Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT202-452-M2024/generic-module-1-git-readings-via-local/grade/vs53

IT202-452-M2024 - [Generic] Module 1 Git Readings via Local

Submissions:

Submission Selection

1 Submission [active] 5/27/2024 6:42:09 PM

Instructions

COLLAPSE

Preliminary Setup:

1. Go to w3schools.com
2. Create an account (preferably with your college account)
3. Visit my-learning.w3schools.com/tutorial/git
4. Complete the following readings:
    1. Essentials 1.1, 1.2, 1.3
    2. Essential Commands 2.1, 2.2, 2.3, 2.4
    3. Branch Management 3.1, 3.2
    4. Remote Collaboration 4.1-4.9
    5. Security Practices 6.1-6.3
    6. Attempt the Git Quiz (aim for <=70%)
5. Verify you're in the main branch via `git status` or `git branch`
6. If not, `git checkout main`
7. Create a branch for this assignment `git checkout -b M1-Git-Readings`
8. **Note**: In this assignment, we'll make the pull request later. In future assignments, we'll likely open it earlier so we can use the URL for assignments
9. Fill in the items in the worksheet below (save as often as necessary)
10. Once finished, export the worksheet
11. Take the exported file and add it anywhere in your repository (a Module1 folder is best, but not required)
12. Make sure git detects it by checking with `git status`
13. If everything is good, continue to submit
    1. Track the file either with `git add path/to/file` or `git add .`
    2. Commit changes via `git commit -m "some relevant message"`
    3. Push the changes via `git push origin M1-Git-Readings`

14. Go to GitHub and use the dropdown in the top left to find the M1-Git-Readings branch and ensure the file is present
15. If the file is there, either use the pull request popup or go to the pull request tab and open a request where main is base and M1-Git-Readings is compare
16. Open and complete the merge of the pull request (it should turn purple)
17. Go to Canvas and upload the same PDF that you just downloaded and pushed to GitHub

Branch name: M1-Git-Readings

Tasks: 2 Points: 10.00

Github Readings (10 pts.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Complete Reading of Below Topics

#1) Complete Essentials Lessons 1.1-1.3

Caption (required) ✓
Describe/highlight what's being shown
This is a screenshot of the "Essentials Lessons 1.1-1.3" module table, showing I competed all of them.

#2) Complete Essential Commands

Caption (required) ✓
Describe/highlight what's being shown
This is a screenshot of the "Essential Commands Lessons 2.1-2.4" module table, showing I competed all of them.

#3) Complete Branch Management

Caption (required) ✓
Describe/highlight what's being shown
This is a screenshot of the "Branch Management Lessons 3.1-3.2" module table, showing I competed all of them.

#4) Complete Remote Collaboration

Caption (required) ✓
Describe/highlight what's being shown
This is a screenshot of the " Complete Remote Collaboration Lessons 4.1-4.9" module table, showing I competed all of them.

#5) Complete Security Practices

#6) Attempt Quiz (>=70%)

Caption (required) ✓
Describe/highlight what's being shown
This is a screenshot of the "Security Practices Lessons 6.1-6.3" module table, showing I competed all of them.

Caption (required) ✓
Describe/highlight what's being shown
This is the results after I took the git quiz, I screenshoted my results and I got above 70 percent (100 percent)

COLLAPSE

Task #2 - Points: 1

Text: Reflection

Details:
Each prompt should have a few reasonable sentences.

#1) Mentions issues or no issues

Explanation (required) ✓
Explain in concise steps how this logically works

EDIT RESPONSE

I only have a couple of issues doing the github excerises in the w2s website, but that was more of an expected learning experience rather than an issue. Overall, this module was smooth and I encountered no real errors.

#2) Summarize core concepts related to the readings

Explanation (required) ✓
Explain in concise steps how this logically works

EDIT RESPONSE

The readings basically taught me everything about Git, showed me what exactly it is, what it's used for, the commands commonly used in git, it's purposes, and other standard git practices.

For example, one of the core concepts talked about in the reading is Git's relationship to Github, and working using something called the "Github Flow". It taught us the basics, such as how to create a branch, then make change or "commits" , make prs (pull requests), review, and merge. In a real world setting it would look like this: you create a repo in github. They're usually two main branches, the dev branch and the main branch. Dev branch is where all the development work happens and the main branch is the always functioning version of the product. Developers can create a sub branch off the dev branch, and make changes or commits to that branch, such as adding a new feature of somesort. Then once its all working, you can push those changes to your branch and make a pr (pull request), and have a reviewier review your code before it gets deployed. If

all is good, then the code has passed all the tests and can be safely merged to the main branch.

Another core concept talked about was how git commits or its version control system actually works. And how we can use the git log to revert, reset, or amend past commits. This is incredibly useful in scenarios where you made accidental changes and you wished you could revert your project back to a time where everything worked. For example, we were working on a project, added a feature successfully and then made a commit. Then worked on another feature, and now everything is broken. You can do git log to find the hash of that commit you made, and then revert back its changes using either git reset or get revert. You can also use git amend to correct the most recent commit if you made a mistake

End of Assignment