PER SCHOLAS

SBA 308A:

JavaScript Web Application

Version 1.0, 08/18/23

Click here to open in a separate window.

Introduction

This assessment measures your capability to implement advanced JavaScript tools and 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 asynchronous JavaScript tools to build a responsive web application.
- Demonstrate understanding of the JavaScript event loop.
- Generate asynchronous code using Promises and async/await syntax.
- Use fetch and/or Axios to interact with an external web API.
- Organize files using modules and imports.

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.

You will need access to an external API in order to complete the requirements listed below. Since the content of the API will largely determine the content of your application, it should be chosen first and carefully. You can look for available APIs across the internet, but make sure they include features that will allow you to implement the requirements listed below.

Click here for a list of free APIs you can use. Two of our favorites, which contain all of the features necessary for this assessment are The Cat API and The Dog API. **You are not limited to these APIs or any others in the list above. Feel free to get creative!**

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
Use the fetch API or Axios to communicate with an external web API. Use the data provided by this API to populate your application's content and features.	**20%**
Create user interaction with the API through a search feature, paginated gallery, or similar. This feature should use GET requests to retrieve associated data.	**15%**
Enable user manipulation of data within the API through the use of POST, PUT, or PATCH requests. Ensure your chosen API supports this feature before beginning.	**15%**
Make use of Promises and async/await syntax as appropriate.	**15%**
Organize your JavaScript code into at least three (3) different module files, and import functions and data across files as necessary.	**3%**
Ensure the program runs as expected, without any undesired behavior caused by misunderstanding of the JavaScript event loop (such as race conditions, API calls being handled out of order, etc.).	**5%**
Create an engaging user experience through the use of HTML and CSS.	**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 to 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.*