Integration Specialist Exercise

Task One: Create a form in a local web application that collects users first name, last name, email, and phone number. When you submit the form, an event should be sent into Google Tag Manager. Configure Google Tag Manager to send the trigger into Google Analytics. Create and configure a goal in Google Analytics so you can display said event as a conversion.

Task Two: Use the form from Task one and push the information collected into Zapier upon submission. In Zapier, validate that the email and phone number are valid using a code block. If the email and phone number are valid then use the credentials below to push the lead into Further. If the email and phone number are not valid then add the user's information into a Google Sheet.

✅ Do not use the TalkFurther application in Zapier. Use a code block to make the API request.

🔑API Key: jbhVuPYn.vHR5WMQjy60k0fsAqJFqm0yPHGek86PT

Community ID: 142430

Documentation: https://api-docs.talkfurther.com/#746b4bf4-a785-4b90-9a85-a3202b787e66

Success looks like: { "success": "Created new Lead"}

Task Three: Implement deduplication logic in your Zap, so that if a lead with the same phone number resubmits themselves, a new lead is not created. Instead an email notification is sent out to notify that a lead has revisited the form.

Task Four: Please use CSS, HTML, and/or Javascript to style your project in a way that is aesthetically pleasing, simple, and creates a great user experience.

✅ Create a loom https://www.loom.com/ demoing your application and how it works. Please ensure you demo the events in Google Tag Manager and Google Analytics as well as the results from the Zap run. Then explain how you implemented each task including challenges and solutions.

Task Five:

Implement the following queries in SQL, given the schema below.

1. List authors(id, first_name, last_name, country_name), book name, ISBN, price, discount, is_hard_copy - if they have books, or null if they don't. Order by author last_name, first_name.
2. List authors (id, first_name, last_name, country_name) where country code is the USA.
3. List authors(id, first_name, last_name, country_name) with books. Order by the number of books descending.
4. Select how many books are from USA authors.
5. Select books (title, isbn, discount, price) where 20 <= discount <=30, order by price increasing.
6. List the cheapest book (price) of every author (first_name, last_name). If an author does not have books, display -1 as the price.