**Title:** Loan Calculator

**Tool Details:**

* **Backend:** Node.js, Express.js
* **Frontend:** WebComponents (Vanilla JS or Lit)
* **Database (optional):** Not required

**Goal:**  
Candidates will gain experience in building a full-stack application using Express.js for backend API development and WebComponents for a modern frontend. They will learn how to handle HTTP requests, process user input, and dynamically update the UI based on backend responses.

**Assignment Description:**  
Develop a Loan Calculator that allows users to input loan amount, interest rate, and tenure. The frontend will collect user inputs, send them to the backend, and display the calculated monthly installment (EMI). The backend will process the loan details, calculate the EMI, and return the result to the frontend for display.

**Tasks & Steps:**

1. **Backend Development:**
   * Set up an Express.js server.
   * Create an endpoint (/calculate-loan) to accept user input (loan amount, interest rate, tenure).
   * Implement logic to compute EMI based on the provided data.
   * Return the calculated EMI as a JSON response.
2. **Frontend Development:**
   * Create a WebComponent-based UI with a form to collect loan details.
   * Implement event handling to send form data to the backend via an API request.
   * Receive and display the calculated EMI in the UI.
3. **Integration & Testing:**
   * Ensure the API receives correct input, processes data, and returns accurate results.
   * Test the frontend to confirm correct request-response handling and UI updates.

**Mathematical Calculation/Steps:**

* EMI Calculation Formula:

EMI=P×r×(1+r)n(1+r)n−1EMI = \frac{P \times r \times (1 + r)^n}{(1 + r)^n - 1}

Where:

* + **P** = Loan Amount
  + **r** = Monthly Interest Rate (Annual Rate / 12 / 100)
  + **n** = Number of Months (Tenure in years × 12)

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

* express for backend server
* cors to enable cross-origin requests
* body-parser for handling JSON data
* lit (optional) for creating WebComponents

**Acceptance Criteria:**

* The backend API correctly processes loan calculations and returns accurate results.
* The frontend correctly collects input, sends it to the API, and displays the calculated EMI.
* The UI updates dynamically based on backend responses.
* The solution is properly structured and follows best practices.

**Submission Guidelines:**

1. Fork the provided repository.
2. Create a folder named loan-calculator-[your-name].
3. Implement the solution inside this folder.
4. Push the code to your forked repository.
5. Submit a pull request with a brief description of your implementation.

**Reference:**

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