

## CZĘŚĆ OBOWIĄZKOWA

### 1. (max. 30%)

Najpierw stworzyłam konto na stronie [Current weather and forecast - OpenWeatherMap](#)

Potem stworzyłam plik app.py i index.html

Po uruchomieniu aplikacji w konsoli wyświetlają się data, imię oraz port (5000), a także tworzy się strona internetowa, na której można wybrać miasto z listy i zobaczyć pogodę.

**Plik: app.py:**

```
from flask import Flask, request, render_template
import logging
import datetime
import requests
import os

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

#Inicjalizacja aplikacji Flask
app = Flask(__name__)

#Dane autora i port
AUTHOR = "Yelyzaveta Zlydnieva"
PORT = os.getenv("PORT", 5000)

#Słownik z krajami i miastami
COUNTRIES_CITIES = {
    "Poland": ["Warszawa", "Krakow", "Gdansk", "Wroclaw",
"Poznan"],
    "Germany": ["Berlin", "Munich", "Hamburg", "Cologne",
"Frankfurt"],
    "France": ["Paris", "Lyon", "Marseille", "Toulouse", "Nice"],
    "Italy": ["Rome", "Milan", "Naples", "Turin", "Florence"]
}

#Logowanie informacji przy starcie aplikacji
startup_time = datetime.datetime.now().strftime("%Y-%m-%d
%H:%M:%S")
logger.info(f"Data uruchomienia: {startup_time}")
logger.info(f"Autor: {AUTHOR}")
logger.info(f"Port TCP: {PORT}")

#Strona główna aplikacji
@app.route("/", methods=["GET", "POST"])
def index():
    weather_data = None
    selected_country = None
    cities = []
```

```

#Jeśli użytkownik wysłał formularz
if request.method == "POST":
    selected_country = request.form.get("country")
    city = request.form.get("city")

    #Zaktualizujemy listę miast na podstawie wybranego kraju
    if selected_country in COUNTRIES_CITIES:
        cities = COUNTRIES_CITIES[selected_country]

        #Pobieramy dane pogodowe, jeśli miasto zostało wybrane
        if city and city in cities:
            api_key = "e3752818441fd5ae6c1fb0940dbe8f5d"
            url =
f"http://api.openweathermap.org/data/2.5/weather?q={city},{selected_
d_country}&appid={api_key}&units=metric"
            response = requests.get(url)
            if response.status_code == 200:
                data = response.json()
                weather_data = {
                    "city": city,
                    "country": selected_country,
                    "temperature": data["main"]["temp"],
                    "description":
data["weather"][0]["description"],
                    "humidity": data["main"]["humidity"]
                }

        #Jeśli żaden kraj nie jest wybrany, ustawimy pierwszy z listy
        if not selected_country:
            selected_country = list(COUNTRIES_CITIES.keys())[0]
            cities = COUNTRIES_CITIES[selected_country]

    #Strona HTML
    return render_template("index.html",
countries=COUNTRIES_CITIES.keys(), cities=cities,
selected_country=selected_country, weather_data=weather_data)

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=int(PORT))

```

#### Plik index.html:

```

<!DOCTYPE html>
<html>
<head>
    <title>Pogoda</title>
</head>

```

```

<body>
  <!-- Nagłówek strony -->
  <h1>Sprawdź pogodę</h1>

  <!-- Formularz do wyboru kraju i miasta -->
  <form method="POST">
    <!-- Wybór kraju -->
    <label for="country">Wybierz kraj:</label>
    <select name="country" id="country"
onchange="this.form.submit()">
      {% for country in countries %}
        <option value="{{ country }}" {% if country ==
selected_country %}selected{% endif %}>{{ country }}</option>
      {% endfor %}
    </select>

    <!-- Wybór miasta -->
    <label for="city">Wybierz miasto:</label>
    <select name="city" id="city">
      {% for city in cities %}
        <option value="{{ city }}">{{ city }}</option>
      {% endfor %}
    </select>

    <!-- Przycisk do zatwierdzenia wyboru -->
    <button type="submit">Sprawdź</button>
  </form>

  <!-- Wyświetlenie pogody, jeśli dane są dostępne -->
  {% if weather_data %}
    <h2>Pogoda w {{ weather_data.city }}, {{
weather_data.country }}</h2>
    <p>Temperatura: {{ weather_data.temperature }}°C</p>
    <p>Opis: {{ weather_data.description }}</p>
    <p>Wilgotność: {{ weather_data.humidity }}%</p>
  {% endif %}
</body>
</html>

```

```

PS C:\Users\Who\OneDrive\Рабочий стол\weather_app> docker logs
weather-container

```

```
>>
```

```
INFO:__main__:Data uruchomienia: 2025-04-26 10:28:57
```

```
INFO:__main__:Autor: Yelyzaveta Zlydnieva
```

```
INFO:__main__:Port TCP: 5000
```

