CereBI
Junior Intern Assignment

Objective

This assignment evaluates your ability to analyze a complex business problem in logistics, propose an effective logical and technological solution, design an intuitive user interface, and create an initial system architecture with an emphasis on modern system integration and data management. We are looking for clear thinking, creative problem-solving, and a foundational understanding of software development principles.

What to Submit

Code and answers should be stored in a GitHub repo. A link to Figma/Canva/Miro can be included in the answers file.
Make sure your answers include:

- **Your complete approach description:** Problem understanding, logical solution, and justifications for your choices.
- **A designed screen:** A screenshot or link to your UX/UI design.
- **A fictitious data table.**
- **Detailed and reasoned answers** to all guiding questions, integrated logically into your explanations or as a clear, separate section.

Emphasis on:

- **Clarity of thought and communication:** The ability to articulate complex ideas clearly and concisely.
- **Readable and usable design:** Not just visual, but structural – how the screen will actually serve users.
- **Basic understanding of modern information system architecture:** How different components interact.
- **Presenting a holistic solution:** A successful combination of business, logical, design, and technical understanding.

Estimated Time for Assignment: Up to 4 hours of focused work. We appreciate your time and understand this is a short project.

Integrity in Your Assignment

We expect this assignment to reflect **your own work and understanding**. You're welcome to use standard learning resources and productivity tools to help you, but the solutions, code, and explanations should ultimately be **yours** and demonstrate **your full comprehension**. This ensures a fair and accurate evaluation of your skills.

Due Date: As mentioned in the interview invitation email.
Please submit your assignment to: career@cerebi.ai

The Assignment

Business Background: LogiDog is a leading global logistics company managing thousands of shipments daily across a wide range of sectors (consumer goods, light industry, medical equipment). Recently, we've seen an increase in customer complaints due to delays in shipment deliveries. The main challenge is proactively identifying shipments at risk of delay and the sources of these delays, which currently prevents quick and effective response. Our goal is to develop a simple yet powerful system that provides early alerts for shipments likely to be delayed, enabling our operations teams to take corrective actions before problems become critical.

1. In-Depth Problem Understanding and Logical Solution Proposal

- **Analysis of Delay Factors:** Briefly describe the common and potential causes of shipment delays for a logistics company of this scale. Focus on both operational factors and external factors.
- **Early Delay Identification Logic:** Present a simple **flowchart** or a clear **textual description** outlining your proposed logic for early delay identification. (For example: "If the delivery date is within X upcoming days, and the shipment is still in stage Y, then trigger an alert.")
    - **Guiding Questions for Discussion and Analysis:**
        - **What types of data (existing or to be collected) could serve as strong indicators for accurate delay prediction?** Explain the rationale for each data type.
        - **Considering a future implementation of a Machine Learning (ML) model:** Which existing data fields, or fields that need to be specifically collected, would be critical for training such a model? Explain why.
        - **Alert Triggering Approach:** Would you prefer alerts to be triggered by **rule-based** mechanisms or **dynamic analysis** (e.g., a basic statistical/ML model)? Explain the advantages and disadvantages of each approach in the context of this initial logistics system, and justify your choice.

2. User Interface (UX/UI) Design for a Primary Alerts Screen

- **Visual Design:** Design a **single blocks screen (Dashboard)** for the system (you can use Draw.io/Canva/Miro). The screen should include:
    - A **clear list of at-risk shipments.**
    - **Filtering and Search Capabilities**
    - **Visual Indicators.**
    - **Guiding Questions for Design and Technical Specification:**
        - **Real-time Data Updates:** Choose a communication approach to ensure the dashboard data is as up-to-date as possible. Explain your chosen approach and justify why it's the best fit for this system.
        - **Data Structure for Display:** Which data structure (schema) is most suitable for displaying this type of data on the screen? Would you prefer a **Normalized** or **Denormalized** structure? Provide a brief example. Explain the advantages of your chosen approach for this scenario.

3. Sample Data and Integration

- **Fictitious Data Creation:** Create a data table (JSON or SQL table(s)) for at least **15 sample shipments**, where some exhibit potential "delay" scenarios that align with your created logic and design. Ensure all fields defined in your screen design are included.
- **API Design**: Please design an API that will serve the client. Focus on input and output format and communication protocol.

4. Implement Delay Logic

Write a small code snippet (in any language of your choice) that:

- **Defines a function** (e.g., isShipmentAtRisk(shipmentData)).
- This function should **take a single shipment object** (using the structure from your fictitious data) as an input.
- **Implement your proposed logic** from Section 1 (e.g., "If deliveryDate is within X days AND currentStage is not Y") to determine if the shipment is at risk of delay.
- The function should **return true if the shipment is at risk, and false otherwise**.
- **Provide example calls** to this function with 2-3 different shipment objects from your sample data, demonstrating both "at risk" and "not at risk" scenarios, and print the results.