

Sprawozdanie z projektu

Serwis Pogodowy

Aplikacja webowa do zarządzania lokalizacjami
i przeglądania prognozy pogody

Michał Pawlik 272566, Mateusz Moskal 272596

Repozytorium GitHub:

https://github.com/MateuszMoskal/Pogoda_NET.git

Marzec – Maj 2025

Spis treści

1	Technologie	3
1.1	Czego dotyczył projekt	3
2	Opis projektu	3
2.1	Cel projektu	3
2.2	Założenia projektowe	3
3	Funkcjonalności aplikacji	4
3.1	Zarządzanie użytkownikami	4
3.1.1	Rejestracja nowych użytkowników	4
3.1.2	Logowanie	4
3.1.3	Wylogowanie	4
3.2	Zarządzanie lokalizacjami	4
3.2.1	Wyszukiwanie miast	4
3.2.2	Dodawanie miast do listy ulubionych	4
3.2.3	Usuwanie miast	4
3.3	Dane pogodowe	5
3.3.1	Aktualna pogoda	5
3.3.2	Prognoza 7-dniowa	5
4	Architektura aplikacji	5
4.1	Struktura bazy danych	5
4.1.1	Tabela Users	5
4.1.2	Tabela Cities	5
4.1.3	Tabela WeatherData	6
5	Implementacja kluczowych komponentów	6
5.1	Kontrolery	6
5.1.1	CityController	6
5.1.2	UserController	6
5.2	Serwisy	7
5.2.1	CityService	7
5.3	Repozytoria	7
5.3.1	RemoteWeatherRepository	7
6	Bezpieczeństwo	7
6.1	Autentykacja i autoryzacja	7
6.2	Walidacja danych	7
6.3	Bezpieczeństwo API	8
7	Interfejs użytkownika	8
7.1	Design	8
7.2	Responsywność	8
7.3	Interaktywność	8

7.4	Komponenty UI	8
8	API Documentation (Swagger)	9
8.1	Endpoints	9
8.1.1	User API	9
8.1.2	City API	9
9	Konfiguracja i wdrożenie	9
9.1	Wymagania systemowe	9
9.2	Konfiguracja	9
9.3	Migracje bazy danych	10
9.4	Obsługa błędów	10
10	Zrzuty ekranu aplikacji	11
10.1	Widok główny aplikacji	11
10.2	Pogoda długoterminowa i dodatkowe informacje	11

Technologie

Czego dotyczył projekt

Serwis Pogodowy – Aplikacja webowa do zarządzania lokalizacjami i przeglądania prognozy pogody

- **Backend:** ASP.NET Core MVC (.NET 9.0)
- **Frontend:** Bootstrap 5.0, Chart.js, FontAwesome
- **Baza danych:** SQL Server z Entity Framework Core
- **API:** OpenWeatherMap API
- **Dokumentacja API:** Swagger/OpenAPI

Opis projektu

Cel projektu

Celem projektu było stworzenie kompleksowej aplikacji webowej umożliwiającej użytkownikom:

- Zarządzanie ulubionymi lokalizacjami
- Przeglądanie aktualnej pogody dla wybranych miast
- Analizę 7-dniowej prognozy pogody z wizualizacją danych
- Bezpieczne przechowywanie danych użytkowników

Założenia projektowe

- Responsywny interfejs użytkownika
- Bezpieczna autentykacja i autoryzacja
- Optymalizacja zapytań do zewnętrznego API poprzez cache'owanie
- Czytelna architektura oparta na wzorcach projektowych
- Skalowalność i łatwość utrzymania kodu

Funkcjonalności aplikacji

Zarządzanie użytkownikami

Rejestracja nowych użytkowników

- Walidacja formatu email
- Wymóg hasła minimum 6 znaków
- Potwierdzenie hasła
- Sprawdzanie unikalności email

Logowanie

- Bezpieczne hashowanie hasła (SHA256)
- Zarządzanie sesją użytkownika
- Automatyczne przekierowania dla zalogowanych

Wylogowanie

- Czyszczenie sesji
- Przekierowanie do strony logowania

Zarządzanie lokalizacjami

Wyszukiwanie miast

- Integracja z OpenWeatherMap Geocoding API
- Prezentacja szczegółów: nazwa, województwo/stan, kraj, współrzędne

Dodawanie miast do listy ulubionych

- Zapisywanie w bazie danych powiązane z użytkownikiem
- Automatyczne pobieranie aktualnej pogody
- Walidacja danych geograficznych

Usuwanie miast

- Potwierdzenie przed usunięciem
- Kaskadowe usuwanie powiązanych danych pogodowych

Dane pogodowe

Aktualna pogoda

- Temperatura rzeczywista i odczuwalna
- Wilgotność powietrza
- Ciśnienie atmosferyczne
- Prędkość wiatru
- Opis warunków pogodowych
- Ikony pogodowe

Prognoza 7-dniowa

- Dane aktualizowane co 3-6 godzin w zależności od danych
- Interaktywne wykresy temperatury
- Wykresy wilgotności i ciśnienia
- Szczegółowa tabela z wszystkimi danymi

