

APLIKACJA DO WIZUALIZACJI DANYCH METEOROLOGICZNYCH DLA MIASTA BYDGOSZCZ

Dawid Bobowski, Katarzyna Kazimierska-Drobny

Uniwersytet Kazimierza Wielkiego
Wydział Informatyki, Wydział Mechatroniki
ul. Kopernika 1, 85-074 Bydgoszcz
e-mail: kkd@ukw.edu.pl

Streszczenie: Niniejsza praca przedstawia projekt aplikacji, która umożliwia wczytywanie i wizualizację danych meteorologicznych dla miasta Bydgoszcz. W tym celu wykorzystano dane meteorologiczne z lat 2001-2021 z terenu Kujaw. Stworzona aplikacja ma za zadanie w przystępny sposób dokonać wizualizacji danych pomiarowych. Ponadto za pomocą aplikacji możliwe jest porównanie wybranych parametrów w celu przeanalizowania i zaobserwowania zmian klimatycznych na terenie Bydgoszczy na przestrzeni lat.

Słowa kluczowe: dane meteorologiczne, wizualizacja, TypeScript, SQLite3

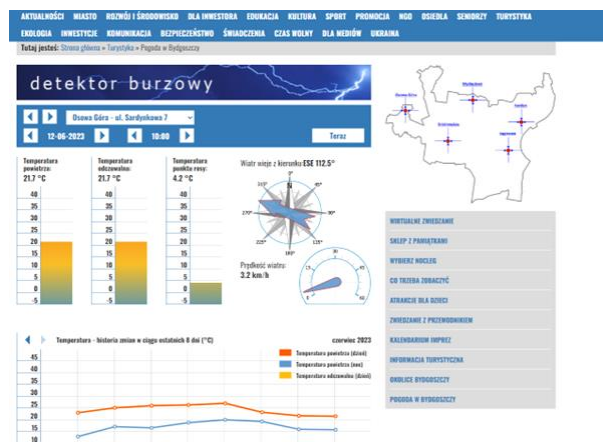
Application for visualization of meteorological data for the city of Bydgoszcz

Abstrakt: This work presents the design of an application for loading and visualization of meteorological data for the city of Bydgoszcz. For this purpose, meteorological data from 2001-2021 from Kujawy were used. The created application is designed to visualize measurement data in an accessible way. Moreover, using the application it will be possible to compare selected parameters in order to analyze and observe climate changes in Bydgoszcz over the years.

Keywords: meteorological data, visualization, TypeScript, SQLite3

1. WSTĘP

Z Na terenie Bydgoszczy znajduje się 5 automatycznych stacji meteorologicznych w tym: Stacja Śródmieście (ul. Grudziądzka); Stacja Fordon (ul. Golloba); Stacja Osowa Góra (ul. Sardynkowa); stacja Myślęcinek (teren Bydgoskiego Centrum Targowo-Wystawienniczego) oraz stacja Łęgnowo (ul. Toruńska 324A). Odczyty z wymienionych stacji są udostępnione publicznie na oficjalnej stronie internetowej miasta. Odczyty są prezentowane po wyborze stacji pogodowej, daty, godziny (w odstępie czasowym co 1h). Dodatkowo wyświetlana jest mapa lokalizacji stacji na terenie miasta, wykresy temperatur, kierunku i prędkości wiatru, historia zmian temperatury z ostatnich 8 dni, wykresy ciśnienia atmosferycznego, opadów, wilgotności powietrza oraz barometr (rys. 1). Oprócz tego strona oferuje narzędzie do monitorowania burzy, które przy pomocy sygnałów radiowych wykrywa wyładowania w promieniu do 480 km [1].



Rysunek. 1 Prezentacja pogody w Bydgoszczy na oficjalnej stronie miasta

Niniejsza praca przedstawia projekt aplikacji, która umożliwia wczytywanie i wizualizację danych meteorologicznych dla miasta Bydgoszcz. W tym celu

wykorzystano dane meteorologiczne z lat 2001-2021 z terenu Kujaw. Stworzona aplikacja ma za zadanie w przystępny sposób dokonać wizualizacji danych pomiarowych. Ponadto za pomocą aplikacji możliwe będzie porównanie wybranych parametrów w celu przeanalizowania i zaobserwowania zmian klimatycznych na terenie Bydgoszczy na przestrzeni lat. Na podstawie danych meteorologicznych aplikacja umożliwi obliczenia wielkości parowania i ewapotranspiracji. Aplikacja została napisana w języku TypeScript, wykorzystując framework React do części interaktywnej oraz Node wraz z frameworkiem Next do obsługi warstwy logicznej. Baza danych została zaimplementowana przy pomocy SQLite3 i obsługiwana biblioteką Knex.

