Assignment 6: Fetch API and JSON - Shopping Cart Data

Objective: In this assignment, students will learn how to use the Fetch API in JavaScript to load and display data from an external JSON file. The assignment will focus on fetching sample shopping cart data from a JSON file and dynamically rendering it onto the webpage.

Concept: Reading and Displaying Shopping Cart Data

Requirements:

- **HTML Structure:**
  - Create a new HTML file named fetch_cart.html.
  - The page should include the following sections:
    i. Shopping Cart List: A section that dynamically displays the items in the shopping cart loaded from the JSON file.
    ii. Cart Summary: A section showing the total number of items and price.
- **JSON File:**
  - Create a JSON file named cart_data.json that contains an array of shopping cart items.
  - Each shopping cart item should have the following properties:
    i. name (string)
    ii. price (number)
    iii. quantity (number)
- **JavaScript Fetch API:**
  - Create a new JavaScript file named fetch_cart.js and link it to your HTML file.
  - Use the Fetch API to read the cart_data.json file when the page loads.
  - Parse the JSON response and dynamically display the shopping cart items in the Shopping Cart List section.
- **Displaying the Shopping Cart:**
  - Dynamically create and display each cart item's name, price, quantity, and total cost (price * quantity).
  - Update the Cart Summary to show the total number of items and the total price of all items in the cart.
- **Handling Fetch Errors:**

- Implement error handling for the Fetch API to gracefully display a message if the JSON file cannot be loaded (e.g., file not found or network error).

Deliverables:

- fetch_cart.html - The HTML file containing the structure for the shopping cart display.
- fetch_cart.js - The JavaScript file containing the Fetch API code.
- cart_data.json - The JSON file containing the sample shopping cart data.
- Ensure all files are included in a ZIP or RAR file together.

Grading Criteria:

- **Fetch API Usage (50%)**
  - Correct implementation of the Fetch API to retrieve and display data from the JSON file.
  - Proper parsing and handling of the JSON response.
- **DOM Manipulation (30%)**
  - Dynamic creation and display of cart items and summary.
  - Efficient updating of the DOM with the fetched data.
- **Error Handling (10%)**
  - Graceful handling of errors during the fetch operation (e.g., displaying an error message if the data fails to load).
- **Code Quality (10%)**
  - Clean and well-organized code with proper comments and error handling.

Conclusion: By completing this assignment, students will gain experience with the Fetch API, working with external JSON files, and dynamically displaying data on a webpage. This assignment helps students understand how modern web applications interact with external data sources, laying the foundation for more advanced data handling in future assignments.