**Title: Random Quote Generator**

**Tool Details:**

* **Backend:** Node.js with Express.js
* **Frontend:** Web Components (vanilla JavaScript, no frameworks)
* **Database:** Not required
* **AI Integration:** Randomly generate and return motivational or famous quotes

**Goal:** Candidates will learn how to build an Express.js backend that serves random quotes and integrate it with a WebComponent-based frontend. They will practice API request handling, random data selection, and dynamic UI updates.

**Assignment Description:** Candidates will create a Random Quote Generator where users can click a button to fetch a random quote from the backend. The backend will return a randomly selected quote from a predefined list, and the frontend will display it dynamically.

**Tasks & Steps:**

1. **Backend Development:**
   * Set up an Express.js server.
   * Create an API endpoint that returns a random quote from a predefined list.
   * Ensure the response format includes both the quote text and author.
2. **Frontend Development:**
   * Create a Web Component that displays quotes dynamically.
   * Implement a button that fetches a new quote on click.
   * Send API requests to the backend and update the UI with the received quote.
3. **Result Display:**
   * Show the quote text and author.
   * Provide an option to fetch a new quote.

**Mathematical Calculation/Steps:**

* Use Math.random() to select a random index from the quotes array.

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

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

**Acceptance Criteria:**

* The backend should return a random quote on each request.
* The frontend should capture the response and dynamically update the UI.
* The API should provide valid quote data (text and author).
* The Web Component should encapsulate the display logic.
* Users should be able to generate new quotes by clicking a button.

**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 correctly serves random quotes, and the frontend successfully updates the UI with API responses.