**Title: Percentage Calculator**

**Tool Details:**

* **Backend:** Node.js with Express.js
* **Frontend:** WebComponent-based UI
* **Database:** Not required
* **AI Integration:** AI dynamically generates input details and suggestions

**Goal:** By completing this assignment, candidates will learn how to:

* Build a simple backend API using Express.js
* Develop a frontend using WebComponents
* Integrate frontend and backend for seamless communication
* Implement dynamic AI-generated input handling

**Assignment Description:** Candidates will develop a Percentage Calculator tool that takes user input (obtained marks and total marks), sends it to an Express.js backend, processes the data, and returns the calculated percentage. The WebComponent-based frontend should dynamically adjust based on AI-generated inputs and provide an intuitive user experience.

**Tasks & Steps:**

1. **Backend Development:**
   * Set up an Express.js server.
   * Create an API endpoint (/calculate-percentage) that receives obtained marks and total marks.
   * Implement request validation to ensure both inputs are numbers.
   * Process the input and calculate the percentage.
   * Return the calculated percentage as a JSON response.
2. **Frontend Development:**
   * Create a WebComponent-based UI with a form.
   * Implement input fields for obtained marks and total marks.
   * Use AI to generate random sample inputs dynamically.
   * Add a submit button that sends data to the backend via a fetch request.
   * Display the calculated percentage dynamically on the page.
3. **Integration:**
   * Ensure the frontend properly sends a POST request with user input.
   * Handle responses and display the result in real time.
   * Implement basic error handling for invalid inputs.

**Mathematical Calculation/Steps:** Formula: Percentage=(Obtained MarksTotal Marks)×100\text{Percentage} = \left( \frac{\text{Obtained Marks}}{\text{Total Marks}} \right) \times 100

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

* express (for backend API handling)
* cors (for cross-origin requests, if needed)
* body-parser (for handling JSON request bodies)

**Acceptance Criteria:**

* The backend API correctly processes requests and returns percentage calculations.
* The frontend dynamically generates input data using AI.
* The WebComponent-based UI properly integrates with the backend.
* Errors (such as non-numeric input or missing fields) are handled gracefully.
* The final percentage is displayed in a user-friendly manner.

**Submission Guidelines:**

1. Fork the provided repository.
2. Create a new folder with your name inside the repo.
3. Implement the backend and frontend solution within your folder.
4. Ensure the application runs successfully with npm install and npm start.
5. Push your changes to your forked repo.
6. Submit a pull request with a brief description of your implementation.

**Reference:**

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