**Branching Development Model**

Project 1

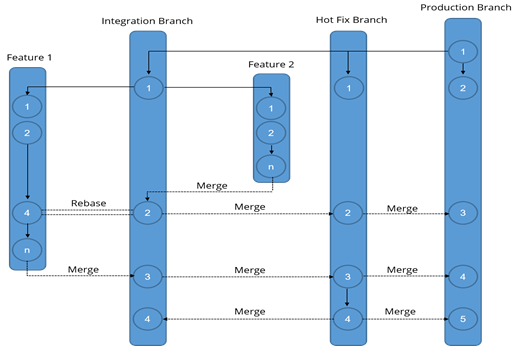
DESCRIPTION

Create a branching model to help your team understand the Git Feature Branch Workflow for faster and efficient integration of work

**Background of the problem statement:**

M-theta Technology Solutions hired you as a DevOps Architect. It is undergoing an infrastructural change to implement DevOps to develop and deliver the products. Since M-theta is an Agile organization, they follow the Scrum methodology to develop the projects incrementally. Hence, the company wants to adopt Git as a Source Code Management (SCM) tool for faster integration of work and smooth transition into DevOps.

So, as a DevOps Architect, you have been asked to build a branching model to demonstrate the Git Feature Branch Workflow for the company’s engineering team. In the branching model, you are required to create a Production branch which will act as the main (master) branch, an Integration branch which will again have two branches inside it namely Feature 1 and Feature 2, and a Hotfix branch which will be used for fixing any issues that could come up from Integration or Production branches.

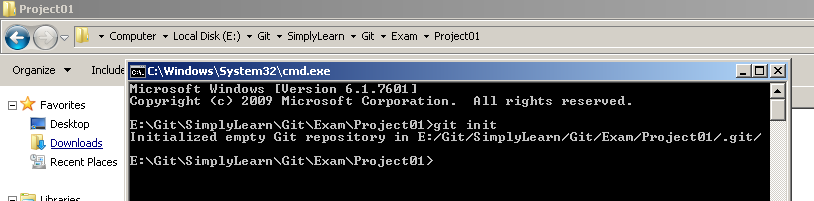


**You must use the following:**

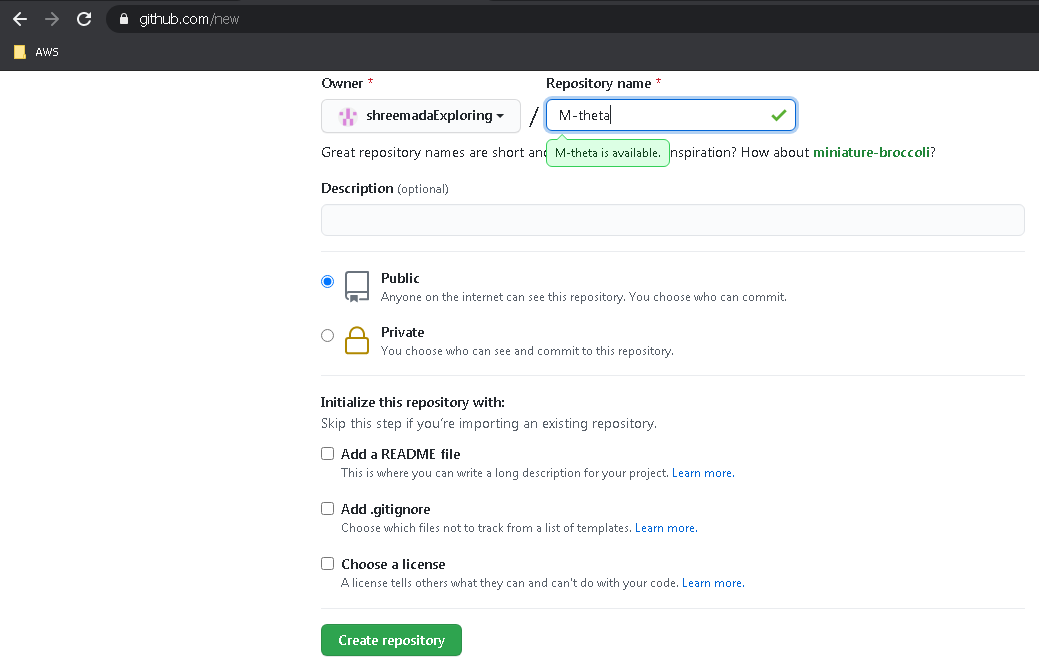
Git: To build the branching model

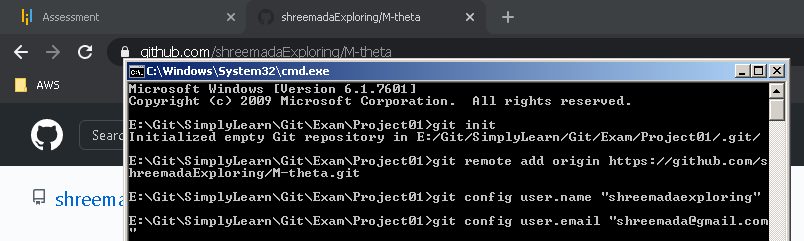
**Steps to perform:**

1. Start with the Production branch (master branch), and then create a HotFix  and Integration branch
   1. Initialize git in folder

