**Przedmiot:**

**Aplikacje internetowe 2**

**Dokumentacja projektowa aplikacji do predykcji i wizualizacji pogody na podstawie danych historycznych**

**Wykonali:**

**Rejman Konrad**

**Marzec Adam**

**Matyaszek Mateusz**

**Wojdyła Marek**

**Prowadzący: dr inż. Piotr Lasek**

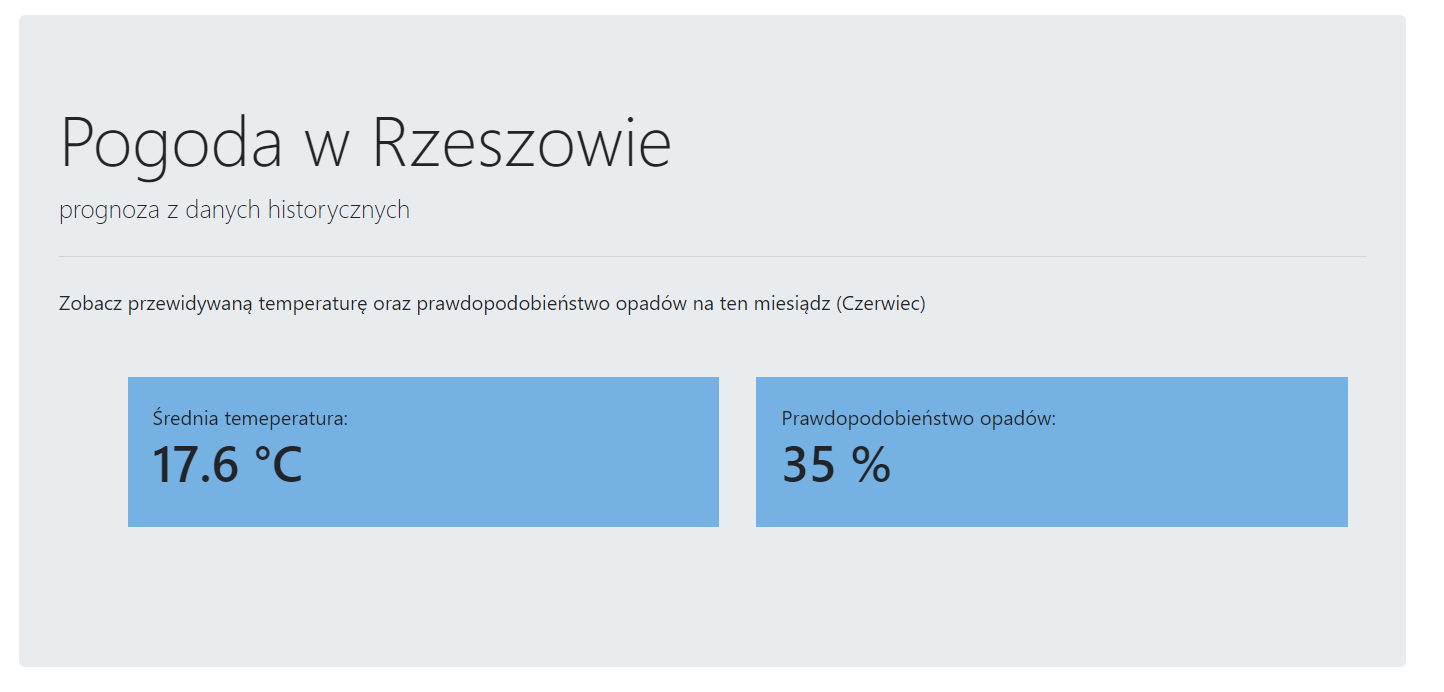
