

**UNIWERSYTET WARMIŃSKO MAZURSKI W OLSZTYNIE**  
**WYDZIAŁ MATEMATYKI I INFORMATYKI**

**Dominik Mateusz Lewczyński**

**Kierunek: Informatyka**

**Specjalność: Data Science w Praktyce**

**Implementacja aplikacji internetowej do analizy danych  
pogodowych w celu przewidzenia prognozy pogody**

Praca magisterska wykonana  
w Katedrze Metod Matematycznych  
Informatyki  
pod kierunkiem  
dr Pawła Drozdy

**Olsztyn 2024**

**UNIVERSITY OF WARMIA AND MAZURY IN OLSZTYN**  
**THE FACULTY OF MATHEMATICS AND COMPUTER**  
**SCIENCE**

**Dominik Mateusz Lewczyński**

**Field of study: Computer science**

**Specialty: Data Science in Practice**

**Implementation of a web application to analyse weather  
data to predict the weather forecast**

Master Thesis written in

Chair of Mathematical Methods of  
Computer Science

under the supervision of

dr Paweł Drozda

**Olsztyn 2024**

# Spis treści

Streszczenie.....	4
Abstract .....	5
1. Wstęp.....	6
1.1. Struktura pracy magisterskiej.....	7
2. Wymagania ogólne .....	8
2.1. Cel pracy.....	8
2.2. Podstawowe wymagania Użytkownika.....	8
3. Dokumentacja projektowa .....	9
3.1. Wymagania sprzętowe.....	9
3.2. Słownik pojęć systemowych.....	9
3.3. Aktorzy.....	9
3.4. Wymagania funkcjonalne .....	10
3.6. Diagram przypadków użycia.....	12
3.7. Diagram hierarchii funkcji .....	13
3.8. Opis funkcjonalności .....	14
4. Badania algorytmów prognostycznych w kontekście prognozowania pogody.....	21
4.1. Algorytmy Machine Learning Użyte w Badaniach.....	21
4.2. Metryki Użyte do Oceny Algorytmów.....	21
4.3. Przebieg badania.....	22
4.4. Wyniki i Analiza .....	30
4.5. Wnioski.....	31
5. Implementacja .....	32
5.1. Strategia realizacji projektu .....	32
5.2. Główne moduły aplikacji .....	33
5.3. Wzorce wykorzystane w projekcie.....	33
5.4. Frontend .....	34
5.5. Backend.....	34
5.6. Bazy danych.....	35
5.7. Testy.....	39
5.7.1. Testowanie endpointów.....	39
5.7.2. Testowanie manualne.....	39
6. Instrukcja Użytkownika .....	40
6.1. Interfejs strony głównej.....	40
6.2. Konto Użytkownika .....	43
6.3. Przeglądanie danych pogodowych.....	47
6.4. Przeprowadzenie prognozy.....	48
7. Licencje .....	54
7.1. Narzędzia.....	54
7.2. Grafiki.....	54
7.3. Biblioteki zewnętrzne.....	55
7.4. Źródła wiedzy programistycznej .....	55
8. Podsumowanie.....	56
Bibliografia.....	57

## Spis rysunków

Rysunek 1: Diagram związków encji (ERD).....	10
Rysunek 2: Diagram przypadków użycia .....	12
Rysunek 3: Diagram FHD .....	13
Rysunek 4: Porównanie algorytmów - MSE (Berlin).....	23
Rysunek 5: Porównanie algorytmów - MAE (Berlin).....	23
Rysunek 6: Porównanie algorytmów - $R^2$ (Berlin).....	24
Rysunek 7: Porównanie algorytmów - Explained Variance (Berlin) .....	24
Rysunek 8: Porównanie algorytmów - MSE (Warszawa).....	25
Rysunek 9: Porównanie algorytmów - MAE (Warszawa).....	26
Rysunek 10: Porównanie algorytmów - $R^2$ (Warszawa).....	26
Rysunek 11: Porównanie algorytmów - Explained Variance (Warszawa).....	27
Rysunek 12: Porównanie algorytmów - MSE (Olsztyn).....	28
Rysunek 13: Porównanie algorytmów - MAE (Olsztyn).....	29
Rysunek 14: Porównanie algorytmów - $R^2$ (Olsztyn) .....	29
Rysunek 15: Porównanie algorytmów - Explained Variance (Olsztyn) .....	30
Rysunek 16: Strona główna aplikacji.....	40
Rysunek 17: Strona About company .....	41
Rysunek 18: Strona About App .....	41
Rysunek 19: Strona Contact .....	42
Rysunek 20: Strona Register .....	42
Rysunek 21: Strona Login.....	43
Rysunek 22: Strona Register .....	43
Rysunek 23: Strona Register z wypełnionym formularzem.....	44
Rysunek 24: Strona Login z wypełnionym formularzem.....	44
Rysunek 25: Panel główny Użytkownika.....	45
Rysunek 26: Panel główny dla konta Admina z dodatkową sekcją nawigacji.....	46
Rysunek 27: Strona Forecast Reports z pusta lista raportów.....	46
Rysunek 28: Panel główny Użytkownika.....	47
Rysunek 29: Strona zawierająca historyczne dane pogodowe .....	48
Rysunek 30: Strona Forecast Weather przed prognozowaniem.....	48
Rysunek 31: Strona Forecast Weather z wypełnionym formularzem.....	49
Rysunek 32: Strona Forecast Weather w trakcie przeprowadzania prognozy .....	49
Rysunek 33: Wykres temperatury z przeprowadzonej prognozy.....	49
Rysunek 34: Wykres wilgotności z przeprowadzonej prognozy.....	50
Rysunek 35: Wykres opadów z przeprowadzonej prognozy .....	50
Rysunek 36: Wykres prędkości wiatru z przeprowadzonej prognozy.....	51
Rysunek 37: Wykres kierunku wiatru z przeprowadzonej prognozy.....	51
Rysunek 38: Tabela z wynikami z przeprowadzonej prognozy .....	52
Rysunek 39: Lista raportów z przeprowadzonej prognozy.....	53
Rysunek 40: Szczegóły raportu.....	53

## Spis tabel

Tabela 1: Dokumentacja przypadku użycia Autoryzacja Użytkownika .....	14
Tabela 2: Dokumentacja przypadku użycia Zarządzanie kontem Użytkowników .....	15
Tabela 3: Dokumentacja przypadku użycia Zarządzanie tabelą miejscowości .....	16
Tabela 4: Dokumentacja przypadku użycia Zarządzanie tabelą danych pogodowych ..	17
Tabela 5: Dokumentacja przypadku użycia Przeglądanie aktualnej pogody.....	18
Tabela 6: Dokumentacja przypadku użycia Przeprowadzenie prognozy .....	19
Tabela 7: Dokumentacja przypadku użycia Tworzenie raportu.....	20
Tabela 8: Porównanie Algorytmów Machine Learning na podstawie metryk ocen dla Berlina.....	22
Tabela 9: Porównanie algorytmów Machine Learning na podstawie metryk ocen dla Warszawy .....	25
Tabela 10: Porównanie algorytmów Machine Learning na podstawie metryk ocen dla Olsztyna.....	28
Tabela 11: Tabela bazy danych Sessions .....	35
Tabela 12: Tabela bazy danych Forecast .....	36
Tabela 13: Tabela bazy danych Location .....	36
Tabela 14: Tabela bazy danych Report.....	37
Tabela 15: Tabela bazy danych Users .....	37
Tabela 16: Tabela bazy danych Weatherdata .....	38

## Streszczenie

Praca przedstawia proces tworzenia opracowania pt. „Implementacja Aplikacji internetowej do analizy danych pogodowych w celu przewidzenia prognozy pogody”. Ma to być aplikacja umożliwiająca Użytkownikom przeglądanie dostępnych danych pogodowych oraz wykonanie prognozy pogody. Dane pogodowe są pobierane z zewnętrznego źródła - API (Application Programming Interface) i są przechowywane w bazie danych w celu dalszej analizy i wizualizacji.

Każdy Użytkownik, który się zarejestruje ma mieć możliwość przeglądania danych historycznych i aktualnych, a także generować prognozy pogody dla wybranych lokalizacji. Admin jest jedynym Użytkownikiem, który nie musi się rejestrować, ponieważ jest już dodany do systemu i ma możliwość zarządzania danymi pogodowymi i danymi lokalizacyjnymi oraz zarządza kontami Użytkowników.

Aplikację stworzyłem przy użyciu biblioteki React.js do frontendu i przy użyciu biblioteki Express.js oraz Flask do backendu oraz przy użyciu PostgreSQL do tworzenia bazy danych. Napisałem oraz przedstawiłem dokumentację projektową powstałą na podstawie wymagań Użytkownika oraz wstępnych założeń. Dodałem obszerną instrukcję Użytkownika wraz z odpowiednimi zrzutami ekranów. Całość wieńczy podsumowanie, mówiące o zrealizowaniu założeń wstępnych wraz ze wskazaniem możliwości dalszego rozwoju pracy.

## Abstract

The work presents the process of creating a study entitled. "Implementation of a Web Application to analyse weather data to predict the weather forecast". It is intended to be an application that allows Users to view available weather data and make a weather forecast. The weather data is taken from an external source - an API (Application Programming Interface) and is stored in a database for further analysis and visualisation.

Any User who registers is to be able to view historical and current data, as well as generate weather forecasts for selected locations. The Admin is the only User who does not need to register as they are already added to the system and have the ability to manage weather and location data and manage user accounts.

I created the application using the React.js library for the frontend and using the Express.js and Flask libraries for the backend and using PostgreSQL to create the database. I wrote and presented design documentation based on User requirements and initial assumptions. I added a comprehensive user manual with relevant screenshots. The whole project is rounded off with a summary of how the initial assumptions were met, together with an indication of how the work can be developed further. with opportunities for further development.

# 1. Wstęp

Prognoza pogody odgrywa kluczową rolę w życiu człowieka, wpływając na wiele aspektów codziennego funkcjonowania oraz kluczowe dziedziny gospodarki, takie jak rolnictwo, transport, turystyka, obrona narodowa czy lotnictwo. Zapewnienie precyzyjnych i niezawodnych prognoz jest niezbędne do podejmowania trafnych decyzji oraz skutecznego planowania działań. W miarę wzrostu zmienności warunków atmosferycznych i złożoności procesów kształtujących pogodę, zapewnienie jak najbardziej precyzyjnych prognoz staje się priorytetem.

Tradycyjne metody prognozowania pogody, oparte głównie na ręcznej analizie danych meteorologicznych oraz doświadczeniu meteorologów, stają się coraz mniej skuteczne w obliczu rosnącej ilości dostępnych danych.

Rozwój technologii, takich jak stacje meteorologiczne, satelity meteorologiczne oraz modele numeryczne, przyczynił się do wzrostu ilości dostępnych danych. Jednakże, efektywne wykorzystanie tych danych do generowania precyzyjnych prognoz wymaga zaawansowanych narzędzi i technik analizy danych.

Jednym z najbardziej obiecujących podejść w prognozowaniu pogody jest wykorzystanie sztucznej inteligencji. Algorytmy uczenia maszynowego mogą efektywnie analizować duże ilości danych pogodowych i identyfikować wzorce, które mogą być wykorzystane do generowania prognoz. Dzięki sztucznej inteligencji możliwe jest także uwzględnianie wielu zmiennych i ich skomplikowanych zależności, co pozwala na bardziej precyzyjne prognozy pogody.

Wraz z rosnącą ilością danych pogodowych dostępnych z różnych źródeł pojawia się pilna potrzeba efektywnych narzędzi do ich analizy i interpretacji. Automatyzacja przetwarzania dużych ilości danych w czasie rzeczywistym oraz generowanie dokładnych prognoz na różne okresy staje się kluczowym wyzwaniem.

W odpowiedzi na to zapotrzebowanie, aplikacje internetowe stają się obiecującym narzędziem, umożliwiając dostęp do danych pogodowych oraz generowanie prognoz za pomocą intuicyjnych interfejsów Użytkownika.



## 1.1. Struktura pracy magisterskiej

Niniejsza praca magisterska opisuje aplikację internetową stworzoną za pomocą technologii React.js, Express.js, Flask, jej możliwości, budowę, projekt oraz zakres działania. Wszystko to opisane jest w następujących rozdziałach:

- Wymagania ogólne – podstawowe założenia powstającej aplikacji,
  - Cel pracy – określono cel powstania aplikacji,
  - Podstawowe wymagania Użytkownika – przedstawienie podstawowych wymagań Użytkownika wobec działania aplikacji,
- Dokumentacja projektowa – część poświęcona dokumentacji projektowej
  - Wymagania sprzętowe – opis wymagań dotyczących użytkowania programu,
  - Słownik pojęć systemowych – opis poszczególnych pojęć systemowych
  - Aktorzy – opis poszczególnych Użytkowników, występujących w systemie,
  - Wymagania funkcjonalne – przedstawienie głównych założeń programu,
  - Struktura bazy danych – część poświęcona diagramowi związków encji (ERD)
  - Diagram przypadków użycia – część poświęcona przedstawieniu interakcji między aktorem a systemem,
  - Diagram hierarchii funkcji – część poświęcona funkcjonalnościom przedstawionym na diagramie hierarchii funkcji,
  - Opis funkcjonalności – część poświęcona szczegółowemu opisaniu wszystkich funkcjonalności.
- Instrukcja Użytkownika – opis działania programu oraz jego użytkowanie,
  - Interfejs strony głównej – część poświęcona przedstawieniu strony głównej i innych stron,
  - Konto Użytkownika – czynności związane z kontem Użytkownika,
  - Przeglądanie danych pogodowych – część poświęcona przeglądaniu danych pogodowych po zalogowaniu do konta Użytkownika,
  - Przeprowadzenie prognozy – prognozowanie pogody dla wybranej miejscowości na wybrane przez Użytkownika dni.
- Licencje – opis licencji dotyczących technologii oraz elementów wykorzystanych w programie,
  - Narzędzia – lista narzędzi wykorzystanych w pracy
  - Grafika – lista linków do grafik wykorzystanych w pracy
  - Biblioteki zewnętrzne – lista bibliotek wykorzystanych w pracy
  - Źródła wiedzy programistycznej – lista adresów jako źródła wiedzy podczas tworzenia aplikacji
- Podsumowanie – bilans pracy, przedstawienie wniosków oraz oszacowanie.
- Bibliografia – Lista materiałów wykorzystanych w pracy magisterskiej.

## **2. Wymagania ogólne**

Celem niniejszego rozdziału jest przedstawienie, jakie były początkowe założenia oraz wymagania Użytkownika docelowego. W podrozdziale 5.1 opisuję, dlaczego tworzona jest aplikacja. W podrozdziale 5.2 wyjaśniam, jakie były wymagania Użytkownika wobec tworzonej aplikacji.

### **2.1. Cel pracy**

Celem niniejszej pracy magisterskiej jest opracowanie aplikacji internetowej, która umożliwia analizę danych pogodowych w celu przewidzenia prognozy pogody. W dobie szybkiego rozwoju technologii informacyjnych i wzrostu dostępności danych meteorologicznych, istnieje coraz większe zapotrzebowanie na narzędzia umożliwiające skuteczną analizę tych danych w celu prognozowania warunków atmosferycznych.

Głównym celem aplikacji będzie zapewnienie Użytkownikom możliwości dostępu do aktualnych i historycznych danych pogodowych, a także ich analizy za pomocą zaawansowanych algorytmów przetwarzania danych.

### **2.2. Podstawowe wymagania Użytkownika**

Poniżej przedstawiam wymagania Użytkownika wobec tworzonej aplikacji internetowej:

- Najnowsza wersja przeglądarki internetowej do poprawnego działania aplikacji. Dostępne przeglądarki to np.: Google Chrome, Mozilla Firefox, Microsoft Edge oraz Opera.
- Komputer, na którym powinna działać aplikacja powinien spełniać minimalne wymagania. (procesor Pentium 4, pamięć RAM 4GB, system operacyjny Windows XP, dysk twardy 20GB)
- Połączenie z Internetem jest wymagane, ponieważ serwer musi stać na szybkim i stabilnym łączu.
- Aplikacja ma być intuicyjna oraz estetyczna. Użytkownik nie powinien mieć problemu z korzystaniem z aplikacji.
- Aplikacja musi być odpowiednio zabezpieczona by chronić dane Użytkowników.
- Aplikacja powinna działać na dużych jak i mniejszych ekranach.
- Aplikacja musi być kompatybilna z innymi systemami IT.
- Aplikacja powinna być wydajna oraz powinna działać szybko.

### 3. Dokumentacja projektowa

#### 3.1. Wymagania sprzętowe

Wymagania aplikacji internetowej do analizy danych pogodowych w celu przewidzenia prognozy pogody dotyczące sprzętu wyglądają następująco:

- Komputer lub urządzenie mobilne z dostępem do Internetu,
- Połączenie internetowe: Dowolne (Wi-Fi lub transfer sieci komórkowej),
- Zainstalowana przeglądarka internetowa: Do wyboru jest Google Chrome, Mozilla Firefox, Microsoft Edge, Safari oraz Opera).

Technologie jakich użyto do stworzenia aplikacji zapewniają działanie w każdej przeglądarce na każdym urządzeniu. Aplikacja jest także responsywna przez co jest dostosowana do ekranu o różnej wielkości.

#### 3.2. Słownik pojęć systemowych:

- **Admin** = imię + nazwisko + dane Użytkownika + status Użytkownika. Administrator systemu.
- **Użytkownik** = adres e-mail + hasło + status Użytkownika. Osoba podlegająca autoryzacji w systemie.
- **Pogoda** = Miejscowość + Dane pogodowe + Przyszłe daty. Aktualne dane pogodowe
- **Prognoza pogody** = Miejscowość + Dane pogodowe + Przyszłe daty. Dane pogodowe na najbliższe dni
- **Raport** = Miejscowość + Prognoza pogody. Prognoza pogody w formie dokumentu do pobrania

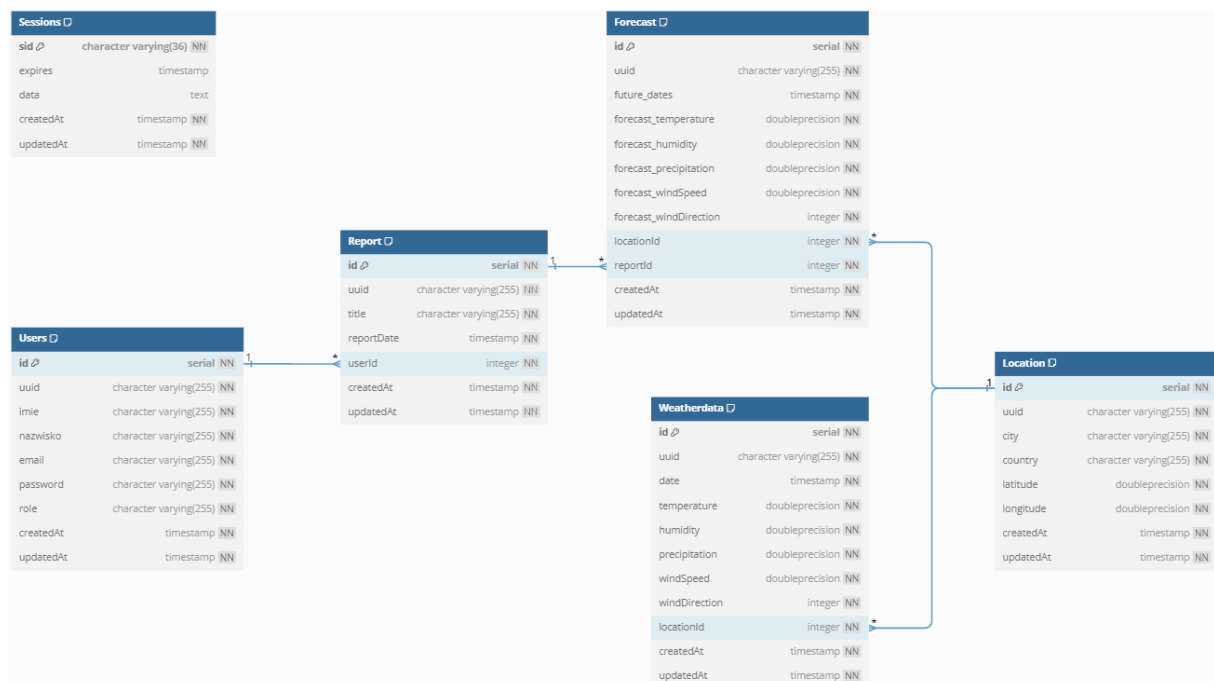
#### 3.3. Aktorzy

- **Admin** – Użytkownik systemu zarządzający danymi innych Użytkowników oraz zarządzający danymi lokalizacyjnymi oraz danymi pogodowymi
- **Użytkownik** – Podlega autoryzacji w tym systemie, przegląda dane pogodowe oraz przeprowadza prognozę pogody

### 3.4. Wymagania funkcjonalne

- **Autoryzuj Użytkownika** - zbiór usług pozwalających na zarządzanie kontem oraz dostęp do niego.
  - **Zarejestruj się** - usługa pozwalająca Użytkownikowi utworzyć nowe konto.
  - **Zaloguj się** - usługa pozwalająca, po podaniu odpowiednich danych, uzyskać dostęp do konta danego Użytkownika lub Admina.
- **Zarządzanie kontem Użytkowników** – usługa pozwalająca Adminowi na zarządzanie kontem Użytkowników.
- **Zarządzanie tabelą miejscowości** – usługa pozwalająca Adminowi na Zarządzanie tabelą miejscowości poprzez ich dodawanie oraz usuwanie.
- **Zarządzanie tabelą danych pogodowych** – usługa pozwalająca Adminowi na zarządzanie tabelą danych pogodowych poprzez aktualizowanie tabeli o nowe dane.
- **Przeglądanie aktualnej pogody** – usługa pozwalająca przejrzanie aktualnej pogody dla wybranej miejscowości.
- **Przeprowadzenie prognozy** – usługa pozwalająca Użytkownikowi na uzyskanie prognozy pogody dla wybranej miejscowości na dane dni.
- **Tworzenie raportu** – Usługa pozwalająca Użytkownikowi na utworzenie raportu po dokonaniu prognozy.

