**Task 1: E-Commerce Cart Functionality**

**Requirement:**  
A client wants a simple shopping cart where users can view and modify their selected items.

1. **Steps:**
   * Use a JSON object to represent the cart. Each item should have name, price, quantity, and category.
   * Allow users to:
     + **View the cart** with item details.
     + **Update the quantity** of items using input fields and dynamically update the total price using forEach() and reduce().
     + **Filter items** by category using filter().
2. **Extra Features:**
   * Apply a 10% discount on the total price if the total exceeds a user-defined threshold. Use a conditional statement and display the updated price dynamically.
3. **Technology:**
   * Use DOM manipulation to display and interact with the cart.

**Task 2: Employee Management Dashboard**

**Requirement:**  
A client wants a dashboard to manage employees in their company.

1. **Steps:**
   * Use an array of objects where each object contains name, age, department, and salary.
   * Implement the following features:
     + Display the list of employees using for...in or for...of.
     + Add a search bar to find employees by name using string methods like includes() or indexOf().
     + Use a dropdown to filter employees by department using filter().
     + Calculate the average salary of employees using reduce().
2. **Extra Features:**
   * Highlight employees earning more than a user-defined salary threshold using map().

**Task 3: Fitness App Progress Tracker**

**Requirement:**  
A fitness app client wants a tracker for daily exercise and calorie burn.

1. **Steps:**
   * Use an object to represent a day's progress, with properties like date, exercises (array of objects), and totalCalories. Each exercise should have name, duration (in minutes), and caloriesBurned.
   * Implement the following:
     + Calculate the total calories burned using reduce() and display it.
     + Allow users to **add exercises** dynamically using a form, updating the total calories in real time.
     + Provide a search feature to find exercises by name.
2. **Extra Features:**
   * If the total calories burned exceed 500, display a congratulatory message using a ternary operator.

**Task 4: Product Comparison Tool**

**Requirement:**  
A client wants a tool to compare different products.

1. **Steps:**
   * Use a JSON object to represent the products with properties like name, price, rating, and category.
   * Display the list of products dynamically using forEach().
   * Add filters for:
     + **Price range:** Using filter().
     + **Category:** Using dropdown options and conditional logic.
2. **Extra Features:**
   * Allow users to sort products by price or rating using a for loop or sort() method.
   * Display the cheapest and most expensive products using Math.min() and Math.max().

**Task 5: Student Result Portal**

**Requirement:**  
A school client needs a portal for students to check their results.

1. **Steps:**
   * Create a JSON object containing student data: name, marks (object with subjects as keys), and averageMarks.
   * Implement the following:
     + Calculate the average marks for each student using reduce().
     + Highlight students who scored above a user-defined threshold using a conditional statement.
     + Display subject-wise marks dynamically using for...in.
2. **Extra Features:**
   * Allow users to search for a student by name using string methods.
   * If the student’s average is below 40, display a "Fail" message using a ternary operator.

**Task 6: Real Estate Inquiry System**

**Requirement:**  
A real estate client wants an inquiry system for available properties.

1. **Steps:**
   * Use an array of objects where each object represents a property with details like location, price, type (apartment/villa), and availability.
   * Implement the following:
     + Display the list of available properties using map().
     + Allow users to filter properties by type or location using filter().
     + Display properties sorted by price using a custom comparator.
2. **Extra Features:**
   * Use while and do...while loops to ask the user if they want to make another search until they select "No".

**Task 7: Quiz App**

**Requirement:**  
A client wants a simple quiz app to test JavaScript knowledge.

1. **Steps:**
   * Use an array of objects where each object represents a question with text, options, correctAnswer, and userAnswer.
   * Implement the following:
     + Display each question one by one using a for loop.
     + Allow users to select their answer using radio buttons.
     + After submission, calculate the score using reduce().
2. **Extra Features:**
   * Highlight incorrect answers in red and correct answers in green using DOM manipulation.