**Rzeszów 2018/2019**

# Opis projektu

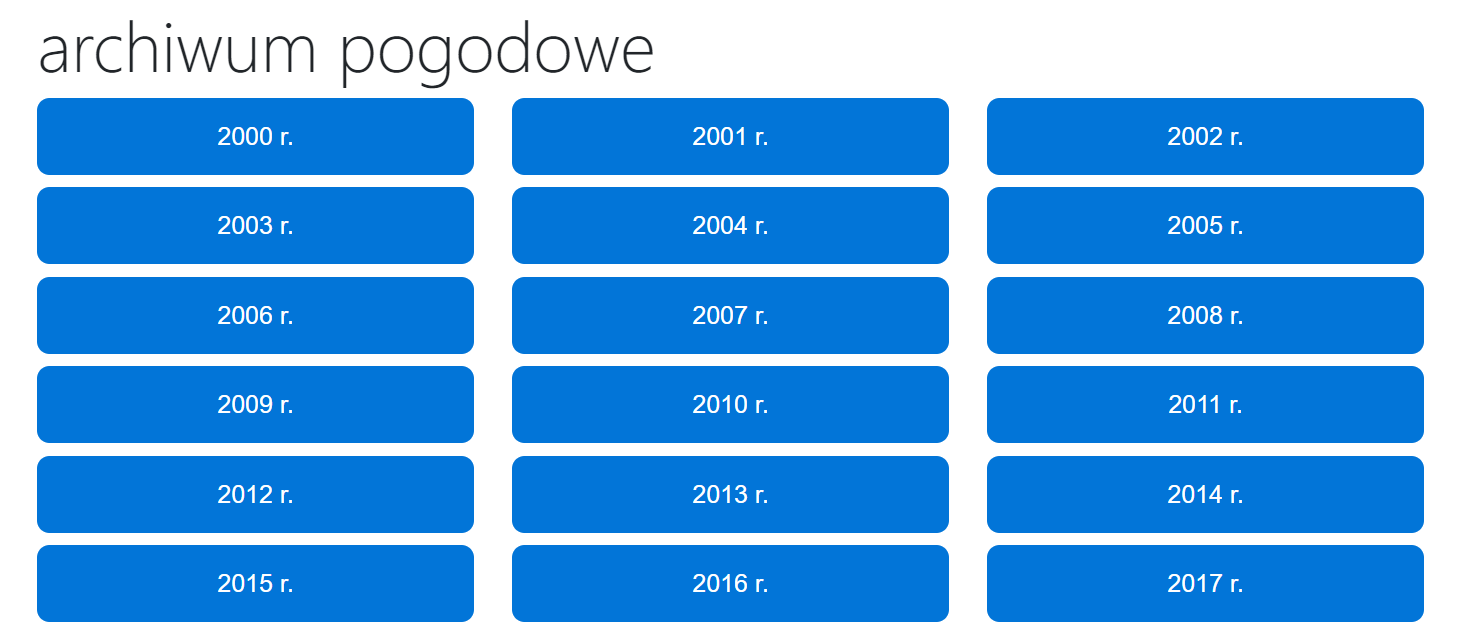
Celem projektu było stworzenie aplikacji webowej opartej na technologii Django. Aplikacja ma służyć do predykcji i wizualizacji pogody na podstawie danych historycznych. Projekt ten został zrealizowany na potrzeby przedmiotu Aplikacje internetowe 2.

