Project Documentation: Calculator/Tool for Iterative Calculations

Overview

This project aims to develop a calculator/tool that runs Python code to perform iterative calculations. The initial version will require users to manually input data, but future versions will integrate API data retrieval and visualization features.

Objectives

1. Iterative Calculations: Implement a system that performs iterative calculations to simulate various scenarios.
2. User Input: Design input fields for users to manually enter data in the initial version.
3. Data Visualization: Create a space for visualizing data results, with plans to integrate more complex visualizations later.
4. Modern Development Practices: Utilize the most modern web development techniques and frameworks.

Initial Version Features

1. User Input Fields: Allow users to input two numbers.
2. Output Calculation: Display a single output number based on the input numbers.
3. Graph Plotting: Provide a graph area to plot the output number on both the x and y coordinates for simplicity.

Future Enhancements

1. API Integration: Pull data from APIs for automated input.
2. Advanced Visualization: Implement more sophisticated data visualization techniques.

Project Documentation: Calculator/Tool for Iterative Calculations

Development Steps

Step 1: Set Up Scaffolding

- Choose a Framework: Select a modern web development framework (e.g., React, Angular, Vue.js).

- Project Initialization: Initialize the project using the chosen framework's CLI tools.

Step 2: Create Basic Components

- Input Fields: Develop components for user input.

- Output Display: Create a component to display the calculated output.

- Graph Component: Implement a basic graph component using a library like Chart.js or D3.js.

Step 3: Implement Calculation Logic

- Python Backend: Set up a backend server (e.g., Flask, Django) to handle Python calculations.

- API Endpoints: Create API endpoints to receive input data and return calculated results.

Step 4: Integrate Frontend and Backend

- Data Flow: Ensure smooth data flow between the frontend input fields and the backend calculation logic.

- Display Results: Update the frontend to display the calculation results and plot them on the graph.

Step 5: Test and Iterate

- Testing: Conduct thorough testing to ensure functionality.

- Iterative Development: Continuously improve the tool based on user feedback and testing results.

Try out these other useful GPTs

Project Documentation: Calculator/Tool for Iterative Calculations