**Title:** Investment Calculator

**Tool Details:**

* **Backend:** Node.js, Express.js
* **Frontend:** WebComponent-based UI
* **Database (if needed):** Optional (for storing calculations)

**Goal:** Candidates will gain hands-on experience in full-stack development by building and integrating a backend API with a WebComponent-based frontend. They will learn about API handling, dynamic data processing using AI, and UI interaction.

**Assignment Description:** Develop an Investment Calculator that allows users to input investment details such as initial amount, interest rate, and duration. The backend will process the request, dynamically generate data using AI, and return calculated investment results. The frontend will display the computed values in a user-friendly interface.

**Tasks & Steps:**

1. **Backend API Development:**
   * Set up an Express.js server.
   * Create an endpoint to receive investment details (initial amount, interest rate, duration).
   * Implement logic to process the request and calculate returns.
   * Use AI to generate dynamic financial insights.
   * Return the processed data in JSON format.
2. **Frontend Integration:**
   * Build a WebComponent-based UI with an interactive form.
   * Capture user input and send it to the backend using fetch API.
   * Display the results dynamically after receiving a response.
3. **Result Display:**
   * Present calculated investment growth over time.
   * Provide AI-generated insights and recommendations.

**Mathematical Calculation/Steps:**

* Compound Interest Formula:

A=P×(1+rn)ntA = P \times (1 + \frac{r}{n})^{nt}

Where:

* + **A** = Final investment value
  + **P** = Initial principal amount
  + **r** = Annual interest rate (decimal form)
  + **n** = Number of times interest applied per year
  + **t** = Number of years

**Third-Party Packages (if required):**

* **express** (for backend API setup)
* **cors** (to enable cross-origin requests)
* **body-parser** (to parse request bodies)
* **openai (or similar AI package)** (for AI-generated insights)

**Acceptance Criteria:**

* The backend correctly handles requests and returns calculated results.
* The frontend successfully captures user input and interacts with the API.
* The UI dynamically displays the returned investment data.
* AI-generated insights are included in the response.
* Proper error handling for invalid input.

**Submission Guidelines:**

1. Fork the provided repository.
2. Create a folder named investment-calculator-<your\_name>.
3. Implement the backend API and frontend WebComponent in the folder.
4. Push your changes to your forked repository.
5. Submit a pull request to the main repository for review.

**Reference:**

1. [**https://www.calculator.net/**](https://www.calculator.net/)