

Za pomocą komendy:

- cd pogoda

przechodzimy do folderu pogoda w którym znajdują się pliki z kodami:

- serwer.py

```
1 from flask import Flask, render_template, request
2 import requests, datetime, logging, os
3
4 aplikacja = Flask(__name__, template_folder="szablony", static_folder="style")
5
6
7 logging.basicConfig(
8     level=logging.INFO,
9     format='%(message)s' # tylko wiadomość, bez poziomu logowania, czasu, modułu itd.
10 )
11 logger = logging.getLogger(__name__)
12
13 AUTOR = "Joanna Jurkowska"
14 PORT = int(os.environ.get("PORT", 8080))
15 KLUCZ_API = "2abc04c28f6034d8fd49c8819f1bd440"
16
17 LOKALIZACJE = {
18     "Polska": ["Lublin", "Warszawa", "Wroclaw"],
19     "Wlochy": ["Mediolan", "Wenecja", "Bologna"],
20     "Hiszpania": ["Madryt", "Barcelona"]
21 }
22
23 @aplikacja.route('/', methods=['GET', 'POST'])
24 def strona_glowna():
25     pogoda = None
26     if request.method == 'POST':
27         kraj = request.form.get('kraj')
28         miasto = request.form.get('miasto')
29         if kraj and miasto:
30             pogoda = pobierz_pogode(miasto)
31     return render_template('strona.html', miasta=LOKALIZACJE, pogoda=pogoda)
32
33 def pobierz_pogode(miasto):
34     url = f"https://api.openweathermap.org/data/2.5/weather?q={miasto}&appid={KLUCZ_API}&units=metric&lang=pl"
35     odpowiedz = requests.get(url)
36     dane = odpowiedz.json()
37     if dane.get("main"):
38         return {
39             "miasto": miasto,
40             "temperatura": dane["main"]["temp"],
41             "wilgotnosc": dane["main"]["humidity"],
42             "cisnienie": dane["main"]["pressure"],
43             "opis": dane["weather"][0]["description"]
44         }
45     return None
46
47 if __name__ == '__main__':
48     logger.info(f"Aplikacja uruchomiona: {datetime.datetime.now()}")
49     logger.info(f"Autor: {AUTOR}")
50     logger.info(f"Nastuchowanie na porcie: {PORT}")
51     aplikacja.run(host='0.0.0.0', port=PORT)
```

- wymagania.txt

Open ▾

wymagania.txt
~/pogoda

```
1 flask
2 requests
3
```

- PlikDockerfile

```
1 # Etap 1: Obraz bazowy z Pythonem (lekka wersja slim dla mniejszego rozmiaru)
2 FROM python:3.10-slim AS base
3
4 # OCI metadata - informacje o obrazie i autorze (zgodne ze standardem OCI)
5 LABEL org.opencontainers.image.title="Aplikacja pogodowa Flask"
6 LABEL org.opencontainers.image.description="Serwer Flask pokazujący prognozę pogody dla wybranego miasta"
7 LABEL org.opencontainers.image.version="1.0"
8 LABEL org.opencontainers.image.authors="Joanna Jurkowska"
9
10 # Ustawiamy katalog roboczy wewnątrz kontenera
11 WORKDIR /aplikacja
12
13 # Kopiujemy tylko plik wymagania.txt - optymalizacja cache
14 COPY wymagania.txt .
15
16 # Instalujemy zależności - tylko raz, przy zmianie wymagania.txt
17 RUN pip install --no-cache-dir -r wymagania.txt
18
19 # Kopiujemy resztę kodu aplikacji - zmiany w kodzie nie psują warstwy pip
20 COPY . .
21
22 # Otwieramy port aplikacji (zdefiniowany w kodzie jako 8080)
23 EXPOSE 8080
24
25 # Zmienna środowiskowa PORT - do użycia w serwerze
26 ENV PORT=8080
27
28 # Healthcheck: sprawdza, czy aplikacja działa pod localhost:8080
29 HEALTHCHECK --interval=30s --timeout=3s --start-period=10s --retries=3 \
30     CMD wget -q --spider http://localhost:8080 || exit 1
31
32 # Domyślna komenda startowa - uruchamia serwer aplikacji
33 CMD ["python", "serwer.py"]
```

