# React Task: Recipe Planner Using Hooks

This task is designed to help you practice and understand all the major React hooks (useState, useEffect, useContext, useReducer, useRef, useMemo, useCallback, and useId). You will create a Recipe Planner app where users can add ingredients, calculate the total cost, and manage a preparation timer. This task ensures comprehensive learning of React hooks.

## Requirements

- Add Ingredients: Users can add ingredients to a recipe along with their quantity and cost.

- Calculate Total Cost: The total cost of the recipe is calculated dynamically.

- Context for Theme: The app supports light and dark themes using useContext.

- Manage Complex State: Use useReducer to manage the list of ingredients.

- Timer for Cooking: A preparation timer is added using useRef and useEffect.

- Memoized Cost Calculation: Optimize total cost calculation using useMemo.

- Callback for Adding Ingredients: Use useCallback to handle adding ingredients.

- Unique IDs: Use useId for form inputs.

## Implementation Plan

The implementation consists of creating several components and utilizing React hooks effectively.

### 1. Setup Components

- App.js: Main component that provides the ThemeContext.

- IngredientForm: Form to add ingredients.

- RecipeList: Displays the list of ingredients.

- Timer: A preparation timer.

### 2. Hooks Usage

#### useState

Manage form state for ingredient name, quantity, and cost. Toggle between light and dark themes.

#### useEffect

Save the recipe list to localStorage on change. Handle the timer countdown.

#### useContext

Provide theme data (light/dark) across components.

#### useReducer

Manage the state of the ingredients list with actions: ADD, REMOVE, CLEAR.

#### useRef

Reference the timer interval and input fields.

#### useMemo

Dynamically calculate and optimize the total cost.

#### useCallback

Handle ingredient addition to avoid unnecessary re-renders.

#### useId

Generate unique IDs for form inputs.