****Tarlac State University

**COLLEGE OF COMPUTER STUDIES**

Case Study

in

Integrative Programming Technology 2

Submitted By:

**FERNANDEZ, Cyrus Ortega**

**LAXAMANA, RYAN**

**LEONES, John Alvin Ferrer**

**QUIAMBAO, Elson Jr. Simbillo**

Submitted To:

**Millet, Axel D.**

Date: []

**Table of Contents**

[**A.** **Definition** 3](#_Toc160378561)

[**B.** **Case Analysis (Git and GitHub Workflow)** 4](#_Toc160378562)

[**B.1** **Fernandez Documentation** 4](#_Toc160378563)

[**A.** **Remote Repository (GitHub) Creating a Milestone** 4](#_Toc160378564)

[**B.** **Local Repository (Git)** 6](#_Toc160378565)

[**C.** **Case Analysis (Git and GitHub Workflow)** 9](#_Toc160378566)

[**C.1** **Laxamana Documentation** 9](#_Toc160378567)

[**A.** **Remote Repository (GitHub) Creating a Milestone** 9](#_Toc160378568)

[**B.** **Local Repository (Git)** 9](#_Toc160378569)

[**D.** **Case Analysis (Git and GitHub Workflow)** 9](#_Toc160378570)

[**D.1** **Leones Documentation** 9](#_Toc160378571)

[**A.** **Remote Repository (GitHub) Creating a Milestone** 9](#_Toc160378572)

[**B.** **Local Repository (Git)** 9](#_Toc160378573)

[**E.** **Case Analysis (Git and GitHub Workflow)** 9](#_Toc160378574)

[**E.1** **Quiambao Documentation** 9](#_Toc160378575)

[**A.** **Remote Repository (GitHub) Creating a Milestone** 9](#_Toc160378576)

[**B.** **Local Repository (Git)** 9](#_Toc160378577)

[**F.** **References** 9](#_Toc160378578)

# **Definition**

The Lab Quiz#2 you did (Static Website) upload the documentation and files in your GitHub repository and add a minimum of 3 more webpages to add in your repository during the development use Git and GitHub and create a documentation of all.

**For the development of the Portfolio Webpages using (Git and GitHub) Issues, Pull Request, Milestone, Branching**

**Quiz # 1 LAB: Portfolio**

**Portfolio Web Pages Upload to GitHub Using Git**

**Requirements:**

* Each member in your group will need to upload **a webpage**
* Create a **Pull Request** for each member in Github with comments.
* Create an **Issue** for each member in Github and comment.
* Create a **Milestone** for the group in Github.
* Create a branch for each member and merge it with your master in Github.

**Define and describe how you use of each command with screenshot in your case study.**

* ~~Git clone~~
* ~~Git Pull~~
* ~~Git Push~~
* ~~Git Fetch~~
* ~~Git Merge~~
* ~~Issue~~
* Pull Request
* ~~Milestone~~
* ~~Branch~~

**Note: List the contributions of your group members**

# **Case Analysis (Git and GitHub Workflow)**

## **Fernandez Documentation**

### **Remote Repository (GitHub) Creating a Milestone**

A screenshot of a computer

Description automatically generated

Since I’m the leader of this project, I’ve created our GitHub repository named IPT2\_Group 5 then later the name changed to BSIT-TSM-3A-IPT2CS-G5. A repository is the most basic element of GitHub. It's a place where you can store your code, your files, and each file's revision history. Repositories can have multiple collaborators and can be either public or private [1].

A screenshot of a computer

Description automatically generated

After creating the repository, it is also my responsibility to add all my members to the GitHub repository also my professor since it is indicated in the instructions for monitoring purposes.

**A screenshot of a computer

Description automatically generated**

I’ve also created one milestone for our project since it is not to complex, I just created one milestone with 5 issues within it every time the issues close the percentage of the Milestone increase indicating that were making a progress**.** You can use milestones to track progress on groups of issues or pull requests in a repository. When you create a milestone, you can associate it with issues and pull requests. To better manage your project, you can view details about your milestone. [2]

**A screenshot of a computer

Description automatically generated**

On this par I’ve created some issues and incorporate it to our milestone this issues is just a simple part by part plan/steps of our projects. Issues are simple to create and flexible to suit a variety of scenarios. You can use issues to track work, give or receive feedback, collaborate on ideas or tasks, and efficiently communicate with others. [3]

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

In this par I just simply uploaded a Main\_Folder for our project’s this folder contains the main html that would be the landing page of our project then a designated folder for our individual Portfolio I made it this way to be more organized and incorporating some Modularity principles.