- szablony/strona.html

```

1 <!DOCTYPE html>
2 <html lang="pl">
3 <head>
4   <meta charset="UTF-8">
5   <title>Prognoza pogody</title>
6   <link rel="stylesheet" href="{ url_for('static', filename='style.css') }}">
7 </head>
8 <body>
9   <div class="container">
10     <h1>🌤 Prognoza pogody</h1>
11
12     <h2>Wybierz lokalizację</h2>
13     <form method="post">
14       <label for="kraj">Kraj:</label>
15       <select name="kraj" id="kraj" onchange="aktualizujMiasto()" required>
16         <option value="" disabled selected>-- wybierz kraj --</option>
17         {% for kraj in miasta %}
18           <option value="{{ kraj }}">{{ kraj }}</option>
19         {% endfor %}
20       </select>
21
22       <label for="miasto">Miasto:</label>
23       <select name="miasto" id="miasto" required>
24         <option value="" disabled selected>-- wybierz miasto --</option>
25       </select>
26
27       <input type="submit" value="Pokaż pogodę">
28     </form>
29
30     {% if pogoda %}
31     <div class="pogoda">
32       <h2>🌤 Pogoda dla {{ pogoda.miasto }}</h2>
33       <ul>
34         <li>🌡 Temperatura: {{ pogoda.temperatura }}°C</li>
35         <li>💧 Wilgotność: {{ pogoda.wilgotnosc }}%</li>
36         <li>📏 Ciśnienie: {{ pogoda.cisnienie }} hPa</li>
37         <li>📖 Opis: {{ pogoda.opis }}</li>
38       </ul>
39     </div>
40     {% endif %}
41 </div>
42
43 <script>
44   const miastaDlaKraju = {{ miasta | tojson }};
45
46   function aktualizujMiasto() {
47     const kraj = document.getElementById('kraj').value;
48     const selectMiasto = document.getElementById('miasto');
49     selectMiasto.innerHTML = "";
50
51     // Dodaj opcję domyślną
52     const opcjaDomyslna = document.createElement("option");
53     opcjaDomyslna.value = "";
54     opcjaDomyslna.text = "-- wybierz miasto --";
55     opcjaDomyslna.disabled = true;
56     opcjaDomyslna.selected = true;
57     selectMiasto.add(opcjaDomyslna);
58
59     // Dodaj miasta
60     miastaDlaKraju[kraj].forEach(miasto => {
61       const opcja = document.createElement("option");
62       opcja.text = miasto;
63       opcja.value = miasto;
64       selectMiasto.add(opcja);
65     });
66   }
67 </script>
68 </body>
69 </html>

```

- style/style.css

```

1 body {
2   font-family: 'Segoe UI', sans-serif;
3   background: linear-gradient(to bottom right, #87CEFA, #f0f8ff);
4   margin: 0;
5   padding: 0;
6   color: #333;
7 }
8
9 .container {
10   max-width: 600px;
11   margin: 60px auto;
12   padding: 30px;
13   background: white;
14   border-radius: 10px;
15   box-shadow: 0 5px 25px rgba(0, 0, 0, 0.1);
16 }
17
18 h1 {
19   text-align: center;
20   color: #0077be;
21 }
22
23 form {
24   display: flex;
25   flex-direction: column;
26   gap: 15px;
27 }
28
29 label {
30   font-weight: bold;
31 }
32
33 select, button {
34   padding: 10px;
35   font-size: 1rem;
36   border: 1px solid #aaa;
37   border-radius: 5px;
38 }
39
40 button {
41   background-color: #0077be;
42   color: white;
43   cursor: pointer;
44   transition: background 0.3s;
45 }
46
47 button:hover {
48   background-color: #005f9e;
49 }
50
51 .pogoda {
52   margin-top: 30px;
53   background-color: #f3f9ff;
54   padding: 20px;
55   border-radius: 10px;
56 }
57
58 .pogoda h2 {
59   margin-top: 0;
60   color: #0077be;
61 }
62
63 .pogoda ul {
64   list-style: none;
65   padding: 0;
66 }
67

```

Zadanie 3

a) zbudowania opracowanego obrazu kontenera,

Za pomocą polecenia:

