**Title: Finance 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 financial calculations dynamically

**Assignment Description:** The objective of this assignment is to build a full-stack Finance Calculator. The frontend will feature a form where users input loan amount, interest rate, and tenure. The backend will process the data, compute the monthly EMI and total payable 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 loan details.
   * Perform financial calculations for EMI and total payable amount.
   * Return computed results to the frontend.
2. **Frontend Development:**
   * Create a WebComponent-based form with input fields for loan amount, interest rate, and tenure.
   * Send form data to the backend using fetch API.
   * Display the computed EMI 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:**

* **Formula for EMI Calculation:** EMI = [P × R × (1 + R)^N] / [(1 + R)^N – 1]
  + P = Principal Loan Amount
  + R = Monthly Interest Rate (Annual Rate / 12 / 100)
  + N = Loan Tenure in Months
* **Total Payable Amount:** EMI × N

**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 EMI calculation and total payable amount are accurate.
* 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.