

# ZAŁOŻENIA DO PROJEKTU ZALICZENIOWEGO

## CEL PROJEKTU

Celem projektu jest stworzenie aplikacji monitorującej jakość powietrza w Polsce.

Podstawowe funkcjonalności aplikacji:

1. możliwość pobrania danych pomiarowych z Internetu
2. możliwość zapisu danych do lokalnej, relacyjnej bazy danych
3. możliwość pobrania danych „historycznych” z bazy
4. możliwość prezentacji danych w formie wykresu
5. możliwość prostej analizy danych

## DANE POMIAROWE

Aplikacja powinna wykorzystywać dane publikowane bezpłatnie przez Główny Inspektorat Ochrony Środowiska. Dane są dostępne poprzez dostępną w Internecie, bezpłatną usługę REST. Usługa zwraca dane w formacie JSON. W zależności od skierowanego żądania (typu GET) można uzyskać następujące informacje:

- **stacje pomiarowe** w Polsce, a dokładniej:
  - *id* – unikalny identyfikator stacji
  - *stationName* – nazwa stacji pomiarowej
  - *gegrLat* – szerokość geograficzna położenia stacji
  - *gegrLon* – długość geograficzna położenia stacji
  - *city* – informacje o adresie stacji, na które składają się:
    - *id* – unikalny identyfikator lokalizacji
    - *name* – nazwa miejscowości, w której znajduje się stacja
    - *commune* – informacje o gminie na które składają się:
      - *communeName* – nazwa gminy
      - *districtName* – nazwa powiatu
      - *provinceName* – nazwa województwa
    - *addressStreet* – nazwa ulicy na której znajduje się stacja

- **stanowiska pomiarowe**, czyli czujniki w danej stacji pomiarowej:
  - *id* – unikalny identyfikator stanowiska pomiarowego
  - *stationId* – identyfikator stacji pomiarowej w której znajduje się dane stanowisko pomiarowe
  - *param* – informacje o tym, co jest mierzone:
    - *paramName* – nazwa mierzonego parametru
    - *paramFormula* – symbol mierzonego parametru
    - *paramCode* – kod parametru
    - *idParam* – identyfikator mierzonego parametru
- **dane pomiarowe**:
  - *key* – kod mierzonego parametru
  - *values* - sekwencja par:
    - *date* – data i czas pomiaru
    - *value* – wartość mierzonego parametru
- **indeks jakości powietrza**:
  - *id* – identyfikator stacji pomiarowej
  - *stCalcDate* – data i czas obliczenia indeksu
  - *stIndexLevel* – najgorszy indeks dla danej stacji:
    - *id* – poziom indeksu (w skali od 0 do 5)
    - *indexLevelName* – tekstowy opis poziomu indeksu
  - *stSourceDataDate* – data i czas zebrania danych na podstawie których liczony był indeks
  - dalej podobne parametry dla każdego stanowiska pomiarowego
  - oraz wartości krytyczne – szczegóły w opisie API

Sposób wywołania usługi REST oraz przykładowe dane są dostępne na stronie:

<https://powietrze.gios.gov.pl/pjp/content/api>

Należy pamiętać, że wszystkie dane pobrane z usługi są tekstowe (nawet jeśli reprezentują wartości liczbowe).

Dane pomiarowe są zbierane standardowo co godzinę. Mogą wystąpić sytuacje, gdy o zadanej porze pomiar nie zostanie dokonany – wówczas zwracana jest wartość *null*.

## DZIAŁANIE PROGRAMU

Program powinien być odporny na sytuacje, gdy nie mamy łączności lub usługa jest niedostępna (należy dodać obsługę wyjątków). W takiej sytuacji użytkownik powinien zostać poinformowany o niedostępności danych i ewentualnie zaproponować skorzystanie z danych „historycznych” (jeśli takie zostały wcześniej zapisane w bazie danych).

Aby aplikacja mogła zaprezentować dane należy najpierw wybrać stację pomiarową.

---

Tu można użytkownikowi podać:

- pełną listę stacji w całej Polsce (odczyt z usługi REST)
- listę stacji w podanej przez użytkownika miejscowości (odczyt z usługi REST i odfiltrowanie stacji po nazwie miejscowości)
- wyszukać stacje w najbliższym, zadanym promieniu (w km) od podanej - opisanej słownie - lokalizacji (np. *Collegium da Vinci*) – można do tego wykorzystać bibliotekę *geopy* (patrz: <https://geopy.readthedocs.io/en/stable/>) i po odczycie wszystkich stacji z usługi odfiltrować te, które są za daleko
- wyświetlić mapę z naniesionymi stacjami i opcjonalnie – po wybraniu rodzaju pomiaru – wyświetlić na mapie dane kodowane skalą kolorystyczną.

Po wskazaniu przez użytkownika stacji pomiarowej (numeru *id*) należy zaprezentować, co stacja potrafi mierzyć (listę stanowisk pomiarowych). Te informacje uzyskamy po skierowaniu zapytania do usługi REST z podanym identyfikatorem stacji.

Po wyborze przez użytkownika stanowiska pomiarowego (numeru *id*) należy odczytać dane pomiarowe – kolejne zapytanie do usługi REST z podanym identyfikatorem stanowiska pomiarowego.

Aplikacja powinna umożliwić zapisanie tych danych w tabeli relacyjnej bazy danych.

Dane pomiarowe należy przedstawić w formie wykresu. Użytkownik powinien mieć możliwość wyboru z jakiego okresu dane mają się pojawić na wykresie.

Dodatkowo aplikacja może dokonać prostej analizy danych, np.:

- obliczyć wartości najmniejsze i największe
- wskazać, kiedy te wartości miały miejsce (data, godzina)
- obliczyć wartość średnią
- obliczyć trend (czy dane mają tendencję do wzrostu, czy maleją)

Aplikacja powinna być wyposażona w interfejs graficzny.

Autor aplikacji ma wolną rękę w kwestii wyboru modelu danych (czy bazujemy na klasach wbudowanych Pythona, modelu obiektowym, czy może obiektach biblioteki *pandas*).

Moduły biblioteki należy pogrupować logicznie w pakiety zgodnie z ich funkcjonalnością.

Projekt powinien być wyposażony w dokumentację (docstring'i).

Należy zastanowić się, czy w aplikacji mogą być przydatne jakieś wzorce projektowe. Jeśli tak, to jakie i gdzie.

Aplikację należy wyposażyć w testy jednostkowe.

---

## WYMAGANIA TECHNICZNE DOTYCZĄCE PROJEKTU

1. Projekt umieszczony w serwisie *GitHub*.
2. Dołączona instrukcja uruchomienia i uruchomienia testów.
3. Baza danych – *SQLite*.
4. Użycie stringów dokumentujących we wszystkich modułach, funkcjach i klasach.
5. Program ma się dać uruchomić w systemie *MS Windows*.
6. Aplikacja może komunikować się z użytkownikiem po polsku lub po angielsku (do wyboru)
7. Do projektu należy dołączyć pliki *README.md* oraz *requirements.txt*