**Task 1: Dynamic Todo List**

* **Objective**: Create a functional Todo List app where users can:
  + Add new tasks.
  + Mark tasks as completed.
  + Delete tasks.
* **Details**:
  + Use a TaskList component that takes an array of tasks as props.
  + Each task is rendered in a Task component.
  + Add buttons for "Complete" and "Delete" actions.
  + Style tasks dynamically using CSS classes (e.g., strike-through for completed tasks).
* **Deliverables**:
  + A functioning Todo List app with the ability to:
    - Add tasks via an input field.
    - Display tasks dynamically using map.
    - Apply dynamic styling for completed tasks.

**Task 2: Product Card Grid**

* **Objective**: Create a grid of product cards that display details like an image, name, price, and an "Add to Cart" button.
* **Details**:
  + Use a ProductGrid component to render multiple ProductCard components.
  + Pass product details (name, price, image) as props to each ProductCard.
  + Display a message (e.g., alert) with the product name when "Add to Cart" is clicked.
  + Style the grid with CSS (e.g., flexbox or grid layout).
* **Deliverables**:
  + A responsive product grid with:
    - Dynamic rendering of product data.
    - CSS styling for the grid and cards.
    - Interaction with "Add to Cart" button.

**Task 3: FAQ Accordion**

* **Objective**: Create an FAQ section where questions can expand/collapse to show answers.
* **Details**:
  + Use an FAQList component to render multiple FAQItem components.
  + Pass a question and answer as props to each FAQItem.
  + Use state in each FAQItem to track if the answer is visible.
  + Add CSS transitions for smooth expand/collapse effects.
* **Deliverables**:
  + A fully functional FAQ section with:
    - Expand/collapse functionality for answers.
    - CSS animations for transition effects.
    - Clean styling for questions and answers.

**Task 4: Image Gallery with Filters**

* **Objective**: Create an image gallery where users can filter images by category.
* **Details**:
  + Use an ImageGallery component to render ImageCard components.
  + Pass image data (URL, category) as props to each ImageCard.
  + Provide buttons or a dropdown for filtering by category.
  + Style the gallery with CSS for a responsive grid layout.
* **Deliverables**:
  + A working gallery with:
    - Dynamic filtering functionality.
    - Responsive styling for grid layout.
    - Clean UI for category filters.