**Individual Assignment 2 (Weight: 5% of the final grade)**

# Overview

In this assignment, you will propose and develop a more advanced JavaScript application that incorporates the concepts covered in Weeks 8 to 13. Your application should demonstrate your understanding of asynchronous programming, HTML and CSS basics, events, objects, JSON, API calls, and classes. The implementation should be cohesive and showcase practical use of each concept to create a functional and interactive application.

# Project Requirements

## Application Title and Description (10%)

1. Provide a title for your application.
2. Write a brief paragraph describing your application. Explain its purpose, the problem it solves, or the need it addresses.

## Implementation (90%)

### Asynchronous Programming (20%)

1. Implement at least two asynchronous operations using Promises, async/await, or callbacks.
2. Demonstrate proper error handling in asynchronous code.

### HTML and CSS (10%)

1. Create a basic HTML structure for your application.
2. Use CSS to style your application, ensuring it is visually appealing and user-friendly.

### Events (15%)

1. Implement at least three event listeners to handle user interactions.
2. Ensure that events trigger meaningful actions within the application.

### Objects, JSON, and Spread Operator (15%)

1. Use objects to organize and manage your application data.
2. Parse JSON data at least twice and stringify JSON data at least twice for storage and transmission.
3. Utilize the spread operator for copying or merging objects and arrays at least once.

### API Calls (20%)

1. Make at least two API calls to retrieve or send data using an API of your choice. You can search for an API at: https://mixedanalytics.com/blog/list-actually-free-open-no-auth-needed-apis/
2. Handle API responses and errors appropriately.
3. Integrate API data into your application meaningfully.

### Classes (10%)

1. Define and use at least two classes to structure your application.
2. Implement class methods and properties to encapsulate functionality.
3. Use inheritance to extend classes where appropriate.

# Submission Guidelines

1. Submit your code as a single ZIP file.
2. Ensure your code is well-documented with comments explaining key sections and logic.
3. Name your ZIP file in the format: YourName\_Assignment2.zip.

# Grading Criteria

## Application Title and Description (10%)

1. Clarity and completeness of the description.

## Implementation (90%)

1. Correct and effective use of asynchronous programming, HTML/CSS, events, objects, JSON, API calls, and classes.
2. Functionality and usability of the application.
3. Meaningful application of concepts learned in Weeks 8 to 13.