### 3.5. Struktura bazy danych



Rysunek 1: Diagram związków encji (ERD)

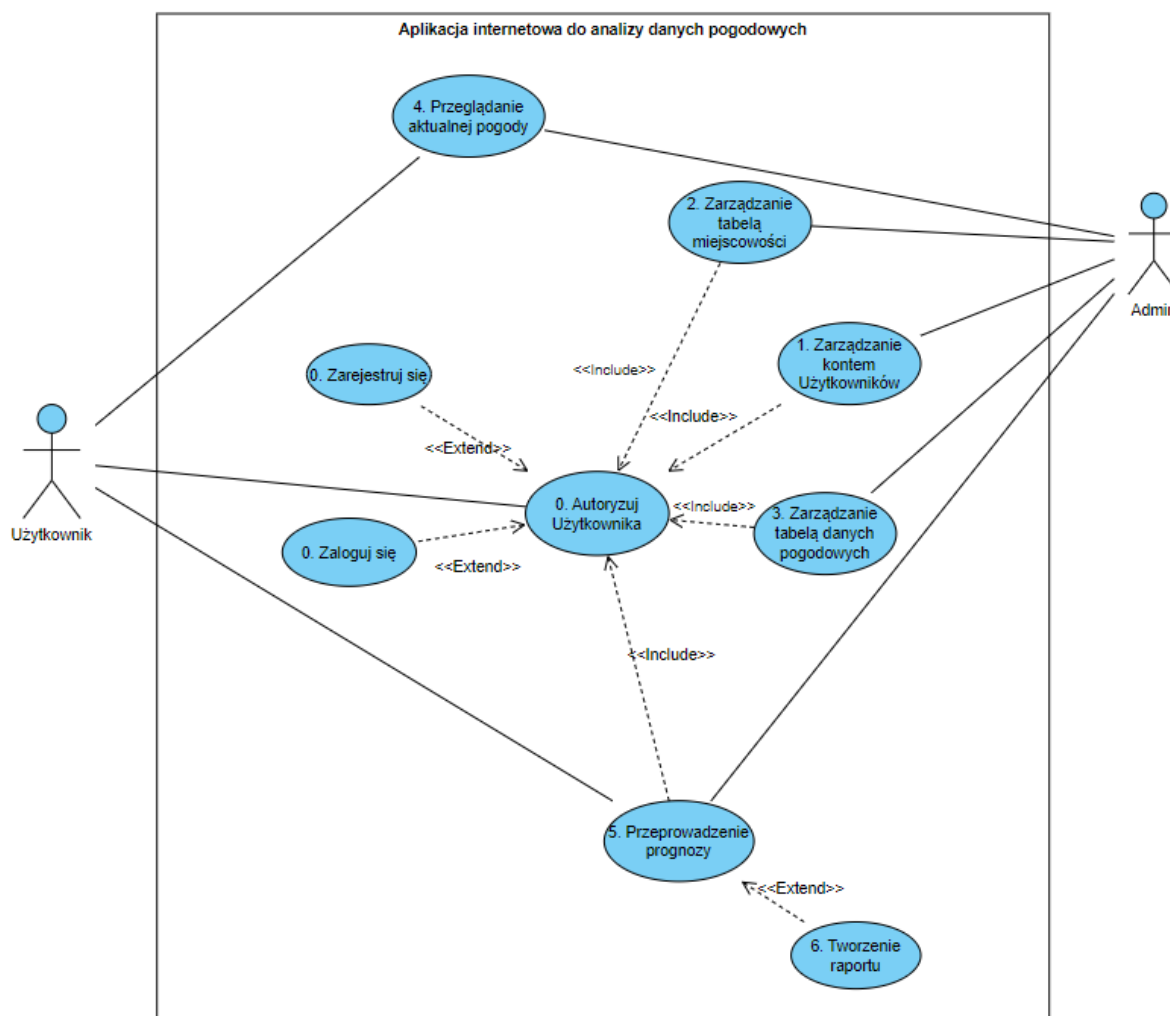
Na powyższym diagramie została przedstawiona struktura bazy danych, w której widać utworzone tabele oraz połączenia między nimi. Na następnej stronie znajdują się krótkie opisy tabel oraz relacji między nimi:

- **Tabela Sessions:** Przechowuje informacje o sesjach Użytkowników, takie jak dane sesji, czas wygaśnięcia oraz daty utworzenia i aktualizacji.
- **Tabela Forecast:** Zawiera prognozy pogodowe, w tym temperaturę, wilgotność, opady, prędkość i kierunek wiatru. Jest powiązana z tabelami location i report.
- **Tabela Location:** Przechowuje dane lokalizacji, takie jak miasto, kraj oraz współrzędne geograficzne.
- **Tabela Report:** Zawiera raporty Użytkowników, w tym tytuł raportu, datę i identyfikator Użytkownika, który go stworzył.
- **Tabela Users;** Przechowuje informacje o Użytkownikach aplikacji, takie jak imię, nazwisko, email, hasło i rola.
- **Tabela Weatherdata:** Zawiera dane pogodowe dla różnych lokalizacji, w tym temperaturę, wilgotność, opady, prędkość i kierunek wiatru. Jest powiązana z tabelą Location.

#### **Relacje między tabelami**

- **Users ↔ Report:** Jeden Użytkownik może mieć wiele raportów (relacja jeden do wielu).
- **Location ↔ Weatherdata:** Jedna lokalizacja może mieć wiele danych pogodowych (relacja jeden do wielu).
- **Location ↔ Forecast:** Jedna lokalizacja może mieć wiele prognoz (relacja jeden do wielu).
- **Report ↔ Forecast:** Jeden raport może zawierać wiele prognoz (relacja jeden do wielu).

### 3.6. Diagram przypadków użycia



Rysunek 2: Diagram przypadków użycia

Na powyższym diagramie przypadków użycia została przedstawiona funkcjonalność aplikacji, z której mogą korzystać Użytkownicy. Widać jakie funkcjonalności są przypisane do Użytkownika, a jakie do Admina.

### 3.7. Diagram hierarchii funkcji



Rysunek 3: Diagram FHD

Na powyższym diagramie została umiejscowiona hierarchia poszczególnych funkcjonalności i dodana do odpowiednich Aktorów.

### 3.8. Opis funkcjonalności

W tym podrozdziale opisze szczegółowo wszystkie funkcjonalności w postaci scenariuszy.

*Tabela 1: Dokumentacja przypadku użycia Autoryzacja Użytkownika*

Nazwa	Autoryzacja Użytkownika
Numer	0
Twórca	Dominik Lewczyński
Poziom ważności	Wysoki
Typ przypadku użycia	Ogólny, niezbędny
Aktorzy	Admin, Użytkownik
Krótki opis	Przeprowadzenie autoryzacji Użytkownika
Warunki wstępne	Dostęp do systemu informatycznego
Warunki końcowe	Zalogowanie lub zarejestrowanie się do systemu
Główny przepływ zdarzeń	1) Użytkownik loguje się do systemu. Podaje e-mail oraz hasło. Jeżeli dane są poprawne to Użytkownik zostaje zalogowany. 2) Użytkownik rejestruje się do systemu. Podaje imię, nazwisko oraz e-mail i hasło. Jeżeli dane zostały podane prawidłowo to Użytkownik zostaje zarejestrowany.
Alternatywne przepływy zdarzeń	1a) Błędne dane – system wyświetla błąd. Użytkownik nie jest zalogowany do systemu. System ponawia czynność logowania 2a) Błędne dane – system wyświetla błąd. Konto Użytkownik nie zostaje utworzone. System ponawia czynność rejestracji
Wyjątki wymagania	Brak
Specjalne wymagania	Brak
Notatki i kwestie	Brak



Tabela 2: Dokumentacja przypadku użycia Zarządzanie kontem Użytkowników

Nazwa	Zarządzanie kontem Użytkowników
Numer	1
Twórca	Dominik Lewczyński
Poziom ważności	Wysoki
Typ przypadku użycia	Ogólny, niezbędny
Aktorzy	Admin
Krótki opis	Funkcja pozwalająca na zarządzanie kontem Użytkowników (zmianę hasła, danych itp.)
Warunki wstępne	Admin jest zalogowany
Warunki końcowe	Admin zarządził danymi Użytkowników
Główny przepływ zdarzeń	<ol style="list-style-type: none"> <li>1) Admin przechodzi do strony „Users”</li> <li>2) Po wybraniu opcji Adminowi zostanie wyświetlona lista Użytkowników</li> <li>3) Admin ma możliwość zmiany emaila hasła oraz innych danych, a także możliwość usunięcia Użytkownika</li> <li>4) Admin edytuje dane Użytkowników. Wybiera opcję „Edit” przy wybranym Użytkowniku. Wypełnia w formularzu pola jakie chce zaktualizować, a potem klika opcję „Save”.</li> <li>5) Admin usuwa Użytkownika. Klika przycisk „Delete” przy wybranym Użytkowniku.</li> </ol>
Alternatywne przepływy zdarzeń	<ol style="list-style-type: none"> <li>3a) Admin nie musi zmieniać danych. Może w każdej chwili opuścić stronę zarządzania kontem.</li> <li>4a) Admin może też kliknąć opcję „Anuluj” co anuluje wprowadzone zmiany.</li> <li>5a) Admin nie musi usuwać Użytkownika. Może w każdej chwili opuścić stronę zarządzania kontem.</li> </ol>
Wyjątki wymagania	Brak
Specjalne wymagania	Brak
Notatki i kwestie	Brak

Tabela 3: Dokumentacja przypadku użycia Zarządzanie tabelą miejscowości

Nazwa	Zarządzanie tabelą miejscowości
Numer	2
Twórca	Dominik Lewczyński
Poziom ważności	Wysoki
Typ przypadku użycia	Ogólny, niezbędny
Aktorzy	Admin
Krótki opis	Funkcja pozwalająca na zarządzanie tabelą miejscowości (dodawanie, usuwanie)
Warunki wstępne	Admin jest zalogowany
Warunki końcowe	Admin zarządził tabelą miejscowości
Główny przepływ zdarzeń	<ol style="list-style-type: none"> <li>1) Admin przechodzi do strony „Locations”</li> <li>2) Admin ma możliwość dodawania nowej miejscowości oraz możliwość usunięcia wybranej miejscowości</li> <li>3) Admin, kiedy chce dodać nową miejscowość wpisuje w formularz jej nazwę i zatwierdza przyciskiem</li> <li>4) Po odczekaniu nowa lokalizacja zostaje dodana do tabeli Locations oraz dane pogodowe do tabeli „Weather data”.</li> <li>5) Admin, kiedy chce usunąć miejscowość naciska przycisk „Delete” przy wybranej lokalizacji.</li> <li>6) Po naciśnięciu dane pogodowe z tabeli Weather data oraz dane lokalizacyjne zostają usunięte.</li> </ol>
Alternatywne przepływy zdarzeń	<p>3a) Admin nie musi dodawać nowej lokalizacji. Może w każdej chwili opuścić stronę zarządzania tabelą miejscowości.</p> <p>5a) Admin nie musi usuwać lokalizacji. Może w każdej chwili opuścić stronę zarządzania tabelą miejscowości.</p>
Wyjątki wymagania	Brak
Specjalne wymagania	Brak
Notatki i kwestie	Brak

Tabela 4: Dokumentacja przypadku użycia Zarządzanie tabelą danych pogodowych

Nazwa	Zarządzanie tabelą danych pogodowych
Numer	3
Twórca	Dominik Lewczyński
Poziom ważności	Wysoki
Typ przypadku użycia	Ogólny, niezbędny
Aktorzy	Admin
Krótki opis	Funkcja pozwalająca na zarządzanie tabelą danych pogodowych (odświeżenie)
Warunki wstępne	Admin jest zalogowany
Warunki końcowe	Admin zarządził tabelą danych pogodowych
Główny przepływ zdarzeń	1) Admin przechodzi do strony „Weather data” 2) Admin ma możliwość odświeżenia danych w celu pobrania najnowszych danych pogodowych 3) Admin, Kiedy chce odświeżyć dane naciska na przycisk „Refresh Data”. 4) Po odczekaniu dane pogodowe są zaktualizowane
Alternatywne przepływy zdarzeń	3a) Admin nie musi odświeżać danych. Może w każdej chwili opuścić stronę zarządzania tabelą danych pogodowych.
Wyjątki wymagania	Brak
Specjalne wymagania	Brak
Notatki i kwestie	Brak

Tabela 5: Dokumentacja przypadku użycia Przeglądanie aktualnej pogody

Nazwa	Przeglądanie aktualnej pogody
Numer	4
Twórca	Dominik Lewczyński
Poziom ważności	Wysoki
Typ przypadku użycia	Ogólny, niezbędny
Aktorzy	Admin, Użytkownik
Krótki opis	Przegląd pogody dla wybranej miejscowości
Warunki wstępne	Użytkownik przegląda listę publikacji
Warunki końcowe	Użytkownik po wybraniu publikacji może ją przeczytać
Główny przepływ zdarzeń	1) Użytkownik na stronie głównej ma wyświetloną listą miejscowości. 2) Użytkownik ma możliwość wybrania nazwy miejscowości by zobaczyć aktualną dla niej pogodę. 3) Po wybraniu nazwy miejscowości Użytkownikowi pokazuje się aktualna pogoda dla wybranej miejscowości.
Alternatywne przepływy zdarzeń	Brak
Wyjątki wymagania	Brak
Specjalne wymagania	Brak
Notatki i kwestie	Użytkownik może w każdej chwili opuścić stronę

Tabela 6: Dokumentacja przypadku użycia Przeprowadzenie prognozy

Nazwa	Przeprowadzenie prognozy
Numer	5
Twórca	Dominik Lewczyński
Poziom ważności	Wysoki
Typ przypadku użycia	Ogólny, niezbędny
Aktorzy	Użytkownik, Admin
Krótki opis	Przeprowadzenie prognozy pogody dla wybranej miejscowości
Warunki wstępne	Użytkownik musi być zalogowany
Warunki końcowe	Użytkownik otrzymał prognozę pogody
Główny przepływ zdarzeń	<p>1) Użytkownik po zalogowaniu na stronie głównej wybiera opcje „Forecast Weather”</p> <p>2) Po wybraniu opcji Użytkownik jest przekierowany do formularza, w którym wybiera nazwę miejscowości oraz na ile dni ma być prognoza. Następnie klika przycisk „Predict forecast weather”.</p> <p>3) Po wypełnieniu formularza system przeprowadzi prognozę. Po jej przeprowadzeniu Użytkownik otrzymuje wynik w postaci danych pogodowych na najbliższe wybrane dni</p> <p>4) Wyniki z prognozy będą automatycznie zapisane na koncie Użytkownika w formie raportu do wglądu</p>
Alternatywne przepływy zdarzeń	2a) Jeśli formularz nie zostanie wypełniony w całości przycisk „Predict forecast weather” nie będzie aktywny.
Wyjątki wymagania	Brak
Specjalne wymagania	Brak
Notatki i kwestie	Brak

Tabela 7: Dokumentacja przypadku użycia Tworzenie raportu

Nazwa	Tworzenie raportu
Numer	6
Twórca	Dominik Lewczyński
Poziom ważności	Wysoki
Typ przypadku użycia	Ogólny, niezbędny
Aktorzy	Użytkownik, Admin
Krótki opis	Wygenerowanie raportu po przeprowadzeniu prognozy
Warunki wstępne	Prognoza pogody została wykonana
Warunki końcowe	Raport został wygenerowany
Główny przepływ zdarzeń	<p>1) Po przeprowadzeniu prognozy Użytkownikowi pokazują się wyniki działania systemu z dostępnym przyciskiem wygeneruj raport.</p> <p>2) Po kliknięciu w przycisk „Download report” raport, zawierający dane z przeprowadzonej prognozy, zostanie pobrany na urządzenie w formie pliku PDF. Użytkownik może mieć możliwość wglądu do raportu po pobraniu na urządzenie.</p>
Alternatywne przepływy zdarzeń	Brak
Wyjątki wymagania	Brak
Specjalne wymagania	Brak
Notatki i kwestie	Użytkownik nie musi pobierać raportu. Może w każdej chwili opuścić stronę z wynikami

## 4. Badania algorytmów prognostycznych w kontekście prognozowania pogody

Prognozowanie pogody jest jednym z największych wyzwań stojących przed nauką i technologią. W celu osiągnięcia jak najlepszej dokładności, wykorzystuje się zaawansowane metody uczenia maszynowego (Machine Learning - ML). W niniejszym rozdziale przedstawiono badania przeprowadzone w celu wyłonienia najlepszego algorytmu ML do prognozowania temperatury. Analizie poddano cztery popularne algorytmy: Random Forest Regressor, Linear Regression, Support Vector Machine (SVM), oraz Gradient Boosting Regressor. Oceniono ich wydajność za pomocą metryk takich jak Mean Squared Error (MSE), Mean Absolute Error (MAE),  $R^2$  oraz Explained Variance.

### 4.1. Algorytmy Machine Learning Użyte w Badaniach

- **Random Forest Regressor** to metoda ensemble learning, która tworzy wiele drzew decyzyjnych podczas trenowania i zwraca średnią prognozę z tych drzew. Jest wysoce odporna na przeuczenie i skuteczna w radzeniu sobie z dużymi zbiorami danych oraz wieloma cechami wejściowymi.
- **Linear Regression** to jedna z najprostszych i najbardziej intuicyjnych metod regresji. Model ten zakłada liniową zależność między zmienną zależną a jedną lub więcej zmiennymi niezależnymi. Pomimo swojej prostoty, jest często używany jako punkt odniesienia dla bardziej złożonych algorytmów.
- **Support Vector Machine (SVM)** to wszechstronny algorytm, który może być stosowany do klasyfikacji i regresji. Jego głównym celem jest znalezienie optymalnego hiperplanu, który maksymalizuje odległość (margines) między różnymi klasami danych, minimalizując ryzyko błędnej klasyfikacji.
- **Gradient Boosting Regressor** to metoda, która tworzy model predykcyjny poprzez łączenie wielu słabych modeli (zwykle drzew decyzyjnych) w silny model. Każde kolejne drzewo stara się skorygować błędy popełnione przez poprzednie drzewa, co prowadzi do bardzo dokładnych prognoz.

### 4.2. Metryki Użyte do Oceny Algorytmów

- **Mean Squared Error (MSE, Średni błąd kwadratowy):** Jest to średnia arytmetyczna kwadratów różnic między wartościami rzeczywistymi a przewidywanymi przez model. Im mniejsza wartość MSE, tym lepszy model.
- **Mean Absolute Error (MAE, Średni błąd bezwzględny):** To średnia arytmetyczna bezwzględnych różnic między wartościami rzeczywistymi a przewidywanymi. Mniejsza wartość MAE oznacza lepszą dokładność modelu.
- **$R^2$  (Współczynnik Determinacji):** Mierzy, jak dobrze przewidywane wartości odpowiadają rzeczywistym wartościom. Wartość  $R^2$  bliska 1 oznacza, że model dobrze wyjaśnia wariancję danych.
- **Explained Variance (Wyjaśniona wariancja):** Wskaźnik ten mierzy proporcję całkowitej wariancji wyjaśnionej przez model. Wyższa wartość wskazuje na lepszą wydajność modelu.

### 4.3. Przebieg badania

W celu oceny dokładności prognoz, modele trenowano i testowano na zbiorze danych pogodowych pochodzących z zewnętrznego API Open-Meteo, a następnie zapisanych w bazie danych PostgreSQL dla trzech miast: Berlin, Warszawa i Olsztyn. Proces obejmował następujące kroki:

1. **Zbieranie danych:**
  - Dane pogodowe zostały pobrane z API Open-Meteo, która dostarcza szczegółowe informacje meteorologiczne.
  - Zebrane dane obejmowały różne zmienne pogodowe, takie jak temperatura, wilgotność, prędkość wiatru i inne, zapisane w bazie danych PostgreSQL.
2. **Przetwarzanie danych:**
  - Przeprowadzono czyszczenie i wstępne przetwarzanie danych, aby usunąć brakujące wartości.
  - Dane zostały znormalizowane i podzielone na zestawy treningowe i testowe w stosunku 80:20.
3. **Trenowanie modeli:**
  - Algorytmy Machine Learning (Random Forest Regressor, Linear Regression, Support Vector Machine, Gradient Boosting Regressor) zostały zaimplementowane i wytrenowane na zestawie treningowym.
  - Każdy model został dostosowany i zoptymalizowany pod kątem hiperparametrów, aby uzyskać jak najlepsze wyniki.
4. **Ocena modeli:**
  - Modele zostały ocenione na zestawie testowym przy użyciu metryk takich jak Mean Squared Error (MSE), Mean Absolute Error (MAE),  $R^2$  oraz Explained Variance.

