the local repository with the comment “First commit for E-value”. Then in order to push the solution to the Git Hub we used the command git push origin master. The screenshot for the same and the updated documents in the GitHub is shown in the below screenshots.



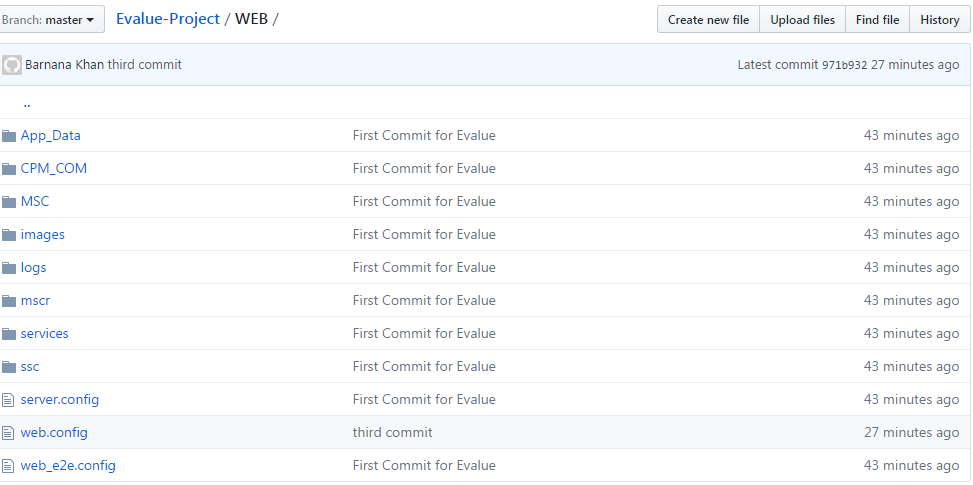
GitHub screenshot after first push:



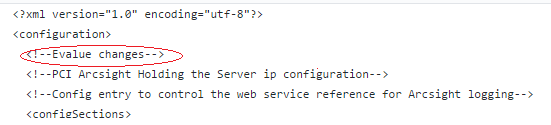
Now we made some changes in web.config file and checked in the same to GitHub. After making changes in the respective file we navigated to Git Bash Here window and on checking the status we could see that one file has been modified.



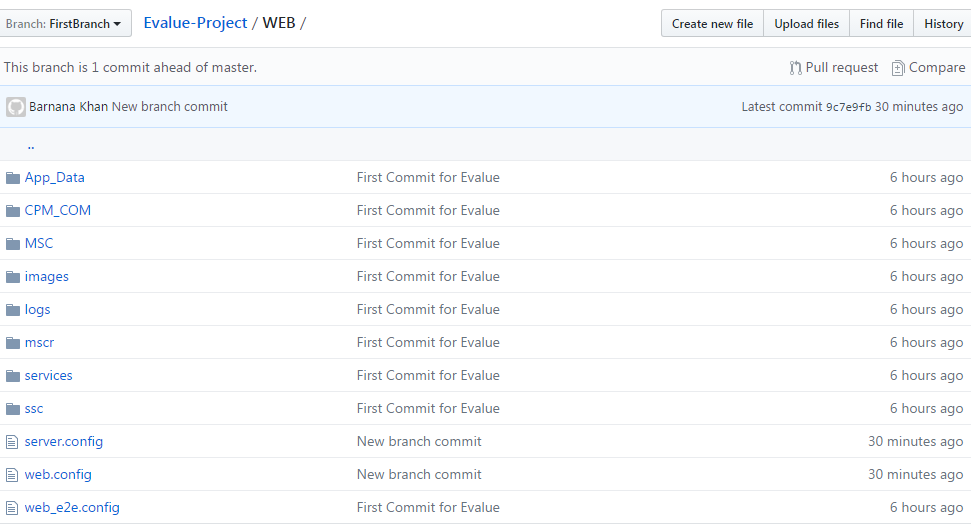
Then on committing to the local repository with the comment “third commit” and again pushing the solution in the GitHub we get the updated form in the GitHub. The screenshot for the same is attached below:



On opening the web.config file we could see the changes done in visual studio available in the GitHub. The screenshot for the same is attached below, the red circled part is the change that was made in order to test the scenario:



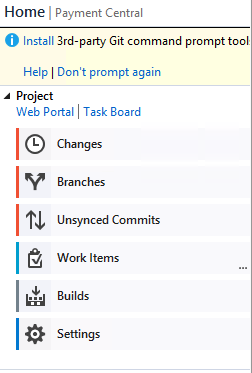
Moreover, we have also created branch with the branch name ***FirstBranch*** and with the comment New Branch Commit have made changes with respective to that branch. The screenshots for the same is attached below:



