**Title: URL Encoder & Decoder**

**Tool Details:**

* **Backend:** Node.js with Express.js
* **Frontend:** Web Components (vanilla JavaScript, no frameworks)
* **Database:** Not required
* **AI Integration:** Encode and decode URLs dynamically

**Goal:** Candidates will learn how to build an Express.js backend that encodes and decodes URLs and integrates with a WebComponent-based frontend. They will practice API request handling, string manipulation, and UI updates.

**Assignment Description:** Candidates will create a URL Encoder & Decoder tool where users can input a URL to be encoded or decoded. The backend will process the request and return the transformed URL, which the frontend will display dynamically.

**Tasks & Steps:**

1. **Backend Development:**
   * Set up an Express.js server.
   * Create API endpoints for encoding and decoding URLs.
   * Use built-in JavaScript functions to encode (encodeURIComponent) and decode (decodeURIComponent) URLs.
   * Return the processed data as a JSON response.
2. **Frontend Development:**
   * Create a Web Component containing an input field and buttons for encoding and decoding.
   * Send a request to the backend with the user input.
   * Display the encoded or decoded result dynamically.
3. **Result Display:**
   * Show the encoded or decoded URL in a structured format.
   * Provide an option to copy the result to the clipboard.

**Mathematical Calculation/Steps:**

* Use encodeURIComponent to replace special characters in the URL.
* Use decodeURIComponent to convert encoded values back to their original form.

**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 encodes and decodes URLs.
* The frontend captures user input and dynamically updates the UI.
* The API should handle both encoding and decoding requests.
* The Web Component should encapsulate the form and request logic properly.
* Users should see clear feedback when a URL is processed.

**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 encodes and decodes URLs, returns accurate responses, and the frontend successfully updates the UI based on API interactions.