**Title: Dice Roller**

**Tool Details:**

* **Backend:** Node.js with Express.js
* **Frontend:** Web Components (vanilla JavaScript, no frameworks)
* **Database:** Not required
* **AI Integration:** Simulate rolling a die and generate random values dynamically

**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, generating random values, and dynamically updating the UI.

**Assignment Description:** Candidates will create a Dice Roller where users can select the number of dice to roll and view the generated random values. The frontend form collects user input and sends it to the backend via an API request. The backend processes the request, rolls the dice by generating random numbers, and returns the results. The frontend then displays the rolled numbers.

**Tasks & Steps:**

1. **Backend Development:**
   * Set up an Express.js server.
   * Create an API endpoint to accept user input (number of dice to roll).
   * Generate random numbers between 1 and 6 for each die roll.
   * Return the rolled numbers to the frontend.
2. **Frontend Development:**
   * Create a Web Component for the input form (number field to select dice count and a roll button).
   * Send the user input to the backend via a fetch API request.
   * Receive the response and update the UI dynamically by displaying the rolled numbers.
3. **Result Display:**
   * Show the rolled numbers visually (e.g., as dice images or text output).
   * Provide an option to roll again.

**Mathematical Calculation/Steps:**

* Generate a random number using Math.floor(Math.random() \* 6) + 1 for each die.
* Repeat the process based on the selected number of dice.

**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 dice rolls based on user input.
* The frontend should capture user input and dynamically update the UI.
* The API should return valid random numbers between 1 and 6.
* The Web Component should encapsulate form logic and interaction.
* The output should be visually clear and engaging.

**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 the dice roll request, returns valid results, and the frontend successfully updates the UI based on API responses.