Frontend Developer Assessment

Mid-Level Full-Stack Position (Frontend Focus) — smti ERP System

Context

We are building an ERP system for a B2B event management company. The system includes an Inquiry Process module where staff manage hotel requests through multiple phases. You will build the **Inquiry Kanban Board** — a visual interface for tracking and managing inquiries.

Task: Inquiry Kanban Board

Build a Next.js application that displays inquiries as a Kanban board with drag-and-drop functionality, filtering, and detail views.

Business Requirements

- Inquiries move through 4 phases: **New → Sent to Hotels → Offers Received → Completed**
- Users can drag inquiry cards between columns to change their phase
- Each card shows: client name, event date, number of guests, potential value (CHF)
- Clicking a card opens a detail panel/modal with full information
- Users can filter by: client name, date range, minimum value
- Board should show column totals (count and sum of potential value)

Sample Data Structure

```
{
  "id": "INQ-2026-0034",
  "clientName": "Novartis AG",
  "contactPerson": "Anna Mueller",
  "eventType": "Conference",
  "eventDate": "2026-03-15",
  "guestCount": 120,
  "potentialValue": 48500,
  "phase": "offers_received",
  "hotels": ["Grand Hotel Zurich", "Hotel Schweizerhof"],
  "notes": "Client prefers city center location",
  "createdAt": "2026-01-10T09:00:00Z",
  "updatedAt": "2026-01-12T14:30:00Z"
}
```

Technical Requirements

1. Kanban Board Component

- 4 columns representing phases
- Drag and drop between columns
- Visual feedback during drag
- Column headers with count and total value
- Responsive design (works on tablet)

2. Inquiry Card Component

- Compact view with key information
- Visual indicator for high-value inquiries (>50K)
- Date formatting (relative: "2 days ago")
- Click to open detail view

3. Filter Panel

- Search by client name (debounced)
- Date range picker
- Minimum value slider or input
- Clear all filters button
- Show active filter count

4. Detail Panel/Modal

- Full inquiry information
- List of associated hotels
- Notes section
- Phase change dropdown (alternative to drag)
- Timestamps (created, last updated)

5. State Management

- Use Zustand or React Context
- Optimistic updates on phase change
- Persist filter state in URL params
- Handle loading and error states

6. API Integration

- Mock API with Next.js API routes
- GET /api/inquiries (with filter params)
- PATCH /api/inquiries/:id (phase update)
- Simulate 500ms network delay

Evaluation Criteria

Criteria	Weight	What We Look For
Component Design	25%	Reusability, prop design, composition patterns
State Management	25%	Data flow, optimistic updates, filter logic
UI/UX Quality	20%	Visual design, interactions, responsive behavior
TypeScript Usage	15%	Type definitions, type safety, no "any" abuse
Code Quality	15%	Readability, organization, error handling

Deliverables

GitHub Repository

Working code that we can clone and run with npm install && npm run dev.

DECISIONS.md

- Why you chose your drag-and-drop approach

README.md

- Setup instructions
- Project structure overview
- Libraries used and why
- How you structured state management
- UX decisions you made and alternatives considered
- What you would improve with more time

Technical Stack

Next.js 14 • React 18 • TypeScript • Tailwind CSS • Zustand or React Context • Any drag-drop library

Important: We will schedule a 30-minute technical call after reviewing your submission. During this call, we will ask you to walk us through your component architecture, explain your state management decisions, and make a small live modification (e.g., add a new filter or card field).