Wyniki badań dla wybranych miast przedstawiono poniżej w formie tabel oraz wykresów:

### Wyniki dla Berlina

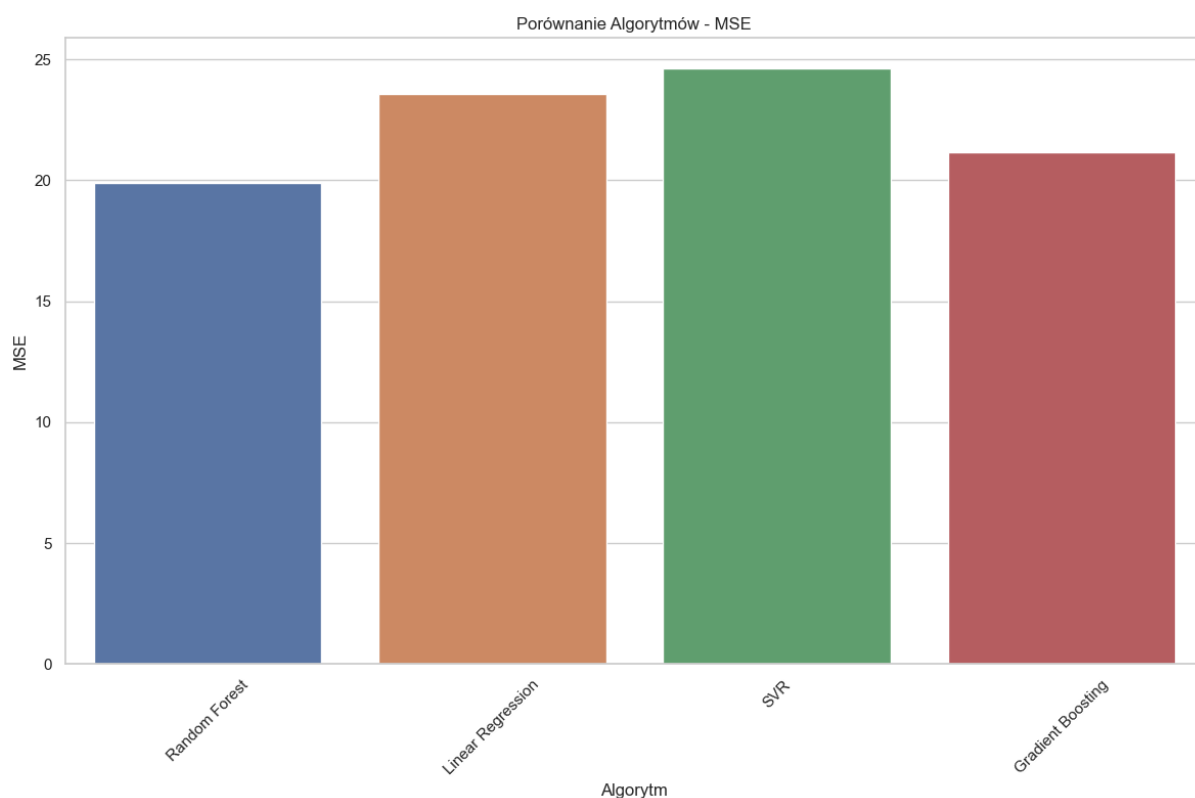
#### Porównanie algorytmów Machine Learning na podstawie metryk ocen dla Berlina

Tabela 8: Porównanie Algorytmów Machine Learning na podstawie metryk ocen dla Berlina

Algorytm	MSE	MAE	$R^2$	Explained Variance
Random Forest	19.893195	3.374996	0.444673	0.445605
Linear Regression	23.581893	3.758851	0.341702	0.347215
SVR	24.639876	3.751111	0.312168	0.346399
Gradient Boosting	21.134548	3.499016	0.410021	0.413198

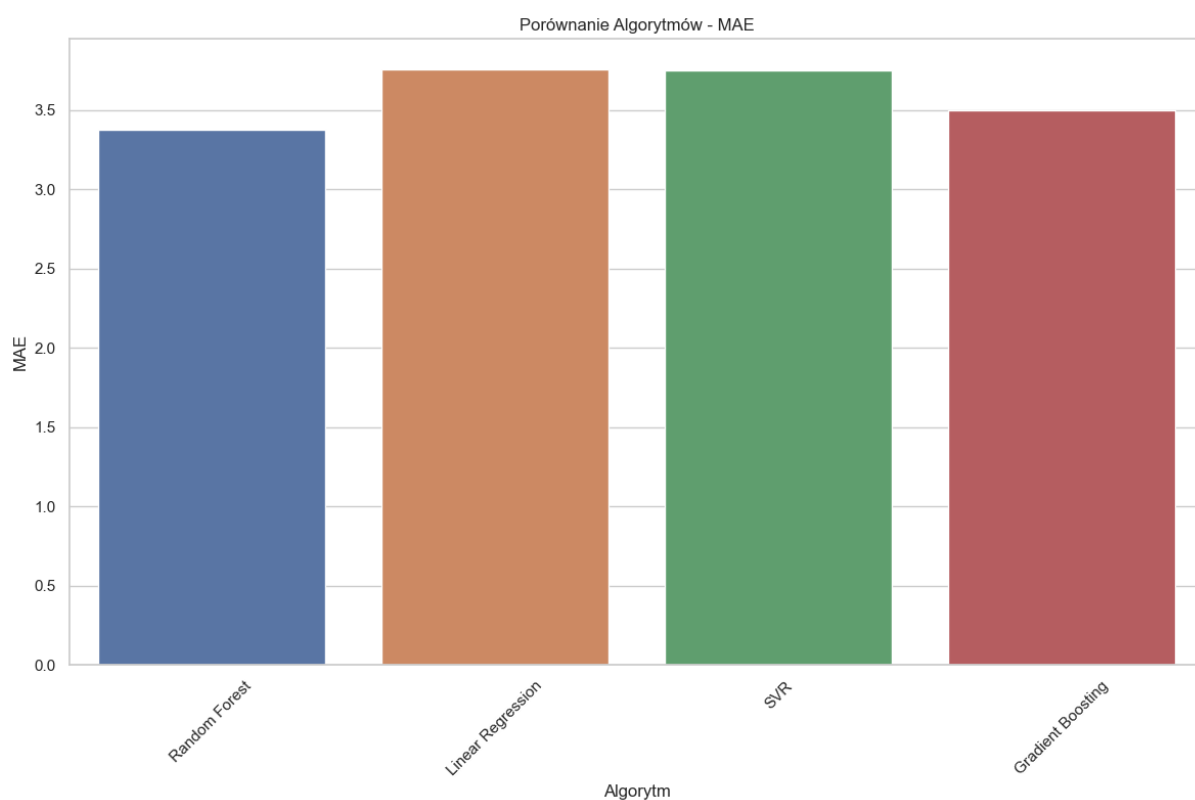


## Porównanie algorytmów na podstawie MSE (Berlin)



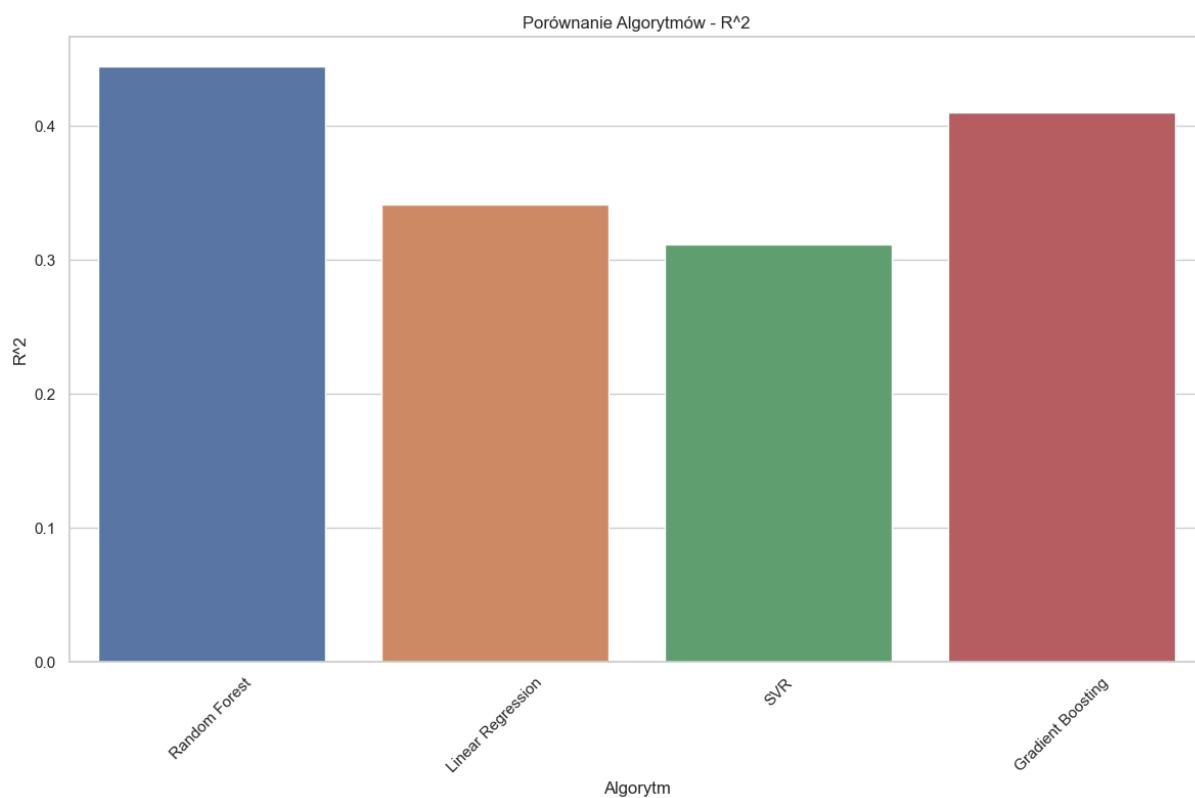
Rysunek 4: Porównanie algorytmów - MSE (Berlin)

## Porównanie algorytmów na podstawie MAE (Berlin)



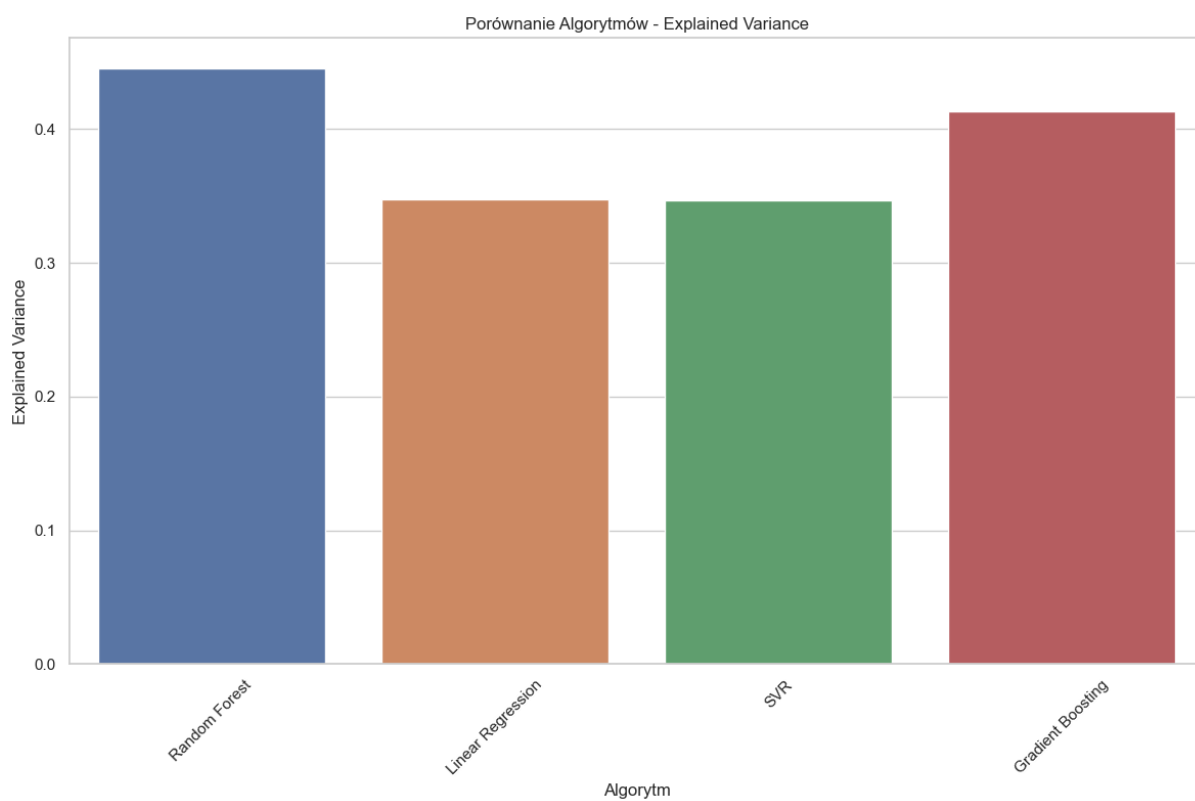
Rysunek 5: Porównanie algorytmów - MAE (Berlin)

## Porównanie algorytmów na podstawie $R^2$ (Berlin)



Rysunek 6: Porównanie algorytmów -  $R^2$  (Berlin)

## Porównanie algorytmów na podstawie Explained Variance (Berlin)



Rysunek 7: Porównanie algorytmów - Explained Variance (Berlin)

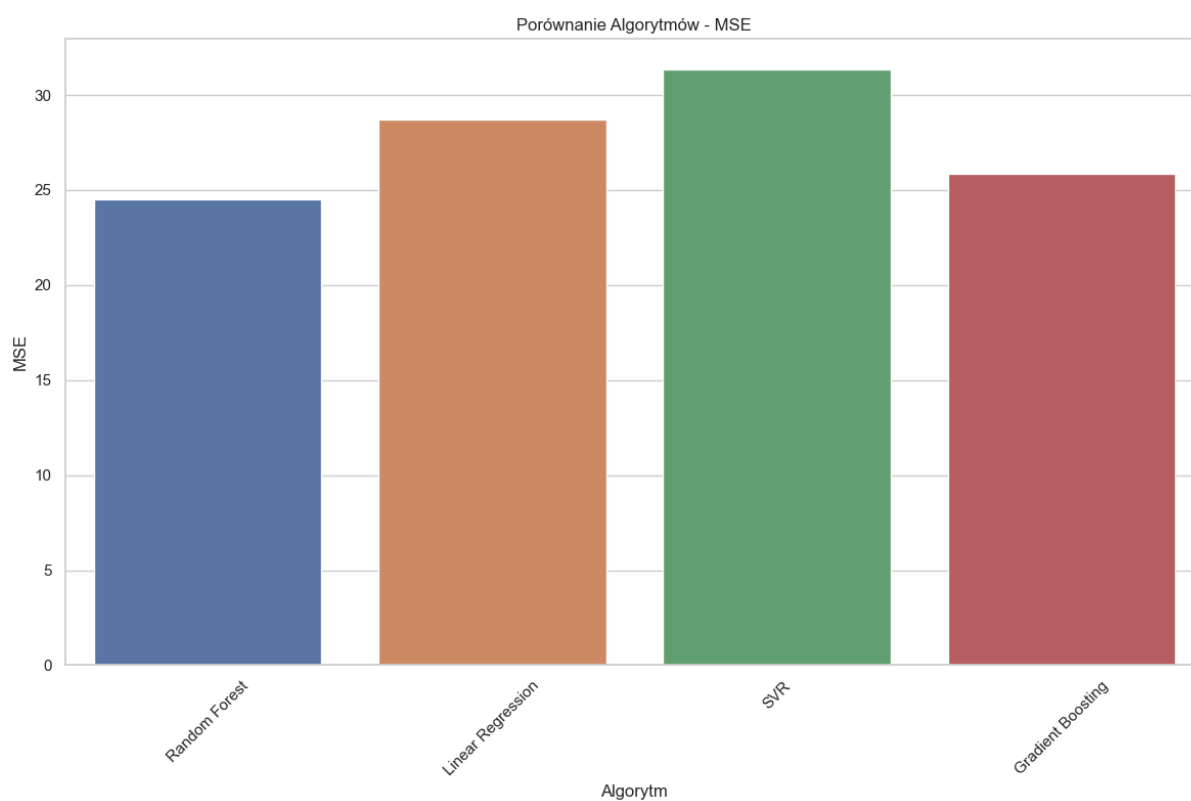
## Wyniki dla Warszawy

### Porównanie algorytmów Machine Learning na podstawie metryk ocen dla Warszawy

Tabela 9: Porównanie algorytmów Machine Learning na podstawie metryk ocen dla Warszawy

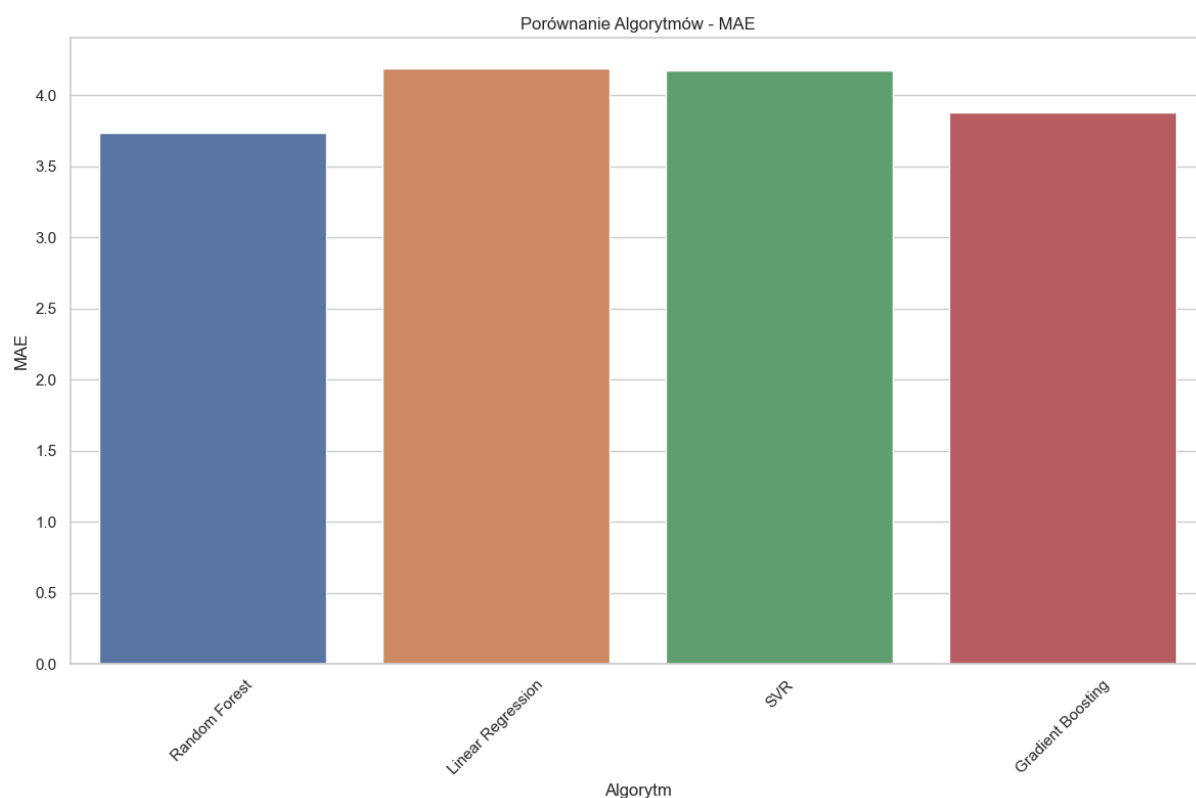
Algorytm	MSE	MAE	R <sup>2</sup>	Explained Variance
Random Forest	24.529275	3.732797	0.421845	0.423000
Linear Regression	28.701884	4.191712	0.323496	0.332168
SVR	31.370282	4.176520	0.260602	0.320531
Gradient Boosting	25.832121	3.878793	0.391137	0.397730

### Porównanie algorytmów na podstawie MSE (Warszawa)



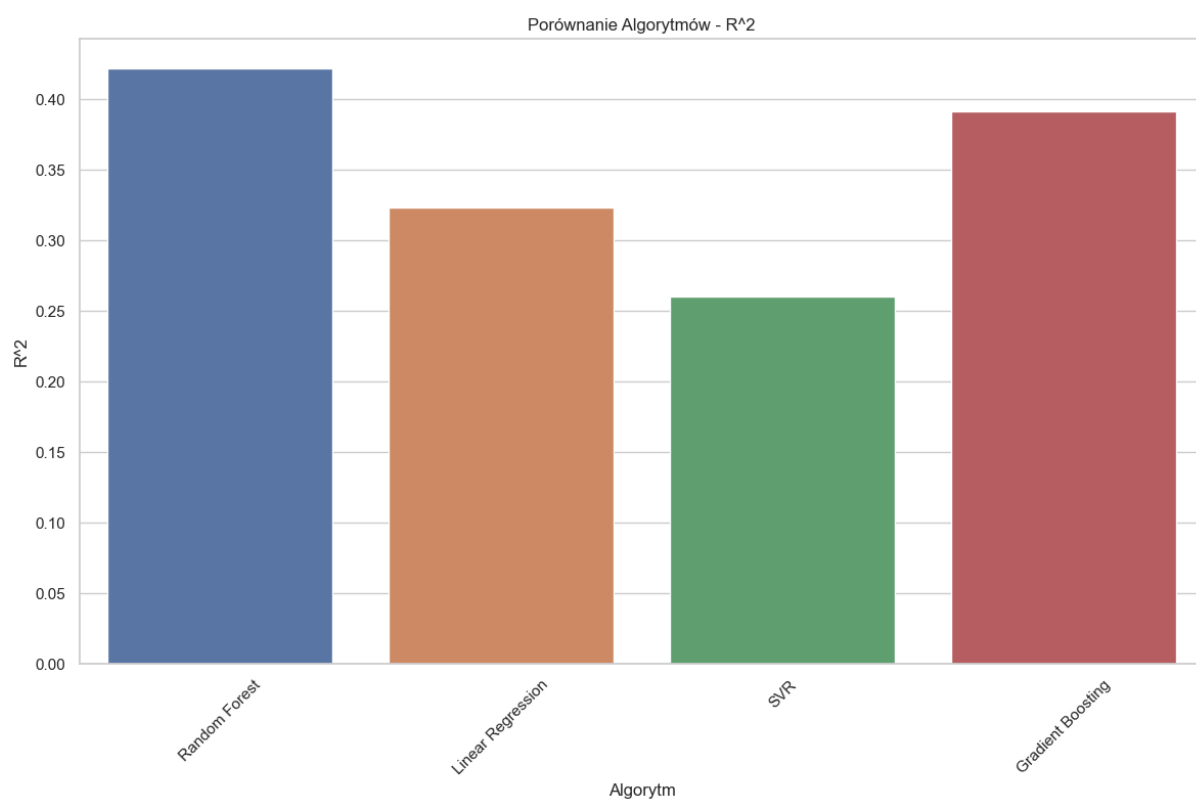
Rysunek 8: Porównanie algorytmów - MSE (Warszawa)