2. PROJEKT APLIKACJI

Projekt aplikacji bazuje na danych pogodowych zebranych za pomocą stacji meteorologicznej Bydgoszcz na przełomie lat 2001-2021. Dane zostały uzyskane z systemu Woda dla Kujaw opracowanego przez Grupę Operacyjną „Woda dla Kujaw” realizowaną w ramach Europejskiego partnerstwa innowacyjnego na rzecz wydajnego i zrównoważonego rolnictwa (EIP-AGRI). Każdego dnia od 1 kwietnia do 31 października w postaci tabeli zostały dostarczone dane: "Data" – data wykonania pomiaru; "T" – temperatura powietrza; "RH" – wilgotność względna powietrza; "V" – prędkość wiatru; "P" – opad atmosferyczny; "Ra" – promieniowanie słoneczne na górnej granicy atmosfery; "es" – ciśnienie nasyconej pary wodnej; "ea" – aktualne ciśnienie pary wodnej; "VPD" – niedosyt wilgotności powietrza; "RS0" – promieniowanie słoneczne przy braku zachmurzenia; "Rs" – promieniowanie słoneczne; "RNL" – promieniowanie długofalowe; "RNS" – promieniowanie krótkofalowe netto; "ET0" – ewapotranspiracja referencyjna.

W celu ułatwienia wizualizacji i porównywania danych meteorologicznych z tabeli została stworzona aplikacja zaimplementowana w formie aplikacji przeglądarkowej dla komputerów stacjonarnych lub laptopów (bez obsługi interfejsu graficznego dla urządzeń mobilnych). Technologie wykorzystane do stworzenia interfejsu graficznego to język TypeScript, framework React, biblioteka Redux oraz biblioteka komponentów Material UI.

TypeScript to język programowania, który jest rozszerzeniem języka JavaScript. Dodaje on statyczne typowanie, dzięki czemu umożliwia zdefiniowanie typów

dla zmiennych, parametrów funkcji i obiektów. Język ten kompiluje się do czystego JavaScriptu, co pozwala na uruchamianie kodu na dowolnej platformie, która go obsługuje [2]. React to popularny framework JavaScript do budowy interfejsów użytkownika. Opiera się na komponentach, które pozwalają tworzyć modułowe i łatwo zarządzane struktury UI. Wykorzystuje wirtualny DOM i efektywnie aktualizuje tylko zmienione elementy, co przyspiesza renderowanie i poprawia wydajność aplikacji. Dzięki deklaracywnemu podejściu React jest często wybieranym rozwiązaniem do tworzenia nowoczesnych i responsywnych aplikacji internetowych [3]. Redux to biblioteka do zarządzania stanem aplikacji w połączeniu z biblioteką React. Wprowadza jednokierunkowy przepływ danych, w którym stan globalny jest przechowywany w jednym miejscu – w tzw. store. Komponenty React mogą odczytywać i modyfikować ten stan poprzez akcje, które są wysyłane do store'a i przetwarzane przez reduktory. Dzięki tej architekturze React Redux ułatwia zarządzanie stanem aplikacji, zapewniając jednoznaczność, skalowalność i łatwą testowalność [4]. Material UI to biblioteka komponentów interfejsu użytkownika, oparta na Material Design, stworzona dla frameworka React. Oferuje gotowe do użycia i estetycznie zaprojektowane komponenty, takie jak przyciski, formularze, nawigacja, karty i wiele innych. Zapewnia spójny i responsywny wygląd aplikacji, a także dostarcza zaawansowane funkcje takie jak animacje, układy siatki i stylowanie [5].

Technologie wykorzystane do stworzenia serwera i obsługi bazy danych: Node; Express; Knex oraz SQLite3. Node to środowisko uruchomieniowe oparte na języku JavaScript, które umożliwia programistom uruchamianie kodu JavaScript poza przeglądarką internetową [6]. Express to minimalistyczny framework aplikacji sieciowych dla Node, który pozwala na szybkie tworzenie serwerów API i aplikacji internetowych. Zapewnia prostą i intuicyjną składnię do obsługi tras, żądań i odpowiedzi HTTP, umożliwiając łatwe budowanie i obsługę aplikacji webowych [7]. Knex to biblioteka zapytań SQL dla języka JavaScript, która umożliwia łatwe tworzenie zapytań i manipulowanie danymi w relacyjnych bazach danych [8]. SQLite3 to lekka, samodzielna i bezserwerowa relacyjna baza danych, która działa bez potrzeby zewnętrznego serwera. Jest to popularna i szeroko stosowana baza danych, idealna do prostych projektów lub aplikacji o niewielkim rozmiarze. SQLite3 oferuje pełną obsługę SQL, transakcje, indeksowanie, a także wsparcie dla wielu języków programowania, w tym dla języka JavaScript [9]. Program wykorzystany do napisania

aplikacji to Visual Studio Code. Jest to darmowy i otwarty edytor kodu stworzony przez firmę Microsoft. Jest dostępny na wielu platformach i oferuje szeroką gamę funkcji, takich jak podświetlanie składni, automatyczne uzupełnianie, debugowanie, integracja z systemem kontroli wersji i wiele innych. VS Code obsługuje wiele języków programowania i posiada rozbudowany ekosystem rozszerzeń ułatwiających pracę [10].

