**Title: Random Number Generator**

**Tool Details:**

* **Backend:** Node.js with Express.js
* **Frontend:** Web Components (vanilla JavaScript, no frameworks)
* **Database:** Not required
* **AI Integration:** Generate random numbers 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, and dynamically updating the UI.

**Assignment Description:** Candidates will create a Random Number Generator where users specify a minimum and maximum range. The frontend form collects these inputs and sends them to the backend via an API request. The backend processes the request, generates a random number within the range, and returns it to the frontend. The frontend then displays the generated number.

**Tasks & Steps:**

1. **Backend Development:**
   * Set up an Express.js server.
   * Create an API endpoint to accept user input (min and max values).
   * Validate inputs and generate a random number within the given range.
   * Return the generated number to the frontend.
2. **Frontend Development:**
   * Create a Web Component for the input form (two number fields for min and max values and a button to generate a number).
   * Send the user input to the backend via a fetch API request.
   * Receive the response and update the UI dynamically by displaying the random number.
3. **Result Display:**
   * Show the generated number in a designated display area on the frontend.

**Mathematical Calculation/Steps:**

* Random Number Formula: Math.floor(Math.random() \* (max - min + 1)) + min;

**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 requests and return valid random numbers.
* The frontend should capture user input and dynamically update the UI.
* The API should respond correctly for various input ranges.
* The Web Component should encapsulate form logic and interaction.

**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 processes the input and returns a valid random number, and the frontend successfully updates the UI based on API responses.