Architektura aplikacji

Struktura bazy danych

Tabela Users

```
1 CREATE TABLE Users (  
2     Id INT IDENTITY(1,1) PRIMARY KEY,  
3     Email NVARCHAR(100) NOT NULL UNIQUE,  
4     Password NVARCHAR(255) NOT NULL  
5 );
```

Listing 1: Struktura tabeli Users

Tabela Cities

```
1 CREATE TABLE Cities (  
2     Id INT IDENTITY(1,1) PRIMARY KEY,  
3     CantryCode NVARCHAR(3) NOT NULL,  
4     Voivodeship NVARCHAR(50),  
5     Name NVARCHAR(100) NOT NULL,  
6     Longitude FLOAT NOT NULL,  
7     Latitude FLOAT NOT NULL,  
8     UserId INT NOT NULL,  
9     LastUpdatedCurrent DATETIME,  
10    LastUpdatedWeekly DATETIME,
```

```
11 FOREIGN KEY (UserId) REFERENCES Users(Id)
12 );
```

Listing 2: Struktura tabeli Cities

Tabela WeatherData

```
1 CREATE TABLE WeatherData (
2     Id INT IDENTITY(1,1) PRIMARY KEY,
3     Temperature FLOAT NOT NULL,
4     FeelsLike FLOAT NOT NULL,
5     Humidity INT NOT NULL,
6     Pressure INT NOT NULL,
7     WindSpeed FLOAT NOT NULL,
8     Description NVARCHAR(255),
9     Icon NVARCHAR(50),
10    CityId INT NOT NULL,
11    LastUpdated DATETIME NOT NULL,
12    Date DATETIME NOT NULL,
13    FOREIGN KEY (CityId) REFERENCES Cities(Id)
14 );
```

Listing 3: Struktura tabeli WeatherData

Implementacja kluczowych komponentów

Kontrolery

CityController

Odpowiedzialny za zarządzanie miastami:

- IndexAsync() – wyświetla listę miast z pogodą
- AddAsync() – formularz wyszukiwania miast
- SelectCity() – wyszukiwanie miast w API
- AddCity() – dodawanie miasta do bazy
- WeatherForecast() – wyświetlanie prognozy
- DeleteCity() – usuwanie miasta

UserController

Zarządzanie użytkownikami:

- Login() – logowanie użytkownika
- Register() – rejestracja nowego konta
- Logout() – wylogowanie

Serwisy

CityService

```
1 public async Task<WheaterForecastVM> GetWeatherForWeekAsync(int cityId)
2 {
3     // 1. Pobranie miasta z bazy
4     // 2. Sprawdzenie cache
5     // 3. Pobranie z API je li potrzeba
6     // 4. Aktualizacja cache
7     // 5. Przygotowanie ViewModelu
8 }
```

Listing 4: Metoda pobierania prognozy pogody

Repozytoria

RemoteWeatherRepository

Komunikacja z OpenWeatherMap API:

- Wykorzystanie HttpClient
- Parsowanie JSON (Newtonsoft.Json)
- Obsługa błędów
- Konwersja jednostek (Kelvin → Celsius)

Bezpieczeństwo

Autentykacja i autoryzacja

- Hashowanie haseł: SHA256
- Walidacja sesji: sprawdzanie przy każdej akcji
- Ochrona CSRF: AntiForgeryToken we wszystkich formularzach

Walidacja danych

- Model Validation: DataAnnotations
- Sanityzacja: trim(), ToLowerInvariant() dla email
- Sprawdzanie zakresów: współrzędne geograficzne
- Długość stringów: ograniczenia w bazie i modelach

Bezpieczeństwo API

- **Ukrycie klucza API:** w appsettings.json
- **Limit zapytań:** cache minimalizuje wywołania
- **Obsługa błędów:** try-catch z logowaniem

Interfejs użytkownika

Design

- **Framework CSS:** Bootstrap 5
- **Ikony:** FontAwesome 6.4.0
- **Czcionki:** Inter (Google Fonts)
- **Paleta kolorów:**
 - Primary: #0066cc
 - Secondary: #00a8ff
 - Success: #51cf66
 - Danger: #ff6b6b

Responsywność

- Grid system Bootstrap
- Dostosowanie wykresów do rozmiaru ekranu

Interaktywność

- **Wykresy:** Chart.js 3.9.1
 - Wykres liniowy temperatury
 - Wykres wieloliniowy wilgotności i ciśnienia

Komponenty UI

- **Karty pogodowe:** z gradientowym tłem
- **Tabele:** z hover effect
- **Formularze:** z walidacją

API Documentation (Swagger)

Endpoints

User API

- POST /api/UserApi/login – logowanie
- POST /api/UserApi/register – rejestracja
- POST /api/UserApi/logout – wylogowanie

City API

- GET /api/CityApi – lista miast użytkownika
- POST /api/CityApi – dodanie miasta
- DELETE /api/CityApi/{id} – usunięcie miasta
- GET /api/CityApi/forecast/{id} – prognoza pogody