- `docker build -f PlikDockerfile -t pogoda .`

budujemy obraz kontenera

```
asia@Ubuntu:~/pogoda$ docker build -f PlikDockerfile -t pogoda .
[+] Building 8.9s (10/10) FINISHED                                docker:default
=> [internal] load build definition from PlikDockerfile           0.0s
=> => transferring dockerfile: 200B                               0.0s
=> [internal] load metadata for docker.io/library/python:3.10-slim 1.4s
=> [auth] library/python:pull token for registry-1.docker.io      0.0s
=> [internal] load .dockerignore                                  0.0s
=> => transferring context: 2B                                       0.0s
=> [1/4] FROM docker.io/library/python:3.10-slim@sha256:57038683f4a259e1 0.0s
=> [internal] load build context                                 0.0s
=> => transferring context: 1.86kB                                   0.0s
=> CACHED [2/4] WORKDIR /aplikacja                               0.0s
=> [3/4] COPY . .                                                 0.1s
=> [4/4] RUN pip install --no-cache-dir -r wymagania.txt         7.0s
=> exporting to image                                             0.2s
=> => exporting layers                                              0.2s
=> => writing image sha256:f42054612e21c6094c24ac711cbc4e10333414e5a4741 0.0s
=> => naming to docker.io/library/pogoda                          0.0s

What's Next?
View a summary of image vulnerabilities and recommendations -> docker scout quickview
```

Jak widac na screenie, operacja powiodla sie, obraz zostal poprawnie zbudowany

b) uruchomienia kontenera na podstawie zbudowanego obrazu,

Za pomocą polecenia:

- `docker run -e KLUCZ_API_POGODOWY=2abc04c28f6034d8fd49c8819f1bd440 -p 8080:8080 pogoda`

kontener Docker zostaje uruchomiany z obrazem pogoda, mapując port 8080 i przekazując do niego zmienna środowiskowa KLUCZ_API_POGODOWY o wartości 2abc04c28f6034d8fd49c8819f1bd440

```
asia@Ubuntu:~/pogoda$ docker run -e KLUCZ_API_POGODOWY=2abc04c28f6034d8fd49c8819f1bd440 -p 8080:8080 pogoda
Aplikacja uruchomiona: 2025-05-05 11:51:02.514706
Autor: Joanna Jurkowska
Nasłuchiwanie na porcie: 8080
* Serving Flask app 'serwer'
* Debug mode: off
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://127.0.0.1:8080
* Running on http://172.17.0.3:8080
Press CTRL+C to quit
```

Jak widac na screenie wymagania z podpunktu a zostały spełnione. Wyświetlona zostaje informacja o dacie uruchomienia, informacje o autorze są wyświetlane, informacje o porcie TCP są przedstawione

c) sposobu uzyskania informacji, które wygenerował serwer w trakcie uruchamiania kontenera (patrz: punkt 1a): `docker run -it -e KLUCZ_API_POGODOWY=2abc04c28f6034d8fd49c8819f1bd440 -p 8080:8080 pogoda`

```
^Casia@Ubuntu:~/pogoda$ docker run -it -e KLUCZ_API_POGODOWY=2abc04c28f6034d8fd49c8819f1bd440 -p 8080:8080 pogoda
Aplikacja uruchomiona: 2025-05-06 07:06:10.379814
Autor: Joanna Jurkowska
Nasłuchiwanie na porcie: 8080
* Serving Flask app 'serwer'
* Debug mode: off
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://127.0.0.1:8080
* Running on http://172.17.0.3:8080
Press CTRL+C to quit
```

