✅ 1. Smart Todo List Web App

Objective:
Create a responsive to-do list application that allows users to add, delete, and filter tasks.

Must Cover:

- Input validation using **Control Flow**
- Add/remove tasks using **DOM Manipulation**
- Store tasks in an array and display using **Array Methods**
- Use **Functions** to structure logic (addTask, removeTask, filterTasks)
- Fetch a motivational quote from an API on load (**Asynchronous JavaScript**)

✅ 2. Student Report Card Generator

Objective:
Build a tool to enter multiple student names and marks, calculate grades, and show them in a styled table.

Must Cover:

- Use **Objects** to represent each student
- Process data using **Array Methods** (`map`, `filter`, `reduce`)
- Use **Functions** for grade calculation and data rendering
- Control flow for grading logic
- Render output in HTML dynamically using **DOM**

✅ 3: Expense Tracker with Data Persistence

🎯 Objective:
Build an expense tracker web app where users can add income and expenses, view their balance, and see a transaction history.

🔧 Must Cover:

- ✅ **JavaScript Basics**: Declare variables for income, expense, balance, etc.
- ✅ **Control Flow**: Validate inputs (non-empty, numeric, etc.) before adding a transaction.

- ✅ **Functions and Scope**: Use reusable functions (`addTransaction()`, `calculateBalance()`, etc.)
- ✅ **DOM Manipulation**: Dynamically add/remove transactions from the UI.
- ✅ **Objects and Arrays**: Store each transaction as an object in an array.
- ✅ **Array Methods**: Use `map()`, `filter()`, and `reduce()` to calculate totals and filter transactions.
- ✅ **Asynchronous JavaScript**: Use `localStorage` (or optionally IndexedDB or mock fetch) to persist and retrieve transaction data asynchronously.
- ✅ **Web Development Integration**: Build a form-based UI, style with CSS/Bootstrap, and make the layout responsive.

✅ 4. Quiz App with Timer and Scoreboard

Objective:
Develop a multiple-choice quiz game with a countdown timer and score tracker.

Must Cover:

- Question data stored in **Arrays of Objects**
- Score calculation and timer handled using **Functions** and **Control Flow**
- **DOM** updates for changing questions and showing scores
- Use of `setInterval()` for timer (**Async**)
- Reset/Replay functionality using **Functions**

✅ 5. Interactive Product Catalog with Search and Filter

Objective:
Build a mini storefront that lists products with search and filter features.

Must Cover:

- Product data as an **Array of Objects**
- Use of `filter()` and `map()` to render product cards dynamically
- User input for search bar and filter dropdown handled using **Event Listeners**
- Card UI updated using **DOM Manipulation**

- Add a simulated delay in rendering filtered results using `setTimeout()` (**Async behavior**)