**Title: Age Calculator**

**Tool Details:**

* **Backend:** Express.js
* **Frontend:** WebComponent-based UI

**Goal:**

Candidates will learn how to create a backend API using Express.js and integrate it with a WebComponent-based frontend. They will gain hands-on experience in handling user input, processing data on the backend, and dynamically updating the frontend with AI-generated results.

**Assignment Description:**

Develop an age calculator where users input their date of birth (DOB) in the frontend form. The backend receives the input, calculates the user's age dynamically using AI-generated logic, and sends the result back to the frontend for display.

**Tasks & Steps:**

**Backend API Development (Express.js)**

1. Set up an Express.js server.
2. Create an endpoint that accepts a date of birth as input.
3. Implement logic to calculate the age dynamically.
4. Return the calculated age in a JSON response.

**Frontend Development (WebComponent)**

1. Create a WebComponent with a form to input DOB.
2. Send the input data to the backend via an API request.
3. Display the returned age dynamically in the UI.

**Integration & Result Display:**

1. Ensure seamless communication between frontend and backend.
2. Display age updates without requiring a full page refresh.

**Mathematical Calculation/Steps:**

* Extract the year, month, and day from the input DOB.
* Get the current date.
* Calculate the difference to determine the exact age.
* AI should generate custom messages based on age ranges (e.g., "You are in your 20s, a great time to explore new skills!").

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

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

**Acceptance Criteria:**

* The backend correctly processes and returns the age based on the given DOB.
* The frontend successfully sends requests and displays the age dynamically.
* The application provides AI-generated insights/messages along with the age.
* The solution follows best coding practices and is well-structured.

**Submission Guidelines:**

1. Fork the repository provided in the assignment.
2. Create a folder with your GitHub username.
3. Implement the backend and frontend inside your folder.
4. Push the code to your forked repository.
5. Submit a pull request for review.

**Reference:**

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