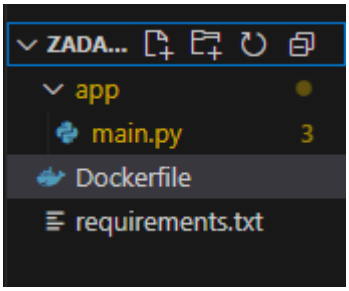
Link do repozytorium GitHub - https://github.com/kkkasia/pawcho_zad1

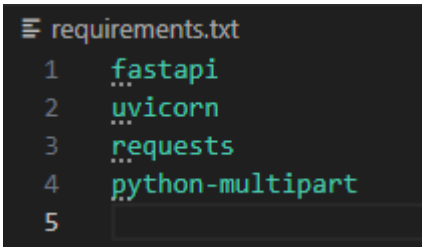
1. Aplikacja pogodowa
   Na zrzutach przedstawiono kod główny aplikacji wyświetlającej pogodę w wybranym mieście – wykonanej w języku Python, a także strukturę projektu wraz z plikiem requierements.txt służącym do definiowania listy bibliotek, niezbędnych do działania aplikacji. Do pobierania pogody użyte przeze mnie zostało darmowe API OpenWeatherMap.

app > main.py > ...

```python
#KLUCZ API - 31c59dd39bfbd8e0db669ff800577f7e

from fastapi import FastAPI, Form
from fastapi.responses import HTMLResponse
import logging
import datetime
import requests
import os

app = FastAPI()

# Konfiguracja logowania
logging.basicConfig(level=logging.INFO)
logger = logging.getLogger("zadanie1")

# Stałe
AUTHOR_NAME = "Kateryna Zinchuk"
PORT = int(os.getenv("PORT", 8000))
OPENWEATHER_API_KEY = os.getenv("OPENWEATHER_API_KEY", "brak-klucza")

# Lista krajów i miast (rozbudowana)
cities = {
    "Polska": ["Warszawa", "Kraków", "Gdańsk"],
    "Niemcy": ["Berlin", "Monachium", "Hamburg"],
    "Francja": ["Paryż", "Marsylia", "Lyon"],
    "USA": ["Nowy Jork", "Los Angeles", "Chicago"],
    "Japonia": ["Tokio", "Osaka", "Kioto"]
}

# Logowanie przy starcie
@app.on_event("startup")
def startup_event():
    logger.info(f"Data uruchomienia: {datetime.datetime.now()}")
    logger.info(f"Autor: {AUTHOR_NAME}")
    logger.info(f"Nasłuch na porcie: {PORT}")

# Strona główna (formularz wyboru kraju)
@app.get("/", response_class=HTMLResponse)
def read_root():
    form_html = """
    <h2>Wybierz kraj:</h2>
    <form action="/select_city" method="post">
    """
    for country in cities.keys():
        form_html += f'<input type="radio" name="country" value="{country}">{country}<br>'
    form_html += """
        <input type="submit" value="Wybierz kraj">
    </form>
    """
```

```python
    return form_html

# Strona wyboru miasta po wybraniu kraju
@app.post("/select_city", response_class=HTMLResponse)
def select_city(country: str = Form(...)):
    city_options = cities.get(country)
    if not city_options:
        return f"<h3>Nieznany kraj: {country}</h3>"

    form_html = f"""
    <h2>Wybierz miasto w kraju {country}:</h2>
    <form action="/weather" method="post">
        <input type="hidden" name="country" value="{country}">
    """
    for city in city_options:
        form_html += f'<input type="radio" name="city" value="{city}">{city}<br>'
    form_html += """
        <input type="submit" value="Pokaż pogodę">
    </form>
    """
    return form_html

# Pokazanie pogody
@app.post("/weather", response_class=HTMLResponse)
def get_weather(country: str = Form(...), city: str = Form(...)):
    if OPENWEATHER_API_KEY == "brak-klucza":
        return "<h3>Brak klucza API do pobrania pogody!</h3>"

    weather_url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={OPENWEATHER_API_KEY}&units=metric&lang=pl"
    response = requests.get(weather_url)
    if response.status_code != 200:
        return "<h3>Nie udało się pobrać pogody</h3>"

    data = response.json()
    temp = data["main"]["temp"]
    description = data["weather"][0]["description"]

    return f"<h2>Pogoda w {city} ({country}):</h2><p>Temperatura: {temp}°C</p><p>Opis: {description}</p>"
```





2. Plik Dockerfile

Dockerfile > ...

```dockerfile
# Etap 1: Budowanie
FROM python:3.12-slim AS builder

LABEL maintainer="Kateryna Zinchuk"

WORKDIR /app

COPY requirements.txt .
RUN pip install --user -r requirements.txt

COPY app ./app

# Etap 2: Finalny obraz
FROM python:3.12-slim

WORKDIR /app

COPY --from=builder /root/.local /root/.local
COPY app ./app

ENV PATH=/root/.local/bin:$PATH
ENV PORT=8000

EXPOSE 8000

CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8000"]
```