## 2. (max. 50%)

Stworzymy plik Dockerfile

```
# Etap 1: budowanie środowiska
FROM python:3.12-slim as builder

# Ustawiamy katalog roboczy
WORKDIR /app

# Kopiujemy wymagania
COPY requirements.txt .

# Instalujemy zależności do katalogu lokalnego
RUN pip install --no-cache-dir --prefix=/install -r requirements.txt

# Etap 2: tworzenie finalnego obrazu
FROM python:3.12-slim

# Autor obrazu
LABEL maintainer="Yelyzaveta Zlydnieva"

# Katalog roboczy
WORKDIR /app

# Kopiujemy aplikację
COPY app.py .
COPY templates/ templates/

# Kopiujemy zainstalowane biblioteki z etapu buildera
COPY --from=builder /install /usr/local

# Ustawiamy zmienną środowiskową
ENV PORT=5000

# Otwieramy port
EXPOSE 5000

# Healthcheck, czy aplikacja działa
HEALTHCHECK --interval=30s --timeout=5s --start-period=5s --retries=3 \
CMD curl --fail http://localhost:5000 || exit 1

# Startujemy aplikację
CMD ["python", "app.py"]
```

### 3. (max. 20%)

Piszemy polecenie do utworzenia obrazu:

PS C:\Users\Who\OneDrive\Рабочий стол\weather\_app> docker build -t weather-app .

```
PS C:\Users\Who\OneDrive\Рабочий стол\weather_app> docker build -t weather-app .
>>
[+] Building 6.9s (13/13) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 950B                             0.0s
=> WARN: FromAsCasing: 'as' and 'FROM' keywords' casing do not match ( 0.0s
=> [internal] load metadata for docker.io/library/python:3.12-slim 1.1s
=> [auth] library/python:pull token for registry-1.docker.io    0.0s
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                    0.0s
=> [builder 1/4] FROM docker.io/library/python:3.12-slim@sha256:858243 0.0s
=> => resolve docker.io/library/python:3.12-slim@sha256:85824326bc4ae2 0.0s
=> [internal] load build context                                0.0s
=> => transferring context: 132B                                  0.0s
=> [builder 3/4] COPY requirements.txt .                        0.0s
=> [stage-1 3/5] COPY app.py .                                  0.0s
=> [builder 4/4] RUN pip install --no-cache-dir --prefix=/install -r r 4.8s
=> [stage-1 4/5] COPY templates/ templates/                   0.0s
=> [stage-1 5/5] COPY --from=builder /install /usr/local      0.1s
=> exporting to image                                           0.5s
=> => exporting layers                                           0.3s
=> => exporting manifest sha256:df259f8a0b7e541566648ec93556862492849e 0.0s
=> => exporting config sha256:56860112e1ac3c2f18086e982bf4d709256a55e9 0.0s
=> => exporting attestation manifest sha256:45d3121de6cfc35f5f4e50baef 0.0s
=> => exporting manifest list sha256:e594e1b64564706945458f98e48bfa6fb 0.0s
```

<input type="checkbox"/>	Name	Tag	Image ID	Created	Size	Actions
<input type="checkbox"/>	weather-app	latest	e594e1b64564	60 seconds ago	197.39 MB	

Polecenie do Uruchomienia kontenera:

PS C:\Users\Who\OneDrive\Рабочий стол\weather\_app> docker run -d -p 5000:5000 --name weather-container weather-app

```
PS C:\Users\Who\OneDrive\Рабочий стол\weather_app> docker run -d -p 5000:5000 --name weather-container weather-app
>> C:\Users\Who\OneDrive\Рабочий стол\weather_app>
e1def6c7dc82e7b32d4f2d75c7af018c5a76bde72fef6798bcc54116b9a0860a
PS C:\Users\Who\OneDrive\Рабочий стол\weather_app>
```

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	Last start...	Actions
<input type="checkbox"/>	weather-container	e1def6c7dc82	weather-app	5000:5000	1 minute ago	

Sprawdzenie logów:

```
PS C:\Users\Who\OneDrive\Рабочий стол\weather_app> docker logs weather-container
>> C:\Users\Who\OneDrive\Рабочий стол\weather_app>
INFO: __main__:Data uruchomienia: 2025-04-26 10:51:29
INFO: __main__:Autor: Yelyzaveta Zlydnieva
INFO: __main__:Port TCP: 5000
* Serving Flask app 'app'
* Debug mode: off
INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
INFO:werkzeug:Press CTRL+C to quit
PS C:\Users\Who\OneDrive\Рабочий стол\weather_app> █
```