OPIS INTERFEJSU, DOSTĘPNYCH FUNKCJI I DZIAŁANIA

Główny interfejs graficzny składa się z dwóch części: panelu głównego z trzema wykresami przedstawiającymi dane o średniej temperaturze, opadzie atmosferycznym i ewapotranspiracji wskaźnikowej na dany okres oraz panelu bocznego, na którym są dostępne wszystkie najważniejsze funkcje aplikacji. Na rysunku 2 przedstawiono projekt interfejsu graficznego do wizualizacji danych meteorologicznych.



Rysunek. 2 Widok głównego interfejsu graficznego aplikacji

Na panelu bocznym dostępne są 4 funkcje umożliwiające manipulowanie wyświetlanymi danymi: „Wybór roczników” – pozwala na wybranie maksymalnie pięciu roczników z listy rozwijanej w celu wyświetlenia danych z określonych lat na wykresach; „Wybór miesiąca” – służy do zautomatyzowanego zawężenia przedziału czasowego na wykresach do wybranego miesiąca; „pobieranie danych” – przycisk, który pozwala pobrać dane meteorologiczne z wykresów z uwzględnieniem naniesionego przedziału czasowego. Jeżeli żadna z powyższych opcji nie zostanie wykorzystana, zostaną wzięte pod uwagę dane z całego dostępnego przedziału (od 1 kwietnia do 31 października). Pobrane dane zapisywane są do pliku CSV.

Ostatni przycisk to „resetowanie przedziału” – przycisk do resetowania niestandardowego przedziału ustawionego na wykresach. Po wciśnięciu wszystkie wykresy zostaną ustawione z powrotem na okres od 1 kwietnia do 31 października.

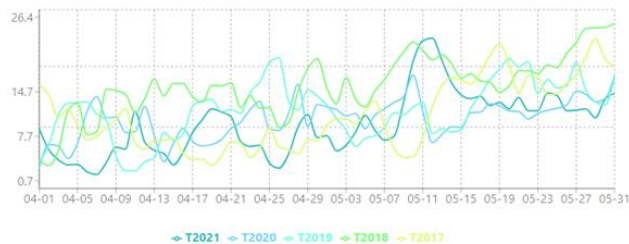
W części głównej aplikacji widoczne są trzy wykresy, które poza wizualizacją pozwalają również na dokładniejsze sprawdzenie wyników w konkretny dzień poprzez najechanie myszką na wybrany wykres. Dzięki temu wyświetli się okienko przedstawiające wartości z danego dnia i zestawu danych. Dodatkowo możliwe jest ręczne zdefiniowanie przedziału wykresu poprzez kliknięcie wykresu i przeciągnięcie w celu zaznaczenia wybranego przedziału, który podświetli się ciemniejszym kolorem. Po puszczeniu przycisku przedział wykresu zostanie zmniejszony i dostosowany. Pozwala to na bardziej szczegółową analizę danych w najmniejszym odstępnie do dwóch dni.

Serwer napisany został w Express i pozwala na połączenie aplikacji z bazą danych i pobranie informacji przy pomocy endpointu API /table/:name, gdzie name to nazwa tabeli z danymi. Na potrzeby uruchomienia projektu w sieci lokalnej wykorzystana została biblioteka cors, która umożliwia wysyłanie zapytań w ramach tej samej sieci bez blokowania żądań przez przeglądarkę. Plik znajduje się w folderze src/backend. Biblioteka Knex obsługuje połączenie do bazy danych oraz obsługę zapytań SQL. Baza danych została zapisana w pliku evapo.sqlite3, a dane źródłowe, które zostały wczytane są dostępne w plikach CSV w folderze src/database/tables. Do obsługi żądań po stronie klienta został stworzony plik index.ts znajdujący się wewnątrz folderu src/api. Składa się z funkcji getDbTable(), która obsługuje wysyłanie żądania o dane z tabeli i zwraca je w formacie JSON do dalszej obsługi przez aplikację. Do zarządzania stanem dla aplikacji wykorzystano bibliotekę Redux. Biorąc pod uwagę wcześniejszą konfigurację reducerów, w aplikacji zostały stworzone tablesSlice oraz chartsSlice w celu odseparowania obsługi stanów i akcji dotyczących tabel oraz samej aplikacji. Główne zadanie funkcji useEffect() dostępnej w ramach Reacta to umożliwienie reagowania na zmiany wewnątrz komponentu, takie jak zmiana stanu, aktualizacja właściwości (props) lub zmiana innych danych. Na potrzeby obsługi panelu bocznego została zaimplementowana jedna funkcja useEffect(), która monitoruje stan wybranego miesiąca selectedMonth. Za każdym razem, gdy na liście wyboru użytkownik wybierze inny miesiąc, zostanie ona aktywowana w celu zaktualizowania zakresów osi na wykresach. Przycisk odpowiedzialny za pobieranie danych został obsłużony