3. Budowa i uruchomienie kontenera

   Budujemy obraz:

```
PS C:\Users\PC\Desktop\zadanie1_docker> docker build -t zadanie1-app .
[+] Building 26.2s (13/13) FINISHED
 => => extracting sha256:e9ddbe7a005f6950194f9ec89ccfff2c86729491901f767d8b03f8062de3daf5
 => [builder 2/5] WORKDIR /app
 => [builder 3/5] COPY requirements.txt .
 => [builder 4/5] RUN pip install --user -r requirements.txt
 => [builder 5/5] COPY app ./app
 => [stage-1 3/4] COPY --from=builder /root/.local /root/.local
 => [stage-1 4/4] COPY app ./app
 => exporting to image
 => => exporting layers
 => => writing image sha256:ff1528f31d75b89e9fe175804005f9771a321f50b32c0a9934eaeb1414d3bb83
 => => naming to docker.io/library/zadanie1-app

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/ioaozvggfpq0l01yp91g8xg20
```

Uruchamiamy kontener poleceniem *docker run -d -p 8000:8000 --name zadanie1-container -e OPENWEATHER_API_KEY=31c59dd39bfbd8e0db669ff800577f7e zadanie1-app* – klucz API ustawiamy jako zmienną środowiskową

```
PS C:\Users\PC\Desktop\zadanie1_docker> docker run -d -p 8000:8000 --name zadanie1-container -e OPENWEATHER_API_KEY=31c59dd39bfbd8e0db669ff800577f7e zadanie1-app
a074817494770a5337e0def70ca1ac61b886540ed4b7000ec8545680ba690d61
```

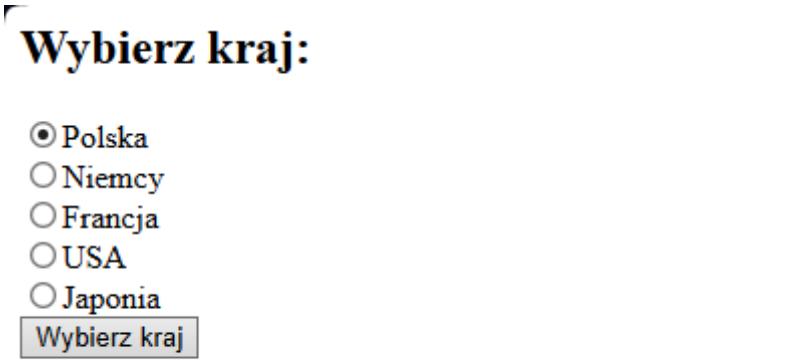
Pobieramy logi z kontenera:

```
PS C:\Users\PC\Desktop\zadanie1_docker> docker logs zadanie1-container
INFO:     Started server process [1]
INFO:     Waiting for application startup.
INFO:zadanie1:Data uruchomienia: 2025-04-27 22:57:54.635883
INFO:zadanie1:Autor: Kateryna Zinchuk
INFO:zadanie1:Nasłuch na porcie: 8000
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
PS C:\Users\PC\Desktop\zadanie1_docker>
```

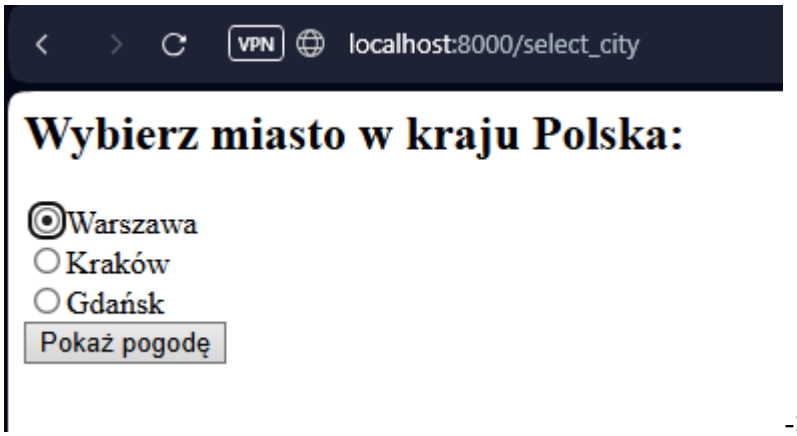
Informacje o liczbie warstw i rozmiarze kontenera:

```
PS C:\Users\PC\Desktop\zadanie1_docker> docker image inspect zadanie1-app --format='Liczba warstw: {{len .RootFS.Layers}}'
>> docker images zadanie1-app
Liczba warstw: 7
REPOSITORY     TAG       IMAGE ID       CREATED         SIZE
zadanie1-app   latest    24922d2c7b79   4 minutes ago   141MB
PS C:\Users\PC\Desktop\zadanie1_docker>
```

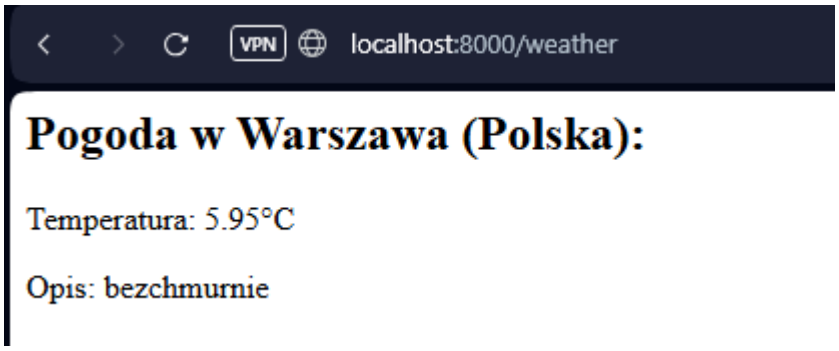
Widok przeglądarki z efektem uruchomienia kontenera:



->



->


VPN
localhost:8000/weather
Pogoda w Warszawa (Polska):
Temperatura: 5.95°C
Opis: bezchmurnie