**Title: Math Quiz Generator**

**Tool Details:**

* **Backend:** Node.js with Express.js
* **Frontend:** Web Components (vanilla JavaScript, no frameworks)
* **Database:** Not required
* **AI Integration:** Dynamically generate math quiz questions and evaluate answers

**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, dynamically generating quiz questions, and processing user responses.

**Assignment Description:** Candidates will create a Math Quiz Generator where users can select the difficulty level and number of questions. The frontend form collects these inputs and sends them to the backend via an API request. The backend generates random math questions, returns them to the frontend, and evaluates the user’s answers when submitted.

**Tasks & Steps:**

1. **Backend Development:**
   * Set up an Express.js server.
   * Create an API endpoint to accept user input (difficulty level and number of questions).
   * Generate random math questions based on the selected difficulty level.
   * Return the generated questions to the frontend.
   * Accept user answers and evaluate them, returning a score.
2. **Frontend Development:**
   * Create a Web Component for the quiz setup form (dropdown for difficulty, number input for questions, and a start button).
   * Display the generated questions dynamically in the UI.
   * Collect user answers and send them back to the backend for evaluation.
   * Receive the score and update the UI dynamically with the result.
3. **Result Display:**
   * Show the generated questions and answer fields.
   * Display the user’s final score after submission.

**Mathematical Calculation/Steps:**

* Difficulty Levels:
  + **Easy:** Addition and subtraction (e.g., 5 + 3, 10 - 4)
  + **Medium:** Multiplication and division (e.g., 6 × 7, 48 ÷ 6)
  + **Hard:** Mixed operations, exponents (e.g., 5 × (3 + 2), 2^3)
* Random Number Generation for Questions:
  + Use Math.random() to create operands within a range based on difficulty.

**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 generate random math questions.
* The frontend should display the quiz dynamically and collect answers.
* The API should evaluate answers and return a valid score.
* The Web Component should encapsulate quiz logic and interaction.
* The quiz should support different difficulty levels.

**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 generates valid quiz questions, evaluates answers correctly, and the frontend updates the UI dynamically based on API responses.