COMP 1073 - Summer 2020

COURSE PROJECT – PHASE TWO – OPTIMIZE YOUR INTERACTIVE TO DO LIST

In Phase One of the Course Project, you had the opportunity to build an interactive To-Do list using HTML, CSS and, of course, JavaScript. In Phase 2 of the project, we'll working on refining your To-Do List and incorporating extra functionality through the use of browser and third-party APIs .

Your Task:

Building on the To-Do List you created for Course Project Phase One, complete the following. *** Please note, if you received a 0 for Phase One, you will need to start your To-Do List from scratch.**

Instructions :

1. **Optimize your existing code in some way.**
    A. If you were not able to get all functionality working in Phase One, this is your opportunity to fix any errors and make sure the user is able to create a task, update a task as complete and move the task to the bottom of the list and delete a task. Remember to review the assignment instructions for Course Project Phase One if you are not sure of all the required functionality needed.
    B. If everything is working well, explore other ways to optimize your code (fix your code formatting, incorporate object-oriented JavaScript, make your code more efficient and DRY (Do Not Repeat Yourself!), modularize your code).

2. Add additional, appropriate functionality by integrating a third-party API AND a browser API.

Consider what functionality makes sense to add to a To-Do List (local storage, weather API, Web Audio API, Drag and Drop API). Although you may choose to use an API that we explored in class, full marks will go to those who incorporate an API that I have not shown an example of.

For a full list of browser APIs : https://developer.mozilla.org/en-US/docs/Web/API

For a full list of third-party APIs: https://www.programmableweb.com/apis/directory

3.) Publish Your Page Using Github Pages.

Include a link to your published GitHub page, your Github Repository, as well your code files in a zipped folder in your submission on Blackboard.

4.) Show What You Know.

Using ScreenCast-O-Matic (https://screencast-o-matic.com/screen-recorder) Create a short video in which you :

A. Describe and explain your code.
B. Describe one way that you optimized your code (how did you improve the code you submitted for Phase One?)
C. Describe and explain the functionality you added using a third-party API and a browser API.

IF YOU HAVE ANY QUESTIONS OR UNCERTAINTIES ABOUT THE PROJECT REQUIREMENTS, PLEASE ASK!

Project Objectives:

- optimize code for increased functionality, performance, readability, and reusability
- devise simple interfaces that allow the user to Create, Read, Update, and Delete (CRUD) data from a data source
- construct a variety of interface elements using unobtrusive client-side scripting techniques.
- manipulate DOM nodes, element attributes, and Cascading Style Sheet (CSS) properties
- identify and debug scripts using validators, DOM inspectors, and error console tools

Project Assessment:

You will be assessed on the following:

	Missing Something	Getting There	Great Work	Awesomesauce

JavaScript **(1 mark)**	Developer used JS that is not valid, properly structured, formatted and commented. Variables, arrays, functions, loops, and conditional structures that are not valid or appropriate to the functional requirements. (0 marks)	Developer used JS that is somewhat valid, properly structured, formatted and commented. Variables, arrays, functions, loops, and conditional structures are somewhat valid and appropriate to the functional requirements. (0.5 marks)	Developer used JS that is mostly valid, properly structured, formatted and commented. Variables, arrays, functions, loops, and conditional structures are mostly valid and appropriate to the functional requirements. (0.75 marks)	Developer used valid, properly structured, formatted and commented JS. The JavaScript includes properly-built variables, arrays, functions, loops, and conditional structures as appropriate to the functional requirements. (1 mark)
Functionality **(8 marks)**	The basic functionality of the application is not complete. Most functionality requirements are not working/not present. No APIs integrated. (2 marks)	The basic functionality of the application is almost complete. Some functionality requirements are not working/not present. APIs are not working as expected or only one type of API included. (4 marks)	The basic functionality of the application is complete. Additional functionality was successfully included using both a third-party API and browser API. APIs included were demonstrated in class. (6 marks)	The basic functionality of the application is complete. Additional functionality was successfully included using both a third-party API and browser API. APIs included were not discussed in class. (8 marks)

Optimization/ Troubleshooting & Debugging **(4 marks)**	Developer did not optimize or improve Phase One code. Functionality issues still persist. (1 mark)	Developer somewhat optimized Phase One code, although other opportunities to optimize were not utilized. Developer was somewhat able to use appropriate debugging tools in order to fix some issues that may be present. (2 marks)	Developer was able to improve and optimize Phase One code in some way with some success. Developer somewhat successfully utilized appropriate debugging tools in order to fix some issues that may be present. (3 marks)	Developer was able to successfully improve and optimize Phase One code in some way. Developer successfully utilized appropriate debugging tools in order to fix any issues that may be present. (4 marks)
Knowledge & Understanding/ Code Review **(2 marks)**	Developer did not complete and include code review. (0 marks)	Code review was completed with a fair degree of detail and professionalism. Some required information was missing. (0.5 marks)	Code review was completed with a good degree of detail and professionalism. Most required information was included. (1 mark)	Code review was completed with a high degree of detail and professionalism. All required information was included. (2 marks)

Project Due Date:

Monday, June 27th at 11:59pm

Project Weight:

15% of final grade

Submission Details:

Please submit all code files as a zipped folder on Blackboard. Please also submit a valid link to your published page with your submission. Your work will not be graded unless it is posted on a web server.

!important

Please ensure that any work you submit is your own unique and independent work. You are not permitted to collaborate with other learners or use code found online (tutorials, resources etc.)

Work submitted that is found to be not your own unique, and independent work will be subjected to a grade of 0 and considered to be academic misconduct.