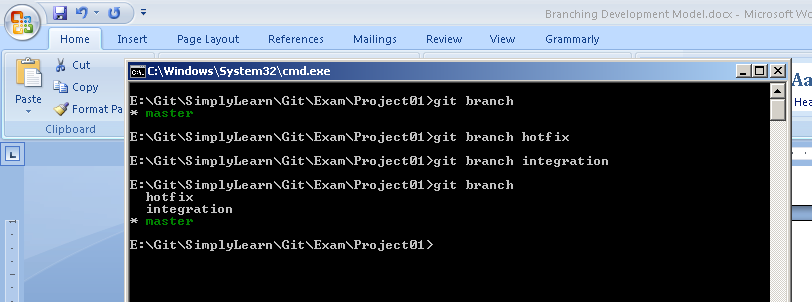


* 1. Create remote repository in git and add to folder





* 1. Create hotfix and integration branch

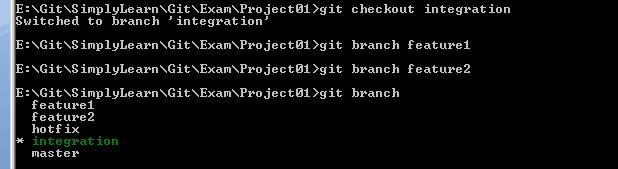


1. Subsequently, create Feature 1 and 2 branches that integrate to the Integration branch as shown in the above figure

**git checkout integration**

**git branch feature1**

**git branch feature2**

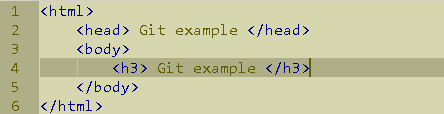


1. Commit some changes in the Feature 2 branch and merge it into the Integration branch. Delete this branch once merging is complete

