**Title: Random Password Generator**

**Tool Details:**

* **Backend:** Node.js with Express.js
* **Frontend:** Web Components (vanilla JavaScript, no frameworks)
* **Database:** Not required
* **AI Integration:** Generate secure random passwords dynamically based on user input

**Goal:** Candidates will learn how to build an Express.js backend and integrate it with a WebComponent-based frontend. They will practice API request handling, processing user input, and dynamically updating the UI.

**Assignment Description:** Candidates will create a Random Password Generator where users specify the password length and character set (uppercase, lowercase, numbers, symbols). The frontend form collects these inputs and sends them to the backend via an API request. The backend processes the request, generates a secure random password based on the criteria, and returns it to the frontend. The frontend then displays the generated password.

**Tasks & Steps:**

1. **Backend Development:**
   * Set up an Express.js server.
   * Create an API endpoint to accept user input (password length and character set options).
   * Validate inputs and generate a random password based on the given criteria.
   * Return the generated password to the frontend.
2. **Frontend Development:**
   * Create a Web Component for the input form (a number field for length and checkboxes for character set options).
   * Send the user input to the backend via a fetch API request.
   * Receive the response and update the UI dynamically by displaying the generated password.
3. **Result Display:**
   * Show the generated password in a designated display area on the frontend.
   * Provide a button to copy the password to the clipboard.

**Mathematical Calculation/Steps:**

* Random Password Generation Logic:
  + Define possible character sets based on user input.
  + Use Math.random() to pick characters randomly from the selected set.
  + Repeat until the password reaches the specified length.

**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 correctly handle requests and return a secure random password.
* The frontend should capture user input and dynamically update the UI.
* The API should respond correctly for various input criteria.
* The Web Component should encapsulate form logic and interaction.
* The password should meet user-specified constraints (length and character set).

**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 the input and returns a valid random password, and the frontend successfully updates the UI based on API responses.