Delivery Tracker (Full-Stack MVP)

Overview

A full-stack **Delivery Tracker MVP** built with **FastAPI**, **SQLite**, **React**, **Tailwind CSS**, and **React Leaflet**. It tracks deliveries on a map, displays status in a sidebar, and updates dynamically to simulate a live dashboard.

Features

- **Interactive Map**: Displays markers for each delivery with color-coded status.
- **Delivery List Sidebar**: Shows all deliveries with details (customer name, address, status, ETA).
- **Filtering**: Filter deliveries by status (All, Preparing, In-Transit, Delivered).
- **Detail Popup**: Click a marker or list item to view delivery details.
- **Live Updates**: Fetches latest data from backend every 10 seconds.

How It Works

Backend (FastAPI + SQLite)

- Stores delivery data (customer info, coordinates, status, ETA).
- Exposes endpoints:
- `/deliveries` → list deliveries (optionally filtered by status)
- `/deliveries/{id}` → get details of a specific delivery
- Backend serves JSON data to the frontend.

Frontend (React + Tailwind + Leaflet)

- Fetches delivery data from backend.
- Renders interactive map markers using Leaflet.
- Sidebar lists all deliveries and highlights markers on click.
- Filtering dropdown updates the list and map dynamically.
- Polling every 10 seconds simulates live updates.

Setup & Implementation

1️⃣Backend

```
cd backend
python -m venv venv          # Create virtual environment
```

```
# Windows
venv\Scripts\activate
# Mac/Linux
# source venv/bin/activate
pip install -r requirements.txt
uvicorn main:app --reload
```

- Runs at: `http://127.0.0.1:8000`
- API docs: `http://127.0.0.1:8000/docs`

2️⃣Frontend

```
cd frontend
npm install
npm start
```

- Runs at: `http://localhost:3000`
- Ensure backend is running simultaneously.

Project Structure

```
delivery_tracker/
├─ backend/        # FastAPI + SQLite
│   ├─ main.py
│   ├─ database.db
│   └─ requirements.txt
├─ frontend/       # React + Tailwind + Leaflet
│   ├─ public/
│   │   └─ index.html
│   ├─ src/
│   │   ├─ App.js
│   │   ├─ MapView.js
│   │   ├─ DeliveryList.js
│   │   └─ index.js
│   └─ package.json
└─ README.md
```

Notes

- Map markers: **Yellow = Preparing**, **Blue = In-Transit**, **Green = Delivered**
- Can be enhanced later with **Firestore** for real-time updates.

- Designed to be **interactive, responsive, and recruiter-friendly**.