**Title: Due Date Calculator**

**Tool Details:**

* **Backend:** Node.js with Express.js
* **Frontend:** WebComponent-based UI
* **Database (Optional):** JSON storage or in-memory data handling

**Goal:** Candidates will learn to:

* Develop a RESTful API using Express.js
* Implement a WebComponent-based frontend for user input and result display
* Integrate frontend and backend through API calls
* Process and return calculated due dates dynamically

**Assignment Description:** The objective of this assignment is to build a full-stack Due Date Calculator. The frontend will feature a form where users input a start date and a duration (in days, weeks, or months). The backend will process the data, compute the due date, and return the result for display on the frontend. AI should generate different input values dynamically for testing purposes.

**Tasks & Steps:**

1. **Backend API Development:**
   * Set up an Express.js server.
   * Create an endpoint to receive start date and duration.
   * Perform due date calculations based on the input.
   * Return the computed due date to the frontend.
2. **Frontend Development:**
   * Create a WebComponent-based form with input fields for start date and duration.
   * Send form data to the backend using fetch API.
   * Display the computed due date received from the backend.
3. **Integration & Testing:**
   * Ensure the frontend correctly sends user input to the backend.
   * Validate responses from the backend.
   * Handle errors gracefully in both frontend and backend.

**Mathematical Calculation/Steps:**

* **Due Date Calculation:**
  + Due Date = Start Date + Duration (in days, weeks, or months)
  + Ensure correct handling of leap years and different month lengths.

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

* express (for backend API)
* cors (to handle cross-origin requests)
* body-parser (for processing JSON input)
* moment.js or date-fns (for date calculations)

**Acceptance Criteria:**

* The backend API correctly receives and processes input data.
* The WebComponent-based frontend interacts seamlessly with the backend.
* The due date calculation is accurate, considering various durations and date formats.
* AI dynamically generates input scenarios for testing.
* The frontend displays results in a user-friendly manner.

**Submission Guidelines:**

1. Fork the provided repository.
2. Create a new folder with your name inside the repository.
3. Implement the solution within the folder.
4. Push the completed code to your forked repository.
5. Submit a pull request to the original repository.

Ensure the solution meets all criteria before submission.