## Porównanie algorytmów na podstawie MAE (Warszawa)



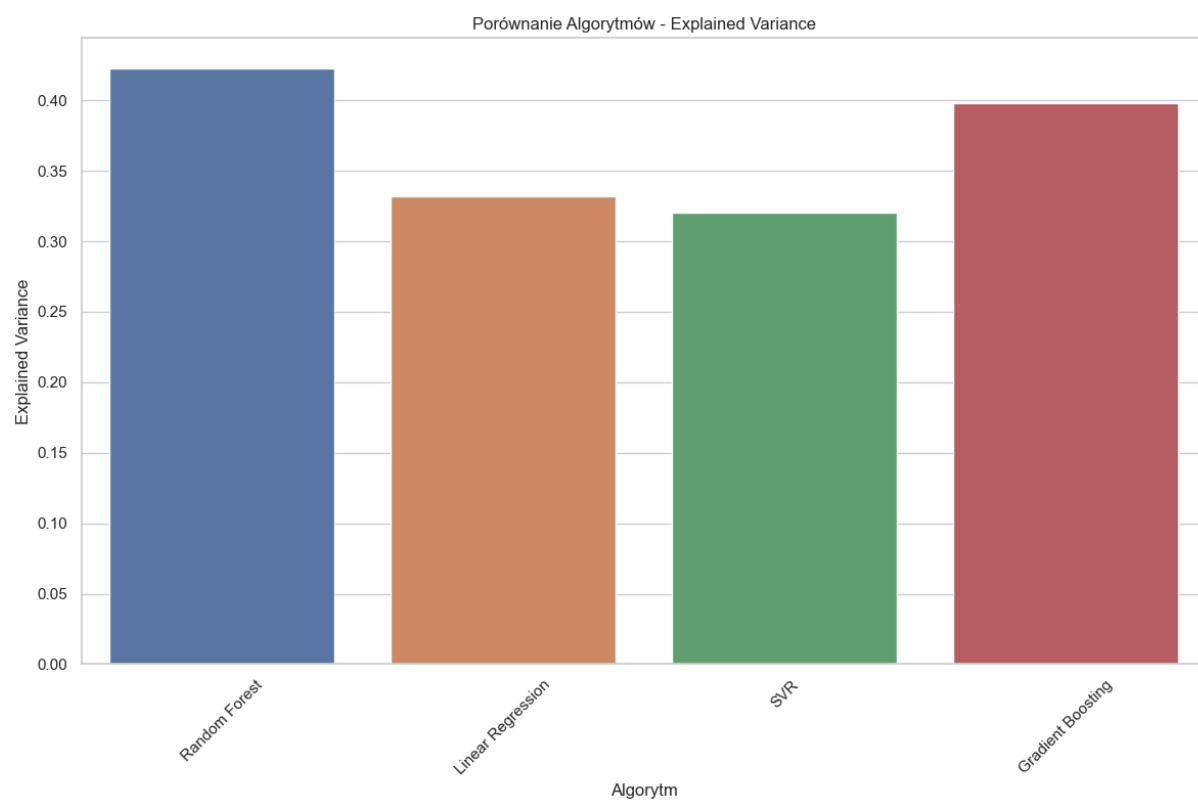
Rysunek 9: Porównanie algorytmów - MAE (Warszawa)

## Porównanie algorytmów na podstawie $R^2$ (Warszawa)



Rysunek 10: Porównanie algorytmów –  $R^2$  (Warszawa)

## Porównanie algorytmów na podstawie Explained Variance (Warszawa)



Rysunek 11: Porównanie algorytmów - Explained Variance (Warszawa)

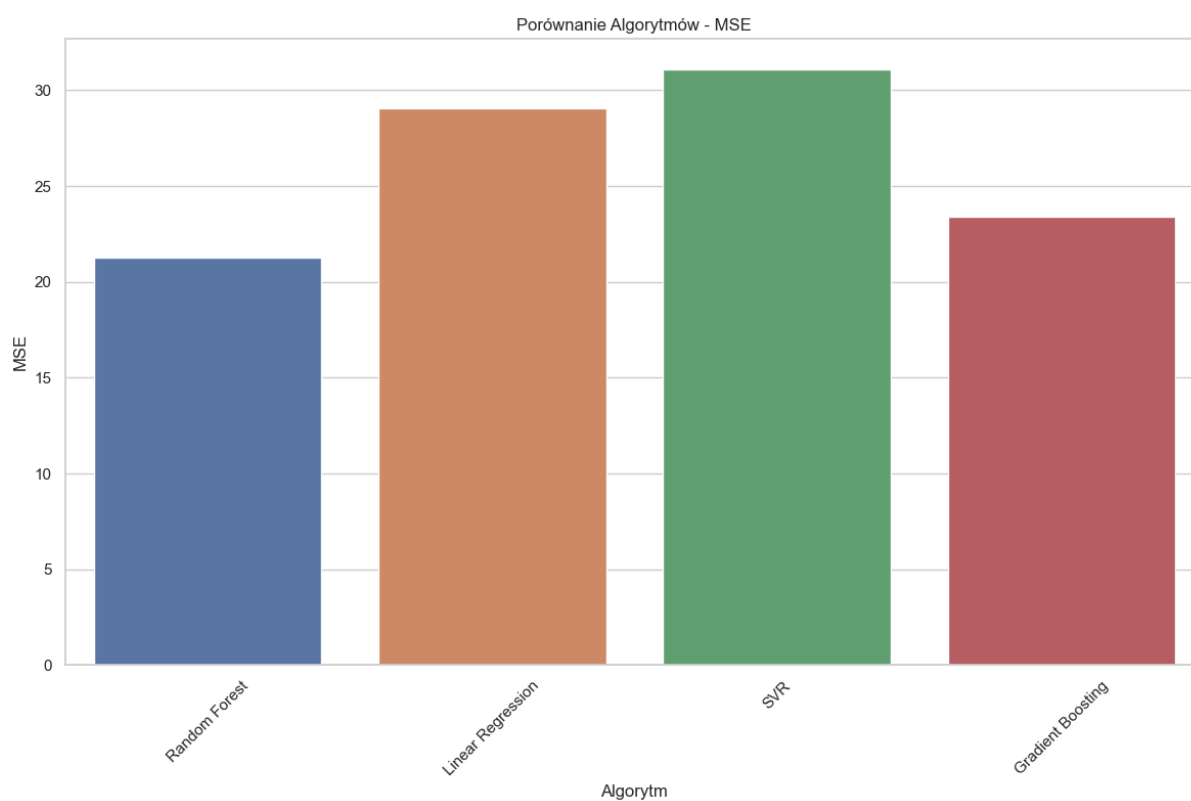
## Wyniki dla Olsztyna

### Porównanie algorytmów Machine Learning na podstawie metryk ocen dla Olsztyna

Tabela 10: Porównanie algorytmów Machine Learning na podstawie metryk ocen dla Olsztyna

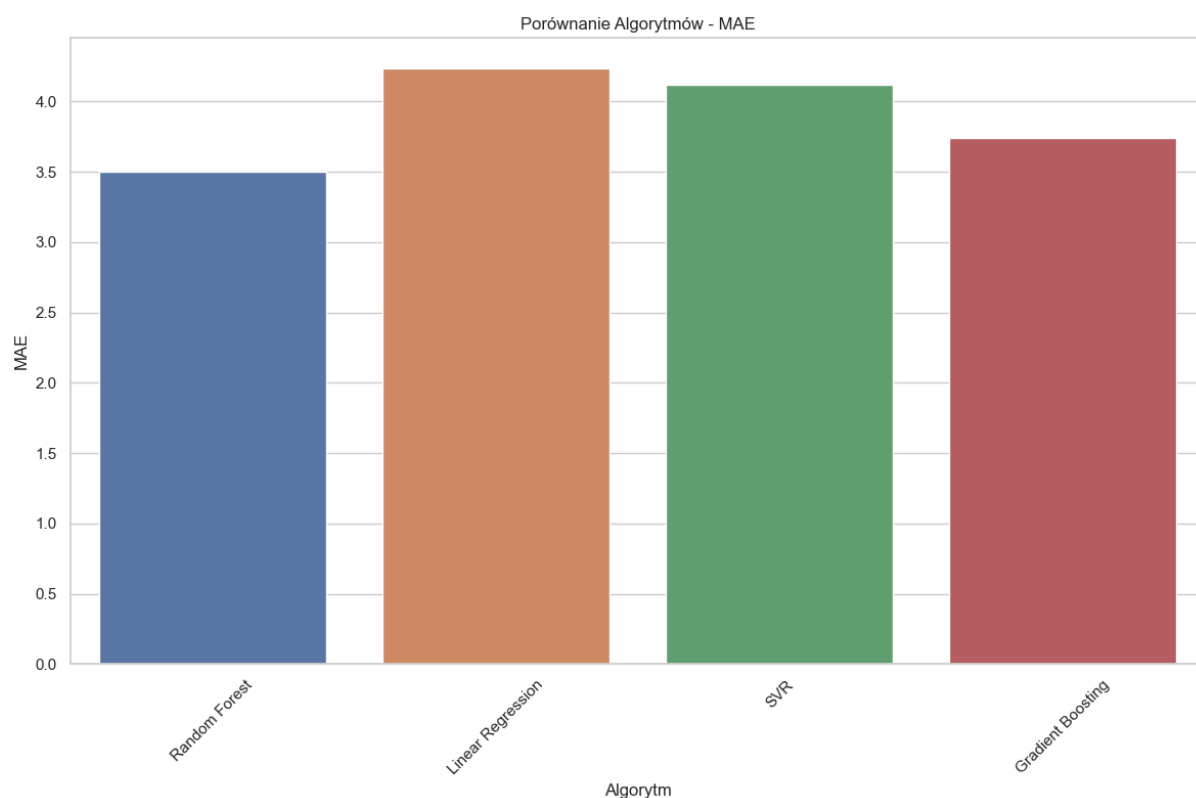
Algorytm	MSE	MAE	R <sup>2</sup>	Explained Variance
Random Forest	21.253210	3.498569	0.493645	0.494880
Linear Regression	29.053680	4.237737	0.307800	0.316643
SVR	31.109866	4.116937	0.258811	0.334822
Gradient Boosting	23.416677	3.739832	0.442101	0.446968

### Porównanie Algorytmów na podstawie MSE (Olsztyn)



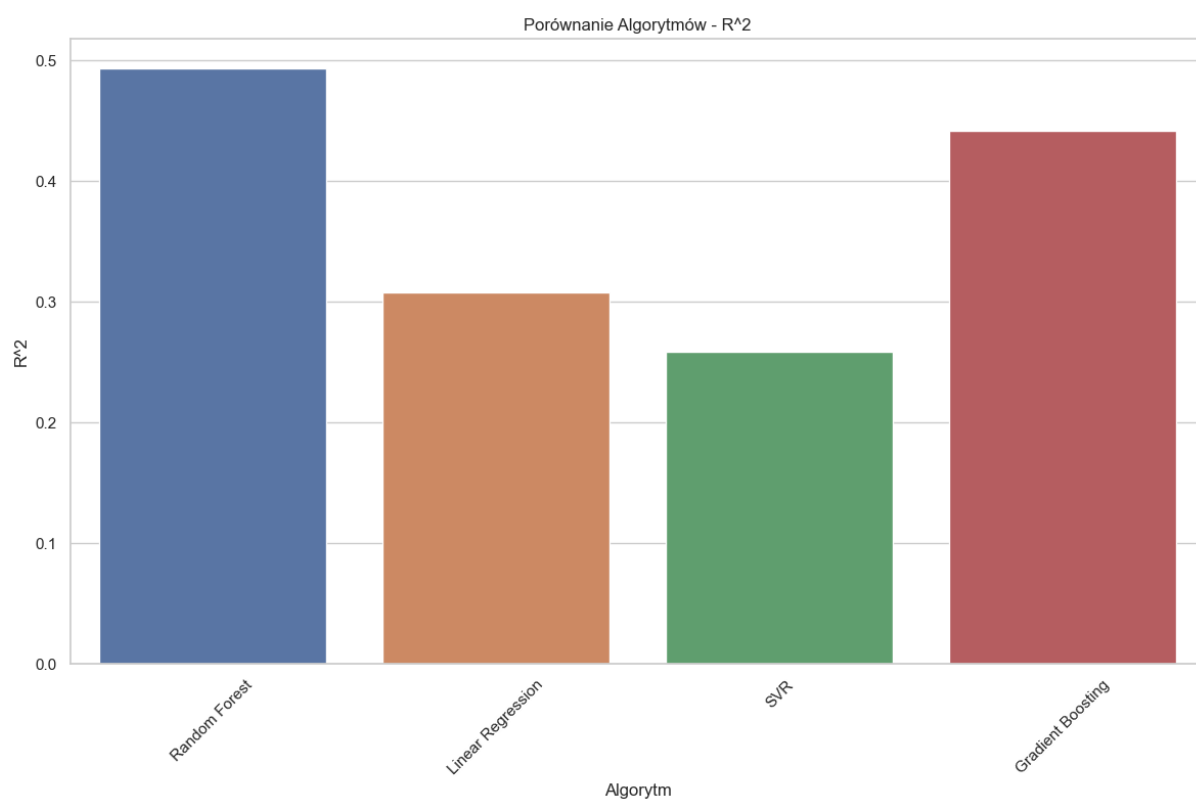
Rysunek 12: Porównanie algorytmów - MSE (Olsztyn)

## Porównanie algorytmów na podstawie MAE (Olsztyn)



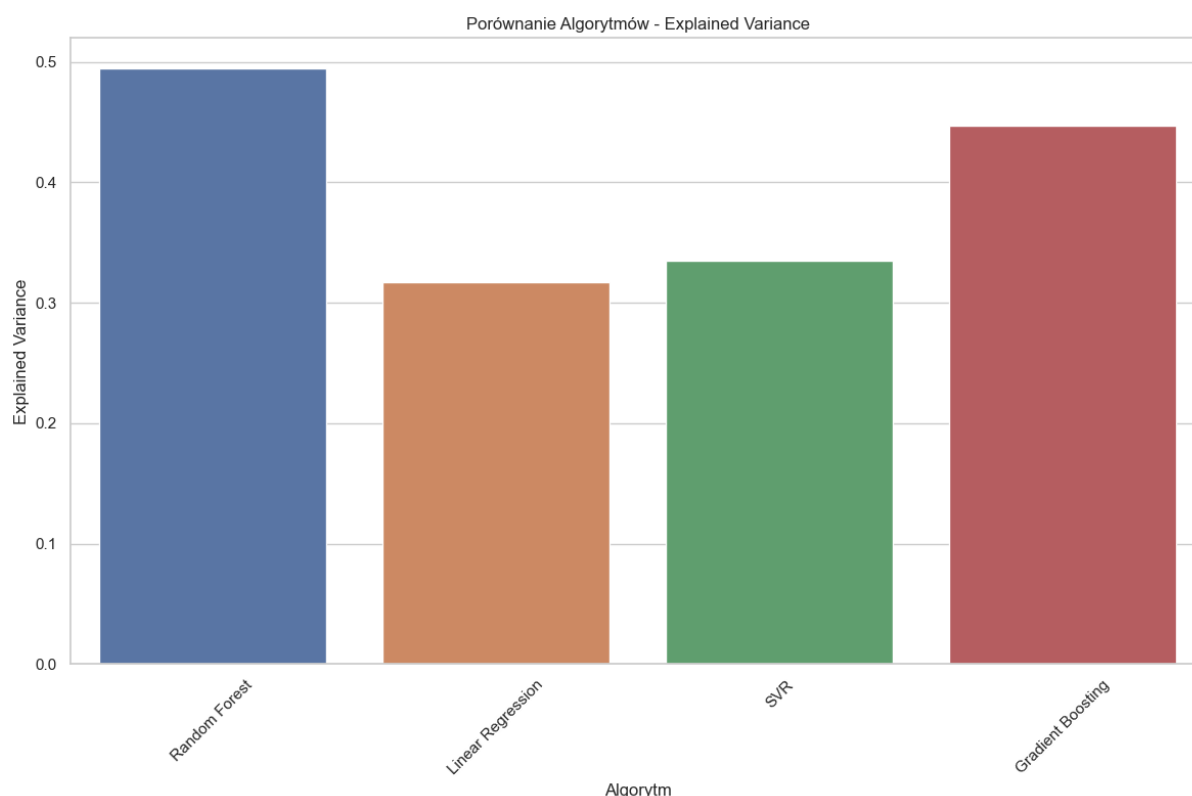
Rysunek 13: Porównanie algorytmów - MAE (Olsztyn)

## Porównanie algorytmów na podstawie $R^2$ (Olsztyn)



Rysunek 14: Porównanie algorytmów –  $R^2$  (Olsztyn)

## Porównanie algorytmów na podstawie Explained Variance (Olsztyn)



Rysunek 15: Porównanie algorytmów - Explained Variance (Olsztyn)

### 4.4. Wyniki i Analiza

#### Mean Squared Error (MSE)

- **Random Forest** osiągnął najniższą wartość MSE dla Berlina (19.893195) i Warszawy (24.529275), co oznacza najmniejsze średnie błędy kwadratowe w prognozach dla tych miast.
- Dla Olsztyna, **Random Forest** również uzyskał niską wartość MSE (21.253210).
- **Gradient Boosting** z wynikiem 21.134548 dla Berlina i 25.832121 dla Warszawy jest nieco gorszy od Random Forest.
- **Linear Regression** i **SVR** wykazały wyższe wartości MSE, co wskazuje na większe błędy w prognozach dla wszystkich miast.

#### Mean Absolute Error (MAE)

- **Random Forest** osiągnął najniższą wartość MAE dla Berlina (3.374996), Warszawy (3.732797) i Olsztyna (3.498569), co oznacza najmniejsze średnie absolutne błędy.



- **Gradient Boosting** z wynikami 3.499016 dla Berlina i 3.878793 dla Warszawy jest blisko wyników Random Forest.
- **Linear Regression** i **SVR** mają wyższe wartości MAE, wskazując na większe odchylenia od rzeczywistych wartości dla wszystkich miast.

## R<sup>2</sup>

- **Random Forest** osiągnął najwyższą wartość R<sup>2</sup> dla Olsztyna (0.493645) i Berlina (0.444673), co oznacza, że najlepiej wyjaśnia wariancję danych w porównaniu do innych modeli.
- **Gradient Boosting** z wartością R<sup>2</sup> równą 0.410021 dla Berlina i 0.391137 dla Warszawy jest nieco gorszy.
- **Linear Regression** i **SVR** mają niższe wartości R<sup>2</sup>, co wskazuje na gorsze dopasowanie do danych.

## Explained Variance

- **Random Forest** ma najwyższą wartość Explained Variance dla Olsztyna (0.494880) i Berlina (0.445605), co oznacza, że najlepiej wyjaśnia wariancję w danych.
- **Gradient Boosting** z wartością 0.413198 dla Berlina i 0.397730 dla Warszawy jest niewiele gorszy.
- **Linear Regression** i **SVR** mają niższe wartości Explained Variance dla wszystkich miast.

## 4.5. Wnioski

- **Random Forest** okazał się najlepszym algorytmem do prognozowania temperatury, osiągając najniższe wartości MSE i MAE oraz najwyższe wartości R<sup>2</sup> i Explained Variance w różnych miastach.
- **Gradient Boosting** również wykazał się dobrą wydajnością, choć nieco gorszą niż Random Forest.
- **Linear Regression** i **SVR** wykazały wyższe błędy i gorsze dopasowanie do danych w porównaniu do Random Forest i Gradient Boosting.
- W przyszłych badaniach warto skupić się na dalszej optymalizacji hiperparametrów dla Random Forest i Gradient Boosting w celu jeszcze lepszego dopasowania i dokładniejszych prognoz.
- Wyniki dla Berlina, Warszawy i Olsztyna potwierdziły stabilność i skuteczność wybranych algorytmów, co podkreśla ich potencjalne zastosowanie w różnych lokalizacjach.

## 5. Implementacja

W niniejszym rozdziale opisałem szczegóły dotyczące implementacji aplikacji internetowej do analizy danych pogodowych. W kolejnych podrozdziałach przedstawiłem jaką była strategia realizacji projektu, z jakich głównych modułów składa się aplikacja, jakie wzorce wykorzystano w projekcie, a także szczegółowo opisałem technologie zastosowane na frontendzie i backendzie oraz jaką bazę danych wykorzystałem i w jaki sposób przeprowadziłem testy. Całą aplikację napisałem za pomocą języka programowania JavaScript i Pythona oraz frameworków opartych o podane języki.

