COMP 1073 - Winter 2020

MODULE FOUR PROJECT - CLEANER ASYNCHRONOUS CODE, PROMISE!

In Module Four, we learned that JavaScript is a single-threaded language. We also became more acquainted with writing JavaScript in an asynchronous way. At first, we explored older methods like callbacks, although we found that there are newer (and cleaner) methods available including promises and promises with async and await.

Your Task:

Revisit Lab 4 and complete the following:

Instructions :

1. Instead of using XMLHTTPRequest to fetch data from a server, use the Fetch API (which uses promises and represents a more modern approach) to retrieve your data. Learn how to implement this by watching the Browser API screencast located under the Module Five folder. This article is also a great resource : https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Fetching_data

2. Create a short screencast in which you review and explain your code and discuss any challenges/successes.

3. Ensure that all your HTML, CSS, and JS is well-commented, formatted, and organized.

4. Publish your page on a web server (AWS, Github pages or your own web server)

TAKE IT FURTHER:

1.) Instead of Lab 4, refractor another Module Assignment to incorporate a more asynchronous approach (2 bonus marks)

2.) Find another creative way to optimize your code. Explain why your approach helps to make your code more performant (2 bonus marks)

HELPFUL RESOURCES:

https://screencast-o-matic.com/ (free screencast software)
https://javascript.info/async
https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous/Promises
https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Fetching_data

Project Objectives:

- identify and incorporate a modern asynchronous approach when writing Javascript
- demonstrate the ability to optimize code for increased functionality and performance using an asynchronous approach

Project Assessment:

You will be assessed on the following:

	Missing Something	Getting There	Great Work	Awesomesauce
JavaScript **(4 marks)**	Developer used JS that is not valid, properly structured, formatted or commented. (1 mark)	Developer used JS that is somewhat valid, properly structured, formatted and commented. (2 marks)	Developer used JS that is mostly valid, properly structured, formatted and commented. (3 marks)	Developer used valid, properly structured, formatted and commented JS. (4 marks)
Asynchronous JavaScript (6 marks)	Developer does not successfully use Fetch & promises in order to make asynchronous HTTP requests with few issues. (0 mark)	Developer uses Fetch & promises in order to make asynchronous HTTP requests with some issues. (2 marks)	Developer uses Fetch & promises in order to make asynchronous HTTP requests with few issues. (4 marks)	Developer successfully uses Fetch & promises in order to make asynchronous HTTP requests (6 marks)
Code Review (2 marks)	No code review provided. (0 marks)	Developer accurately and effectively reviews code with little detail. (1 mark)	Developer accurately and effectively reviews code with some detail. (1.5 marks)	Developer accurately and effectively reviews code. (2 marks)

Project Due Date:

March 26th @ 11:59pm (Lakehead Students)
April 2nd @ 11:59pm (Georgian Students)

Project Weight:

12% of final grade

Submission Details:

Please submit all all code files as a zipped folder on Blackboard (D2L for Lakehead students) , a valid link to your published page and a link to your screencast video.

!important

Please ensure that any work you submit is your own unique work. Work submitted that is found to be not your own unique will be subjected to a grade of 0 and considered to be academic misconduct.