Sprawdzamy liczby warstw i rozmiar:

```
PS C:\Users\Who\OneDrive\Рабочий стол\weather_app> docker history weather-app
>>


| IMAGE        | CREATED        | CREATED BY                                      | SIZE   | COMMENT                |
|--------------|----------------|-------------------------------------------------|--------|------------------------|
| e594e1b64564 | 2 minutes ago  | CMD ["python" "app.py"]                         | 0B     | buildkit.dockerfile.v0 |
| <missing>    | 2 minutes ago  | HEALTHCHECK &[["CMD-SHELL" "curl --fail http... | 0B     | buildkit.dockerfile.v0 |
| <missing>    | 2 minutes ago  | EXPOSE map[5000/tcp:{}]                         | 0B     | buildkit.dockerfile.v0 |
| <missing>    | 2 minutes ago  | ENV PORT=5000                                   | 0B     | buildkit.dockerfile.v0 |
| <missing>    | 2 minutes ago  | COPY /install /usr/local # buildkit             | 8.88MB | buildkit.dockerfile.v0 |
| <missing>    | 2 minutes ago  | COPY templates/ templates/ # buildkit           | 16.4kB | buildkit.dockerfile.v0 |
| <missing>    | 2 minutes ago  | COPY app.py . # buildkit                        | 12.3kB | buildkit.dockerfile.v0 |
| <missing>    | 24 minutes ago | WORKDIR /app                                    | 8.19kB | buildkit.dockerfile.v0 |
| <missing>    | 24 minutes ago | LABEL maintainer=Yelyzaveta Zlydnieva           | 0B     | buildkit.dockerfile.v0 |
| <missing>    | 2 weeks ago    | CMD ["python3"]                                 | 0B     | buildkit.dockerfile.v0 |
| <missing>    | 2 weeks ago    | RUN /bin/sh -c set -eux; for src in idle3 p...  | 16.4kB | buildkit.dockerfile.v0 |
| <missing>    | 2 weeks ago    | RUN /bin/sh -c set -eux; savedAptMark="\$(a...  | 44.9MB | buildkit.dockerfile.v0 |
| <missing>    | 2 weeks ago    | ENV PYTHON_SHA256=07ab697474595e06f06647417d... | 0B     | buildkit.dockerfile.v0 |
| <missing>    | 2 weeks ago    | ENV PYTHON_VERSION=3.12.10                      | 0B     | buildkit.dockerfile.v0 |
| <missing>    | 2 weeks ago    | ENV GPG_KEY=7169605F62C751356D054A26A821E680... | 0B     | buildkit.dockerfile.v0 |
| <missing>    | 2 weeks ago    | RUN /bin/sh -c set -eux; apt-get update; a...   | 10.3MB | buildkit.dockerfile.v0 |
| <missing>    | 2 weeks ago    | ENV LANG=C.UTF-8                                | 0B     | buildkit.dockerfile.v0 |
| <missing>    | 2 weeks ago    | ENV PATH=/usr/local/bin:/usr/local/sbin:/usr... | 0B     | buildkit.dockerfile.v0 |
| <missing>    | 2 weeks ago    | # debian.sh --arch 'amd64' out/ 'bookworm' '... | 85.3MB | debuerreotype 0.15     |


PS C:\Users\Who\OneDrive\Рабочий стол\weather_app>
```

Działanie aplikacji:

# Sprawdź pogodę

Wybierz kraj: Polska ▾ Wybierz miasto: Warszawa ▾ Sprawdź

# Pogoda w Lublin, Polska

Temperatura: 12.36°C

Opis: overcast clouds

Wilgotność: 71%

Apr 26, 12:31pm

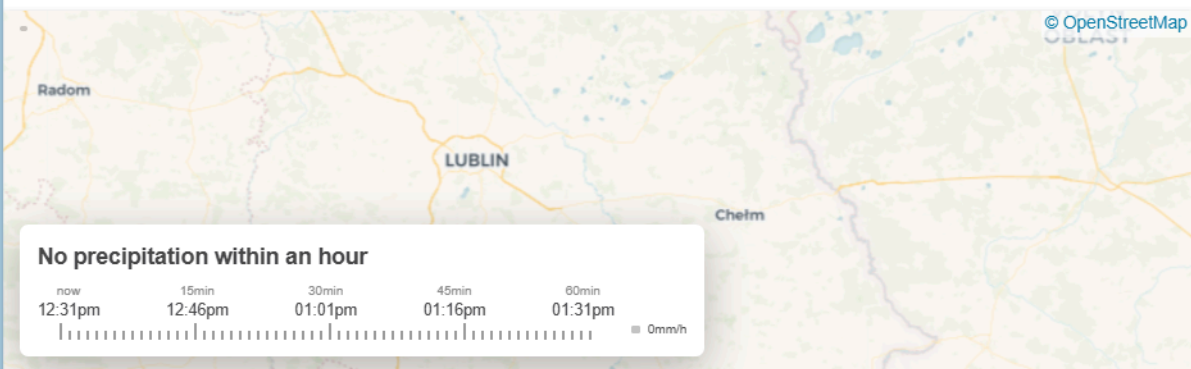
**Województwo Lubelskie, PL**

Feels like 12°C. Overcast clouds. Moderate breeze



12°C

➤ 5.8m/s W    ☉ 1023hPa    Humidity: 71%  
UV: 6    Dew point: 7°C    Visibility: 10.0km



### Hourly forecast

12pm 1pm 2pm 3pm 4pm 5pm 6pm 7pm 8pm 9pm 10pm 11pm Apr 27 1am 2am