### 5.1. Strategia realizacji projektu

Implementowanie aplikacji internetowej do analizy danych pogodowych w celu przewidzenia prognozy pogody realizowałem zgodnie z następującą strategią:

- **Projektowanie bazy danych:** W pierwszej kolejności została zaprojektowana baza danych PostgreSQL, która będzie przechowywać dane dotyczące lokalizacji, pogody, Użytkowników, oraz inne dane potrzebne do działania aplikacji. Szczegóły odnośnie do tego co wykorzystałem do stworzenia bazy danych znajduje się w folderze 5.6 Bazy danych,
- **Tworzenie backendu:** Stworzyłem backend za pomocą frameworków Express.js oraz Flask (Framework Pythona). Express.js odpowiada za komunikację z bazą danych, przetwarzanie danych oraz udostępnianie danych do części frontendowej. Flask jest używany do realizacji specyficznych zadań analizy danych, w tym wykorzystania algorytmu RandomForestRegressor do przewidywania prognozy pogody. Więcej szczegółów o tym, co użyłem do stworzenia backendu, znajduje się w podrozdziale 5.5 Backend,
- **Testowanie backendu:** Po stworzeniu backendu zacząłem testowanie poprzez dodawanie nowych Użytkowników oraz Dodanie nowych Lokalizacji, aby upewnić się, że wszystko działa prawidłowo oraz że integracja z bazą danych istnieje. Więcej szczegółów o tym czego użyłem do testowania znajduje się w podrozdziale 5.7 Testy.
- **Tworzenie frontendu:** Opracowałem frontend aplikacji przy użyciu React.js ze zdefiniowanym szablonem Redux. Do wyglądu aplikacji zastosowałem framework CSS o nazwie Bulma, natomiast do wysyłania wiadomości e-mail wykorzystałem EmailJS. Więcej szczegółów o tym co użyłem do stworzenia frontendu znajduje się w podrozdziale 5.4 Frontend,
- **Integracja.** Następnie frontend i backend połączyłem ze sobą, aby umożliwić komunikację między nimi,
- **Testowanie aplikacji:** Na koniec testowałem aplikację korzystając z niej, aby upewnić się, że całość działa prawidłowo i nie występują problemy w czasie użytkowania.

## 5.2. Główne moduły aplikacji

Aplikacja składa się z kilku głównych modułów:

- **Panel administracyjny:** Umożliwia Adminowi na zarządzanie Lokalizacjami oraz danymi pogodowymi, a także zapewnia zarządzanie kontami Użytkowników.
- **System Prognozowania pogody:** Pozwala Użytkownikom na przeprowadzenie prognozy pogody poprzez wybranie miejscowości z listy oraz wybranie ilości dni na jakie chce się przeprowadzić prognozę.
- **Witryna publiczna:** Ten moduł zawiera stronę internetową, która jest dostępna dla wszystkich Użytkowników i umożliwia przeglądanie pogody na aktualną porę dnia oraz sprawdzenia informacji na temat projektu

## 5.3. Wzorce wykorzystane w projekcie

W aplikacji zastosowano następujące wzorce projektowe:

- **Wzorec Model-View-Controller (MVC):** Wzorec ten pozwolił na oddzielenie logiki biznesowej od logiki prezentacji. W aplikacji, backend odpowiada za logikę biznesową, natomiast frontend zajmuje się prezentacją danych.
- **Wzorec Store:** Wzorec ten został wykorzystany w połączeniu z React i Redux. Umożliwia on centralizację stanu aplikacji i ułatwia jego zarządzanie.
- **Wzorec Singleton:** Wzorec ten został użyty do klasy odpowiedzialnej za połączenie z bazą danych, zapewnił on, że w całej aplikacji jest tylko jedna instancja tej klasy, co umożliwia łatwiejsze zarządzanie połączeniem z bazą danych.
- **Wzorec Strategy:** Wzorec ten został zastosowany w module Flask odpowiedzialnym za prognozowanie pogody. Pozwala on na dynamiczne wybieranie i używanie różnych algorytmów prognozowania.

## 5.4. Frontend

Frontend aplikacji został stworzony z użyciem biblioteki React. Jest ona oparta o język JavaScript. Komenda **create-react-app** z argumentem **template redux** pozwoliła na stworzenie aplikacji w React z predefiniowanym szablonem Redux. Zawiera on niezbędne pliki i konfiguracje do implementacji stanu aplikacji za pomocą biblioteki Redux. Biblioteki wymienione poniżej były najbardziej przydatne w czasie tworzenia interfejsu aplikacji oraz do integracji z backendem.

- **React-router-dom** – pozwala na implementację nawigacji aplikacji.
- **Axios** – służy do komunikacji z API.
- **Moment** – pozwala na wyświetlenie daty w odpowiednim formacie.
- **Bulma** – framework CSS daje gotowe komponenty, które pozwoliły na stworzenie solidnego i responsywnego interfejsu.
- **@Redux.js/tolkit** – ułatwia tworzenie aplikacji z wykorzystaniem biblioteki Redux. Zawiera on skróty dla najczęściej stosowanych operacji, takich jak konfiguracja i tworzenie store'a, tworzenie akcji i reduktorów oraz ułatwia pracę z typami danych.
- **Emailjs** – pozwala na wysyłanie wiadomości e-mail z poziomu aplikacji.

## 5.5. Backend

Backend aplikacji został stworzony z użyciem frameworka Express.js oraz dodatkowego frameworka Flask do przeprowadzania prognoz pogody. Wykorzystałem dodatkowe biblioteki, które pomogły w napisaniu backendu:

- **Dotenv** – pozwala na ładowanie pliku .env z konfiguracją do projektu.
- **Cors** – umożliwia obsługę mechanizmu CORS (Cross-Origin Resource Sharing) w aplikacji Express. Jest to potrzebne, jeśli aplikacja korzysta z zasobów z innych domen.
- **Pg** – zezwala na połączenie z bazą danych PostgreSQL za pomocą Node.js.
- **Sequelize** – ORM (Object-Relational Mapping) dla Node.js, który ułatwia pracę z bazami danych, bez konieczności pisania zapytań SQL.
- **Connect-session-sequelize** – pozwala na trzymanie sesji Użytkownika w bazie danych za pomocą ORM Sequelize.
- **Express-session** – umożliwia obsługę sesji Użytkownika w aplikacji Express.
- **Argon2** – daje możliwość hashowania haseł za pomocą algorytmu Argon2, który jest uważany za jeden z bezpieczniejszych sposobów hashowania.
- **Scikit-learn** – biblioteka do uczenia maszynowego w Pythonie, wykorzystywana do implementacji modelu RandomForestRegressor.
- **Numpy** – biblioteka do obliczeń naukowych w Pythonie.
- **Pandas** – biblioteka do manipulacji danymi i analizy danych w Pythonie.
- **Flask-SQLAlchemy** – rozszerzenie do Flask'a ułatwiające integrację z bazą danych za pomocą SQLAlchemy.

Do przeprowadzania prognoz pogody użyłem algorytmu **RandomForestRegressor**, który jest częścią biblioteki Scikit-learn. Flask jest

wykorzystywany do implementacji tego modelu uczenia maszynowego. Proces ten obejmuje:

- **Pobieranie danych:** Pobieranie danych pogodowych z tabeli **Weatherdata** w bazie danych PostgreSQL.
- **Przygotowanie danych:** Przetwarzanie pobranych danych za pomocą bibliotek Numpy i Pandas.
- **Trenowanie modelu:** Wykorzystanie danych z **Weatherdata** do trenowania modelu **RandomForestRegressor**.
- **Przeprowadzanie prognoz:** Wykorzystanie wytrenowanego modelu do prognozowania przyszłych warunków pogodowych, a następnie przesyłanie wyników do części frontendowej aplikacji.

Dzięki takiemu podejściu backend aplikacji jest w stanie przeprowadzać skomplikowane analizy danych, co stanowi kluczowy element funkcjonalności aplikacji. Każde z użytych modułów pozwolił na stworzenie działającego zaplecza, które łączy się z bazą danych. Dają one możliwość tworzenia konta, logowania się na nie oraz umożliwiają pobieranie danych z zewnętrznego API do bazy danych oraz przeprowadzenie prognozy pogody.

## 5.6. Bazy danych.

Do przechowywania danych Użytkowników oraz danych lokalizacyjnych i danych pogodowych została wykorzystana baza danych **PostgreSQL**. Aby ją uruchomić, zastosowałem narzędzie **pgAdmin** dostarczone wraz z instalatorem **PostgreSQL**, co pozwoliło na proste stworzenie bazy danych potrzebnej do stworzenia aplikacji. Jako dane przekazywane do bazy posłużyłem się zewnętrznym **API Open-Meteo**, które pozwala na dostęp do danych pogodowych na podstawie współrzędnych geograficznych. Poniżej znajdują się szczegółowe opisy poszczególnych tabel w bazie danych:

### 1. Tabela: **Sessions**

**Opis:** Tabela przechowuje informacje o sesjach Użytkowników, w tym dane sesji oraz czas wygaśnięcia. Jest używana do zarządzania sesjami w aplikacji.

*Tabela 11: Tabela bazy danych Sessions*

Nazwa Kolumny	Typ danych	Opis
sid	VARCHAR(36)	Unikalny identyfikator sesji, klucz główny.
expires	TIMESTAMP WITH TIME ZONE	Data i czas wygaśnięcia sesji.
data	TEXT	Dane przechowywane w sesji.
createdAt	TIMESTAMP WITH TIME ZONE	Data i czas utworzenia sesji.
updatedAt	TIMESTAMP WITH TIME ZONE	Data i czas ostatniej aktualizacji sesji.

## 2. Tabela: **Forecast**

**Opis:** Tabela przechowuje prognozy pogodowe dla różnych lokalizacji. Zawiera dane dotyczące temperatury, wilgotności, opadów, prędkości i kierunku wiatru, a także odniesienia do lokalizacji i raportów.

*Tabela 12: Tabela bazy danych Forecast*

Nazwa Kolumny	Typ danych	Opis
id	SERIAL	Unikalny identyfikator prognozy, klucz główny.
uuid	VARCHAR(255)	Unikalny identyfikator UUID prognozy.
future_dates	TIMESTAMP WITH TIME ZONE	Data i czas prognozy.
forecast_temperature	DOUBLE PRECISION	Przewidywana temperatura.
forecast_humidity	DOUBLE PRECISION	Przewidywana wilgotność.
forecast_precipitation	DOUBLE PRECISION	Przewidywane opady.
forecast_windSpeed	DOUBLE PRECISION	Przewidywana prędkość wiatru.
forecast_windDirection	INTEGER	Przewidywany kierunek wiatru.
locationId	INTEGER	Identyfikator lokalizacji (FK).
reportId	INTEGER	Identyfikator raportu (FK).
createdAt	TIMESTAMP WITH TIME ZONE	Data i czas utworzenia prognozy.
updatedAt	TIMESTAMP WITH TIME ZONE	Data i czas ostatniej aktualizacji prognozy.

## 3. Tabela: **Location**

**Opis:** Tabela przechowuje informacje o lokalizacjach, takie jak miasto, kraj oraz współrzędne geograficzne. Używana do identyfikacji miejsc dla których przechowywane są dane pogodowe i prognozy.

*Tabela 13: Tabela bazy danych Location*

Nazwa Kolumny	Typ danych	Opis
id	SERIAL	Unikalny identyfikator lokalizacji, klucz główny.
uuid	VARCHAR(255)	Unikalny identyfikator UUID lokalizacji.
city	VARCHAR(255)	Nazwa miasta.

Nazwa Kolumny	Typ danych	Opis
country	VARCHAR(255)	Nazwa kraju.
latitude	DOUBLE PRECISION	Szerokość geograficzna.
longitude	DOUBLE PRECISION	Długość geograficzna.
createdAt	TIMESTAMP WITH TIME ZONE	Data i czas utworzenia wpisu.
updatedAt	TIMESTAMP WITH TIME ZONE	Data i czas ostatniej aktualizacji wpisu.

#### 4. Tabela: **Report**

**Opis:** Tabela przechowuje raporty generowane przez Użytkowników, zawierające tytuł, datę oraz identyfikator Użytkownika, który stworzył raport.

*Tabela 14: Tabela bazy danych Report*

Nazwa Kolumny	Typ danych	Opis
id	SERIAL	Unikalny identyfikator raportu, klucz główny.
uuid	VARCHAR(255)	Unikalny identyfikator UUID raportu.
title	VARCHAR(255)	Tytuł raportu.
reportDate	TIMESTAMP WITH TIME ZONE	Data raportu.
userId	INTEGER	Identyfikator Użytkownika, który stworzył raport (FK).
createdAt	TIMESTAMP WITH TIME ZONE	Data i czas utworzenia raportu.
updatedAt	TIMESTAMP WITH TIME ZONE	Data i czas ostatniej aktualizacji raportu.

#### 5. Tabela: **Users**

**Opis:** Tabela przechowuje dane o Użytkownikach aplikacji, w tym imię, nazwisko, email, hasło oraz rolę Użytkownika.

*Tabela 15: Tabela bazy danych Users*

Nazwa Kolumny	Typ danych	Opis
id	SERIAL	Unikalny identyfikator Użytkownika, klucz główny.
uuid	VARCHAR(255)	Unikalny identyfikator UUID Użytkownika.
imie	VARCHAR(255)	Imię Użytkownika.

Nazwa Kolumny	Typ danych	Opis
nazwisko	VARCHAR(255)	Nazwisko Użytkownika.
email	VARCHAR(255)	Adres email Użytkownika.
password	VARCHAR(255)	Hasło Użytkownika (przechowywane w formie zaszyfrowanej).
role	VARCHAR(255)	Rola Użytkownika (np. Admin, standardowy).
createdAt	TIMESTAMP WITH TIME ZONE	Data i czas utworzenia konta.
updatedAt	TIMESTAMP WITH TIME ZONE	Data i czas ostatniej aktualizacji konta.

## 6. Tabela: **Weatherdata**

**Opis:** Tabela przechowuje dane pogodowe dla różnych lokalizacji, takie jak temperatura, wilgotność, opady, prędkość i kierunek wiatru. Zawiera odniesienia do lokalizacji.

*Tabela 16: Tabela bazy danych Weatherdata*

Nazwa Kolumny	Typ danych	Opis
id	SERIAL	Unikalny identyfikator danych pogodowych, klucz główny.
uuid	VARCHAR(255)	Unikalny identyfikator UUID danych pogodowych.
date	TIMESTAMP WITH TIME ZONE	Data i czas rejestracji danych pogodowych.
temperature	DOUBLE PRECISION	Temperatura.
humidity	DOUBLE PRECISION	Wilgotność.
precipitation	DOUBLE PRECISION	Opady.
windSpeed	DOUBLE PRECISION	Prędkość wiatru.
windDirection	INTEGER	Kierunek wiatru.
locationId	INTEGER	Identyfikator lokalizacji (FK).
createdAt	TIMESTAMP WITH TIME ZONE	Data i czas utworzenia wpisu.
updatedAt	TIMESTAMP WITH TIME ZONE	Data i czas ostatniej aktualizacji wpisu.



## 5.7. Testy

Testowanie aplikacji było kluczowym etapem procesu w celu zapewnienia, że wszystkie funkcjonalności działają zgodnie z wymaganiami i oczekiwaniami Użytkowników. Testowanie przeprowadzono na kilku poziomach, z wykorzystaniem różnych metod i narzędzi.

### 5.7.1. Testowanie endpointów

Testowanie endpointów aplikacji zostało przeprowadzone za pomocą narzędzia **Postman**. To narzędzie umożliwia wysyłanie zapytań HTTP do różnych endpointów API, co pozwala na weryfikację poprawności odpowiedzi serwera oraz sprawdzenie, czy API spełnia założone wymagania.

- **Testowanie Metod HTTP:** Sprawdzono poprawność metod HTTP takich jak GET, POST, PATCH, DELETE. Każda z metod była testowana pod kątem prawidłowej obsługi żądań i odpowiedzi.
- **Testowanie Autoryzacji:** Weryfikowano, czy mechanizmy autoryzacji działają poprawnie. Testowano dostęp do endpointów w zależności od ról i uprawnień Użytkowników.

### 5.7.2. Testowanie manualne

Testowanie manualne aplikacji miało na celu weryfikację jej funkcjonalności z perspektywy Użytkownika końcowego. Proces ten obejmował:

- **Testowanie Interfejsu Użytkownika:** Sprawdzono, czy interfejs Użytkownika jest intuicyjny i czy wszystkie funkcjonalności są łatwe do znalezienia i użycia. Testowano różne scenariusze użytkownika, takie jak rejestracja, logowanie, dodawanie danych, przeprowadzanie prognoz pogody itp.
- **Testowanie Scenariuszy użytkownika:** Testowano aplikację w rzeczywistych scenariuszach użytkownika, aby upewnić się, że działa ona zgodnie z oczekiwaniami i spełnia wymagania funkcjonalne.

## 6. Instrukcja Użytkownika

W niniejszym rozdziale widnieje instrukcja użytkowania z **aplikacji internetowej do analizy danych pogodowych**. Na opisanych zrzutach ekranu przedstawiony został interfejs graficzny.

### 6.1. Interfejs strony głównej

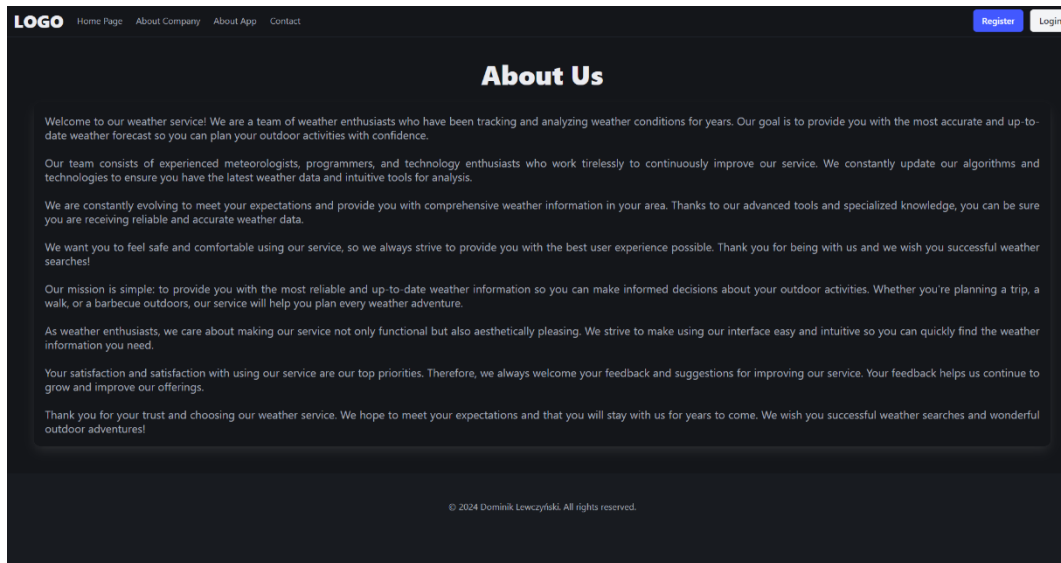
Po uruchomieniu aplikacji w wybranej przeglądarce internetowej pierwszym widokiem jakie Użytkownik zobaczy to jest widok strony głównej, który wygląda następująco:



Rysunek 16: Strona główna aplikacji

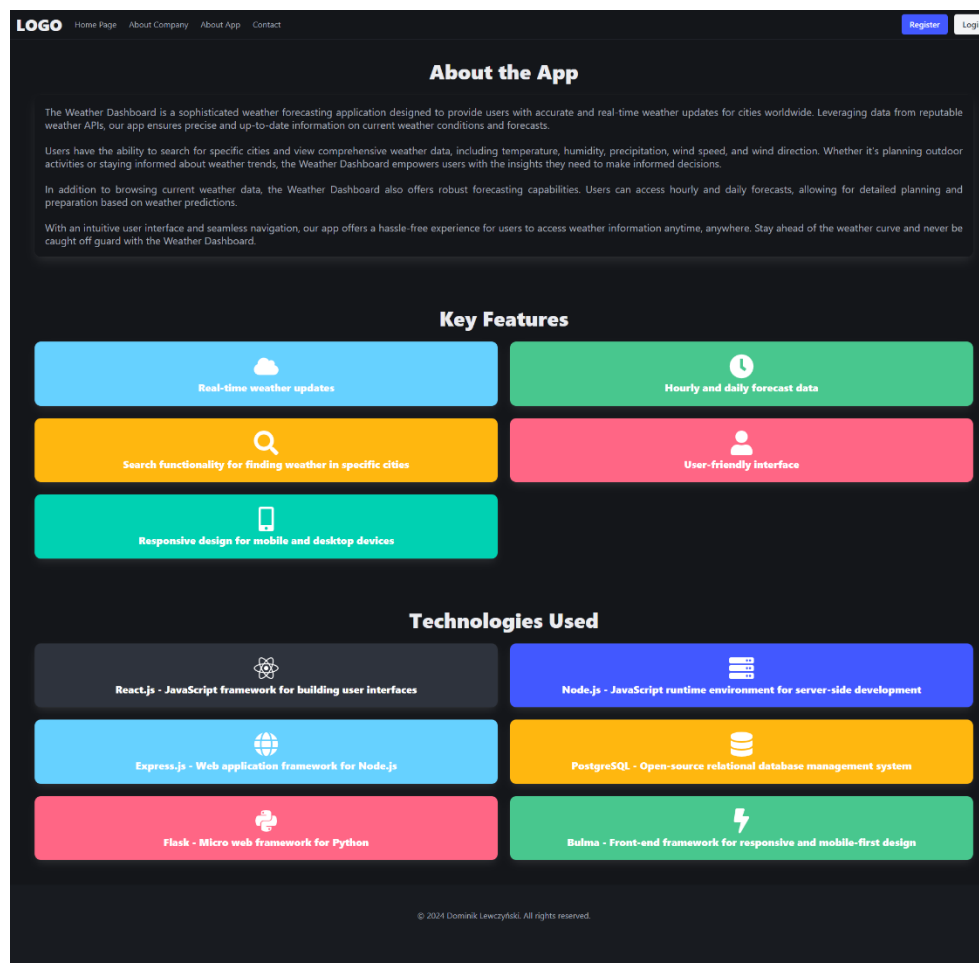
To jest strona główna aplikacji, gdzie jest wyświetlana pogoda na aktualną porę danego dnia. Powyżej napisu „Current Weather in Warsaw for 29.05.2024 | 09:00” znajduje się lista z dostępnymi lokalizacjami. Po wybraniu innej miejscowości zobaczymy inne dane pogodowe. Na górze strony znajduje się nawigacja, za pomocą której zostaniemy przekierowani do odpowiednich stron

- **About Company** – strona, w której jest zawarte informacje o właścicielach aplikacji



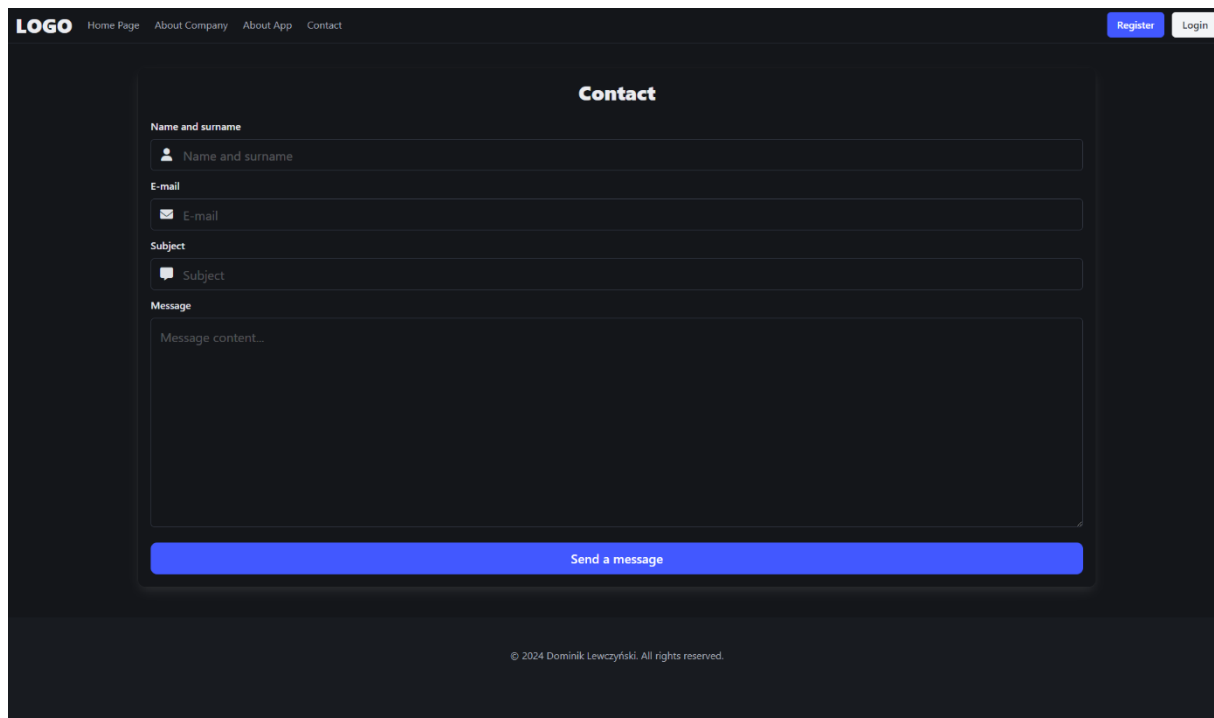
Rysunek 17: Strona About company

- **About App** – strona, w której jest zawarte informacje na temat aplikacji (Na temat działania, użytych technologiach itd.)



