**Title: Gradient Generator**

**Tool Details:**

* **Backend:** Node.js with Express.js
* **Frontend:** Web Components (vanilla JavaScript, no frameworks)
* **Database:** Not required
* **AI Integration:** Dynamic gradient generation based on user input

**Goal:** Candidates will learn how to develop a backend using Express.js and integrate it with a WebComponent-based frontend. They will gain experience in handling API requests, processing data, and dynamically updating the UI.

**Assignment Description:** Candidates will create a Gradient Generator tool where users input two colors and a gradient type (linear or radial). The frontend form collects user input and sends it to the backend via an API request. The backend processes the request and returns a dynamically generated gradient CSS string. The frontend then updates the display accordingly.

**Tasks & Steps:**

1. **Backend Development:**
   * Set up an Express.js server.
   * Create an API endpoint to accept user input (color1, color2, gradient type).
   * Process the request and generate a gradient CSS string.
   * Return the generated CSS to the frontend.
2. **Frontend Development:**
   * Create a Web Component for the input form (two color pickers and a dropdown for gradient type).
   * Send the user input to the backend via a fetch API request.
   * Receive the response and update the UI dynamically by applying the gradient.
3. **Result Display:**
   * Use the returned gradient CSS to style a preview box dynamically.

**Mathematical Calculation/Steps:**

* Linear Gradient: background: linear-gradient(direction, color1, color2);
* Radial Gradient: background: radial-gradient(color1, color2);

**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 CSS gradient values.
* The frontend should capture user input and dynamically update the UI.
* The API should respond correctly with different gradient types and colors.
* 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 the correct gradient, and the frontend successfully updates the UI based on API responses.