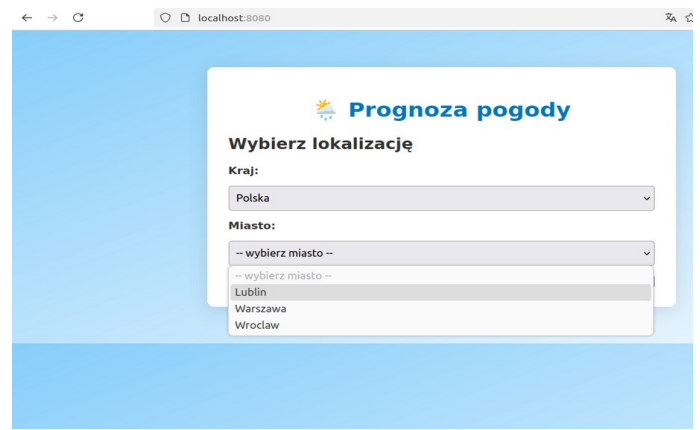
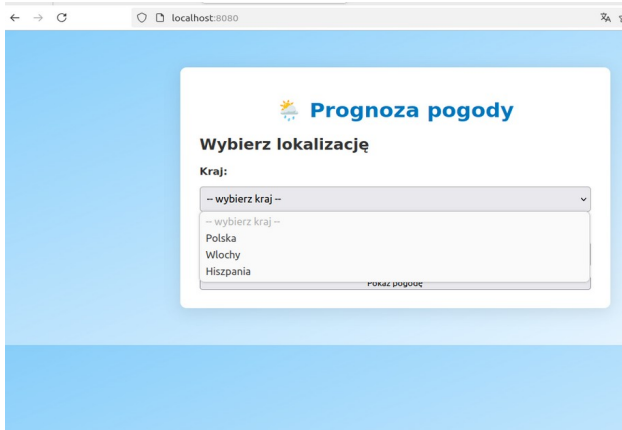
d) sprawdzenia, ile warstw posiada zbudowany obraz.

```
asia@Ubuntu:~/pogoda$ docker history pogoda
IMAGE          CREATED          CREATED BY
SIZE           COMMENT
582fdf76069c   10 minutes ago  CMD ["python" "serwer.py"]
0B             buildkit.dockerfile.v0
<missing>      10 minutes ago  HEALTHCHECK &{["CMD-SHELL" "wget -q --spider...
0B             buildkit.dockerfile.v0
<missing>      10 minutes ago  ENV PORT=8080
0B             buildkit.dockerfile.v0
<missing>      10 minutes ago  EXPOSE map[8080/tcp:{}]
0B             buildkit.dockerfile.v0
<missing>      10 minutes ago  COPY . . # buildkit
6.16kB         buildkit.dockerfile.v0
<missing>      10 minutes ago  RUN /bin/sh -c pip install --no-cache-dir -r...
13.6MB         buildkit.dockerfile.v0
<missing>      10 minutes ago  COPY wymagania.txt . # buildkit
16B           buildkit.dockerfile.v0
<missing>      43 hours ago    WORKDIR /aplikacja
0B             buildkit.dockerfile.v0
<missing>      43 hours ago    LABEL org.opencontainers.image.authors=Joann...
0B             buildkit.dockerfile.v0
<missing>      43 hours ago    LABEL org.opencontainers.image.version=1.0
0B             buildkit.dockerfile.v0
<missing>      43 hours ago    LABEL org.opencontainers.image.description=S...
0B             buildkit.dockerfile.v0
<missing>      43 hours ago    LABEL org.opencontainers.image.title=Aplikac...
0B             buildkit.dockerfile.v0
<missing>      3 weeks ago     CMD ["python3"]
0B             buildkit.dockerfile.v0
<missing>      3 weeks ago     RUN /bin/sh -c set -eux; for src in idle3 p...
36B           buildkit.dockerfile.v0
<missing>      3 weeks ago     RUN /bin/sh -c set -eux; savedAptMark="$(a...
43.2MB         buildkit.dockerfile.v0
<missing>      3 weeks ago     ENV PYTHON_SHA256=4c68050f049d1b4ac5aadd0df5...
0B             buildkit.dockerfile.v0
<missing>      3 weeks ago     ENV PYTHON_VERSION=3.10.17
0B             buildkit.dockerfile.v0
<missing>      3 weeks ago     ENV GPG_KEY=A035C8C19219BA821ECEA86B64E628F8...
0B             buildkit.dockerfile.v0
<missing>      3 weeks ago     RUN /bin/sh -c set -eux; apt-get update; a...
9.23MB         buildkit.dockerfile.v0
<missing>      3 weeks ago     ENV LANG=C.UTF-8
0B             buildkit.dockerfile.v0
<missing>      3 weeks ago     ENV PATH=/usr/local/bin:/usr/local/sbin:/usr...
0B             buildkit.dockerfile.v0
<missing>      3 weeks ago     # debian.sh --arch 'amd64' out/ 'bookworm' '...
74.8MB         debuerreotype 0.15
```


W celu zobaczenia czy wymagania z podpunktu b zostały spełnione: w przeglądarce wpisujemy adres:

- <http://localhost:8080>

ze screenow widac ze jest mozliwosc wybrania z listy rozwijanej zarowno kraju jak miasta



Przycisk: Pokaż pogodę zapisuje wybór miejsca oraz wyświetla pogodę dla dowolnego miejsca zawartego w liście rozwijanej