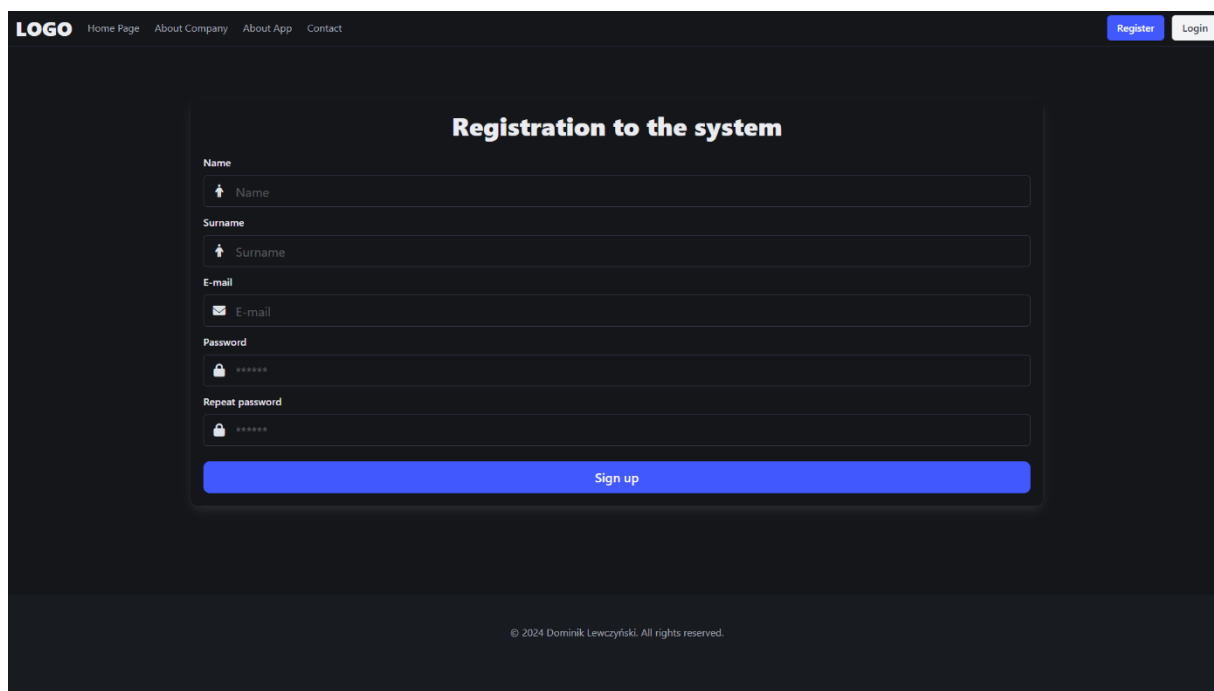
Rysunek 18: Strona About App

- **Contact** – strona, na której umieszczony jest formularz kontaktowy. Za jego pomocą można wysłać wiadomość np. można zapytać się o szczegółowe informacje na temat działania aplikacji.



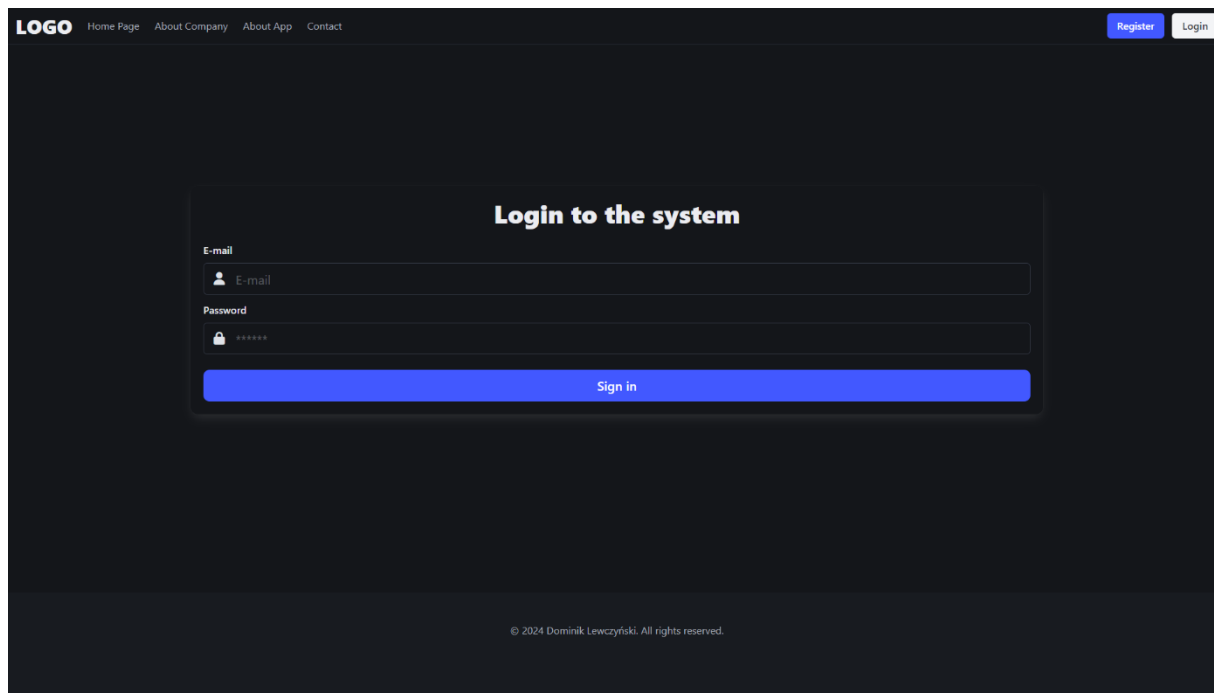
Rysunek 19: Strona Contact

- **Register** – strona, na której jest formularz do rejestracji nowego konta



Rysunek 20: Strona Register

- **Login** – strona, z formularzem do logowania się do swojego konta

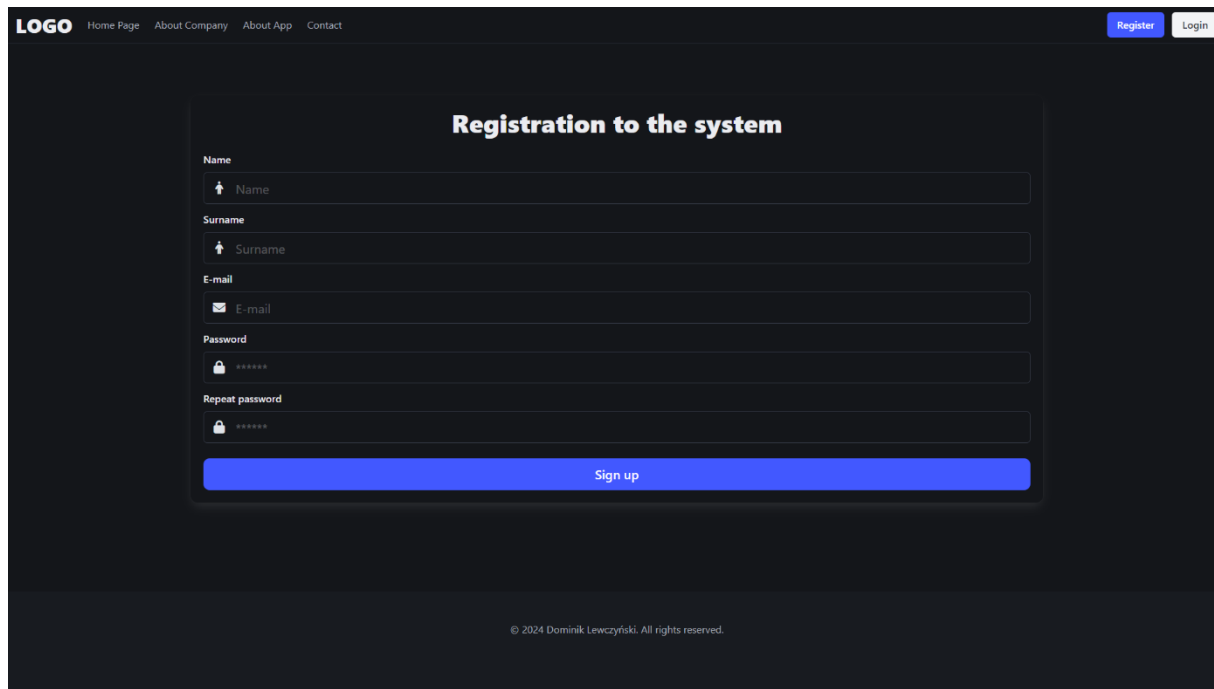


The screenshot shows a dark-themed web application interface. At the top, there is a navigation bar with a 'LOGO' on the left and links for 'Home Page', 'About Company', 'About App', and 'Contact' in the center. On the right side of the navigation bar, there are two buttons: 'Register' (highlighted in blue) and 'Login' (white with a dark border). The main content area features a central card titled 'Login to the system'. This card contains two input fields: 'E-mail' with an email icon and 'Password' with a lock icon. Below these fields is a large blue button labeled 'Sign in'. At the bottom of the page, there is a footer with the text '© 2024 Dominik Lewczyński. All rights reserved.'

Rysunek 21: Strona Login

## 6.2. Konto Użytkownika

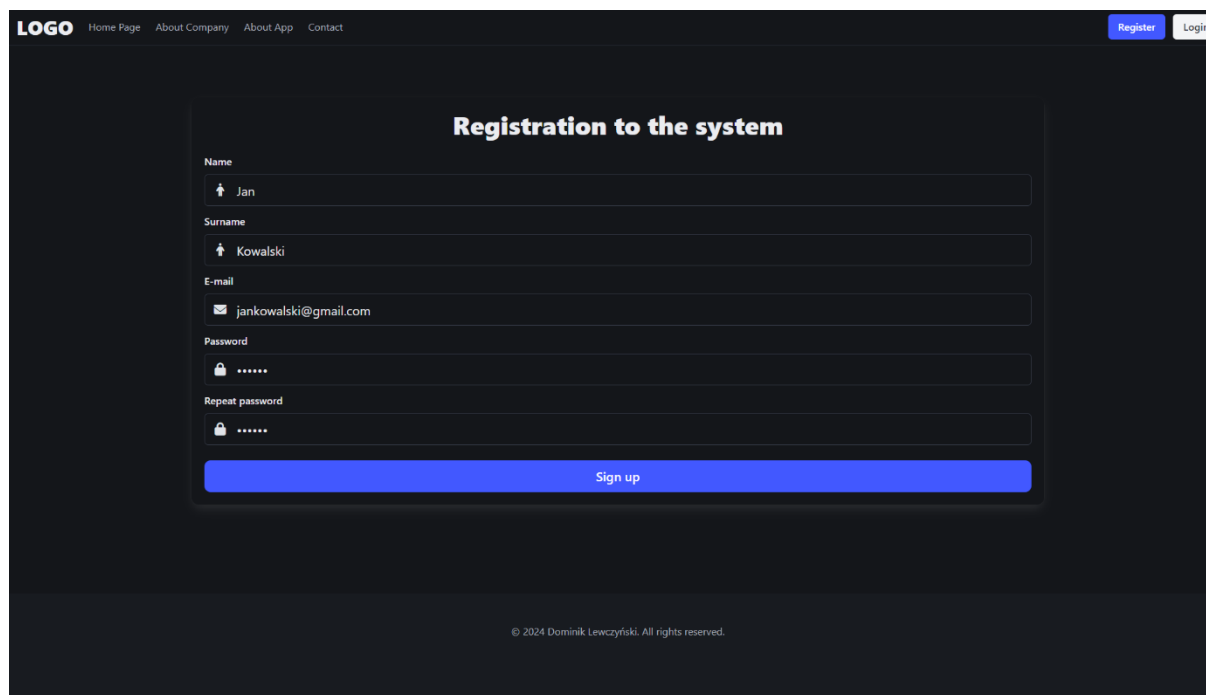
Jeżeli Użytkownik zdecyduje się na utworzenie konta wybiera na stronie głównej przycisk **Register**. Zostanie wówczas otwarty formularz do rejestracji konta.



The screenshot shows the same dark-themed web application interface as the previous one. The navigation bar is identical. The main content area features a central card titled 'Registration to the system'. This card contains five input fields: 'Name' with a person icon, 'Surname' with a person icon, 'E-mail' with an email icon, 'Password' with a lock icon, and 'Repeat password' with a lock icon. Below these fields is a large blue button labeled 'Sign up'. At the bottom of the page, there is a footer with the text '© 2024 Dominik Lewczyński. All rights reserved.'

Rysunek 22: Strona Register

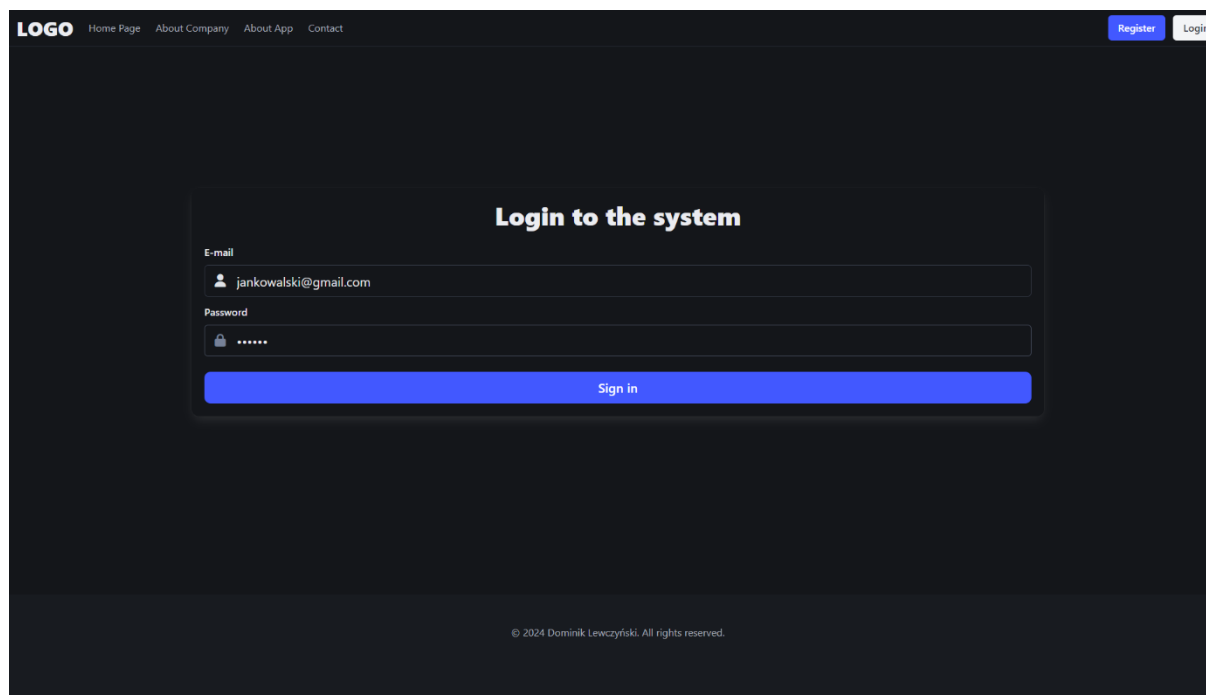
Następnie w formularzu tym wypełnia on niezbędne dane. Na poniższym przykładzie pokazałem utworzenie konta dla Użytkownika Jan Kowalski



The screenshot shows a web application interface with a dark theme. At the top, there is a navigation bar with a 'LOGO' and links for 'Home Page', 'About Company', 'About App', and 'Contact'. On the right side of the navigation bar, there are two buttons: 'Register' (highlighted in blue) and 'Login'. The main content area features a central form titled 'Registration to the system'. The form contains the following fields: 'Name' with the value 'Jan', 'Surname' with the value 'Kowalski', 'E-mail' with the value 'jankowalski@gmail.com', 'Password' with masked characters '\*\*\*\*\*', and 'Repeat password' with the value '\*\*\*\*\*'. A large blue button labeled 'Sign up' is positioned at the bottom of the form. At the very bottom of the page, there is a footer with the text '© 2024 Dominik Lewczyński. All rights reserved.'

Rysunek 23: Strona Register z wypełnionym formularzem

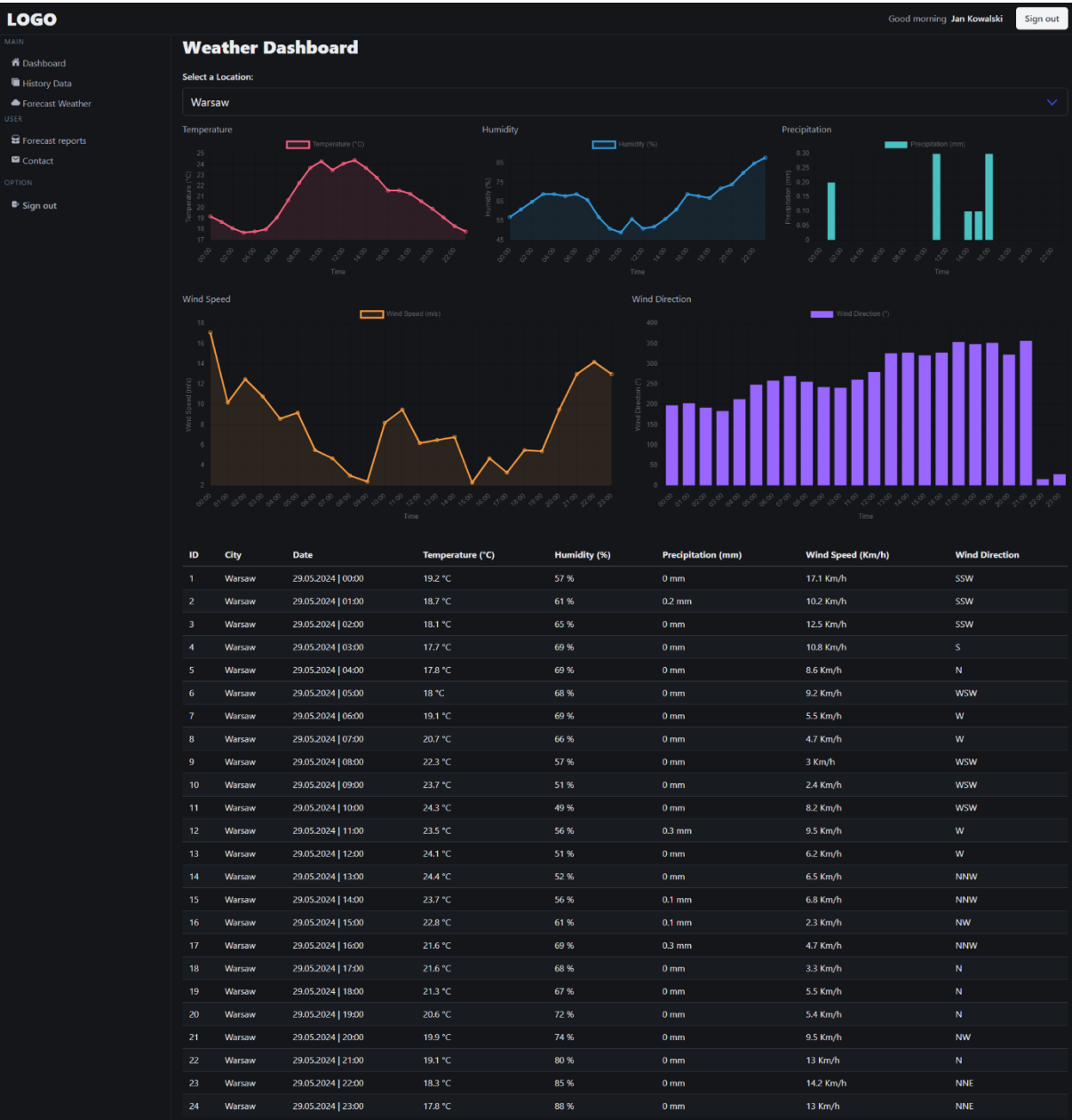
Kiedy konto zostanie utworzone Użytkownik Jan Kowalski wybiera na górze z prawej strony przycisk „Login” i zostaje przekierowany do strony z formularzem do logowania na konto.



