**🧭 Final Comment – TDA-13: Core Map View with Leaflet**

**Summary**  
We successfully implemented the **Core Map View** for the Turkish Diaspora App as part of Epic 4 (Frontend Map & List Application).  
This story delivers the main visual interface of the application — an interactive, high-performance Leaflet map connected to a fully functional backend endpoint that serves verified Turkish-oriented business locations.

**✅ Scope & Deliverables**

1. **Backend integration**
   * Created a new async router at  
     Backend/api/routers/locations.py with prefix /api/v1/locations.
   * Implemented an **async SQLAlchemy query** that retrieves all VERIFIED locations from the database, supporting optional filters for category and geographic radius (via haversine distance).
   * Added a /ping endpoint for quick health verification (/api/v1/locations/ping).
   * Updated Backend/app/main.py to include this router and aligned it with existing dev-routers (dev\_ai, dev\_classify, google\_dev).
   * The router now uses the project’s **AsyncEngine** from app/db.py with an internally defined async\_sessionmaker, ensuring full compatibility with the async DB layer.
   * CORS configuration validated to allow communication from the Vite dev server.
2. **Frontend implementation**
   * Built the **MapView** component (Frontend/src/components/MapView.tsx) using **React + Leaflet**.
   * Added supporting utilities:
     + useUserPosition.ts → detects user location with graceful fallback to NL center.
     + lib/api.ts → implements getLocations() to call /api/v1/locations.
     + types/location.ts → defines the Location type.
     + lib/geo.ts → provides geographic helpers and constants.
   * Implemented **smooth map performance** with preferCanvas, cached icons, and efficient marker rendering (tested with > 200 markers).
   * Verified that markers display correctly, each popup shows the location name + category, and the component remains responsive on mobile.

**🔍 Testing & Validation**

* /api/v1/locations/ping returns { "ok": true, "router": "locations" } → router loaded successfully.
* /api/v1/locations returns JSON list of verified records.
* MapView fetches data, displays OpenStreetMap tiles, renders markers, and opens popups correctly.
* Performance validated up to 200+ markers with smooth panning & zooming.
* CORS verified between frontend (Vite) and backend (Uvicorn).

**🧩 Definition of Done**

| **Requirement** | **Status** | **Notes** |
| --- | --- | --- |
| Map loads with OpenStreetMap tiles | ✅ | Rendered via React-Leaflet |
| Markers displayed for verified locations | ✅ | Connected to async endpoint |
| Popups show name + category | ✅ | Implemented and tested |
| Smooth performance (≥ 200 markers) | ✅ | Canvas renderer + caching |
| Endpoint available under /api/v1/locations | ✅ | Verified via Swagger & frontend |

**🚀 Outcome**

TDA-13 establishes the full end-to-end pipeline between the database and the interactive frontend map.  
The map component now provides a live visual overview of verified businesses and serves as the foundation for **TDA-14 (Location List & Filtering Components)**, where filtering, sorting, and list-map synchronization will be added.

✅ **TDA-13 = Done**