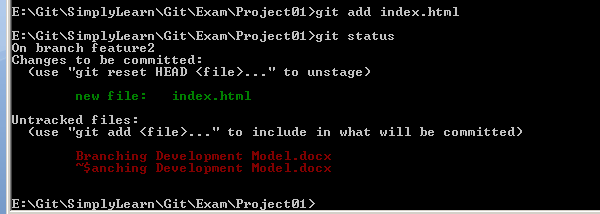
**checkout branch feature2**



create index.html and add following code



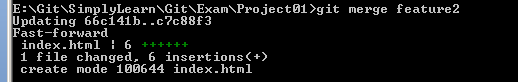
**git add index.html**



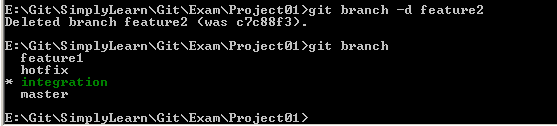
**git commit –m “added index file”** 

**git checkout integration** 

**git merge feature2**



**git branch –d feature2**

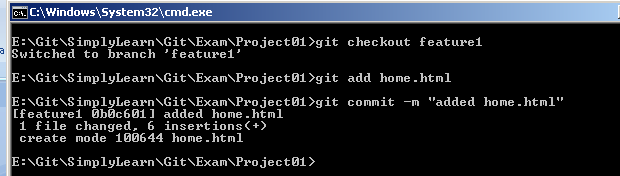


1. Commit some changes in the Feature 1 branch and rebase it to the Integration branch

**git checkout feature1**

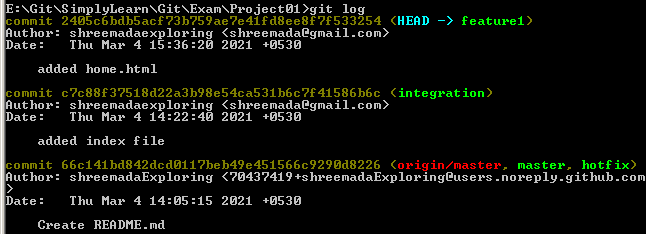
**git add home.html**

**git commit –m “added home.html”**



**git rebase integration**

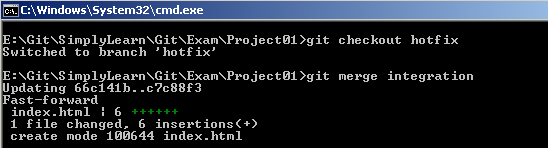




1. Merge the Integration branch into Hotfix and Production branch to update these branches

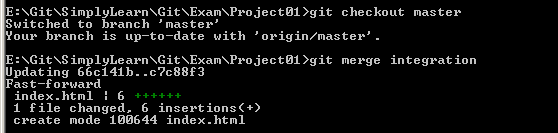
**git checkout hotfix**

**git merge integration**



**git checkout master**

**git merge integration**

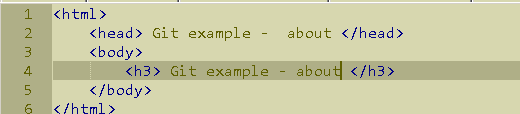


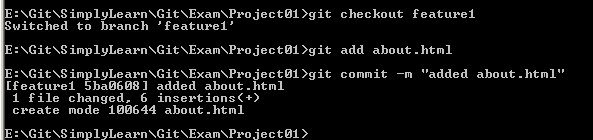
1. Commit some changes in Feature 1 branch, and then merge it into Integration, Hotfix, and Production branch. Delete this branch once merging is complete

**git checkout feature1**

**git add about.html**

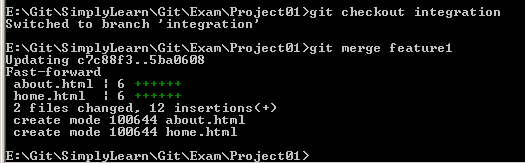
**git commit –m “added about.html”**





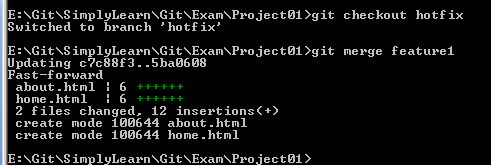
**git checkout integration**

**git merge feature1**



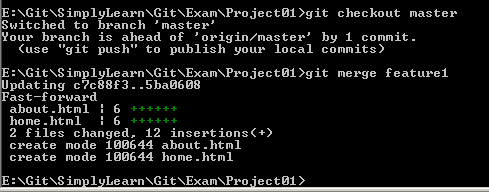
**git checkout hotfix**

**git mege feature1**

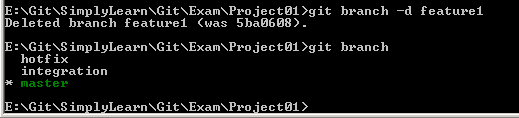
****

**git checkout master**

**git merge feature1**

****

**git branch –d feature1**

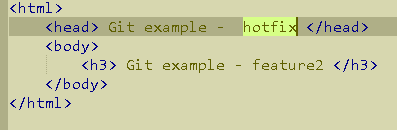


1. Commit some changes in the Hotfix branch and merge it into the Production as well as the Integration branch

**git checkout hotfix**

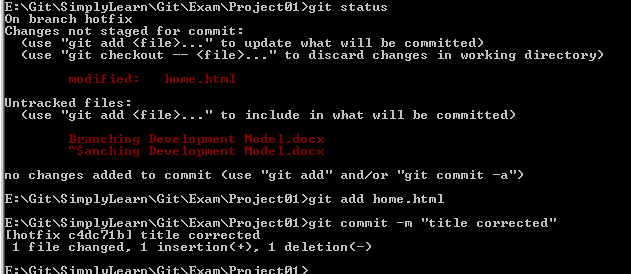


modify home.html – change title to hotfix



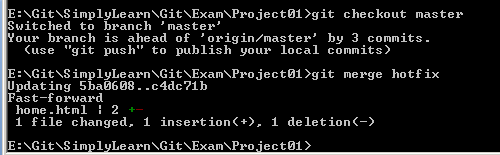
**git add home.html**

**git commit –m “title corrected”**



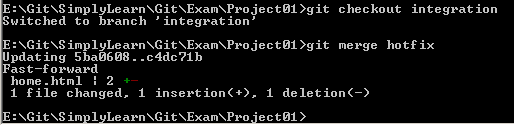
**git checkout master**

**git merge hotfix**



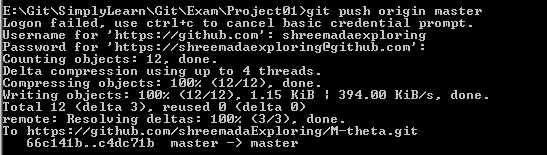
**git checkout integration**

**git merge hotfix**



1. Push to origin

Pushed all changes to remote master



Can be checked from here: <https://github.com/shreemadaExploring/M-theta>