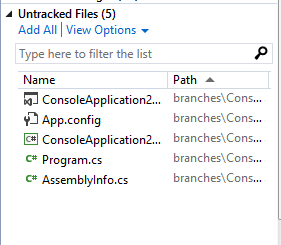
***Connecting Visual Studio and VSTS:***

In order to connect to Visual Studio with VSTS we need to navigate to the Visual Studio solution and we need to open Team Explorer, and then we need to create a local repository the path for our local repository is: C:\Users\xgs35gur\Source\Repos\PAYMENT\_TOOL in the Payment Tool server, where we have cloned all the projects available in VSTS. In order to do that, we need to click on Clone in the Team Explorer and then give the comment whereas the path will be already available, then on click of clone all the solutions available in VSTS will be available in our local repository.

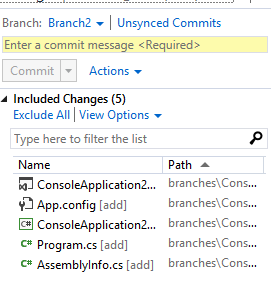
Now if we want to make any changes or add any other solution we will copy the solution folder and paste it in our local repository path. Then on opening the solution, we should navigate to the Team Explorer. In case we want a new branch then we will Home icon and create a new branch. The same is shown in the screenshot:



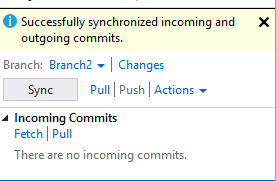
On navigating to the Branches we will create a new branch with our required branch name and will click Create Branch. Then on clicking Changes in the Home Page we can see Untracked Files as shown in the below screenshot:

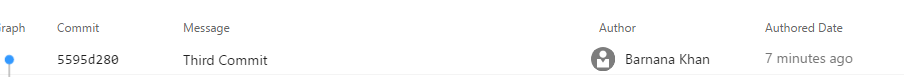


Then we will click Add All. On clicking Add All, our solution will be coming under Included Changes as shown in below screenshot:

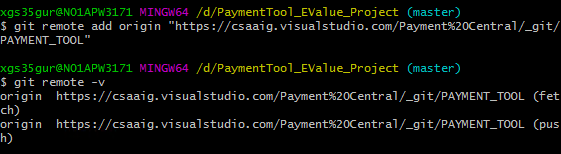


Then we will give some commit Message and will click commit. Then the below screen will appear, then on clicking Sync our files will be available in the remote repository. We will be able to see our files in VSTS after refreshing it. The screenshots are attached below:

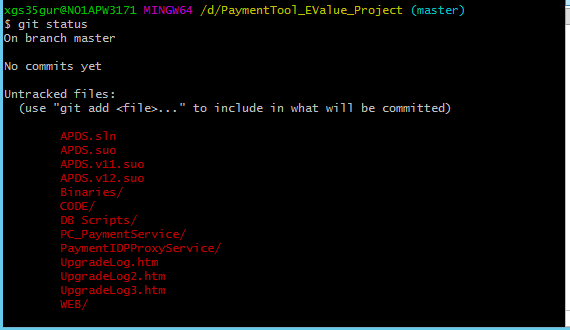




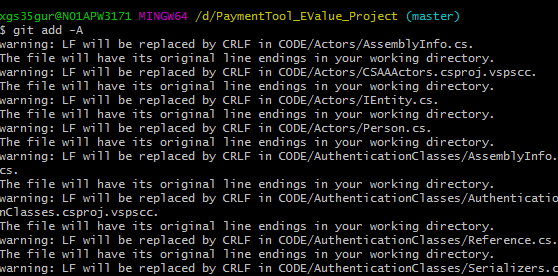
Moreover, we can also add our solution in VSTS on the basis Git Bash. The solution we want to ad we need to navigate to that folder and right click and select Git Bash Here, then the command prompt window will available for us. We need to set up the connection with the VSTS, first we need to create a local repository for which we will give the command git init, then we will establish the remote connection we will give the command as: git remote add origin "https://csaaig.visualstudio.com/Payment%20Central/\_git/PAYMENT\_TOOL” which will establish a connection then on giving command git remote –v we can see the connection. The screenshot for the same is attached below:



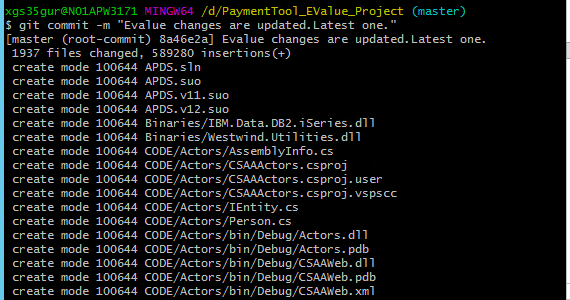
Then we created a branch for our project and names as Evalue. Then we added everything to the staging area and then finally we committed with the message. The entire process is shown in the below screenshot.