funkcją `createTextFile()`, która za pomocą biblioteki `FileSaver` ułatwia implementację funkcjonalności. W pliku `MainCharts.tsx` znajduje się główna część kodu odpowiedzialna za obsługę wykresów w aplikacji. Wykresy zaimplementowano przy pomocy biblioteki `Recharts`, która oferuje gotowe rozwiązania do wizualizacji danych dzięki personalizowanym wykresom.

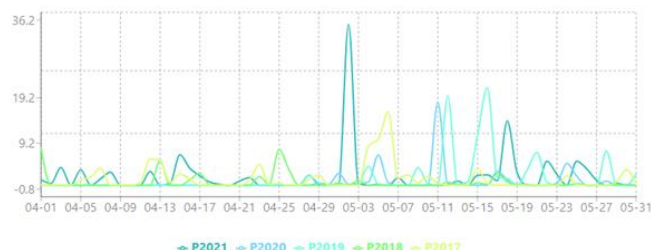
3 TESTOWANIE APLIKACJI. PORÓWNANIE DANYCH METEOROLOGICZNYCH DLA MIASTA BYDGOSZCZ

W celu przetestowania działania aplikacji przeprowadzono porównania warunków pogodowych w Bydgoszczy i okolicy na przestrzeni lat 2017-2021. Na rysunku 3 przedstawiono porównanie wahań się temperatury w latach 2017-2021. Pomiary temperatury dokonywane były od początku miesiąca kwietnia do końca maja. Zaobserwowano niskie temperatury bliskie zeru w kwietniu oraz spore wahania przez cały okres przedwakacyjny.



Rysunek. 3 Wykres średniej temperatury w okresie przedwakacyjnym w latach 2017-2021

Na rysunku 4 przedstawiono porównanie opadów atmosferycznych w latach 2017-2021. Pomiary temperatury dokonywane były od początku kwietnia do końca maja. Zauważono, że opady są sporadyczne i w niskich ilościach. Odczytano pojedyncze opady powyżej 7,5 mm w okresie przedwakacyjnym w każdym z roczników oraz jeden opad nawalny bliski 40 mm. W danym okresie lata 2020 i 2018 nie obfitowały w opady.



Rysunek.4. Wykres opadu atmosferycznego w okresie przedwakacyjnym w latach 2017-2021

4. PODSUMOWANIE I WNIOSKI

W pracy przedstawiono aplikację do wizualizacji danych meteorologicznych z miasta Bydgoszcz. Opracowana aplikacja pozwala na zwiększenia świadomości społecznej na temat zmian klimatycznych i globalnego ocieplenia. Dzięki możliwościom wizualizacji danych i analizy użytkownicy mogą podejmować bardziej świadome decyzje dotyczące zarządzania uprawami i nawadnianiem, co przyczynia się do zrównoważonego rozwoju i ochrony środowiska. Przeprowadzona analiza i porównanie danych pokazują, że z roku na rok klimat podlega powolnej, aczkolwiek zauważalnej zmianie.

Litatura

1. bydgoszcz.pl, *Miejskie stacje meteorologiczne i detektor burz*, <https://www.bydgoszcz.pl/rozwoj-i-srodowisko/inteligentny-rozwoj-miasta/przyjazne-warunki-zycia/miejskie-stacje-meteorologiczne-i-detektor-burz/>, dostęp: 12.06.2023.
2. typescriptlang.org, TypeScript, dostęp: 06.06.2023.
3. legacy.reactjs.org, React.js, dostęp: 06.06.2023.
4. redux.js.org, Redux, dostęp: 06.06.2023.
5. mui.com, MUI, dostęp: 06.06.2023. .
6. nodejs.org, <https://nodejs.org/en/about>, Node, dostęp: 06.06.2023.
7. expressjs.com, Express, dostęp: 6.06.23.
8. knexjs.org, <https://knexjs.org/guide/>, Knex, dostęp: 06.06.2023.
9. sqlite.org, <https://www.sqlite.org/about.html>, SQLite3, dostęp: 06.06.2023.
10. code.visualstudio.com, <https://code.visualstudio.com/>, VS Code, dostęp: 06.06.2023.