**Coding Assignment 3**

**Instructions**

Overview

In this project, students are expected to build a website using the Express/Node.js platform, with the Axios HTTP client, that integrates a chosen public API from the given list: [**Public API Lists**](https://github.com/appbrewery/public-api-lists). The website should interact with the chosen API, retrieve data, and present it in a user-friendly manner.

Learning Objectives

* Develop an understanding of how to integrate public APIs into web projects.
* Gain practical experience using Express/Node.js for server-side programming.
* Enhance understanding of client-server communication using Axios.
* Demonstrate ability to manipulate, present, and work with data retrieved from APIs.

Example Ideas

* Use the [JokeAPI](https://sv443.net/jokeapi/v2/) to Create a website that gives the user a joke based on their name.
* Use the [OpenWeatherMap API](https://openweathermap.org/api/one-call-3) to build a website that tells a user if it will rain tomorrow in their location of choice.
* Use the [Blockchain API](https://api.blockchain.com/v3/#/unauthenticated/getTickerBySymbol) to check the price of a cryptocurrency for the user.
* Use the [CocktailDB API](https://www.thecocktaildb.com/api.php) to make a website that gives the user a random cocktail recipe with images of the cocktail.
* Use the [Open UV API](https://www.openuv.io/) to make a website based on your home location that tells you if you need to apply sunscreen today.
* Bonus points for choosing your own API and creating something of your own. Sometimes, you'll decide on an API and then realise that it doesn't work, that's all a part of the process, just move on to the next one and try again!

Requirements

1. API Choice

* Browse through the [provided list](https://github.com/appbrewery/public-api-lists) and choose an API of interest. This choice should be guided by the potential to retrieve, manipulate, and present data in a meaningful and interactive way. I recommend choosing an API that does not require authentication and is CORS enabled. ([What is CORS?](https://medium.com/@electra_chong/what-is-cors-what-is-it-used-for-308cafa4df1a))

2. Project Planning

* Think through your project, researching the chosen API, its features, what data it will provide, and how it will be used in your web application.

3. Project Setup

* Set up a new Node.js project using Express.js.
* Include Axios for making HTTP requests.
* Ensure that the project has a structured directory and file organization.

4. API Integration

* Implement at least a GET endpoint to interact with your chosen API.
* Use Axios to send HTTP requests to the API and handle responses.

5. Data Presentation

* Design the application to present the retrieved data in a user-friendly way.

6. Error Handling

* Ensure that error handling is in place for both your application and any API requests. You can console log any errors, but you can also give users any user-relevant errors.

7. Documentation

* Include comments throughout your code to explain your logic.

8. Code Sharing

* Use what you have learnt about GitHub to commit and push your project to GitHub
* Include a Readme.md file that explains how to start your server, what commands are needed to run your code. e.g. npm i  and then nodemon index.js

Recommended Resources

* Express.js: [**Getting Started Guide**](https://expressjs.com/en/starter/installing.html)
* Node.js: [**Documentation**](https://nodejs.org/en/docs/)
* Axios: [**Documentation**](https://axios-http.com/docs/intro)
* Public APIs: [**API List**](https://github.com/appbrewery/public-api-lists)
* More free APIs: <https://rapidapi.com/collection/list-of-free-apis>