### **Local Repository (Git)**

**Cloning the repository from GitHub to local repository using bash terminal**

A computer screen with text

Description automatically generated

The first thing that I do is clone our GitHub repository I didn’t initialize my local repository since previously already uploaded a starting point for everyone that’s why I just copy the repository straight, on git bash type “git clone ***URL OF YOUR GITHUB.”***  To copy the files from your GitHub Repository. Clones a repository into a newly created directory, creates remote-tracking branches for each branch in the cloned repository. [4]

**Working on local-repo and creating a local-branch**

A computer screen shot of text

Description automatically generated

On this part after cloning the GitHub repository I’ve navigate my self to my respective folder and typed “git checkout -b ***Fernandez\_Branch***” to create my own branch. Git branches are effectively a pointer to a snapshot of your changes. When you want to add a new feature or fix a bug—no matter how big or how small—you spawn a new branch to encapsulate your changes. [5]

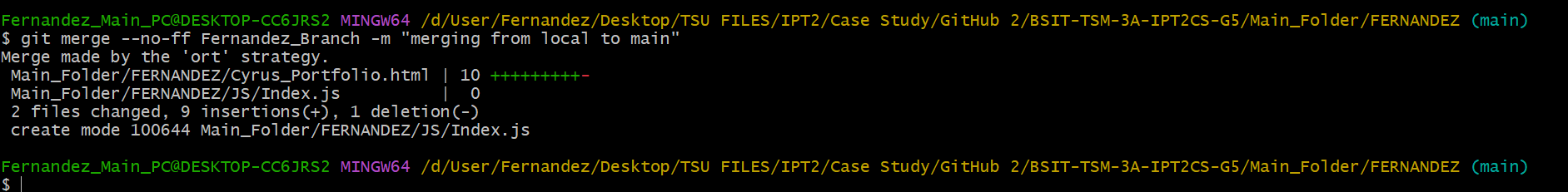
A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

I’ve added an navigation bar to my html file cloned form the GitHub, sine I’ve edited the file it is now modified by using ***“git add .”***  the file would be switched to tracked and after that we can use “git commit -m ***MESSAGE***” to fully commit the file and get a snapshot.

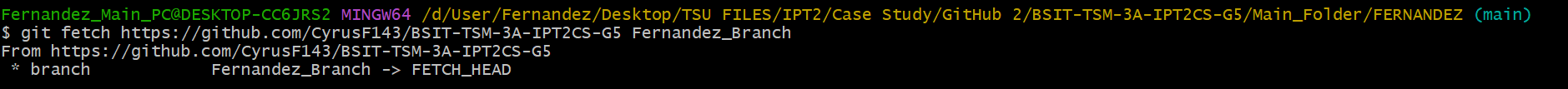


I use “git merge –no-f ***BRANCH*** -m” to merge my local branch to the main branch. Incorporates changes from the named commits (since the time their histories diverged from the current branch) into the current branch. [6]

A black screen with a black background

Description automatically generated

I’ve pushed my local main to the git hub branch with the command “git push -u origin”. git push origin will push the current branch to the branch of the matching name in the remote repository. [7]



The git fetch command downloads commits, files, and refs from a remote repository into your local repo. Fetching is what you do when you want to see what everybody else has been working on. [8]

**Creating the first version of my HTML Portfolio**

**A screenshot of a computer

Description automatically generated**

**A computer screen shot of a program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

**A black rectangle with yellow lines

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

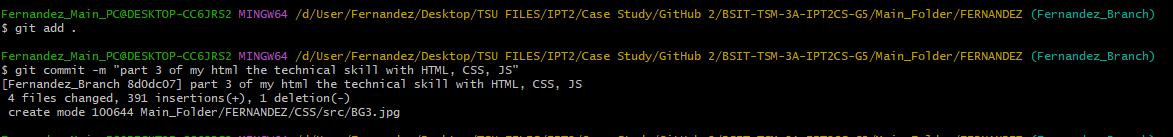
Description automatically generated**

**A black screen with yellow text

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

****

**A black rectangle with yellow text

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a video game

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A black rectangle with yellow text

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

# **Case Analysis (Git and GitHub Workflow)**

## **Laxamana Documentation**

### **Remote Repository (GitHub) Creating a Milestone**

### **Local Repository (Git)**

# **Case Analysis (Git and GitHub Workflow)**

## **Leones Documentation**

### **Remote Repository (GitHub) Milestone**

A screenshot of a computer

Description automatically generated

In this figure. I have accessed the milestone that was created by our Leader in our GitHub Repository.

A screenshot of a computer

Description automatically generated In this figure. It shows the Issues that was made by our Leader and each of the members be able to comment on each of the section in the issue.

A screenshot of a computer

Description automatically generated

In this figure. I have commented on the section of portfolio which contains my that my final portfolio has been merge to our GitHub Repository.

A screenshot of a computer

Description automatically generated

In this figure. I have commented on the section of documentation which is that I’m starting on creating my whole documentation in the creation of website.

A screenshot of a computer

Description automatically generated

In this figure. I have pull-request my Final Portfolio in my own Branch. It was successfully merged.

