

Transport Management System Assignment

Project Overview

Build a full-stack Transport Management System with:

- **Backend:** Django REST Framework
- **Frontend:** Vue.js (with TypeScript & Composition API)

The focus is on **clean architecture**, **proper API design**, and **basic relational data handling**. The system should manage transport orders and associated waypoints.

Core Models

Orders: Order number (unique), customer name, date

Waypoints: Location address, type (Pickup/Delivery), linked to an order

All models require appropriate relationships and constraints.

Primary Features

Transport Order Creation

Users can create a new transport order with:

- Order Number
- Customer Name
- Date

Each order can have **multiple waypoints**. Each waypoint must include:

- Location address
- Type: Pickup or Delivery

Transport Order Listing

- List all transport orders with:
 - Order Number
 - Customer Name
 - Date
 - Number of waypoints
- (Optional) Filter orders by:
 - Date
 - Customer Name

Technical Implementation

Backend (Django):

- Use **Django REST Framework** for API endpoints.
- Create models for Orders and Waypoints with proper relationships.
- Implement the following endpoints:
 - `POST /api/orders/` → Create a transport order with waypoints
 - `GET /api/orders/` → List all orders (with optional filters)
 - `GET /api/orders/{id}/` → Retrieve details of an order with its waypoints

Example request:

```
{
  "order_number": "ORD123",
  "customer_name": "John Doe",
  "date": "2025-08-21",
  "waypoints": [
    {"location": "New York", "type": "Pickup"},
    {"location": "Boston", "type": "Delivery"}
  ]
}
```

Frontend (Vue.js + TypeScript + Composition API):

- Two simple pages:
 1. **Create Transport Order:** Form with dynamic waypoint addition
 2. **Transport Order List:** Table to display all orders (with optional filtering)
- Use **Axios** or **Fetch API** for API calls.
- Keep UI minimalistic (focus on functionality, not styling).

Deliverables

1. GitHub Repository

- Django backend
- Vue.js frontend
- README including:
 - Setup instructions
 - Assumptions and decisions
 - How to run the app

2. Sample Data

(Optional) Provide 2-3 sample transport orders with waypoints for testing.

3. API Documentation

- Auto-generated OpenAPI schema OR simple markdown documentation

3. API Documentation

- Auto-generated OpenAPI schema OR simple markdown documentation

4. Deployment

- Deploy your application using the provided link and ensure both the backend and frontend are accessible

Evaluation Criteria

- Correctness & functionality (meets requirements)
- API design & model relationships
- Code quality and structure (clean, modular, readable)
- Proper usage of Vue Composition API and TypeScript
- Basic error handling

Optional Bonus Features

- Update/Delete transport orders
- Pagination for order listing
- Client-side form validation
- Docker setup for easy deployment