

Sprawozdanie:

Kod app.js

```
const express = require('express'); // Import biblioteki Express, która ułatwia tworzenie serwera HTTP w Node.js.
const axios = require('axios'); // Import biblioteki Axios, która służy do wykonywania zapytań HTTP (np. do API pogodowego).
const dotenv = require('dotenv'); // Import biblioteki dotenv, która pozwala wczytywać zmienne środowiskowe z pliku '.env'.
const app = express(); // Tworzy nową app Express
dotenv.config(); // Wczytuje zmienne z pliku '.env'

// port i klucz API pobierane z pliku .env
const PORT = process.env.PORT || 3000;
const API_KEY = process.env.API_KEY;

// predefiniowana lista krajów i miast
const cities = {
  Poland: ['Warsaw', 'Gdańsk', 'Lublin'],
  Germany: ['Berlin', 'Munich']
};

// użycie folderu do statycznych plików (index.html)
app.use(express.static('public'));
app.use(express.urlencoded({ extended: true }));

app.post('/weather', async (req, res) => {
  const { country, city } = req.body; // pobiera dane z formularza
  const url =
    `https://api.openweathermap.org/data/2.5/weather?q=${city},${country}&appid=${API_KEY}&units=metric`;
  try {
    const response = await axios.get(url); // Wysyła zapytanie GET do API i czeka na odpowiedź.
    const weather = response.data; // zapis danych z odp
    res.send(`<h1>Weather in ${city}, ${country}</h1><p>${weather.weather[0].description}, temp: ${weather.main.temp}°C</p>`);
    // Wyświetla prosty HTML z opisem pogody i temperaturą.
  } catch {
    res.send(`<p>Error getting weather for ${city}</p>`);
  }
});

app.listen(PORT, () => { // uruchamia serwer i nasłuchuje na określonym porcie
  console.log(`App started on port ${PORT}`);
  console.log(`Author: Yuliia Kozlova`);
  console.log(`Start date: ${new Date().toISOString()}`);
});
```

Plik Dockerfile

```
#etap 1: Budowanie
FROM node:20-alpine AS builder
WORKDIR /app
COPY package*.json ./

#instalacja tylko zaleznosci produkcyjnych
RUN npm ci --omit=dev
COPY . .

#etap 2: Minimalny obraz
FROM node:20-alpine
LABEL org.opencontainers.image.authors="Yuliia Kozlova"
WORKDIR /app
COPY --from=builder /app /app
ENV NODE_ENV=production
EXPOSE 3000
HEALTHCHECK --interval=30s --timeout=10s --start-period=5s --retries=3 \
  CMD wget --spider http://localhost:3000 || exit 1
CMD ["node", "app.js"]
```

Zrzuty :

a) Zbudowania opracowanego obrazu kontenera

```
D:\pollub\sem 6\pwcho\zadanie1>docker build -t yuliia/weather-app .
[+] Building 3.3s (11/11) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile               0.0s
=> => transferring dockerfile: 552B                               0.0s
=> [internal] load .dockerignore                                  0.0s
=> => transferring context: 2B                                       0.0s
=> [internal] load metadata for docker.io/library/node:20-alpine 1.0s
=> [builder 1/5] FROM docker.io/library/node:20-alpine@sha256:c628bdc7ebc7f95b1b23249a445eb415ce68ae9def8b68364b 0.0s
=> [internal] load build context                                  0.5s
=> => transferring context: 4.75MB                                   0.5s
=> CACHED [builder 2/5] WORKDIR /app                             0.0s
=> [builder 3/5] COPY package*.json ./                            0.0s
=> [builder 4/5] RUN npm ci --omit=dev                           1.4s
=> [builder 5/5] COPY . .                                         0.1s
=> [stage-1 3/3] COPY --from=builder /app /app                   0.1s
=> exporting to image                                             0.1s
=> => exporting layers                                              0.1s
=> => writing image sha256:e1c831e01dce0d0e0db96f1fea0b7d12c17da2bc9e18f792aef07a02c762287e 0.0s
=> => naming to docker.io/yuliia/weather-app                     0.0s

View build details: docker-desktop:///dashboard/build/default/default/ytgdhoicn6hwci55w2mpfikoe

What's Next?
 1. Sign in to your Docker account → docker login
 2. View a summary of image vulnerabilities and recommendations → docker scout quickview
```

b) uruchomienia kontenera na podstawie zbudowanego obrazu

```
D:\pollub\sem 6\pwcho\zadanie1>docker run -d -p 3000:3000 --name pogoda yuliia/weather-app
b687d8a84c6d1d20242a24e915a03b95540cf9fc793a1d040787e509943f06a8
```


- c) sposobu uzyskania informacji z logów, które wygenerowała opracowana aplikacja podczas uruchamiania kontenera

```
D:\pollub\sem 6\pwcho\zadanie1>docker logs pogoda
App started on port 3000
Author: Yuliia Kozlova
Start date: 2025-05-07T17:35:04.533Z
```

```
D:\pollub\sem 6\pwcho\zadanie1>docker image inspect yuliia/weather-app --format='{{.RootFS.Layers}}'
'[sha256:08000c18d16dadf9553d747a58cf44023423a9ab010aab96cf263d2216b8b350 sha256:75228a052fed2baf07b8652c14c8aba95480f15
6a027bf287f1ad44237d4ca9a sha256:2c9cabe102898883f748afd7f07319db1bcfeed1124c768394ba01d122cd526f sha256:b8185dcc9be5529
abfdea9590093f7474f53a9cbc91a3adf3fbabb3a42a83ec4 sha256:9e3b4e3e7cbe2da94d86fd543d92c4775d8cea3e35b3045b676662eedab48d1
f sha256:0fb2e2f690752c86d36284dc06be15f4c007948e44def06c65ceea6dc064d76f]'
```

```
D:\pollub\sem 6\pwcho\zadanie1>docker image ls yuliia/weather-app
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
yuliia/weather-app  latest     e1c831e01dce  4 minutes ago  139MB
```

```
D:\pollub\sem 6\pwcho\zadanie1>
```




yuliia/weather-app

Running

0.01% 3000:3000 3 minutes ago

<



pogoda

yuliia/weather-app

b687d8a84c6d

3000:3000

STATUS

Exited (255) (1 second ago)

■

▶

↺

🗑

Logs

Inspect

Bind mounts

Exec

Files

Stats

2025-05-07 19:35:04

App started on port 3000

2025-05-07 19:35:04

Author: Yuliia Kozlova

2025-05-07 19:35:04

Start date: 2025-05-07T17:35:04.533Z

Weather in Warsaw, Poland

broken clouds, temp: 9.06°C