

CDAC Mumbai

Module: WPT

Topic: Assignment - 6

Section 1

1. **Question:** Create a simple shopping cart application using let, const, and var to manage items. Use const for constant values (like tax rates) and let for variables that might change (like total price).
2. **Question:** Write a function to calculate the area of a rectangle using both a regular function and an arrow function.
3. **Question:** Create an object to represent a book with properties such as title, author, and year published. Add a method to display the book details.
4. **Question:** Given an object representing a car, use object destructuring to extract its properties.
5. **Question:** Given an array of numbers, use array destructuring to extract the first two numbers.
6. **Question:** Use the map method to create a new array that contains the lengths of the names in the following array.
7. **Question:** Use the filter method to create a new array containing only the even numbers from the given array.
8. **Question:** Use the reduce method to find the total price of items in a shopping cart.
9. **Question:** Create a function that takes any number of arguments and returns their sum using the rest operator.
10. **Question:** Use the spread operator to merge two arrays of fruits.
11. **Question:** Write a function that accepts a callback and executes it after a delay.
12. **Question:** Create a promise that resolves with a message after 3 seconds.
13. **Question:** Create a function that returns another function, demonstrating closure.
14. **Question:** Use async/await to fetch data from a public API and log it to the console.
15. **Question:** Create a function that takes an array of numbers, applies a filter to keep only even numbers, then uses map to double those numbers, and finally returns the total using reduce.

Section 2

Project Title: Personal Budget Tracker

Duration: 30 Minutes

Description:

Create simple Personal Budget Tracker application that allows users to manage their expenses. The application should include functionalities to add, view, and calculate the total expenses. You will utilize various JavaScript concepts to implement this application.

Requirements:

1. **Variables:** Use let, const, and var to manage state variables like expense list and total expense.
2. **Functions and Arrow Functions:** Create functions to add an expense, display all expenses, and calculate the total. Use an arrow function for at least one of these.
3. **JavaScript Objects:** Represent each expense as an object with properties such as description, amount, and date.
4. **Destructuring:** Use array and object destructuring when retrieving expense details for display.
5. **Array Methods (Map, Filter, Reduce):**
 - Use map to display a list of expense descriptions.
 - Use filter to show only expenses above a certain amount (e.g., \$20).
 - Use reduce to calculate the total expenses.
6. **Rest and Spread Operator:** Use the rest operator to allow adding multiple expenses at once. Use the spread operator to create a new expense list when adding new expenses.
7. **Callback Functions:** Implement a function that takes a callback to display a success message after an expense is added.
8. **Promises:** Create a promise that simulates fetching initial expenses from an API (you can just resolve with a hard-coded array).
9. **Closures:** Use a closure to create a function that maintains the state of total expenses.
10. **Async/Await:** Use async/await to fetch initial expenses and display them in the application when it loads.