# WIZUALIZACJA I RAPORTOWANIE DANYCH

## ĆWICZENIE 4

KAMIL BUDZYN 229850

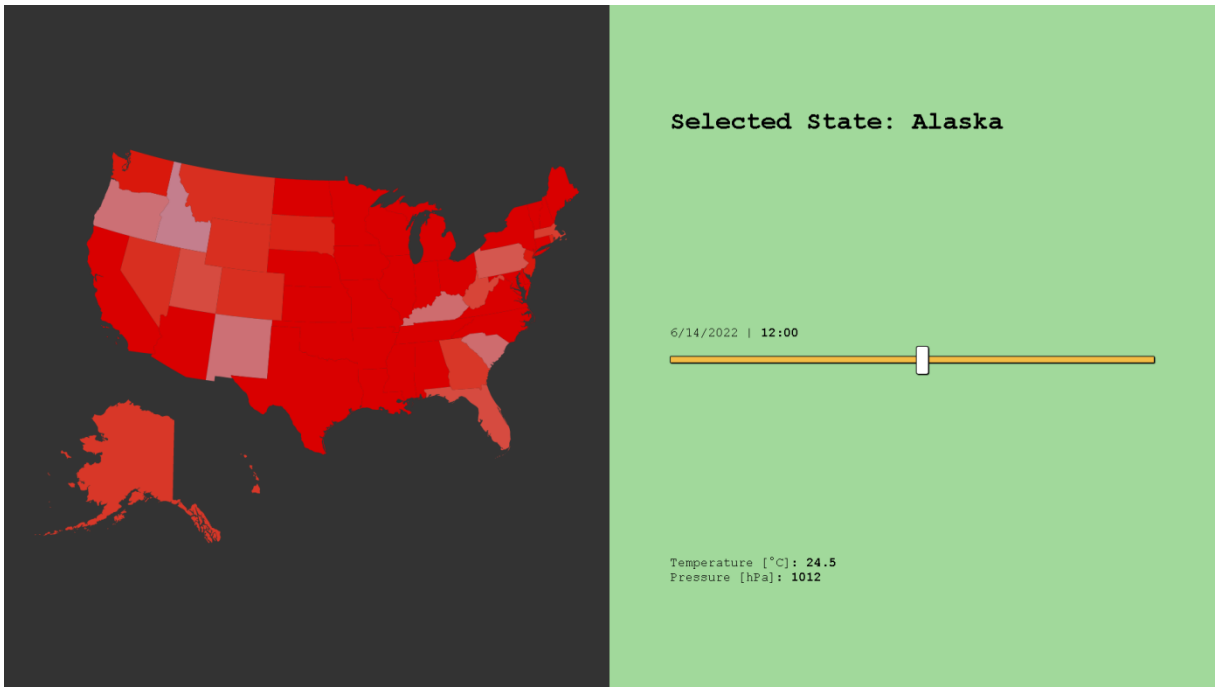
ETAPY PROCESU WIZUALIZACJI, ĆWICZENIE 4 NA PODSTAWIE BEN FRY - THE SEVEN STAGES OF VISUALIZING DATA:

1. **ACQUIRE – POZYSKIWANIE**
    a. Za pomocą API “WEATHERAPI” (HTTPS://WWW.WEATHERAPI.COM/) zostały pobrane dane na temat geolokalizacji oraz wartości pomiarów temperatury i ciśnienia atmosferycznego.
2. **PARSE – PARSOWANIE**
    a. Wszystkie uzyskane dane musiały przejść przez etap parsowania, dzięki któremu udało się odseparować dane związane tylko z konkretną lokalizacją.
3. **FILTER – FILTROWANIE**
    a. Na potrzeby projektu zostały wybrane tylko niezbędne dane z wszystkich dostępnych dla dewelopera.
4. **MINE – POWIĄZANIE**
    a. Połączenie danych w odpowiednie grupy w celu prezentacji wyników.
5. **REPRESENT – REPREZENTACJA**
    a. Wybór formy wizualizacji. Zastosowałem Radio SVG Map.
6. **REFINE – PRZETWORZENIE**
    a. Poprawienie końcowej wersji wizualizacji, aby była wyraźniejsza i bardziej angażująca wizualnie za pomocą kolor kodowania obszarów na mapie w zależności od tempereatury.
7. **INTERACT – INTERAKCJA**
    a. Implementacja interakcji, dzięki których użytkownik aplikacji może wybrać interesujący go zestaw danych. Projekt uwzględnia wybór lokalizacji oraz godziny aktualnego dnia, w celu uzyskania temperatury oraz ciśnienia atmosferycznego.

