**Title: SQL to JSON Converter**

**Tool Details:**

**Technology Stack:**

* **Frontend:** Web Components (Lit or Vanilla JS)
* **Backend:** Express.js (Node.js)
* **Database:** SQLite or MySQL
* **Processing:** Convert SQL query results into JSON format
* **Data Handling:** API requests & response parsing

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

* Learn how to build a **RESTful API** using **Express.js**.
* Implement **SQL query execution and JSON conversion** using Node.js.
* Work with **Web Components** to create an interactive frontend.
* Understand **client-server communication** via API calls.
* Gain experience in **handling database queries and data transformation**.

**Assignment Description:**

Develop an **SQL to JSON Converter** where users can input **SQL queries** via a **WebComponent-based form**. The backend, built with **Express.js**, will execute the query on an **SQLite/MySQL database** and return a **JSON response** with the structured data.

**Tasks & Steps:**

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

* Set up an **Express.js** server with a **SQLite/MySQL database**.
* Implement an API that **accepts SQL queries**, executes them, and converts the results to JSON.
* Handle query validation and prevent **SQL injection attacks**.

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

* Create a **form** using Web Components where users can **input SQL queries**.
* Send the SQL query to the backend using the **fetch API**.
* Display the **converted JSON output** after processing.
* Provide an option to **download the JSON file**.

**3. Integration & Testing:**

* Ensure the frontend **properly communicates** with the backend.
* Handle errors gracefully (e.g., **invalid SQL syntax or database errors**).
* Test the **SQL query execution and JSON conversion** to ensure accuracy.

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

* **SQL Execution:** Run the SQL query against the database.
* **Data Structuring:** Convert **query results** into a **JSON format**.

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

* express (for backend server)
* sqlite3 or mysql2 (for database connection)
* body-parser (for parsing incoming request data)
* lit (for WebComponent development)

**Acceptance Criteria:**

* The **Express.js backend** should successfully **execute SQL queries and return JSON**.
* The **WebComponent-based frontend** should have a **form** for SQL query input.
* The converted JSON output should be **properly formatted and displayed**.
* Proper **error handling** should be in place for **invalid SQL input**.

**Submission Guidelines:**

1. **Fork** the provided GitHub repository.
2. **Create a folder** named sql-to-json-<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 processes SQL queries and integrates seamlessly with the WebComponent-based frontend.**