### **Local Repository (Git)**

**Cloning the repository from GitHub to local repository using bash terminal**

A computer screen shot of a computer code

Description automatically generated

In this figure. We already have our GitHub Repository which was created by our Leader. I did was to clone our Git Repository in Github. To make this process I copied the link of our Git Repository and proceed to the process, “git clone ***URL OF THE GITHUB REPOSITORY***”. To copy all the file from the GitHub Repository. Clones a repository into a newly created directory, creates remote-tracking branches for each branch in the cloned repository. [4]

A screen shot of a computer

Description automatically generated

In this figure. Since we are about to finish our own website in our GitHub Repository. I have updated my local repository based on the files that was pushed in our remote repository, to make this process I followed the command “git pull ***URL OF THE GITHUB REPOSITORY***”. The git pull command is used to fetch and download content from a remote repository and immediately update the local repository to match that content. [9]

A screenshot of a computer

Description automatically generated

In this Figure. I first access my folder on our GitHub Repository, “cd ***FOLDER NAME***””. After that I did create my own branch on respective folders that was assign by my Leader for my website, to make this process I followed this command “git checkout -b ***Name of your Branch***”. Git branches are effectively a pointer to a snapshot of your changes. When you want to add a new feature or fix a bug—no matter how big or how small—you spawn a new branch to encapsulate your changes. [5]

A screenshot of a computer

Description automatically generated

In this figure. I have added my files of my portfolio in the creation my portfolio website from my local to the GitHub Repository, to make this process if did first “git add ***File***” my files and then “git commit -m “***Message***””. The process “git add” will allow you to switch the file status to tracked and “git commit” will allow you to fully commit and get the snapshot of the file.

A screenshot of a computer

Description automatically generated

In this figure. I have merge my local branch to my main branch in our GitHub Repository, to make this process I followed this command, “git merge ***Branch Name***”. Incorporates changes from the named commits (since the time their histories diverged from the current branch) into the current branch. [6]

A screenshot of a computer program

Description automatically generated

In this figure. I have pushed my local main to the GitHub branch, to make this process I followed this command, “git push -u origin ***Branch Name***”. Git push origin will push the current branch to the branch of the matching name in the remote repository. [7]

A screen shot of a computer

Description automatically generated

In this figure. I have fetched my Branch in our GitHub Repository to download my commits, files, and refs, to make this process I followed this command, “git fetch “***URL OF THE GITHUB REPOSITORY” “BRANCH NAME”***”. Fetching is what you do when you want to see what everybody else has been working on. [8]

**Creating the first version of my HTML Portfolio**

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

**A screenshot of a computer

Description automatically generated**

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

**A screenshot of a computer

Description automatically generated**

**A screenshot of a website

Description automatically generated**

**A screenshot of a video game

Description automatically generated**

**A screenshot of a video game

Description automatically generated**

**Screens screenshot of a video game

Description automatically generated**

**A screenshot of a video game

Description automatically generated**

A screenshot of a computer

Description automatically generated

# **Case Analysis (Git and GitHub Workflow)**

## **Quiambao Documentation**

### **Remote Repository (GitHub) Creating a Milestone**

### **Local Repository (Git)**

# **References**

|  |  |
| --- | --- |
| [1] | "About repositories," GitHub Docs, [Online]. Available: https://docs.github.com/en/repositories/creating-and-managing-repositories/about-repositories. [Accessed 3 3 2024]. |
| [2] | "About milestones," GitHub Docs, [Online]. Available: https://docs.github.com/en/issues/using-labels-and-milestones-to-track-work/about-milestones. [Accessed 3 3 2024]. |
| [3] | "About issues," GitHub Docs, [Online]. Available: https://docs.github.com/en/issues/tracking-your-work-with-issues/about-issues. [Accessed 3 3 2024]. |
| [4] | "--local-branching-on-the-cheap," Git, [Online]. Available: https://git-scm.com/docs/git-clone. [Accessed 3 3 2024]. |
| [5] | "Git Branch," ATLASSIAN, [Online]. Available: https://www.atlassian.com/git/tutorials/using-branches. [Accessed 3 3 2024]. |
| [6] | "--everything-is-local," GIT, [Online]. Available: https://git-scm.com/docs/git-merge. [Accessed 3 3 2024]. |
| [7] | A. Khoo, "Git Push Origin," warp, [Online]. Available: https://www.warp.dev/terminus/understanding-git-push-origin#:~:text=git%20push%20origin%20is%20a,a%20remote%20repository%20(origin).&text=origin%20is%20the%20conventional%20shorthand,repository%20provider)%20for%20your%20project.. [Accessed 3 3 2024]. |
| [8] | "Git fetch," ATLASSIAN, [Online]. Available: https://www.atlassian.com/git/tutorials/syncing/git-fetch. [Accessed 3 3 2024]. |
| [9] | "ATLASSIAN," Atlassian, [Online]. Available: https://www.atlassian.com/git/tutorials/syncing/git-pull#:~:text=The%20git%20pull%20command%20is,Git%2Dbased%20collaboration%20work%20flows.. [Accessed 7 March 2024]. |