# Wygląd oraz działanie aplikacji:

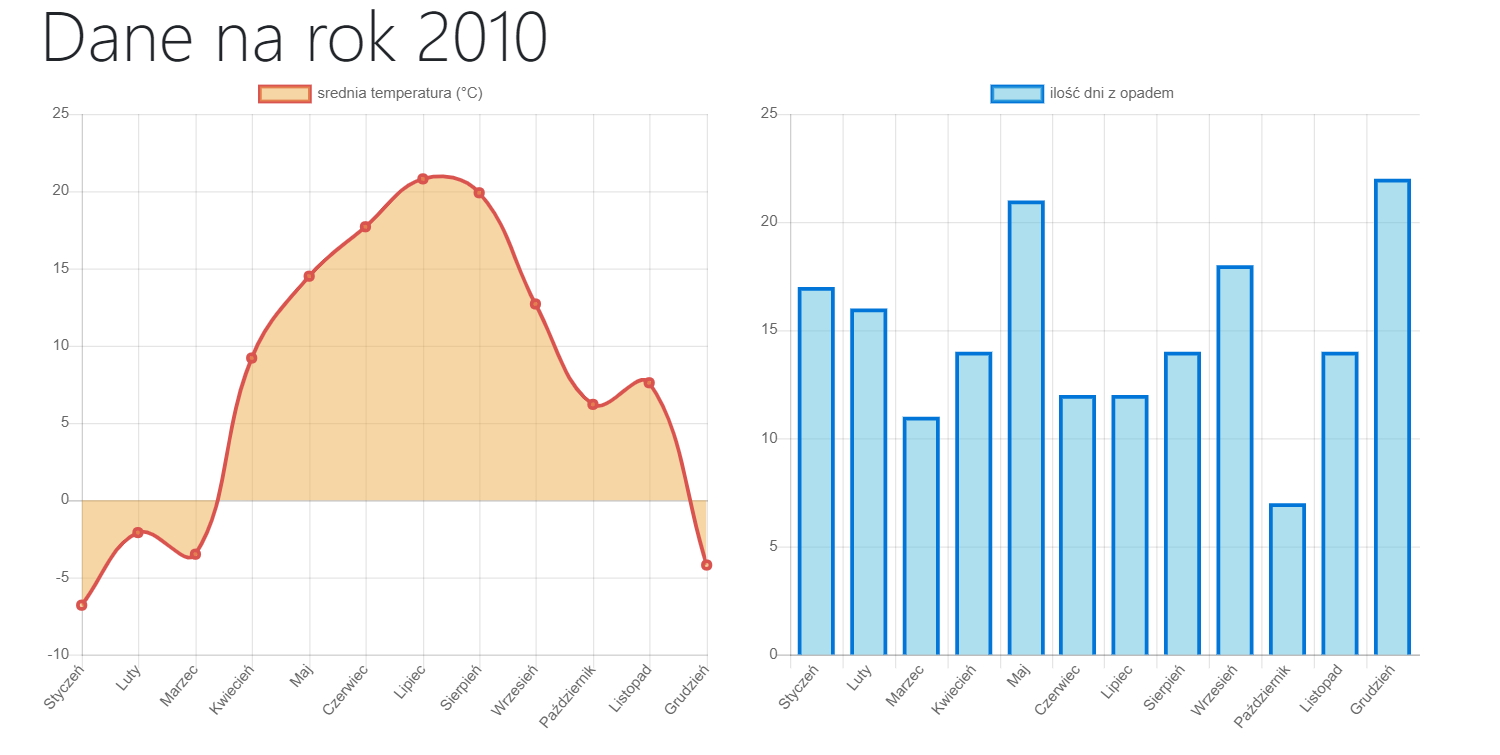
Aplikacja przewiduje pogodę na aktualny miesiąc dla miasta Rzeszów w formie (średnia temperatura na miesiąc oraz szansa opadów):



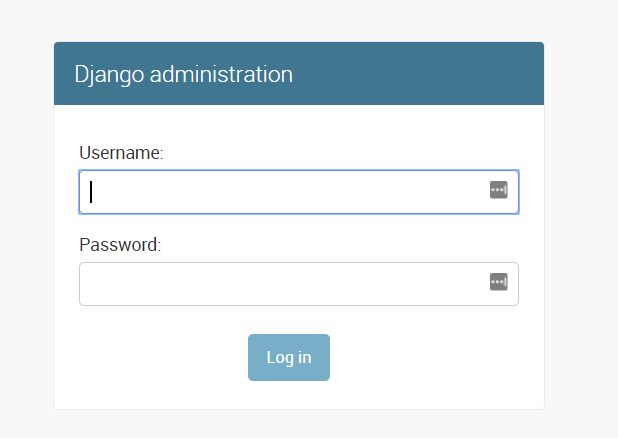
Udostępnia archiwum pogodowe dla Rzeszowa na 18 lat:



