**Title: Income Tax Calculator**

**Tool Details:**

* **Backend:** Node.js, Express.js
* **Frontend:** WebComponent (Vanilla JS, Lit, or Stencil)
* **Database (optional):** None (process data in-memory)

**Goal:** By completing this assignment, candidates will gain hands-on experience in developing and integrating a full-stack web application. They will learn to create a backend using Express.js, develop a WebComponent-based frontend, handle HTTP requests, process user input, and return dynamic responses using AI-generated data.

**Assignment Description:** The Income Tax Calculator will be a simple web application where users enter their annual income, and the backend calculates the applicable tax based on predefined tax slabs. The frontend will feature a form where users input their income, submit the data to the backend, and display the calculated tax result dynamically.

**Tasks & Steps:**

1. **Backend API Development (Express.js):**
   * Set up an Express.js server.
   * Create a POST API endpoint (/calculate-tax) to receive income data.
   * Implement a function to calculate tax based on income slabs.
   * Send the calculated tax amount as a JSON response.
2. **Frontend WebComponent Development:**
   * Create a WebComponent-based form (using Vanilla JS, Lit, or Stencil).
   * Include an input field for the user to enter income.
   * Submit the data to the backend via an API request.
   * Display the calculated tax dynamically within the component.
3. **Integration & Result Display:**
   * Ensure seamless communication between frontend and backend.
   * Implement error handling for invalid inputs.
   * Style the WebComponent using CSS.

**Mathematical Calculation/Steps:**

* Define tax slabs (Example: 0-2.5L: 0%, 2.5L-5L: 5%, 5L-10L: 20%, Above 10L: 30%).
* Calculate tax progressively for each slab.
* Return the total tax payable.

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

* express (for backend server)
* cors (to handle CORS in API calls)
* body-parser (for processing JSON requests, if needed)

**Acceptance Criteria:**

* The backend API correctly calculates and returns tax based on input.
* The WebComponent successfully submits data and displays the calculated tax.
* Proper error handling for invalid inputs.
* Clean, modular, and well-documented code.

**Submission Guidelines:**

1. Fork the provided repository.
2. Create a folder named <your-name>-income-tax-calculator.
3. Implement the backend and frontend in separate directories within the folder.
4. Push the completed code to your forked repository.
5. Submit a pull request with your solution.

**Reference:**

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

Ensure that the backend handles requests correctly, processes data accurately, and integrates seamlessly with the frontend.