**📌 Project Name: Contact Management & Visualization Dashboard**

**🛠️ Tech Stack:**

* **Frontend:** React.js, Tailwind CSS, @react-google-maps/api
* **Backend:** Node.js/Express
* **Mapping:** Google Maps JavaScript API
* **State Management:** React Hooks
* **Data Handling:** Server-side filtering with HubSpot API integration
* **Security**: Helmet.js, CORS, and custom security headers

**🚀 Overview:**

This project is a robust, interactive, and map-integrated contact management system built to allow users to filter, visualize, and interact with contacts based on geographic and role-based attributes. With full HubSpot CRM API integration, it ensures a scalable, real-time experience optimized for responsiveness and performance.

All setup and operations — including contact creation, role assignment, and property configuration — were carried out and verified using HubSpot’s sandbox via Postman, the HubSpot UI, and mobaXterm CLI.

**📍 Key Features:**

**1. Dynamic Contact Filtering**

* Filter contacts by Country, State, City, and Role.
* Real-time updates via controlled inputs.
* Dynamically fetches location options (e.g., states/cities) based on prior selection.
* All data fetched and filtered server-side for scalability.

**2. Google Map Integration**

* Integrated using @react-google-maps/api.
* Displays contacts using distinct role-based icons.
* If a contact has multiple roles, multiple icons are shown at the same location.
* Includes a custom Legend:
  + ⭐ Contractor
  + 🔺 Geo Tech
  + 🏠 Home Owner
  + 🤝 Referral Partner
  + 📣 Affiliate
  + 🌐 Community Partner
* Map dynamically adjusts based on applied filters.

**3. Responsive Design**

* Tailwind CSS ensures mobile-first responsiveness.
* On small screens, region and role filters are placed side-by-side.
* On larger screens, filters revert to a stacked layout for clarity.

**4. Modular and Reusable Codebase**

* Components are modular and reusable:
  + FilterDropdown for dropdown selections.
  + ContactList for rendering filtered contacts.
  + SectionHeader for consistent layout.
* Centralized logic using a custom hook useContactFilters.
* Clean and scalable folder structure for easy maintainability.

**5. HubSpot CRM Integration**

* Fully integrated with a HubSpot test/sandbox account.
* Verified and tested using HubSpot’s CLI (mobaXterm), Postman, and HubSpot UI.
* created a custom property project\_role.
* Populated sample data with contacts having addresses and 1–2 roles each.
* Supported roles:
  + Contractor
  + Home Owner
  + Affiliate
  + Referral Partner
  + Community Partner
  + Geo Tech

**6. Robust Error Handling**

* Handles geocoding issues with user-friendly messages.
* Displays fallback content for API or fetch errors.

**7. Security Enhancements**

* **CORS Integration:** Enabled CORS to control resource access policies across origins.
* **Helmet Middleware:** Applied Helmet.js to automatically set secure HTTP headers.
* **Custom Security Headers:**
  + Disabled x-powered-by to hide Express framework info.
  + Applied Referrer-Policy to control referer data sharing.
  + Enforced X-Content-Type-Options to prevent MIME-sniffing.
  + Enabled X-XSS-Protection for basic XSS attack mitigation.
  + Set Strict-Transport-Security to enforce HTTPS across subdomains.

✅ Deliverables

* Source Code <https://github.com/RyomenDev/GeoContacts>
* Deployed Web Application <https://geo-contacts-main.vercel.app/>

Author: Akash Mishra

Date: 23 April 2025