**Title: Even-Odd Checker**

**Tool Details:**

* **Backend:** Node.js with Express.js
* **Frontend:** Web Components (vanilla JavaScript, no frameworks)
* **Database:** Not required
* **AI Integration:** Dynamically determine whether a number is even or odd

**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, simple mathematical computations, and dynamic UI updates.

**Assignment Description:** Candidates will create an Even-Odd Checker where users can input a number, send it to the backend via an API request, and receive a response indicating whether the number is even or odd. The frontend will display the result dynamically.

**Tasks & Steps:**

1. **Backend Development:**
   * Set up an Express.js server.
   * Create an API endpoint that accepts a number from the frontend.
   * Determine whether the number is even or odd using modulo operation.
   * Return the result to the frontend.
2. **Frontend Development:**
   * Create a Web Component for the input form (text field for number input and a check button).
   * Send the user input to the backend via a fetch API request.
   * Receive the response and dynamically display the result on the UI.
3. **Result Display:**
   * Show "Even" or "Odd" based on the API response.
   * Provide an option to check another number.

**Mathematical Calculation/Steps:**

* Use the modulo operator: number % 2 === 0 → Even, otherwise Odd.

**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 determine whether the input number is even or odd.
* The frontend should capture user input and dynamically update the UI.
* The API should return valid results based on the given number.
* The Web Component should encapsulate form logic and interaction.
* The result should be clearly displayed to the user.

**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 number, returns accurate even/odd results, and the frontend successfully updates the UI based on API responses.