KOD APLIKACJI

```jsx
import { useState, useEffect } from "react";

import { RadioSVGMap } from "react-svg-map";
import USA from "@svg-maps/usa";
import "react-svg-map/lib/index.css";

import "./styles.css";
import "./input-range.css";

const API_KEY = "21a5a33ed3f049f9bbc131308221905";

const colors = [ ···
];

const locations = [ ···
];

const App = () => {
  const [selectedLocation, setSelectedLocation] = useState("Alaska");
  const [selectedHour, setSelectedHour] = useState(12);
  const [temperature, setTemperature] = useState("");
  const [pressure, setPressure] = useState("");

  const date = new Date();

  let elements;
  if (!elements) {
    elements = document.querySelector(":root");
  }

  const getAllWeatherAndColor = async () => {
    for (let state of locations) {
      console.log(state);

      const apiCall = await fetch(
        `https://api.weatherapi.com/v1/forecast.json?key=${API_KEY}&q=${state.name}&days=1&aqi=no&alerts=no`
      );

      const response = await apiCall.json();

      const temperature =
        response.forecast.forecastday[0].hour[selectedHour].temp_c;

      colorState(temperature, state);
    }
  };

  function colorState(temperature, state) {
    let color;
    if (temperature <= 10) color = colors[0];
    else if (temperature >= 29) color = colors[29];
    else color = colors[Number(temperature).toFixed(0)];

    let element = document.getElementById(state.id);
    element.style.fill = color;
  }

  function color_coding() {
    let color;
    if (temperature <= 10) color = colors[0];
    else if (temperature >= 29) color = colors[29];
    else color = colors[Number(temperature).toFixed(0)];
    console.log(Number(temperature).toFixed(0));
    elements.style.setProperty("--fill", color);
  }

  const getWeather = async () => {
    const apiCall = await fetch(
      `https://api.weatherapi.com/v1/forecast.json?key=${API_KEY}&q=${selectedLocation}&days=1&aqi=no&alerts=no`
    );

    const response = await apiCall.json();

    const temperature =
      response.forecast.forecastday[0].hour[selectedHour].temp_c;

    const pressure =
      response.forecast.forecastday[0].hour[selectedHour].pressure_mb;

    setTemperature(temperature);
    setPressure(pressure);
  };

  useEffect(() => {
    getWeather();
  }, [selectedHour, selectedLocation]);

  useEffect(() => {
    color_coding();
    getAllWeatherAndColor();
  }, [temperature]);

  return (
    <div className="app-container">
      <div className="map-container">
        <RadioSVGMap
          map={USA}
          className="map"
          onChange={(event) => {
            const stateName = Object.entries(event)[1][1].name;
            setSelectedLocation(stateName);
          }}
        />
      </div>
      <div className="right-container">
        <div className="inner-right-container">
          <h1>Selected State: {selectedLocation}</h1>
          <div className="range-input-container">
            <label htmlFor="input">
              {date.toLocaleDateString()} |{" "}
              <strong>{selectedHour + ":00"}</strong>
            </label>
            <input
              id="input"
              type="range"
              className="range-input"
              value={selectedHour}
              min="0"
              max="23"
              step="1"
              onChange={(e) => {
                setSelectedHour(e.target.value);
              }}
            />
          </div>
          <div className="data-container">
            <p>
              Temperature [°C]: <strong>{temperature}</strong>
            </p>
            <p>
              Pressure [hPa]: <strong>{pressure}</strong>
            </p>
          </div>
        </div>
      </div>
    </div>
  );
};

export default App;
```

WIDOK APLIKACJI PRZED INTERAKCJĄ




WIDOK APLIKACJI PO INTERAKCJI