**Title: Image to Base64 Converter**

**Tool Details:**

**Technology Stack:**

* **Frontend:** Web Components (Lit or Vanilla JS)
* **Backend:** Express.js (Node.js)
* **File Processing:** Convert image files to **Base64 encoded strings**
* **Data Handling:** File uploads & API requests

**Goal:**  
By completing this assignment, candidates will:

* Learn how to build a **RESTful API** using **Express.js**.
* Implement **image encoding** to **Base64 format**.
* Work with **Web Components** to create a frontend for image upload.
* Understand **client-server communication** via API calls.
* Gain experience in handling **binary data and encoding techniques**.

**Assignment Description:**

Develop an **Image to Base64 Converter** where users upload an **image file** (e.g., PNG, JPG) via a **WebComponent-based form**. The backend, built with **Express.js**, will process the image and return the **Base64 encoded string**. The frontend should display both the **original image** and the **Base64 output**.

**Tasks & Steps:**

**1. Backend API Development (Express.js):**

* Set up an **Express.js** server to handle **file uploads**.
* Implement logic to **convert images into Base64 strings**.
* Implement API routes to **process the file and return the encoded string**.

**2. Frontend (WebComponent-based UI):**

* Create a **form** using Web Components that allows users to **upload an image file**.
* Send the uploaded file to the backend using the **fetch API**.
* Display the **original image** and the **Base64 encoded result** on the UI.

**3. Integration & Testing:**

* Ensure the frontend **properly sends data** to the backend.
* Handle errors gracefully (e.g., **invalid file types, large files**).
* Test **end-to-end functionality** to ensure seamless integration.

**Mathematical Calculation/Steps (if applicable):**

* Base64 encoding represents **binary data in an ASCII string** using a **64-character set**.
* The encoded size is approximately **33% larger** than the original binary file.

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

* express (for backend server)
* multer (for handling file uploads)
* fs (Node.js built-in file system module for reading images)
* lit (for WebComponent development)

**Acceptance Criteria:**

* The **Express.js backend** should successfully process **uploaded image files** and return a **Base64 string**.
* The **WebComponent-based frontend** should have a **responsive form** that allows users to upload files and displays the **Base64 encoded output**.
* Proper **error handling** should be in place for **unsupported file formats**.
* The application should work seamlessly across **modern web browsers**.

**Submission Guidelines:**

1. **Fork** the provided GitHub repository.
2. **Create a folder** named image-to-base64-<your-name>.
3. **Implement the backend and frontend** in the respective subfolders.
4. **Push the code** to your forked repository.
5. **Submit a pull request** with a brief description of your implementation.

**Ensure that the backend correctly handles requests, processes the image file, and seamlessly integrates with the WebComponent-based frontend.**