**Title: Greeting Message API**

**Tool Details:**

* **Backend:** Node.js with Express.js
* **Frontend:** Web Components (vanilla JavaScript, no frameworks)
* **Database:** Not required
* **AI Integration:** Generate personalized greeting messages dynamically

**Goal:** Candidates will learn how to build a simple Express.js API that generates greeting messages based on user input. They will practice API development, request handling, and frontend-backend integration using Web Components.

**Assignment Description:** Candidates will create a Greeting Message API that returns personalized greeting messages based on user input (e.g., name and time of day). The backend will process requests and generate greetings dynamically, while the frontend will allow users to enter their name and receive a customized message.

**Tasks & Steps:**

1. **Backend Development:**
   * Set up an Express.js server.
   * Create an API endpoint to receive a user's name and generate a greeting message.
   * Optionally, modify the greeting based on the current time of day (morning, afternoon, evening).
   * Return the generated message as a JSON response.
2. **Frontend Development:**
   * Create a Web Component with an input field and submit button.
   * Send a request to the backend with the user’s name.
   * Display the personalized greeting message dynamically on the UI.
3. **Result Display:**
   * Show the greeting message in a structured format.
   * Optionally, style the message based on time of day.

**Mathematical Calculation/Steps:**

* Determine the time of day (morning, afternoon, evening) based on the current hour.
* Generate a greeting message accordingly, e.g., "Good morning, John!".

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

* express (for backend server)
* cors (to allow frontend-backend communication)
* body-parser (to handle JSON request data)

**Acceptance Criteria:**

* The backend correctly generates greeting messages based on user input.
* The frontend allows users to enter their name and displays the greeting dynamically.
* The API should return appropriate messages based on time of day.
* The Web Component encapsulates the form and request logic properly.
* Users should receive instant feedback when a greeting is generated.

**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 generates accurate greetings, the frontend successfully interacts with the API, and the message is displayed dynamically.