****Tarlac State University

**COLLEGE OF COMPUTER STUDIES**

Case Study

in

Integrative Programming Technology 2

Submitted By:

**Jay Mark B. Melivo**

**Ma. Virgie D. Petchalin**

**MarkJoseph Fabicon**

**Lucielle Joy Razon**

Submitted To:

**Axel Millet**

Date: []

Table of Contents

[A. Definition 3](#_Toc19107023)

[B. Case Analysis #1 4](#_Toc19107024)

[B.1 Problem 4](#_Toc19107025)

[B.2 Solution/Implementation 4](#_Toc19107026)

[B.3 Explanation 4](#_Toc19107027)

[C. Case Analysis #2 5](#_Toc19107028)

[C.1 Problem 5](#_Toc19107029)

[C.2 Solution/Implementation 5](#_Toc19107030)

[C.3 Explanation 5](#_Toc19107031)

[D. References 6](#_Toc19107032)

# 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**

**Jay Mark Melivo – Repository, landing page, documentation, Website.**

**Ma. Virgie Petchalin – Documentation, Website.**

**Mark Joseph Fabicon – Documentation, Website.**

**Lucielle Joy Razon – Documentation, Website.**

# Case Analysis (Git and GitHub Workflow)

## Documentation

.

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

Delegated task: to keep on tract with the HTML from collaborators

A screenshot of a computer

Description automatically generated

Before the case study runs to its development cycle, the group decided to agree whenever the deadlines are. I am delegated to keep on tract the HTML files of my groupmates and the other elements of a webpage such as PHP, JavaScript and CSS are delegated to them as well. Using GitHub’s Milestone feature, it makes the progress to be tracked, it also tracks all pull request and issues that the collaborators had made [1].

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

As we can see here, there are different milestone that are created.

Local Repository (Git)

Setting Email and Username for Author

A screenshot of a computer screen

Description automatically generated

**Creating the profile’s HTML file**

A person wearing a mask

Description automatically generated

A white background with black text

Description automatically generated

This is my html profile; no other designs are implemented yet and basic information about are added.

**Creating and initializing local repository**

A computer screen with text on it

Description automatically generatedCreation of local working directory and initializing at as git directory.

Cloning the repository from GitHub to local repository using bash terminal

A screenshot of a computer

Description automatically generated

Cloning can be achieved by accessing the github repository by clicking the Code and under HTTPS copy the link.

A screenshot of a computer

Description automatically generatedOn the bash terminal type git clone and paste the link from github. Git clone command copies an existing repository and clones it to new repository or another location [2]. Accessing the repo by $cd folder-name in our case study it will be cd IPT2-CaseStudy.

Working on local-repo and creating a local-branch

A black screen with yellow text

Description automatically generated

A screen shot of a computer

Description automatically generatedOn this part of the study, I created a new branch under clone local repository and named it as axel-branch. Branches are made for new features to add or simply not to interfere with the main branch that can be merge to the main later on [3].

The created html file was then pasted on the local-repo folder, this makes the files as untracked and by using the git command “*git add .`*” , the files are now tracked, using the git commit -m “message” then takes a snapshot of the current code and leaves a message for later documentation, git checkout main then moves us to the main branch.

A screenshot of a computer

Description automatically generated

Merging from local branch to local main. Merging through the command *git merge –no-ff branchname -m “message”* integrates the changes that was made from the other branches to main [4].

A computer screen shot of a computer program

Description automatically generated Pushing the local main to github branch, the command *git push origin HEAD:branch name* sends my local main repo to the specified github branch. Git push uploads local content repository to other location or remote repository [5].

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generatedGit fetch, fetches files from repository and saves it into the local machine [6].

A screenshot of a computer

Description automatically generated**Creating new Issue**

While scanning to the submitted html files of the collaborators, there are unclosed tags that are observed. Notifying them through github using the Issue feature. GitHub issue is intended for discussion, pull request and manages the collaborators on how they will solve the problem [7].

**A screen shot of a computer

Description automatically generatedAddressing the Issue**

In order to address the issue, we expect that our cloned repository has been changed or collaborators added their own works to the clone repository, In order to update our local copy of repository we must use git pull. Git pull downloads the new contents from remote repository and updates the local repository [8] which we will work later on.

A screenshot of a computer

Description automatically generated

**Adding CSS**

A person and person hugging on a beach

Description automatically generated

A computer screen shot of a black screen

Description automatically generatedA person in a striped shirt

Description automatically generated

After ensuring that the HTML file is correct and working properly, it is time to design the webpage. Updating the initial html file and uploading the css which we add, commit and push it again to our branch in github.

**Adding JavaScript**

**A screen shot of a computer program

Description automatically generatedA grey and white screen

Description automatically generated**

**Merging Git Branches to Main**

A screen shot of a computer

Description automatically generated

A screenshot of a chat

Description automatically generated

A screenshot of a computer

Description automatically generated

After the intensive checking by the collaborators, the last stage of application of the created branch is to merge with the github main branch. In this way, there are major changes to the main. In local repository we can merge our local branch to local main, the counterpart of merging in local is the github dedicated pull requests. Github pull request is a thread where the collaborators talks about the proposed changes to the main branch or the repository, where finished and approved works are located [9].

**Grade Matrix:**

Git Command : **20%**

GitHub Implement : **20%**

Documentation : **40%**

Webpage (Development) : **30%**

**100%**