

SmogWatch

System Monitorowania Jakości Powietrza

2025-06-08

1. Skład zespołu oraz role w projekcie

Projekt SmogWatch został zrealizowany przez zespół 2-osobowy:

Członek zespołu	Rola w projekcie	Główne odpowiedzialności
Łukasz Domagała 131971	Backend	Backend Flask, API integration
Igor Barcik 131780	Frontend	Vue.js, UI/UX design

2. Opis tematyki projektu

SmogWatch to system monitorowania jakości powietrza w Polsce, który wykorzystuje dane rzeczywiste z oficjalnego API Głównego Inspektoratu Ochrony Środowiska (GIOŚ). Projekt koncentruje się na analizie i wizualizacji danych dotyczących jakości powietrza w czasie rzeczywistym.

Tematyka projektu obejmuje:

- Monitoring jakości powietrza w polskich miastach
- Analiza poziomów zanieczyszczeń (PM10, PM2.5, NO2, CO, O3, SO2)
- Indeks jakości powietrza (AQI) z kategoryzacją zagrożenia
- Udostępnienie danych przez nowoczesny interfejs webowy

3. Cel realizacji projektu

Głównym celem projektu SmogWatch jest stworzenie łatwo dostępnego narzędzia do monitorowania jakości powietrza w Polsce, które:

- **Agreguje dane rządowe** - Wykorzystuje oficjalne dane z GIOŚ
- **Prezentuje informacje w przystępny sposób** - Poprzez nowoczesny interfejs webowy
- **Umożliwia szybki dostęp do informacji** - O stacjach pomiarowych i aktualnych pomiarach
- **Wspiera świadomość ekologiczną** - Poprzez wizualizację problemów związanych z jakością powietrza

4. Opis realizacji i sposób osiągnięcia celu projektowego

Projekt został zrealizowany w architekturze klient-serwer z następującymi komponentami:

Backend (Flask)

- **API Gateway** - Pośredniczenie między frontendem a API GIOŚ
- **Endpointy RESTful** - Standaryzowane interfejsy komunikacji
- **Error handling** - Obsługa błędów i logowanie
- **CORS** - Umożliwienie komunikacji cross-origin

Frontend (Vue.js)

- **Single Page Application** - Nowoczesny interfejs użytkownika
- **Logowanie i autoryzacja** - OIDC do uwierzytelniania użytkowników (Authentik + postgres)
- **Responsywny design** - Dostosowanie do różnych urządzeń
- **Komponenty wielokrotnego użytku** - StationsList, StationDetails
- **Asynchroniczna komunikacja** - Z backendem poprzez Axios

Integracja danych

- **API GIOŚ** - <https://api.gios.gov.pl/pjp-api/rest>
- **Real-time data** - Aktualne pomiary jakości powietrza
- **Geolokalizacja stacji** - Współrzędne geograficzne stacji pomiarowych

5. Opis zadań projektowych, stos technologiczny i sposób realizacji

Stos technologiczny

Backend

```
Flask=3.0.0           # Mikro-framework webowy
Flask-CORS=4.0.0      # Cross-Origin Resource Sharing
requests=2.31.0       # HTTP client do komunikacji z API
```

Frontend

```
vue: "^3.3.4"         # Progressive JavaScript framework
vue-router: "^4.2.4"  # Routing dla SPA
axios: "^1.5.0"        # HTTP client
oidc-client-ts: "^3.0.1" # OpenID Connect client
```

Realizacja zadań

Część 1: Analiza danych

1. **Pozyskanie danych** - Integracja z API GIOŚ
2. **Przetwarzanie danych** - Normalizacja i walidacja
3. **Analiza jakości powietrza** - Kalkulacja indeksów AQI

Część 2: Aplikacja webowa

1. **System routingu** - Vue Router dla nawigacji SPA
2. **Komponenty UI** - Lista stacji i szczegóły pomiarów
3. **Wizualizacja danych** - Kolorowe oznaczenia poziomów zagrożenia
4. **Responsywność** - Dostosowanie do urządzeń mobilnych

Architektura systemu

Frontend (Vue.js)	→	Backend (Flask)	→	API GIOŚ
↓		↓		↓
Komponenty		REST Endpoints		Dane rządowe
↓		↓		↓
Interfejs		Przetwarzanie		Stacje i pomiary

6. Wnioski i możliwości dalszego rozwoju

Wnioski z realizacji projektu

1. **Skuteczność** - API GIOŚ dostarcza stabilne i aktualne dane
2. **Architektura** - Podział frontend/backend umożliwia łatwe skalowanie
3. **Użyteczność** - Prosty interfejs zwiększa dostępność informacji o smogu

Możliwości dalszego rozwoju

Funkcjonalności

- **Powiadomienia** - Alerty o przekroczeniu norm jakości powietrza
- **Analiza historyczna** - Wykresy trendów w czasie
- **Prognozy** - Przewidywanie jakości powietrza za pomocą ML
- **Mapy interaktywne** - Wizualizacja geograficzna stacji

Techniczne ulepszenia

- **Baza danych** - Cache'owanie danych dla lepszej wydajności
- **Autentykacja** - System logowania użytkowników
- **PWA** - Progressive Web App dla urządzeń mobilnych
- **Testy automatyczne** - Unit i integration testy

Integracje

- **Dodatkowe źródła danych** - Europejskie agencje środowiskowe
- **API społecznościowe** - Udostępnianie danych innym aplikacjom
- **Eksport danych** - CSV, JSON, PDF reports

7. Bibliografia

1. Główny Inspektorat Ochrony Środowiska. (2025). *API systemu monitoringu jakości powietrza*. Pobrano z: <https://api.gios.gov.pl/pjp-api/rest>
2. Vue.js Team. (2025). *Vue.js Documentation*. Pobrano z: <https://vuejs.org/>
3. Pallets Team. (2025). *Flask Documentation*. Pobrano z: <https://flask.palletsprojects.com/>
4. Główny Urząd Statystyczny. (2024). *Ochrona środowiska 2024*. Warszawa: GUS.
5. Europejska Agencja Środowiska. (2024). *Air quality in Europe — 2024 report*. Kopenhaga: EEA.
6. World Health Organization. (2021). *WHO global air quality guidelines*. Geneva: WHO Press.