

Assignments:

1. **Closures and Scope**

Create a function that generates a sequence of Fibonacci numbers. Use closures to maintain the state of the sequence between function calls. Also, write a function that calculates the nth number in the Fibonacci sequence using memoization to optimize performance.

2. **Promises and Async/Await**

Write a program that simulates fetching data from multiple APIs. Use Promises and async/await to handle asynchronous operations. Create a function that fetches data from two different APIs concurrently and combines the results.

3. **Object-Oriented Programming**

Implement a complex object hierarchy, such as a library management system. Define classes for books, users, and transactions. Incorporate inheritance, encapsulation, and polymorphism. Add methods for borrowing, returning, and searching for books.

4. **Functional Programming**

Create a program that manipulates a list of data using functional programming principles. Use higher-order functions like `map`, `filter`, and `reduce` to transform and process the data. For example, you could work with a list of products and perform various calculations like finding the total cost or filtering out items based on certain criteria.

5. **Web APIs and DOM Manipulation**

Build a single-page web application that interacts with a public API. Create a dashboard that displays real-time data, such as stock prices or weather information. Use AJAX to fetch data, and manipulate the DOM to dynamically update the UI without requiring a full page reload.

Projects:

1. **Task Management App**

Develop a task management application where users can create, update, and delete tasks. Implement features like setting due dates, categorizing tasks, and marking tasks as completed. Use local storage to persist data across sessions.

2. **Interactive Data Visualization:**

Build a data visualization dashboard using libraries like D3.js or Chart.js. Fetch data from an API or use sample data, and create dynamic charts, graphs, and maps to display the information. Allow users to customize the visualization's parameters.

3. **Collaborative Note-Taking App:**

Create a real-time collaborative note-taking application similar to Google Docs. Use technologies like WebSockets or a real-time database to enable multiple users to edit the same document simultaneously. Implement features like user cursors, live updates, and version history.

4. **E-Commerce Platform:**

Develop a complete e-commerce platform with product listings, user authentication, shopping cart functionality, and secure payment processing. Implement advanced features like recommendations based on user behavior, order tracking, and inventory management.

5. **Social Media Platform:**

Build a social media platform that includes user profiles, posts, comments, and likes. Implement a news feed that displays content from users a person follows. Incorporate features like notifications, hashtags, and user search.