**Title: Interest Rate Calculator**

**Tool Details:**

* **Backend:** Node.js with Express.js
* **Frontend:** WebComponent-based UI
* **Database (Optional):** JSON storage or in-memory data handling

**Goal:** Candidates will learn to:

* Develop a RESTful API using Express.js
* Implement a WebComponent-based frontend for user input and result display
* Integrate frontend and backend through API calls
* Process and return interest rate calculations dynamically

**Assignment Description:** The objective of this assignment is to build a full-stack Interest Rate Calculator. The frontend will feature a form where users input principal amount, time period, and interest type (simple or compound). The backend will process the data, compute the interest amount, and return the result for display on the frontend. AI should generate different input values dynamically for testing purposes.

**Tasks & Steps:**

1. **Backend API Development:**
   * Set up an Express.js server.
   * Create an endpoint to receive principal, time, and interest type.
   * Perform interest calculations (simple or compound) based on the input.
   * Return computed results to the frontend.
2. **Frontend Development:**
   * Create a WebComponent-based form with input fields for principal, time period, and interest type.
   * Send form data to the backend using fetch API.
   * Display the computed interest amount and total amount payable received from the backend.
3. **Integration & Testing:**
   * Ensure the frontend correctly sends user input to the backend.
   * Validate responses from the backend.
   * Handle errors gracefully in both frontend and backend.

**Mathematical Calculation/Steps:**

* **Simple Interest Formula:** SI = (P × R × T) / 100
  + P = Principal Amount
  + R = Annual Interest Rate
  + T = Time in Years
* **Compound Interest Formula:** CI = P × (1 + R/100)^T - P
  + P = Principal Amount
  + R = Annual Interest Rate
  + T = Time in Years

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

* express (for backend API)
* cors (to handle cross-origin requests)
* body-parser (for processing JSON input)

**Acceptance Criteria:**

* The backend API correctly receives and processes input data.
* The WebComponent-based frontend interacts seamlessly with the backend.
* The interest calculation is accurate for both simple and compound interest.
* AI dynamically generates input scenarios for testing.
* The frontend displays results in a user-friendly manner.

**Submission Guidelines:**

1. Fork the provided repository.
2. Create a new folder with your name inside the repository.
3. Implement the solution within the folder.
4. Push the completed code to your forked repository.
5. Submit a pull request to the original repository.

Ensure the solution meets all criteria before submission.