AMALITECH

Week 2.2

Lab Activity: Dictionary Web App

Learning Objectives:

- **ES6+ JavaScript:** Gain proficiency in modern JavaScript features to write cleaner, more maintainable code.
- **Asynchronous Programming:** Understand and implement asynchronous operations using Promises, `fetch`, and `async/await`.
- **API Interaction:** Learn how to interact with external APIs to fetch and display data.
- **DOM Manipulation:** Master techniques for dynamically updating the content and structure of web pages.
- **Error Handling:** Develop strategies for gracefully handling errors in your front-end application.
- **Responsive Design:** Create a user interface that adapts seamlessly to different screen sizes and devices.

Introduction:

In this lab session, you will build a dictionary web app. The app will allow users to search for words and view their definitions, pronunciations, and potentially other related information from a dictionary API.

Project Setup:

1. **Get the project files:** Download the project files from the link given by the trainer.
2. **Familiarize with the UI:** Review the provided HTML, CSS, and design assets to understand the layout and desired functionality.

Tasks:

1. **HTML Structure**:
    - Design a user-friendly layout using semantic HTML elements.
2. **CSS Styling:**
    - Style the elements according to the design provided.
    - Ensure the layout is responsive and works well on different screen sizes.
    - Consider using a CSS preprocessor (like Sass or Less) for easier style organization.
3. **JavaScript Functionality (with ES6 Emphasis):**
    - **API Interaction:**
        - Use fetch to make asynchronous requests to the dictionary API.
        - Use async/await or .then() / .catch() for handling Promise-based responses.
        - Parse the JSON data returned from the API.
    - **DOM Manipulation:**
        - Display the retrieved word definitions, pronunciations, and other relevant information.
        - Dynamically update the UI based on user interactions (e.g., search, theme toggle, font selection).
    - **Error Handling**:
        - Gracefully handle API errors (e.g., word not found, network issues).
        - Provide informative feedback to the user in case of errors.
4. **Testing and Refinement:**
    - Thoroughly test the functionality to ensure it works as expected.
    - Refine the design and functionality based on your ideas and feedback.

Evaluation

Your dictionary app will be evaluated based on the following criteria:

- **ES6+ Proficiency:** Effective use of modern JavaScript features throughout the codebase.

- **Functionality:** Correct implementation of API interaction, data display, and error handling.
- **Design:** Visual appeal, adherence to design guidelines, and responsiveness.
- **User Experience:** Ease of use, clear feedback, and overall satisfaction.
- **Code Quality:** Clean, organized, well-documented, and maintainable code.