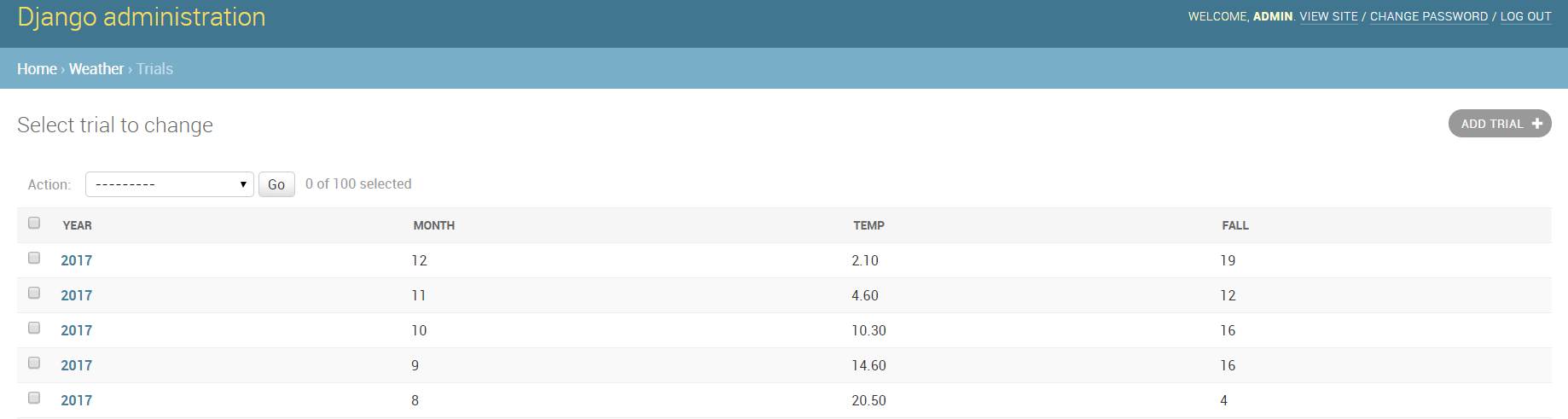
Przedstawia dane pogodowe w formie dwóch wykresów:



# Panel administracyjny:



Po zalogowaniu do panelu administracyjnego możemy zarządzać naszą bazą danych:



# Mechanika działania aplikacji:

Lokalna baza danych pogody dla miasta Rzeszów zawiera takie informacje jak: średnia temperatura oraz ilość dni z opadem w danym miesiącu w okresie 18 lat (od roku 2000 do 2017).

Program sprawdza aktualny miesiąc i na podstawie bazy danych – wylicza średnią temperaturę wszystkich temperatur z wcześniejszych lat dla tego miesiąca.

Podobnie wyliczana jest predykcja opadów. Wynik jest podawany w procentach. Średnia ilość dni z opadem jest dzielona na ilość dni danego miesiąca – w ten sposób otrzymujemy prawdopodobieństwo opadów na aktualny miesiąc.

# Technologie

W realizacji aplikacji wykorzystano następujące technologie oraz narzędzia:

## Python

To język skryptowy, interpretowany - co oznacza, że piszemy skrypt a następnie wykonujemy go za pomocą interpretera. Python jest łatwy w nauce, lecz mimo to jest bardzo potężny. Działa na wielu systemach, w tym na systemach wbudowanych. Oprócz Pythona (napisanego w C) istnieje Jython napisany w Javie i działający wszędzie tam, gdzie jest Java oraz Iron Python napisany w .NET i działający wszędzie tam gdzie jest dostępne środowisko .NET Microsoftu. Python posiada bogatą bibliotekę standardową umożliwiającą wykonanie większości czynności.

## Django

To darmowy i open-source'owy framework do tworzenia aplikacji webowych, napisany w Pythonie. To taki zbiór narzędzi, dzięki którym tworzenie stron internetowych jest szybsze i łatwiejsze.

## HTML5

To skrót od *Hypertext Markup Language*, co oznacza, że jest on hipertekstowym językiem znaczników. HTML opisuje strukturę strony internetowej, a jego elementami składowymi są znaczniki. Warto pamiętać, że HTML nie jest językiem programowania, bo nie można w nim wykonywać np. obliczeń. HTML pozwala nam zbudować strukturę, czyli szkielet np. naszej strony internetowej.

