**Title: Basic Contact Form Backend**

**Tool Details:**

* **Backend:** Node.js with Express.js
* **Frontend:** Web Components (vanilla JavaScript, no frameworks)
* **Database:** Not required (store data in memory or log it to a file)
* **AI Integration:** Validate and process form data dynamically

**Goal:** Candidates will learn how to build an Express.js backend that handles contact form submissions and integrates with a WebComponent-based frontend. They will practice API request handling, form validation, and dynamic UI updates.

**Assignment Description:** Candidates will create a Basic Contact Form Backend where users can submit their contact details (name, email, message). The backend will validate the input, process it, and return a confirmation response. The frontend will allow users to input their details and display the submission result dynamically.

**Tasks & Steps:**

1. **Backend Development:**
   * Set up an Express.js server.
   * Create an API endpoint to handle contact form submissions.
   * Validate the incoming form data (ensure name, email, and message are provided).
   * Process the data (store in memory, log to a file, or simulate database storage).
   * Return a success or error response.
2. **Frontend Development:**
   * Create a Web Component containing a form with fields for Name, Email, and Message.
   * Send a request to the backend with form data.
   * Display success or error messages dynamically.
3. **Result Display:**
   * Show a confirmation message on successful form submission.
   * Display error messages if the form validation fails.
   * Optionally, provide a way to reset the form.

**Mathematical Calculation/Steps:**

* Validate email format using regex (/^[\w-\.]+@([\w-]+\.)+[\w-]{2,4}$/).
* Ensure the message length meets minimum requirements.

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

* express (for backend server)
* cors (to allow frontend-backend communication)
* body-parser (to handle JSON request data)
* nodemailer (optional, for sending emails)

**Acceptance Criteria:**

* The backend correctly validates and processes form submissions.
* The frontend captures user input and dynamically updates the UI.
* The API correctly handles errors and returns appropriate responses.
* The Web Component encapsulates the form and submission logic properly.
* Users should see clear feedback after form submission.

**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 processes contact form submissions correctly, returns appropriate responses, and the frontend successfully updates the UI based on API interactions.