**Title: Sales Tax 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 data submission and display
* Integrate frontend and backend through API calls
* Process and return data dynamically using AI-generated inputs

**Assignment Description:** The goal of this assignment is to build a full-stack Sales Tax Calculator. The frontend will feature a simple form where users input product details (price, category, and quantity). The backend will receive this data, calculate the total price including sales tax, and return the result to be displayed on the frontend. AI should generate product details dynamically to test different scenarios.

**Tasks & Steps:**

1. **Backend API Development:**
   * Set up an Express.js server.
   * Create an endpoint to receive form data.
   * Process tax calculations based on product category.
   * Return the final amount including sales tax.
2. **Frontend Development:**
   * Create a WebComponent-based form with input fields for price, category, and quantity.
   * Send form data to the backend using fetch API.
   * Display the computed result 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:** Total Price = (Price \* Quantity) + (Tax Rate \* Price \* Quantity)
* **Example Tax Rates:**
  + General Goods: 10%
  + Food Items: 5%
  + Luxury Goods: 20%

**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 sales tax calculation is accurate and category-based.
* 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.