## CSS3

Odpowiada za wygląd naszych stron internetowych. Używając tego języka możemy zmienić na naszych stronach między innymi kolory i wielkości fontów. Kod CSS można umieścić w oddzielnym pliku z rozszerzeniem .css i "wstawić" go poprzez specjalny tag HTML. Można też umieścić go bezpośrednio w dokumencie HTML.

## JavaScript

JavaScript to język programowania, który umożliwia wdrożenie na stronie internetowej skomplikowanych elementów, dzięki którym strona ta może nie tylko wyświetlać statyczne informacje, ale również obsługiwać zmianę treść odpowiednio do sytuacji, wyświetlać interaktywne mapy i animacje grafiki 2D/3D , wyświetlać video itd. Jest to trzecia warstwa standardowych technologii internetowych

## Bootstrap

F[reamwork](https://pl.wikipedia.org/wiki/Framework) CSS, rozwijany przez programistów [Twittera](https://pl.wikipedia.org/wiki/Twitter" \o "Twitter). Zawiera zestaw przydatnych narzędzi ułatwiających tworzenie [interfejsu graficznego](https://pl.wikipedia.org/wiki/Graficzny_interfejs_u%C5%BCytkownika) [stron](https://pl.wikipedia.org/wiki/Strona_internetowa) oraz aplikacji internetowych. Bazuje głównie na gotowych rozwiązaniach [HTML](https://pl.wikipedia.org/wiki/HTML) oraz [CSS](https://pl.wikipedia.org/wiki/Kaskadowe_arkusze_styl%C3%B3w) (kompilowanych z plików [Less](https://pl.wikipedia.org/wiki/Less_(j%C4%99zyk_arkuszy_styl%C3%B3w))[[2]](https://pl.wikipedia.org/wiki/Bootstrap_(framework)#cite_note-2)) i może być stosowany m.in. do [stylizacji](https://pl.wikipedia.org/wiki/Kaskadowe_arkusze_styl%C3%B3w) takich elementów jak teksty, formularze, przyciski, wykresy, nawigacje i innych komponentów wyświetlanych na stronie. Framework korzysta także z języka [JavaScript](https://pl.wikipedia.org/wiki/JavaScript).

## Chart.js

Biblioteka ta zawiera animowane i interaktywnych wykresy oparte na elemencie canvas (HTML5) oraz zestawu skryptów JavaScript.

# Pliki wchodzące w skład projektu:

Serwis pogody składa się z kilku widoków HTML odpowiedzialnych za wyświetlanie konkretnych informacji w zależności od zapytania URL.

* base.html – jest to bazowy plik, do którego wczytywane są konkretne widoki. Zawiera w sobie stopkę strony internetowej. Dodatkowo posiada meta tagi w sekcji head, odpowiedzialne za import skryptów Chart.js, Bootstrap oraz posiada lokalne style css.
* index.html – jest to plik zawierający widok dla strony głównej serwisu. Wczytuje on informacje na temat przewidywanej temperatury oraz prawdopodobieństwo opadów w aktualnym miesiącu. Są tam również zamieszczone linki do poszczególnych lat archiwa pogodowego.
* year.html – plik ten odpowiada za wyświetlanie archiwum pogodowego na konkretne lata w formie wykresów

Za wyświetlaniem konkretnych widoków odpowiedzialne są pliki:

* urls.py – w nim określany jest adres URL pod którym będzie wczytywany konkretny widok.
* views.py – plik, w który zdefiniowane są funkcje dla widoków. Najczęściej pełnią one funkcję kontrolerów wskazujących końcowe pliki html przekazując do nich dane.
* models.py – ten plik reprezentuję obiekty w bazie danych

baza zawiera jedną tabelę (Trials) zawierającą kolumny:

- year – rok

- month – miesiąc

- temp – średnia temperatura

- fall – ilość dni z opadem w danym miesiącu