PER SCHOLAS

SBA 316:

The Document Object Model

Version 1.0, 06/26/23

Click here to open in a separate window.

Introduction

This assessment measures your understanding of the Document Object Model (DOM) and your capability to implement its features in a practical manner. You have creative freedom in the topic, material, and purpose of the web application you will be developing, so have fun with it! However, remember to plan the scope of your project to the timeline you have been given.

This assessment has a total duration of **two (2) days**. This is a **take-home assessment**.

You have **two total days** (including weekends and holidays) to work on this assessment. This assessment will be due at **5:00pm** on the second day after it is assigned. Your instructor may provide you with class time to work on the assessment, schedule permitting.

Objectives

- Use DOM properties, methods, and techniques to create a web application that provides a dynamic user experience.
- Use BOM properties, methods, and techniques to facilitate creation of a dynamic web application.
- Demonstrate proficiency with event-driven programming and DOM events.
- Implement basic form validation using any combination of built-in HTML validation attributes and DOM-event-driven JavaScript validation.

Submission

Submit the link to your completed assessment using the **Start Assignment** button on the Assignment page in Canvas.

Instructions

You will create a small single-page web application. The topic and content of this application is entirely up to you; be creative!

Your work will be graded according to the technical requirements listed in the following section. Creativity and effort always work in your favor, so feel free to go beyond the scope of the listed requirements if you have the time.

Keep things simple. Like most projects you will encounter, you should finish the absolute minimum requirements *first*, and then add additional features and complexity if you have the time to do so. This will also help you understand what you can get done in a specific allotment of time if you were to be asked to do something similar in the future.

Once you have an idea in mind, approach your design through the user's perspective. User experience is one of the most important aspects of successful web design. If users enjoy their time on with your application, they are more likely to trust whatever services or information you offer, and more likely to come back and use the application again in the future.

Since topic and content are secondary to functionality for this assessment, we have included some resources below for free content that you can use to populate your application. Once you have gotten your functionality in place, you can return and fill in the content with something interesting.

Resources for free content:

- **Text:** Lipsum, a Lorem Ipsum text generator.
- **Images:** Pexels, a resource for stock photos (and other media).
- **GIFs:** Motion Elements, a resource for GIFs (and other media).

Requirements

The requirements listed here are **absolute minimums**. Ensure that your application meets these requirements before attempting to further expand your features.

Create your application locally, and initialize a local git repo. Make frequent commits to the repo. When your application is complete, **push your repo to GitHub and submit the link to the GitHub page** using the submission instructions at the top of this document.

Requirement	Weight
Cache at least one element using selectElementById.	5%
Cache at least one element using querySelector or querySelectorAll.	5%
Use the parent-child-sibling relationship to navigate between elements at least once (firstChild, lastChild, parentNode, nextElementSibling, etc.).	5%
Iterate over a collection of elements to accomplish some task.	10%
Create at least one element using createElement.	5%
Use appendChild and/or prepend to add new elements to the DOM.	5%
Use the DocumentFragment interface or HTML templating with the cloneNode method to create templated content.	2%
Modify the HTML or text content of at least one element in response to user interaction using innerHTML, innerText, or textContent.	10%
Modify the style and/or CSS classes of an element in response to user interactions using the style or classList properties.	5%
Modify at least one attribute of an element in response to user interaction.	3%
Register at least two different event listeners and create the associated event handler functions.	10%
Use at least two Browser Object Model (BOM) properties or methods.	3%
Include at least one form and/or input with HTML attribute validation.	5%
Include at least one form and/or input with DOM event-based validation. (This can be the same form or input as the one above, but should include event-based validation in addition to the HTML attribute validation.)	5%
Ensure that the program runs without errors (comment out things that do not work, and explain your blockers - you can still receive partial credit).	10%
Commit frequently to the git repository.	5%
Include a README file that contains a description of your application.	2%

Level of effort displayed in creativity, presentation, and user experience.	**5%**

Reflection (Optional)

Once you have completed your project, answer the following questions to help solidify your understanding of the process and its outcomes, as well as improve your ability to handle similar tasks in the future.

- *What could you have done differently during the planning stages of your project to make the execution easier?*

- *Were there any requirements that were difficult to implement? What do you think would make them easier to implement in future projects?*

- *What would you add to, or change about your application if given more time?*

- *Use this space to make notes for your future self about anything that you think is important to remember about this process, or that may aid you when attempting something similar again:*