2e Systems

Front End Developer – Technical Assignment

Notes (before you begin)

This assignment is designed to evaluate your technical thinking, problem-solving skills, and your ability to deliver well-structured solutions across backend and/or frontend layers. While the functionality may seem straightforward, the challenge lies in applying best practices, handling edge cases, and demonstrating code clarity and architectural awareness.

Time Estimate: Approximately 8–12 hours.

This is not a timed challenge – we encourage you to work at your own pace and focus on delivering a thoughtful and maintainable solution. Clean, understandable code and pragmatic decisions are valued more than perfection or overengineering.

If you run out of time or aren't able to finish every feature, that's okay. We encourage you to submit your work still, especially if it demonstrates thoughtful design, clean implementation, or highlights specific areas of your strength. Partial submissions are perfectly acceptable, and we'd rather see what you've done well than receive nothing at all.

Assignment: TravelHub – Travel Management System

Your task is to build a small application that allows users to manage airports, airlines, and routes between airports, all tied to countries.

Required Functionality

Entities

- Airport
  - Name
  - IATA or ICAO code
  - Country (selected from a static list)
  - GPS Location (via map click)

- Airline
  - Name
  - Base Country
  - Serviced Airports (one or more)

- Route
  - From Airport → To Airport
  - Operated by an Airline

Minimum Features

- Add / Edit / Delete airports, airlines, and routes
- View details for each entity
- Manage relationships (e.g. assign airports to airlines, link routes)
- Store and retrieve data (via API or embedded logic)
- User Interface to view and manage all entities

Technology Guidelines

You are free to choose the technology stack that best reflects your strengths and experience. The goal is to build a working application, and we're more interested in how you approach problems than which framework you use.

Backend API

Build a working RESTful API to support the required functionality.

While we primarily use PHP with Laravel, we encourage you to use the backend framework with which you're most comfortable (e.g., Symfony, Node.js with Express, Next.js API routes, etc.).

Focus on:

- RESTful routing and structure
- Validation and data modeling
- Error handling and response design

Frontend UI

Build a usable web interface that communicates with the backend API (or uses mocked data, if needed).

While our preferred choice is React, you are free to use any frontend approach you're familiar with, including Vue, Laravel Blade, Inertia.js, or full-stack frameworks like Next.js.

Focus on:

- UI structure and navigation
- Data display and CRUD actions
- UX considerations

Optional Enhancements (Bonus Points)

These are not required, but can showcase your experience and initiative.

User Experience & UI

- Interactive map for selecting or displaying airports (e.g. Leaflet, Google Maps)
- Search or filtering by country or airline
- Pagination or infinite scroll
- Responsive layout

Logic & Backend

- Route distance calculation (e.g. Haversine formula)
- Basic authentication (email + password)
- Extended data relationships

Developer Quality

- API documentation (e.g. Swagger, Postman collection)
- Unit tests (backend or frontend)
- State management (Redux, Zustand, etc.)

Submission Instructions

You may submit your solution in one of the following ways:

- Share a Git repository (GitHub, GitLab, Bitbucket)
- Send a ZIP archive via email

Please include a `README.md` file with:

- Setup & run instructions
- Stack used and reasoning (brief)
- Time taken and any limitations

Evaluation Criteria

Area	What We're Looking For
Code Quality	Clarity, consistency, naming, formatting
App Structure	Separation of concerns, organization
Data Handling	Validation, routing, response structure
UI & UX Basics	Usability, layout, responsiveness
Data Modeling & Logic	Entity modeling, linking, and correctness
Optional Features (Bonus)	Search, maps, tests, documentation, auth, etc.