**Title: Temperature Converter**

**Tool Details:**

* **Backend:** Node.js with Express.js
* **Frontend:** Web Components (vanilla JavaScript, no frameworks)
* **Database:** Not required
* **AI Integration:** Convert temperature units dynamically based on user input

**Goal:** Candidates will learn how to build an Express.js backend and integrate it with a WebComponent-based frontend. They will practice API request handling, processing user input, performing unit conversions, and dynamically updating the UI.

**Assignment Description:** Candidates will create a Temperature Converter where users can input a temperature value and select the conversion type (Celsius to Fahrenheit, Fahrenheit to Celsius, etc.). The frontend form collects these inputs and sends them to the backend via an API request. The backend processes the request, performs the conversion, and returns the result to the frontend. The frontend then displays the converted temperature.

**Tasks & Steps:**

1. **Backend Development:**
   * Set up an Express.js server.
   * Create an API endpoint to accept user input (temperature value and conversion type).
   * Perform the temperature conversion based on the selected type.
   * Return the converted value to the frontend.
2. **Frontend Development:**
   * Create a Web Component for the input form (text field for temperature and dropdown for conversion type).
   * Send the user input to the backend via a fetch API request.
   * Receive the response and update the UI dynamically by displaying the converted temperature.
3. **Result Display:**
   * Show the converted temperature in a designated display area on the frontend.
   * Provide an option to reset the form.

**Mathematical Calculation/Steps:**

* **Celsius to Fahrenheit:** F = (C × 9/5) + 32
* **Fahrenheit to Celsius:** C = (F - 32) × 5/9
* **Celsius to Kelvin:** K = C + 273.15
* **Kelvin to Celsius:** C = K - 273.15

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

* express (for backend server)
* cors (to allow frontend-backend communication)
* body-parser (to handle JSON requests)

**Acceptance Criteria:**

* The backend should correctly handle temperature conversion requests.
* The frontend should capture user input and dynamically update the UI.
* The API should return correct conversion values.
* The Web Component should encapsulate form logic and interaction.
* The conversion should be accurate based on mathematical formulas.

**Submission Guidelines:**

1. Fork the provided repository.
2. Create a folder with your name inside the repo.
3. Implement both backend and frontend in the designated folder.
4. Push the completed code to your forked repo.
5. Submit a pull request for review.

The assignment is complete when the backend correctly processes input and returns valid conversion results, and the frontend successfully updates the UI based on API responses.