

Mapa Obiektów – Dokumentacja README

Cel aplikacji: Interaktywna mapa hal przemysłowych, która łączy dane geograficzne z bazą Supabase, umożliwiając filtrowanie, dodawanie/edycję wpisów, eksport CSV oraz szybkie analizy wizualne (Chart.js). Całość to jeden plik HTML (z wbudowanym CSS i JS) + zasoby CDN, więc może działać na dowolnym serwerze statycznym lub lokalnie z pliku.


Spis treści

- 1. Schemat danych w Supabase
- 2. Zewnętrzne biblioteki
- 3. Przepływ aplikacji
- 4. Zmienne globalne
- 5. Opis kluczowych funkcji
- 6. Warstwa wizualna (CSS)

Schemat danych w Supabase

Kolumna	Typ	Opis
id	int (PK, auto)	Unikalny identyfikator wpisu.
investor	text	Nazwa inwestycji / firmy. <i>Wymagana.</i>
city	text	Miasto (pokazywane w popupie). <i>Wymagane.</i>
region	text	Województwo (używane w filtrach). <i>Wymagane.</i>
area	numeric	Powierzchnia w m ² (opcjonalna, używana w statystykach).
industry	text	Branża (opcjonalnie).
object_type	text	Rodzaj obiektu (np. „magazyn”, „produkcja”).
contract_type	text	Typ kontraktu (np. „zaprojektuj i wybuduj”).
link_url	text	Publiczny link do materiałów o inwestycji.

lat	numeric	Szerokość geo (generowana automatycznie po geokodowaniu).
lon	numeric	Długość geo.

Tip: jeśli w Supabase ustawisz regułę RLS „Anon select”, aplikacja będzie tylko odczytywać dane , ale dodawanie/edycja/usuwanie wymaga ustawienia odpowiednich polityk bezpieczeństwa.

Zewnętrzne biblioteki

Biblioteka	Do czego służy?
Supabase JS v2	Komunikacja z bazą danych i storage.
Leaflet	Podstawa interaktywnej mapy.
Leaflet.markercluster	Grupowanie wielu markerów w klastry gwiazd (lepsza wydajność).
Chart.js	Wykresy kołowe i słupkowe w pop-upie „Analiza”.
Font Awesome	Ikony UI (przyciski, karty, popupy).

Ładowanie `<script>` i `<link>` z CDN gwarantuje brak zależności build-time.

Przepływ aplikacji

flowchart TD

```

A(Start strony) --> B(fetchObjects)
B --> C[allObjects wypełnione]
C --> D[populateFilters]
C --> E[addOrUpdateMarker dla każdego]
E --> F[updateSummary]
```

subgraph UI

```

G{Użytkownik}
G --> |zmiana filtra| H(filterMarkers)
G --> |Dodaj/Edytuj| I(add-popup)
G --> |Usuń| J(delete-popup)
G --> |Eksport CSV| K(exportCSV)
G --> |Analiza| L(open-analysis)
```

end

H --> E
I -->|po zapisie| B
J -->|po usunięciu| B

1. **Pobranie danych (fetchObjects)** – SELECT * z tabeli objects.
2. **Render mapy** – tworzenie markerów + klastra.
3. **Filtry** – onchange → filterMarkers filtruje tablicę i re-renderuje markery + karty podsumowania.
4. **Add/Edit** – popup zbiera dane, opcjonalnie geokoduje (Nominatim) i INSERT/UPDATE do Supabase.
5. **Delete** – potwierdzenie w popupie → DELETE.
6. **Eksport CSV** – generowanie pliku in-browser z aktualnym widokiem (uwzględnia filtry).
7. **Analiza** – otwarcie popupu, inicjalizacja kanw i updateAnalysisCharts().

Zmienne globalne

Zmienna	Typ	Znaczenie
supabaseUrl	string	URL instancji Supabase.
supabaseKey	string	Klucz publiczny „anon” (ukryj w .env przy deployu!).
supabaseClient	object	Instancja klienta.
allObjects	array	Aktualna lista obiektów (z bazy).
editingObject	object null	Obiekt w trybie edycji.
deletingObject	object null	Obiekt oczekujący na usunięcie.
map	Leaflet Map	Referencja do mapy.
markerClusterGroup	MC Group	Warstwa z klastrami markerów.
chartAreaRange, chartTopRegions, chartTopIndustries, chartContractTypes	Chart.js instances	Wykresy w panelu analizy.

Opis kluczowych funkcji

Funkcja	Co robi?	Parametry istotne
fetchObjects()	SELECT * z Supabase, zapisuje w <code>allObjects</code> , renderuje markery + filtry.	–
getUniqueValues(arr, key)	Zwraca unikalne, posortowane wartości dla kolumny (pomaga w budowie filtrów).	<code>arr</code> , <code>key</code>
populateFilters(data)	Wypełnia <code><select></code> danymi z bazy.	<code>data</code> – tablica obiektów
filterMarkers()	Czyści klastry, filtruje <code>allObjects</code> wg wybranych kryteriów, renderuje nowe markery i statystyki.	–
addOrUpdateMarker(obj)	Tworzy marker Leaflet → popup z danymi i przyciskami „Edytuj/Usuń”.	<code>obj</code> – rekord z bazy
generatePopupContent(obj)	HTML popupu, sanitizuje URL i tworzy przyciski akcji.	<code>obj</code>
updateSummary(arr)	Liczy liczbę obiektów + sumę powierzchni, formatuje z separatorami tysięcy.	<code>arr</code>
clearAddPopupFields()	Resetuje inputy w formularzu add/edit.	–
updateAnalysisCharts()	Liczy dane stat. i aktualizuje/rysuje 4 wykresy Chart.js.	–
exportCSV() (inline)	Generuje CSV w przeglądarce dla przefiltrowanego datasetu.	–

Warstwa wizualna (CSS)

- **Design tokens** – kolory i cienie w `:root` dla łatwej zmiany motywu.
- **Layout** – Flex + CSS Grid (karty analizy) + responsywne media queries ≤ 768 px.
- **Komponenty**: `.controls-panel`, `.summary-cards`, `.card`, `.map-container`, 3 rodzaje popupów.
- **Interakcje** – delikatny transform: `translateY(-2px)` na przyciskach/kartach, `box-shadow`.