**Title:** Grade Calculator

**Tool Details:**

* **Backend:** Node.js, Express.js
* **Frontend:** Web Components (Vanilla JS, HTML, CSS)
* **Database (Optional):** In-memory storage (JSON or array)

**Goal:** By completing this assignment, candidates will learn how to develop a backend using Express.js, integrate it with a WebComponent-based frontend, and handle user input dynamically. This assignment will reinforce knowledge of API creation, request handling, and dynamic frontend updates.

**Assignment Description:** Candidates will build a Grade Calculator tool where users input their scores through a WebComponent-based frontend. The input will be sent to an Express.js backend, which will process the data and return a calculated grade based on predefined grading criteria. The result will be displayed dynamically in the frontend.

**Tasks & Steps:**

1. **Backend API Development (Express.js):**
   * Set up an Express.js server.
   * Create a route (/calculate-grade) to handle form submissions.
   * Process the input scores and calculate the grade.
   * Return the calculated grade in JSON format.
2. **Frontend Development (WebComponent):**
   * Create a WebComponent with an HTML form for user input (subject marks).
   * Handle form submission and send a request to the backend.
   * Display the calculated grade dynamically within the WebComponent.
3. **Integration & Result Display:**
   * Ensure the frontend correctly interacts with the backend.
   * Use fetch API to send user input and receive the calculated grade.
   * Dynamically update the UI with the received result.

**Mathematical Calculation/Steps:**

* The backend should calculate the grade based on the following scale:
  + **90-100:** A
  + **80-89:** B
  + **70-79:** C
  + **60-69:** D
  + **Below 60:** F
* The final grade should be determined as the average of all input scores.

**Third-Party Packages (if required):**

* **Express.js** (for backend API development)
* **Cors** (if necessary, to allow frontend-backend communication)
* **Body-parser** (to handle JSON data)

**Acceptance Criteria:**

* The Express.js backend correctly handles user input and processes the grade calculation.
* The WebComponent-based frontend sends user input via API calls.
* The received grade is displayed dynamically within the frontend UI.
* The implementation should follow best practices for clean and modular code.

**Submission Guidelines:**

1. Fork the provided repository.
2. Create a folder named <your-github-username> inside the repo.
3. Implement the backend and frontend solution within this folder.
4. Push your code to the forked repository.
5. Submit a pull request for review.

**Reference:**

1. [**https://www.calculator.net/**](https://www.calculator.net/)