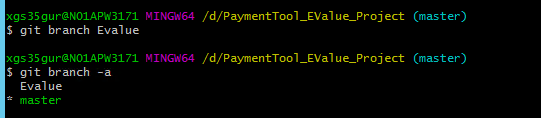
**Adding to staging area:**



**Checking the status and committing:**

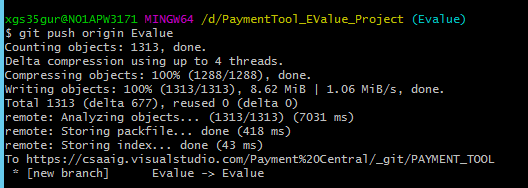


**We created a branch and named as Evalue:**

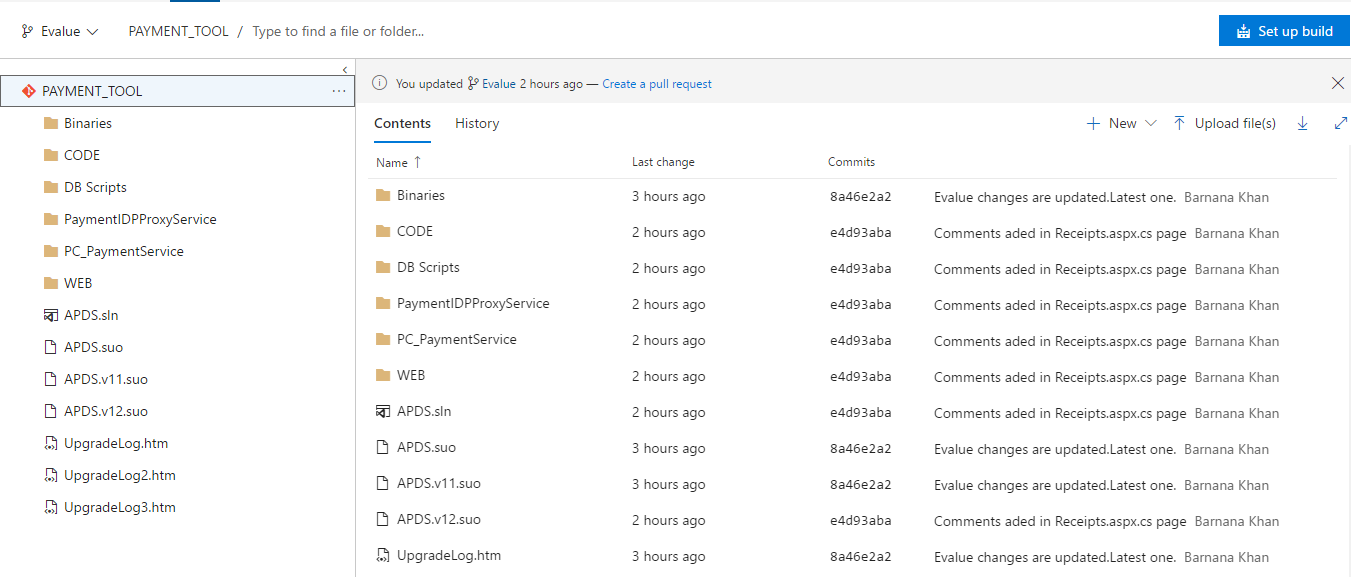




Finally, we pushed it to our remote repository, on giving the command git push origin Evalue we will be asked to login to our VSTS Branch, then we need to give our account credentials and finally all the folders will be pushed in the VSTS. On refreshing the VSTS page, we can see our folders available.

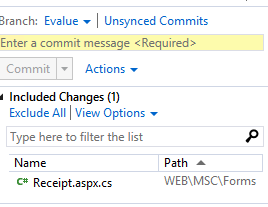


**VSTS:**

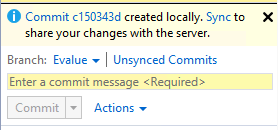


Now in case if we want to make any changes in our Evalue folder then we can also update it in our solution folder and then see the changes in VSTS.

For example: we have made some changes in Receipts.aspx.cs page, then on saving the changes we can see our changes available under Included Changes in the Team Explorer. The screenshot for the same is shown below:



Then we need to give a commit message and enter commit. On clicking commit, we will be asked to sync up with the VSTS as shown in the below screenshot:



Then we need to click on Sync and navigate back to the command prompt and give the command git push origin Evalue. Then we will be able to see our changes in the VSTS on refreshing it.