Konfiguracja i wdrożenie

Wymagania systemowe

- .NET 9.0 SDK
- SQL Server (LocalDB minimum)
- Visual Studio 2022
- Klucz API OpenWeatherMap

Konfiguracja

```
1 {  
2   "WeatherApi": {  
3     "ApiKey": "YOUR_API_KEY",  
4     "BaseUrl": "https://api.openweathermap.org",  
5     "CacheTimeoutMinutes": 180  
6   },  
7   "ConnectionStrings": {  
8     "DefaultConnection": "Server=YOUR_SERVER;Database=WeatherService  
9     ;..."  
10  }
```

Listing 5: Plik konfiguracyjny appsettings.json

Migracje bazy danych

Listing 6: Komendy migracji

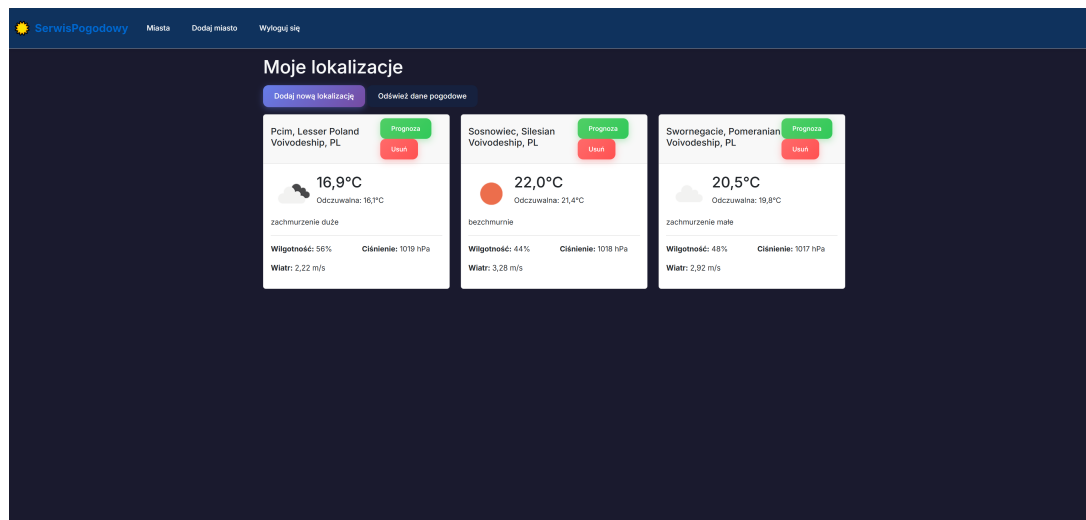
```
dotnet ef migrations add InitialCreate  
dotnet ef database update
```

Obsługa błędów

- Try-catch blocks w krytycznych miejscach
- Logging z wykorzystaniem ILogger
- User-friendly komunikaty błędów

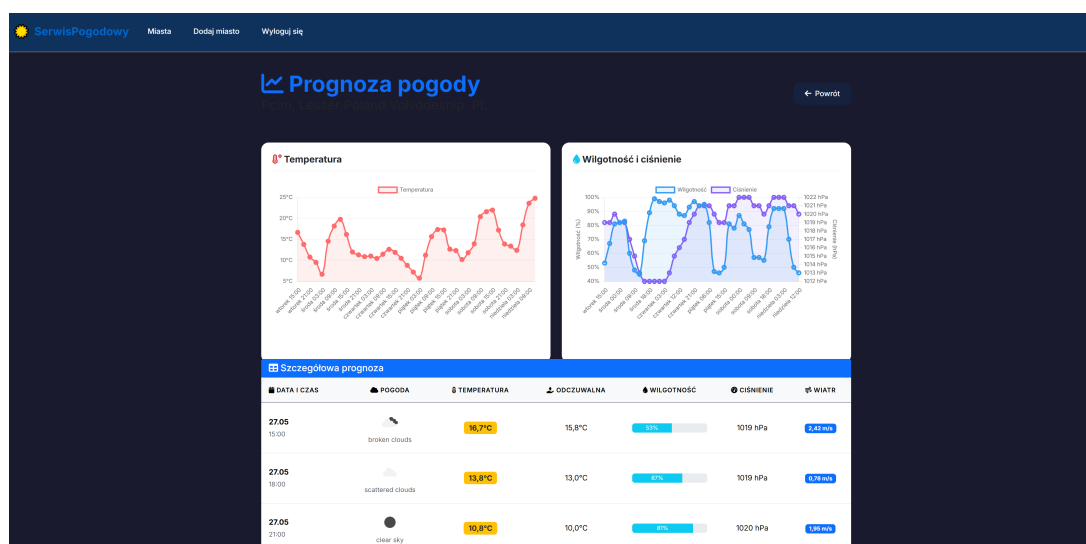
Zrzuty ekranu aplikacji

Widok główny aplikacji



Rysunek 1: Strona główna serwisu pogodowego z listą lokalizacji

Pogoda długoterminowa i dodatkowe informacje



Rysunek 2: Interfejs dodawania i edycji lokalizacji