The screenshot shows the same web application interface as before, but now the 'Login' button in the top navigation bar is highlighted in blue. The main content area features a central form titled 'Login to the system'. The form contains the following fields: 'E-mail' with the value 'jankowalski@gmail.com' and 'Password' with masked characters '\*\*\*\*\*'. A large blue button labeled 'Sign in' is positioned at the bottom of the form. The footer at the bottom of the page remains the same, with the text '© 2024 Dominik Lewczyński. All rights reserved.'

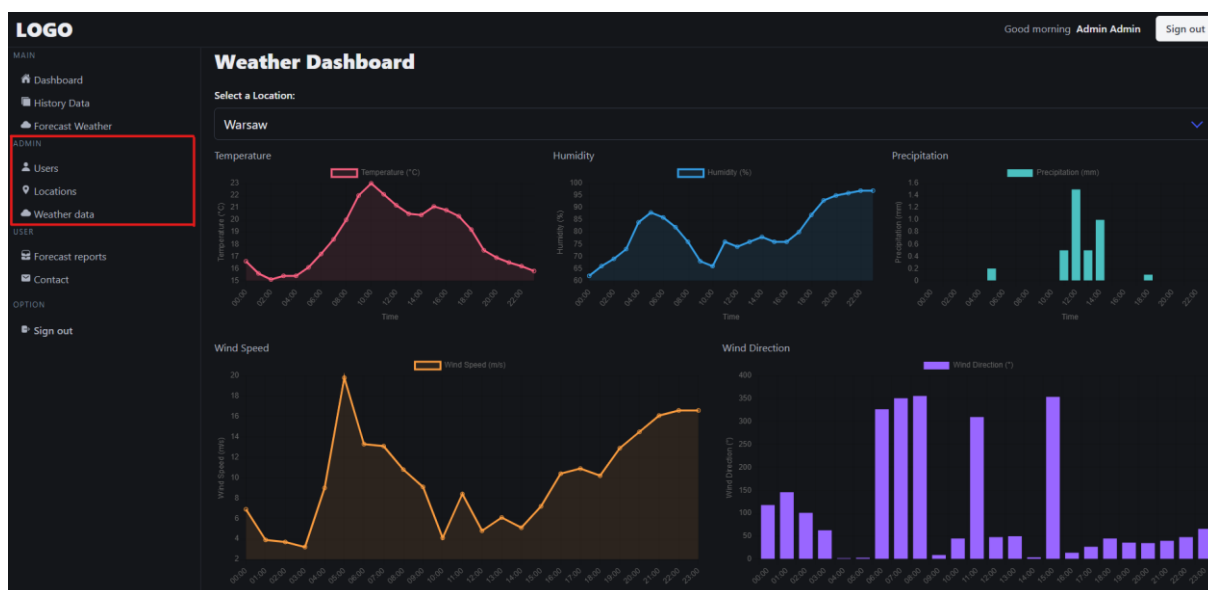
Rysunek 24: Strona Login z wypełnionym formularzem

Po naciśnięciu przycisku „Login” na formularzu, Użytkownik zostaje zalogowany na swoje konto.



Rysunek 25: Panel główny Użytkownika

W ten oto sposób konto Użytkownika zostało utworzone oraz zalogowano się na nie. Na powyższym zdjęciu znajduje się główny panel konta Użytkownika. W główmy panelu wyświetlają się dane pogodowe na aktualny dzień z większą ilością szczegółów i możliwością zobaczenia danych w formie wykresów oraz tabeli. Istnieje opcja wybrania innej miejscowości Po lewej stronie jest nawigacja, która zmienia się w zależności czy zalogowano się na konto Użytkownika czy konto Admina. W tym przykładzie znajduje się link do strony **Dashboard**, link do strony **History Data** zawierający historyczne dane pogodowe, link **Forecast Weather** do przeprowadzenia prognozy pogody i link **Forecast reports** do przechowywania raportów z przeprowadzonej prognozy. Strona **Contact** zawiera formularz do kontaktowania się z Adminem. Dla konta Admina dodatkowo są linki: do listy Użytkowników , do listy z dostępnymi lokalizacjami oraz listy z danymi pogodowymi



Rysunek 26: Panel główny dla konta Admina z dodatkową sekcją nawigacji

Dla nowo utworzonego konta strona z raportami z przeprowadzonej prognozy wygląda następująco:

The screenshot shows a 'Forecast Reports' page for a user named Jan Kowalski. The top right corner displays 'Good morning Jan Kowalski' and a 'Sign out' button. The left sidebar is identical to the previous screenshot. The main content area is titled 'Forecast Reports' and contains a table with the following columns: ID, Title, Date, User, and Options. The table is currently empty.

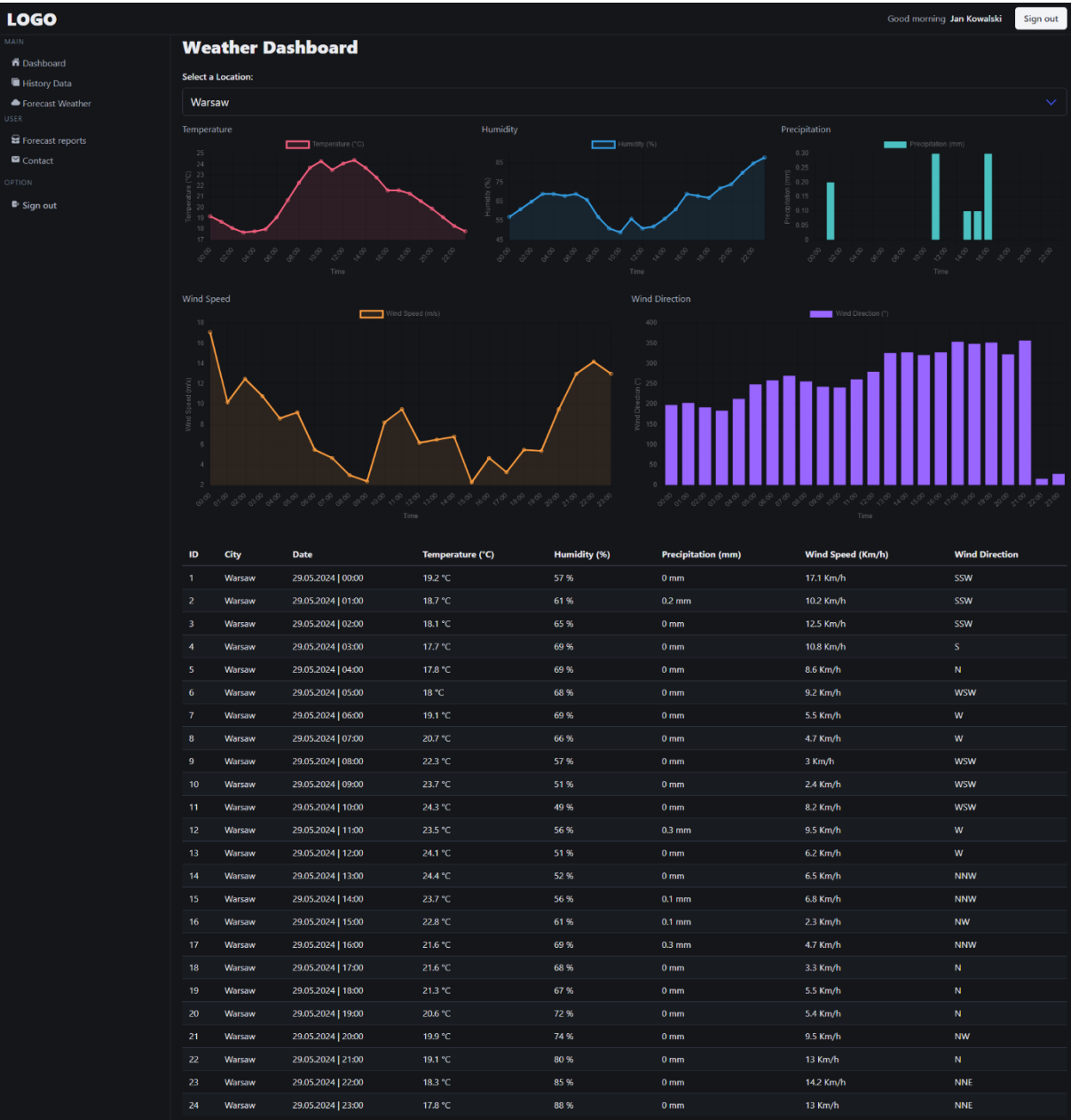
Rysunek 27: Strona Forecast Reports z pusta listą raportów

Jest to pusta lista, która będzie zawierać raporty po dokonaniu procesu prognozowania pogody na stronie **Forecast Weather**. Więcej szczegółów na temat prognozowania są w rozdziale 6.4. Przeprowadzenie prognozy



### 6.3. Przeglądanie danych pogodowych

Jeśli Użytkownik będzie chciał przejrzeć pogodę na aktualny dzień po zalogowaniu przechodzi do strony Dashboard



Rysunek 28: Panel główny Użytkownika

Dane są przedstawione w formie wykresów oraz tabeli pozwalającej przejrzania pogody na aktualny dzień na daną godzinę. Jeśli Użytkownik będzie chciał przeglądać historyczne dane pogodowe przechodzi na stronę **History data**, którą wybiera z nawigacji po lewej stronie. Po kliknięciu w opcje zostaniemy przekierowani do strony z dostępną tabelą oraz opcjami

LOGO

Good morning Jan Kowalski

Sign out

MAIN

Dashboard

History Data

Forecast Weather

USER

Forecast reports

Contact

OPTION

Sign out

History data list

Select City:

Select city

Select Date Range:

From - To

Apply Filters

ID	City	Date	Temperature (°C)	Humidity (%)	Precipitation (mm)	Wind Speed (Km/h)	Wind Direction
1	Berlin	01.04.2024   00:00	9.4 °C	97 %	0 mm	17.6 Km/h	ENE
2	Berlin	01.04.2024   01:00	9.2 °C	97 %	0 mm	24.3 Km/h	ENE
3	Berlin	01.04.2024   02:00	8.6 °C	96 %	0 mm	19.8 Km/h	E
4	Berlin	01.04.2024   03:00	8.5 °C	95 %	0 mm	21.8 Km/h	E
5	Berlin	01.04.2024   04:00	8.8 °C	94 %	0 mm	25.3 Km/h	E
6	Berlin	01.04.2024   05:00	9.3 °C	93 %	0 mm	28.5 Km/h	E
7	Berlin	01.04.2024   06:00	10.8 °C	91 %	0 mm	32.4 Km/h	E
8	Berlin	01.04.2024   07:00	12.4 °C	88 %	0 mm	28.3 Km/h	ESE
9	Berlin	01.04.2024   08:00	13 °C	85 %	0 mm	28 Km/h	E
10	Berlin	01.04.2024   09:00	14.5 °C	80 %	0 mm	23.5 Km/h	ESE
11	Berlin	01.04.2024   10:00	16.9 °C	71 %	0 mm	21.5 Km/h	ESE
12	Berlin	01.04.2024   11:00	18.9 °C	61 %	0 mm	21.3 Km/h	SE
13	Berlin	01.04.2024   12:00	19.2 °C	58 %	0 mm	16.5 Km/h	SSE

Rysunek 29: Strona zawierająca historyczne dane pogodowe

Na powyższym zdjęciu jest wyświetlona lista danych historycznych posortowana według nazwy miejscowości. Użytkownik ma możliwość przejrzenia tabeli. Posiada również opcje filtrowania tabeli poprzez wybranie miejscowości z listy oraz ustawienia zakresu dat.

## 6.4. Przeprowadzenie prognozy

Jeśli Użytkownik będzie chciał przeprowadzić prognozę pogody przechodzi do strony Forecast Weather poprzez kliknięcie linku w nawigacji po lewej stronie. Po wybraniu opcji zostaniemy przekierowani do strony z formularzem do przeprowadzenia prognozy pogody

LOGO

Good morning Jan Kowalski

Sign out

MAIN

Dashboard

History Data

Forecast Weather

USER

Forecast reports

Contact

OPTION

Sign out

Forecast Weather

Select city to forecast weather:

Select Location

Write a number of days to forecast weather

1

Predict forecast weather

Rysunek 30: Strona Forecast Weather przed prognozowaniem

Na powyższym zdjęciu jest wyświetlony formularz, z którego Użytkownik wybiera miejscowość oraz wpisuje ilość dni, na które chce przeprowadzić prognozę. Dla przykładu przeprowadzę prognozę pogody dla miejscowości Szczytno na 2 dni.

Strona 48

The screenshot shows a web application interface for weather forecasting. On the left is a sidebar with a 'MAIN' section containing 'Dashboard', 'History Data', and 'Forecast Weather' (highlighted). Below this is a 'USER' section with 'Forecast reports', 'Contact', and 'Sign out'. The top right corner displays 'Good morning Jan Kowalski' and a 'Sign out' button. The main content area is titled 'Forecast Weather' and contains a form with two fields: 'Select city to forecast weather:' with a dropdown menu showing 'Szcztyno', and 'Write a number of days to forecast weather' with a text input containing '2'. A blue button labeled 'Predict forecast weather' is positioned below the second field.

Rysunek 31: Strona Forecast Weather z wypełnionym formularzem

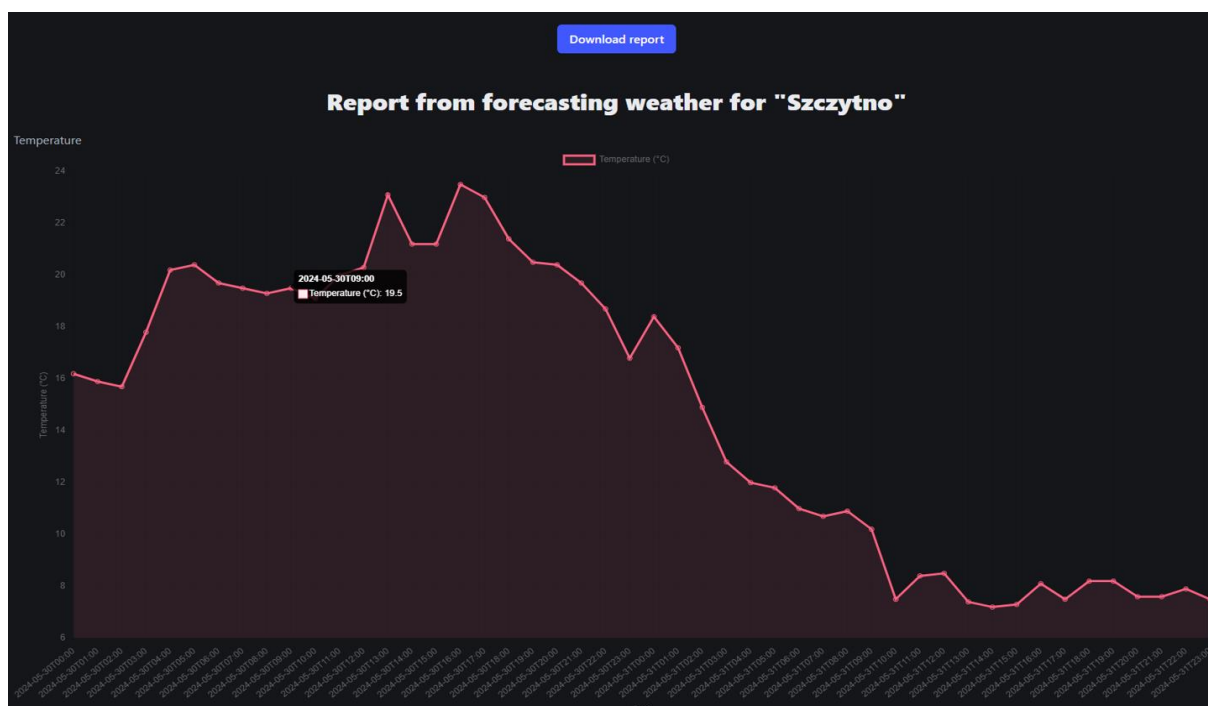
Po wypełnieniu formularza naciskamy przycisk **Predict forecast weather**. Po kliknięciu pokaże się napis **Loading**, który informuje że trzeba czekać na zakończenie procesu. Po przeprowadzeniu prognozy pokaże się komunikat o dodaniu raportu do konta

This screenshot shows the same 'Forecast Weather' page as before, but with a 'Loading...' message centered at the bottom. A dark notification box at the top center displays the message 'Report added to account' with an 'OK' button. The sidebar and top navigation elements remain the same.

Rysunek 32: Strona Forecast Weather w trakcie przeprowadzania prognozy

Po kliknięciu **Ok** przy komunikacie ukaże się na wynik procesu przeprowadzenia prognozy pogody w formie raportu, który też jest automatycznie zapisany do konta Użytkownika.

## Wykres Temperatury



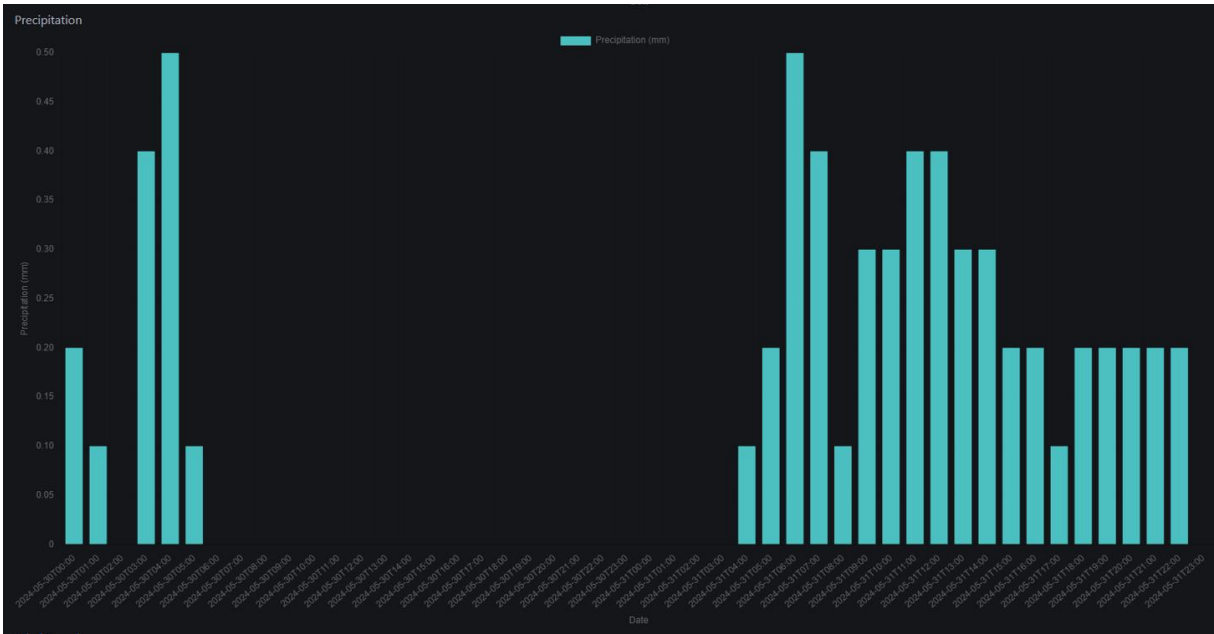
Rysunek 33: Wykres temperatury z przeprowadzonej prognozy

Wykres wilgotności



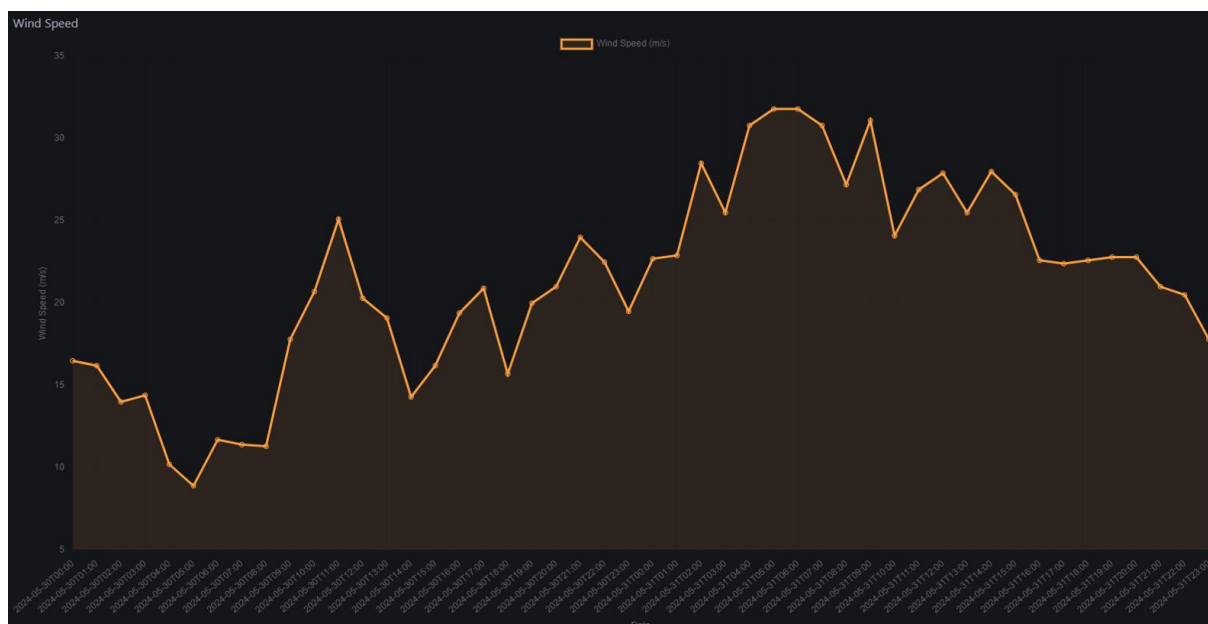
Rysunek 34: Wykres wilgotności z przeprowadzonej prognozy

