### ****Level 2 JavaScript Curriculum (8 Weeks): Day-by-Day Breakdown****

### ****Month 1: Deep Dive into JavaScript for Dynamic Websites (Weeks 1-4)****

**Objective:** Transition from CLI to browser-based JavaScript, focusing on DOM manipulation, user interactions, and building real-world projects.

#### ****Week 1: Introduction to Browser-Based JavaScript****

* **Day 1:**
  + **Topic:** Introduction to JavaScript in the Browser.
  + **Focus:** Understanding the Document Object Model (DOM).
  + **Activity:** Explore and manipulate the DOM using querySelector and event listeners.
* **Day 2:**
  + **Topic:** Event Handling.
  + **Focus:** User interactions (click, input events).
  + **Activity:** Create a simple interactive counter.
* **Day 3:**
  + **Topic:** Styling with JavaScript.
  + **Focus:** Changing CSS dynamically, adding/removing classes.
  + **Activity:** Build a color-changing background feature.
* **Day 4:**
  + **Topic:** Building with Functions.
  + **Focus:** Creating reusable JavaScript functions.
  + **Activity:** Apply functions to handle repetitive tasks in DOM manipulation.
* **Day 5:**
  + **Topic:** Mini Project Kickoff – **To-Do List Application**.
  + **Focus:** Set up project structure, basic UI with HTML/CSS.
  + **Activity:** Start implementing add and display task functionality.

#### ****Week 2: Enhancing the To-Do List with Advanced JavaScript Concepts****

* **Day 1:**
  + **Topic:** State Management with Arrays.
  + **Focus:** Using arrays to manage To-Do list items.
  + **Activity:** Implement adding tasks to an array and displaying them.
* **Day 2:**
  + **Topic:** Data Persistence with Local Storage.
  + **Focus:** Saving tasks in the browser’s local storage.
  + **Activity:** Implement saving and loading tasks.
* **Day 3:**
  + **Topic:** Editing and Deleting Items.
  + **Focus:** Add functionality to update and delete tasks.
  + **Activity:** Complete To-Do List features, focusing on state updates.
* **Day 4:**
  + **Topic:** Error Handling and Form Validation.
  + **Focus:** Validating input and handling errors gracefully.
  + **Activity:** Ensure that tasks are not empty or duplicated.
* **Day 5:**
  + **Topic:** Project Polishing and Presentation.
  + **Focus:** Final touches, styling improvements, and showcasing work.
  + **Activity:** Present the To-Do List project, highlighting key learning points.

#### ****Week 3: Start of Major Project – Recipe Book Application****

* **Day 1:**
  + **Topic:** Project Setup and Planning.
  + **Focus:** Planning application structure and user flow.
  + **Activity:** Set up the initial HTML/CSS layout.
* **Day 2:**
  + **Topic:** Dynamic Content with JavaScript.
  + **Focus:** Rendering data from JavaScript objects and arrays.
  + **Activity:** Display recipes dynamically on the page.
* **Day 3:**
  + **Topic:** Advanced DOM Manipulation.
  + **Focus:** Create, update, and remove elements from the DOM.
  + **Activity:** Implement adding new recipes and displaying them.
* **Day 4:**
  + **Topic:** Modular JavaScript.
  + **Focus:** Organizing code into functions and modules.
  + **Activity:** Refactor code to make it modular and maintainable.
* **Day 5:**
  + **Topic:** Styling and User Experience Enhancements.
  + **Focus:** Use Tailwind CSS for styling and improving the user interface.
  + **Activity:** Enhance the look and feel of the Recipe Book.

#### ****Week 4: Completing and Presenting the Recipe Book Project****

* **Day 1:**
  + **Topic:** Working with Form Inputs.
  + **Focus:** Collecting user input for new recipes.
  + **Activity:** Implement a form to add recipes.
* **Day 2:**
  + **Topic:** Introduction to Error Handling.
  + **Focus:** Adding validation and feedback for user inputs.
  + **Activity:** Validate forms and display errors when necessary.
* **Day 3:**
  + **Topic:** Saving Data with Local Storage.
  + **Focus:** Persisting recipe data across sessions.
  + **Activity:** Store and retrieve recipes from local storage.
* **Day 4:**
  + **Topic:** Final Enhancements and Testing.
  + **Focus:** Testing functionality, fixing bugs, and enhancing usability.
  + **Activity:** Thoroughly test the application and refine code.
* **Day 5:**
  + **Topic:** Project Presentation and Feedback.
  + **Focus:** Showcase completed Recipe Book Project.
  + **Activity:** Present the project and receive feedback.

### ****Month 2: Advanced JavaScript and APIs (Weeks 5-8)****

**Objective:** Introduce APIs and integrate them into real-world projects to deepen understanding of JavaScript in web development.

#### ****Week 5: Introduction to APIs and Fetching Data****

* **Day 1:**
  + **Topic:** Introduction to APIs and Fetch API.
  + **Focus:** Understanding what APIs are and how to fetch data from an external source.
  + **Activity:** Fetch data from a public API (e.g., JSONPlaceholder).
* **Day 2:**
  + **Topic:** Working with Asynchronous JavaScript.
  + **Focus:** Promises and async/await.
  + **Activity:** Practice fetching and displaying data with async/await.
* **Day 3:**
  + **Topic:** Error Handling in API Calls.
  + **Focus:** Handling errors gracefully during data fetching.
  + **Activity:** Implement error handling for failed API requests.
* **Day 4:**
  + **Topic:** Rendering API Data in the DOM.
  + **Focus:** Displaying fetched data dynamically.
  + **Activity:** Build a simple news or weather dashboard with fetched data.
* **Day 5:**
  + **Topic:** Mini Project Kickoff – API Integration.
  + **Focus:** Integrate API data into a basic project.
  + **Activity:** Start a simple application (e.g., Movie Search or GitHub User Finder).

#### ****Week 6: Enhancing API-Based Project****

* **Day 1:**
  + **Topic:** Advanced Data Manipulation.
  + **Focus:** Filtering, sorting, and searching API data.
  + **Activity:** Add features like search and filter to the API project.
* **Day 2:**
  + **Topic:** UI Enhancements and State Management.
  + **Focus:** Managing UI state and improving UX.
  + **Activity:** Add loading spinners, error messages, and empty states.
* **Day 3:**
  + **Topic:** Working with Multiple APIs.
  + **Focus:** Combining data from multiple sources.
  + **Activity:** Integrate additional data from a second API.
* **Day 4:**
  + **Topic:** Storing API Data Locally.
  + **Focus:** Caching API responses with local storage.
  + **Activity:** Implement local data storage for offline access.
* **Day 5:**
  + **Topic:** Presentation of API Project.
  + **Focus:** Finalizing and presenting the mini project.
  + **Activity:** Showcase the project and discuss improvements.

#### ****Weeks 7-8: Major Project – E-commerce Product Page with API Integration****

* **Week 7, Day 1:**
  + **Topic:** Project Kickoff and Planning.
  + **Focus:** Define the scope and features of an e-commerce product page.
  + **Activity:** Set up initial layout and structure.
* **Week 7, Day 2-3:**
  + **Topic:** Fetching Product Data from an API.
  + **Focus:** Use an API to display products dynamically.
  + **Activity:** Implement product listing with API data.
* **Week 7, Day 4:**
  + **Topic:** Implementing a Shopping Cart.
  + **Focus:** Add to cart functionality, managing state with JavaScript.
  + **Activity:** Create a dynamic cart system.
* **Week 7, Day 5:**
  + **Topic:** Enhancing Product Interactions.
  + **Focus:** Add features like product filters and search.
  + **Activity:** Complete basic product interaction features.
* **Week 8, Day 1-2:**
  + **Topic:** Checkout Process.
  + **Focus:** Building forms and handling checkout logic.
  + **Activity:** Implement order summary and form validation.
* **Week 8, Day 3:**
  + **Topic:** API Error Handling and Final Tweaks.
  + **Focus:** Refine API calls, handle errors, and optimize performance.
  + **Activity:** Final touches and debugging.
* **Week 8, Day 4:**
  + **Topic:** Performance Optimization and Testing.
  + **Focus:** Optimize JavaScript and API performance.
  + **Activity:** Test the project thoroughly.
* **Week 8, Day 5:**
  + **Topic:** Project Presentation and Feedback.
  + **Focus:** Final presentation of the e-commerce project.
  + **Activity:** Present to peers and receive constructive feedback.