Wykres Opadów



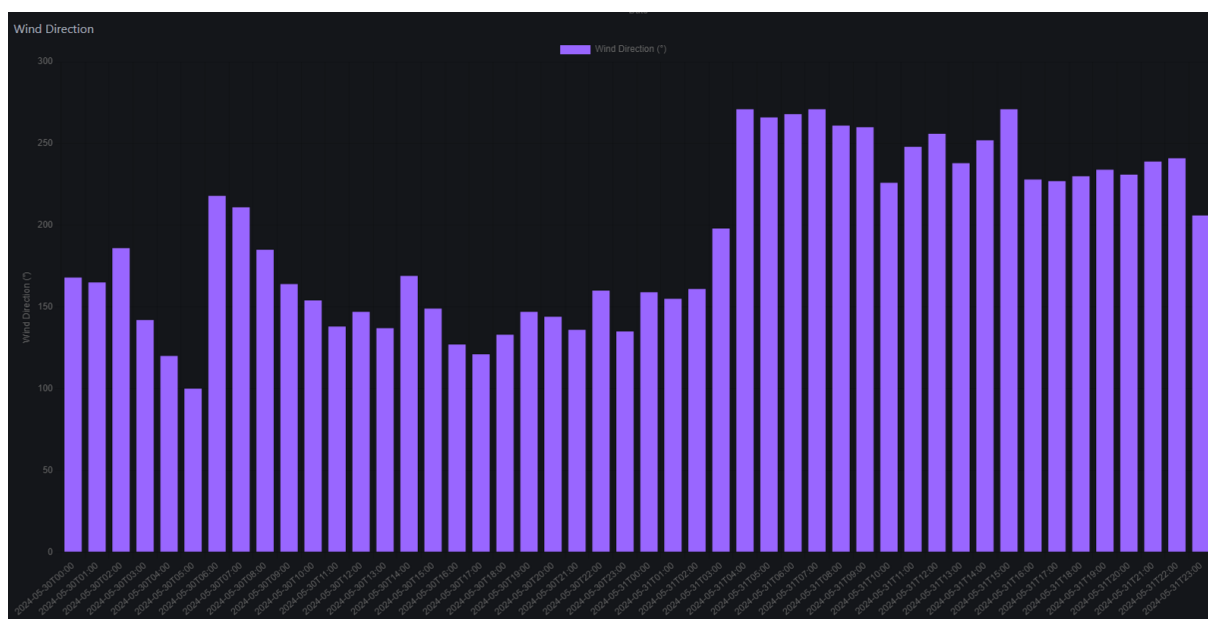
Rysunek 35: Wykres opadów z przeprowadzonej prognozy

## Wykres prędkości wiatru



Rysunek 36: Wykres prędkości wiatru z przeprowadzonej prognozy

## Wykres kierunku wiatru



Rysunek 37: Wykres kierunku wiatru z przeprowadzonej prognozy

Tabela z wynikami prognozy

ID	Date	Temperature (°C)	Humidity (%)	Precipitation (mm)	Wind Speed (m/s)	Wind Direction
1	30.05.2024   00:00	16.2 °C	69 %	0.2 mm	16.5 m/s	N
2	30.05.2024   01:00	15.9 °C	52 %	0.1 mm	16.2 m/s	SSE
3	30.05.2024   02:00	15.7 °C	55 %	0.0 mm	14.0 m/s	S
4	30.05.2024   03:00	17.8 °C	72 %	0.4 mm	14.4 m/s	SE
5	30.05.2024   04:00	20.2 °C	73 %	0.5 mm	10.2 m/s	ESE
6	30.05.2024   05:00	20.4 °C	73 %	0.1 mm	8.9 m/s	E
7	30.05.2024   06:00	19.7 °C	72 %	0.0 mm	11.7 m/s	SW
8	30.05.2024   07:00	19.5 °C	73 %	0.0 mm	11.4 m/s	SSW
9	30.05.2024   08:00	19.3 °C	72 %	0.0 mm	11.3 m/s	S
10	30.05.2024   09:00	19.5 °C	69 %	0.0 mm	17.8 m/s	SSE
11	30.05.2024   10:00	19.1 °C	68 %	0.0 mm	20.7 m/s	SSE
12	30.05.2024   11:00	20.0 °C	63 %	0.0 mm	25.1 m/s	SE
13	30.05.2024   12:00	20.3 °C	66 %	0.0 mm	20.3 m/s	SSE
14	30.05.2024   13:00	23.1 °C	49 %	0.0 mm	19.1 m/s	SE
15	30.05.2024   14:00	21.2 °C	65 %	0.0 mm	14.3 m/s	S
16	30.05.2024   15:00	21.2 °C	63 %	0.0 mm	16.2 m/s	SSE
17	30.05.2024   16:00	23.5 °C	44 %	0.0 mm	19.4 m/s	SE
18	30.05.2024   17:00	23.0 °C	42 %	0.0 mm	20.9 m/s	ESE
19	30.05.2024   18:00	21.4 °C	61 %	0.0 mm	15.7 m/s	SE
20	30.05.2024   19:00	20.5 °C	60 %	0.0 mm	20.0 m/s	SSE
21	30.05.2024   20:00	20.4 °C	61 %	0.0 mm	21.0 m/s	SE
22	30.05.2024   21:00	19.7 °C	59 %	0.0 mm	24.0 m/s	SE
23	30.05.2024   22:00	18.7 °C	59 %	0.0 mm	22.5 m/s	SSE
24	30.05.2024   23:00	16.8 °C	67 %	0.0 mm	19.5 m/s	SE
25	31.05.2024   00:00	18.4 °C	68 %	0.0 mm	22.7 m/s	SSE
26	31.05.2024   01:00	17.2 °C	69 %	0.0 mm	22.9 m/s	SSE
27	31.05.2024   02:00	14.9 °C	83 %	0.0 mm	28.5 m/s	SSE
28	31.05.2024   03:00	12.8 °C	81 %	0.0 mm	25.5 m/s	SSW
29	31.05.2024   04:00	12.0 °C	82 %	0.1 mm	30.8 m/s	W
30	31.05.2024   05:00	11.8 °C	83 %	0.2 mm	31.8 m/s	W
31	31.05.2024   06:00	11.0 °C	86 %	0.5 mm	31.8 m/s	W
32	31.05.2024   07:00	10.7 °C	83 %	0.4 mm	30.8 m/s	W
33	31.05.2024   08:00	10.9 °C	81 %	0.1 mm	27.2 m/s	W
34	31.05.2024   09:00	10.2 °C	83 %	0.3 mm	31.1 m/s	W
35	31.05.2024   10:00	7.5 °C	86 %	0.3 mm	24.1 m/s	SW
36	31.05.2024   11:00	8.4 °C	85 %	0.4 mm	26.9 m/s	WSW
37	31.05.2024   12:00	8.5 °C	85 %	0.4 mm	27.9 m/s	WSW
38	31.05.2024   13:00	7.4 °C	86 %	0.3 mm	25.5 m/s	WSW
39	31.05.2024   14:00	7.2 °C	84 %	0.3 mm	28.0 m/s	WSW
40	31.05.2024   15:00	7.3 °C	81 %	0.2 mm	26.6 m/s	W
41	31.05.2024   16:00	8.1 °C	81 %	0.2 mm	22.6 m/s	SW
42	31.05.2024   17:00	7.5 °C	81 %	0.1 mm	22.4 m/s	SW
43	31.05.2024   18:00	8.2 °C	82 %	0.2 mm	22.6 m/s	SW
44	31.05.2024   19:00	8.2 °C	82 %	0.2 mm	22.8 m/s	SW
45	31.05.2024   20:00	7.6 °C	80 %	0.2 mm	22.8 m/s	SW
46	31.05.2024   21:00	7.6 °C	81 %	0.2 mm	21.0 m/s	WSW
47	31.05.2024   22:00	7.9 °C	80 %	0.2 mm	20.5 m/s	WSW
48	31.05.2024   23:00	7.5 °C	79 %	0.0 mm	17.8 m/s	SSW

Download report

Rysunek 38: Tabela z wynikami z przeprowadzonej prognozy

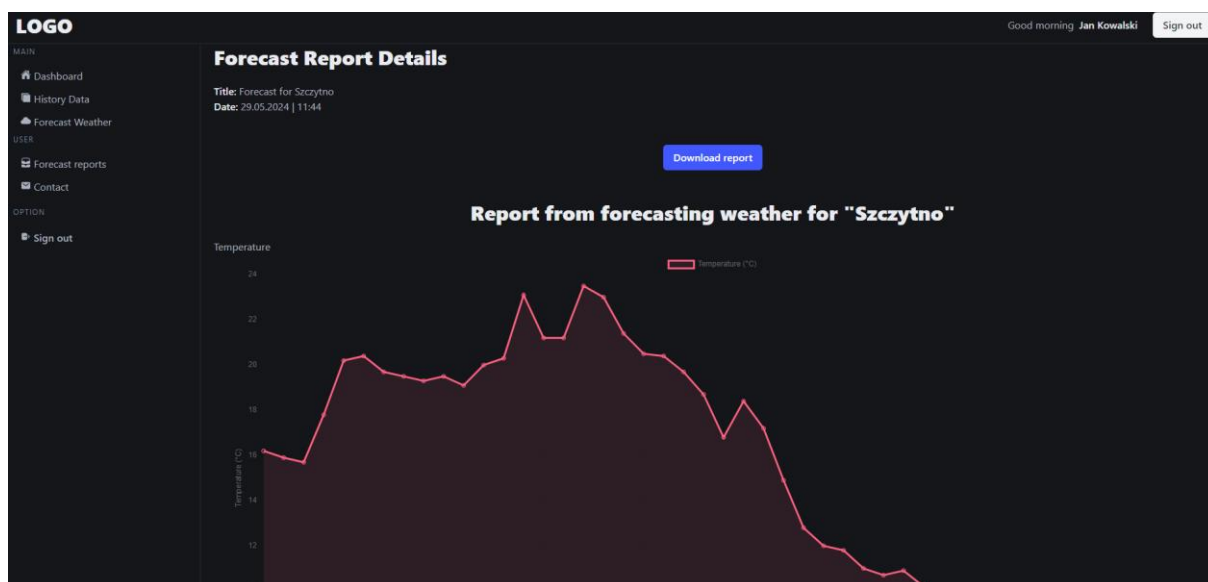
Użytkownikowi zostają wyświetlone wyniki przeprowadzonej prognozy na w formie wykresów oraz tabeli z możliwością pobrania raportu w formie pliku PDF poprzez naciśnięcie przycisku **Download report**.

Jeśli chcę zobaczyć raport z przeprowadzonej prognozy po ponownym zalogowaniu do panelu klika w opcje **Forecast reports** z dostępnej nawigacji Po kliknięciu zostaje przekierowany do strony zawierającej listę zapisanych raportów.

ID	Title	Date	User	Options
1	Forecast for Szczytno	29.05.2024   11:44	Jan Kowalski	<a href="#">Show More</a> <a href="#">Delete</a>

Rysunek 39: Lista raportów z przeprowadzonej prognozy

Po przekierowaniu będzie dostępny raport z przewozonej prognozy. Po kliknięciu w przycisk **Show More** można zobaczyć szczegóły raport oraz pobrać raport w formie pliku pdf za pomocą przycisk **Download report**.



Rysunek 40: Szczegóły raportu

## 7. Licencje

Celem niniejszego rozdziału jest przedstawienie środowisk, grafiki, bibliotek zewnętrznych oraz źródeł wiedzy programistycznej jakie zostały wykorzystane w celu stworzenia aplikacji.

### 7.1. Narzędzia

- **Visual Studio Code** – darmowy edytor kodu z kolorowaniem stworzony przez firmę Microsoft, o otwartym kodzie źródłowym na licencji MIT.
- **Google Chrome** – darmowa przeglądarka internetowa na urządzenia stacjonarne i mobilne, za pomocą, której wpisując hasło w wyszukiwarce Google, możemy wyszukiwać i poruszać się po stronach internetowych oraz przeglądać na nich treści w celu znalezienia potrzebnych dla nas informacji.
- **Postman** – darmowe narzędzie, które służy do testowania i debugowania interfejsów API (Application Programming Interface). Pozwala na tworzenie, edytowanie i wysyłanie żądań HTTP oraz przeglądanie odpowiedzi z serwera. Może być używane do testowania różnych typów żądań, takich jak GET, POST, PUT i DELETE, a także do wysyłania danych w różnych formatach, takich jak JSON czy XML.
- **Node.js** – darmowa platforma programistyczna oparta na języku JavaScript, która pozwala na tworzenie aplikacji serwerowych. Dzięki Node.js, programiści mogą korzystać z jednego języka (JavaScript) na obu stronach aplikacji (frontend i backend) co pozwala na zwiększenie produktywności i jednolity styl kodowania.
- **Open-Meteo** – Darmowe API pogodowe o otwartym kodzie źródłowym, z którego może korzystać każdy bez potrzeby rejestracji czy uzyskiwania klucza API. Platforma kieruje się ideą, że dostęp do rzetelnych informacji pogodowych powinien być powszechnie dostępny

### 7.2. Grafiki

- **Font awesome** - źródło darmowych ikon w dowolnym formacie.



### 7.3. Biblioteki zewnętrzne

- **React.js** – biblioteka języka programowania JavaScript, która umożliwia tworzenie interfejsów graficznych w aplikacjach internetowych.
- **Express.js** – back-endowy framework aplikacji internetowych do budowania interfejsów API RESTful z Node.js, wydany jako darmowe oprogramowanie o otwartym kodzie źródłowym na licencji MIT. Przeznaczony jest do budowania aplikacji internetowych i API.
- **Flask** – back-endowy framework dla języka Python, który umożliwia szybkie tworzenie aplikacji internetowych. Jest lekki i łatwy w użyciu, co czyni go popularnym wyborem dla projektów, które wymagają szybkiego prototypowania i prostego zarządzania.
- **Bulma** - bezpłatna platforma typu open source, która zapewnia gotowe do użycia komponenty frontendowe, które można łatwo łączyć w celu tworzenia responsywnych interfejsów internetowych.
- **EmailJS** - biblioteka JavaScript, która pomaga wysyłać wiadomości e-mail przy użyciu technologii po stronie klienta. Zaletą EmailJS jest to, że nie są wymagane żadne serwery; wszystko, co należy zrobić, to podłączyć go do jednej z obsługiwanych usług e-mail, utworzyć szablon wiadomości e-mail i użyć EmailJS do uruchomienia wiadomości e-mail.

### 7.4. Źródła wiedzy programistycznej

- **StackOverflow** – internetowa społeczność programistyczna z całego świata. Opisane tam artykuły stanowiły pomoc przy rozwiązywaniu problemów programistycznych jak i przy nauce pisania czytelnego kodu.

## 8. Podsumowanie

Niniejsza praca magisterska dotyczyła stworzenia aplikacji internetowej do analizy danych pogodowych i prognozowania pogody. Celem projektu było opracowanie nowoczesnego i funkcjonalnego narzędzia, które umożliwi Użytkownikom łatwe przeglądanie danych pogodowych oraz samodzielne prognozowanie pogody.

Aplikacja została stworzona zgodnie z początkowymi założeniami. Wszystkie aspekty jej funkcjonalności realizują początkowe założenia. Aplikacja Pozwala na przeglądanie danych pogodowych. Aplikacja pobiera dane pogodowe z zewnętrznego źródła i prezentuje je w przejrzysty sposób, umożliwiając Użytkownikom łatwe sprawdzenie pogody w dowolnym miejscu na świecie. Za pomocą aplikacji można Analizować dane pogodowe. Aplikacja posiada narzędzia do analizy danych pogodowych, takie jak wykresy i tabele, które pozwalają Użytkownikom na identyfikację trendów i zależności w danych. Użytkownik ma także możliwość prognozowania pogody. Aplikacja wykorzystuje modele uczenia maszynowego do prognozowania pogody na różne dni. Użytkownicy mogą generować prognozy dla konkretnych lokalizacji i sprawdzać szczegółowe informacje, takie jak temperatura, opady, wiatr.

Aplikacja spełnia wszystkie założone cele i stanowi funkcjonalne narzędzie do analizy danych pogodowych i prognozowania pogody. Aplikacja internetowa do analizy danych pogodowych może zostać rozbudowana na wiele sposobów. Można na przykład zintegrować aplikację z dodatkowymi źródłami danych pogodowych, np. z lokalnymi stacjami meteorologicznymi lub satelitami. Można także rozszerzyć możliwości analizy, czyli Zaimplementowanie bardziej zaawansowanych narzędzi do analizy danych, takich jak modele statystyczne i algorytmy uczenia maszynowego. Istnieje także możliwość Udoskonalenia modeli prognozowania, czyli opracowanie bardziej precyzyjnych modeli prognozowania pogody wykorzystując zaawansowane techniki uczenia maszynowego.

W trakcie realizacji projektu zastosowałem najnowsze technologie po stronie frontendu, takie jak React.js z narzędziami Redux Toolkit oraz frameworkiem Bulma, oraz technologie po stronie backendu takie jak Express.js oraz Flask, co pozwoliło na stworzenie szybkiej i responsywnej w pełni działającej aplikacji. Przeprowadzone testy poprzez korzystanie z aplikacji oraz narzędzia Postman wykazały, że aplikacja jest stabilna i spełnia wszystkie wymagania funkcjonalne.

Proces stworzenia aplikacji internetowej do analizy danych pogodowych zakończył się sukcesem. Otrzymano nowoczesne i funkcjonalne narzędzie, które umożliwia Użytkownikom łatwy dostęp do danych pogodowych i samodzielne prognozowanie pogody. Aplikacja ta jest stabilna i spełnia wszystkie założone wymagania funkcjonalne.

Aplikacja stanowi wartościowe narzędzie dla osób, które potrzebują dostępu do rzetelnych danych pogodowych i chcą samodzielnie prognozować pogodę. Może być ona wykorzystywana przez osoby prywatne, jak również przez firmy i instytucje działające w różnych branżach, np. rolnictwie, turystyce, lotnictwie czy transporcie.

## Bibliografia

1. Piero Paialunga, rok 2024, „Weather forecasting with Machine Learning, using Python”,  
<https://towardsdatascience.com/weather-forecasting-with-machine-learning-using-python-55e90c346647>
2. Jon Reilly, rok 2024, „Using Machine Learning for Accurate Weather Forecasts in 2023”,  
<https://www.akkio.com/post/weather-prediction-using-machine-learning>.
3. The Weather Company, rok 2023, „AI in weather forecasting, prediction and communication”,  
<https://www.weathercompany.com/blog/ai-in-weather-forecasting-prediction-and-communication/> .
4. Maarten Laureyssen, rok 2022, „Developing a Weather Model with Machine Learning in Python”,  
<https://medium.com/@maarten/developing-a-weather-model-with-machine-learning-in-python-ed1b741dc553> .
5. Learn AI, rok 2023, „Random Forest Algorithm”,  
<https://medium.com/@divakar1591/random-forest-algorithm-772d3e5568af>
6. Jawara Gordon, rok 2023, „Getting Started with Data Visualization and Chart.js”,  
<https://medium.com/@jawaragordon/data-visualization-with-chart-js-a-step-by-step-guide-cd2aedc6a378>
7. Mauro Di Pietro, rok 2021, „Web Development with Python: Dash (complete tutorial)”,  
<https://towardsdatascience.com/web-development-with-python-dash-complete-tutorial-6716186e09b3>
8. Anshumaan Tiwari, rok 2023, „Fetching Weather Data with Node.js and Express: A Beginner's Guide”,  
<https://medium.com/javarevisited/fetching-weather-data-with-node-js-and-express-a-beginners-guide-9420e4cc2f8b>
9. Benedict Neo, rok 2023, „Create Beautiful Graphs with Python”,  
<https://medium.com/geekculture/create-beautiful-graphs-with-python-4235f50b2adb>
10. Tristan Joshua Alba, rok 2020, „5 Reasons why I'm learning Web Development, as a Data Scientist”,  
<https://towardsdatascience.com/5-reasons-why-im-learning-web-development-as-a-data-scientist-33bd61601b62>

11. Muhammad Usman, rok 2020, „Data Visualization With React & Chart.js”  
**<https://medium.com/swlh/data-visualization-with-react-chart-js-be5e238bc302>**
12. Jordan Walke., rok 2024, „React – A JavaScript library for building user interfaces”,  
**<https://reactjs.org/docs/getting-started.html>**
13. TJ Holowaychuk, rok 2024, „Express.js – Node.js web application framework”,  
**<http://expressjs.com/>**
14. Jeremy Thomas, rok 2024, „Bulma – A modern CSS framework based on Flexbox”,  
**<https://bulma.io/documentation/>**
15. Andris Reinman, rok 2024, „EmailJS – Send your emails with a simple javascript API”,  
**<https://www.emailjs.com/docs/>**
16. Armin Ronacher, rok 2024, „Flask’s documentation”,  
**<https://flask.palletsprojects.com/en/3.0.x/>**