**Kolegium Nauk Przyrodniczych  
Uniwersytet Rzeszowski**

**Przedmiot:**

**Wybrane Zagadnienia Współczesnej Informatyki**

**Tytuł projektu:**

***Aplikacja webowa do akwizycji i analizy danych z kanału RSS portalu gofin.pl (prawo pracy).***

**Wykonał:**

**Kamil Filar, Informatyka, rok III, lab1**

**Prowadzący: dr hab. inż. Krzysztof Pancerz, prof. UR**

**Rzeszów 2020**

Spis treści:

1. Opis serwisu/aplikacji internetowej ………………………………………………………………………………………… 3
   1. Główne funkcjonalności ……………………………………………………………………………………………………. 3
2. Szczegółowy opis funkcjonalności/wyglądu programu …………………………………………………………….. 4

2.1 Graficzny interfejs użytkownika ………………………………………………………………………………………….. 5

2.2 Apache Solr ………………………………………………………………………………………………………………………… 7

2.3 Występowanie danego słowa w wybranej tabeli ………………………………………………………………… 7

2.4 Lematyzacja ..………………………………………………………………………………………………………………………. 8

2.5 Przygotowanie danych ………………………………………………………………………………………………………… 9

2.6 Chmura słów ……………………………………………………………………………………………………………………….. 9

2.7 Częstotliwość występowania danego słowa ……………………………………………………………………… 11

2.8 Klasteryzacja danych ………………………………………………………………………………………………………… 12

# Opis serwisu/aplikacji internetowej

Nazwa aplikacji: Aplikacja webowa do akwizycji i analizy danych z kanału RSS portalu gofin.pl (prawo pracy).

Aplikacja została stworzona z wykorzystaniem środowiska R oraz Apache Solr. Aplikacja webowa oparta jest o zaawansowane GUI (możliwość ustawiania parametrów) zrealizowane z wykorzystaniem pakietu Shiny. Ponadto posiada funkcjonalności takie jak:

- Możliwość pobierania wiadomości i magazynowania ich w Apache Solr.

- Możliwość analizy częstości występowania słów i wizualizacji chmury słów w wiadomościach zmagazynowanych w Apache Solr.

- Możliwość klasteryzacji wiadomości zmagazynowanych w Apache Solr oraz wizualizacji klastrów.

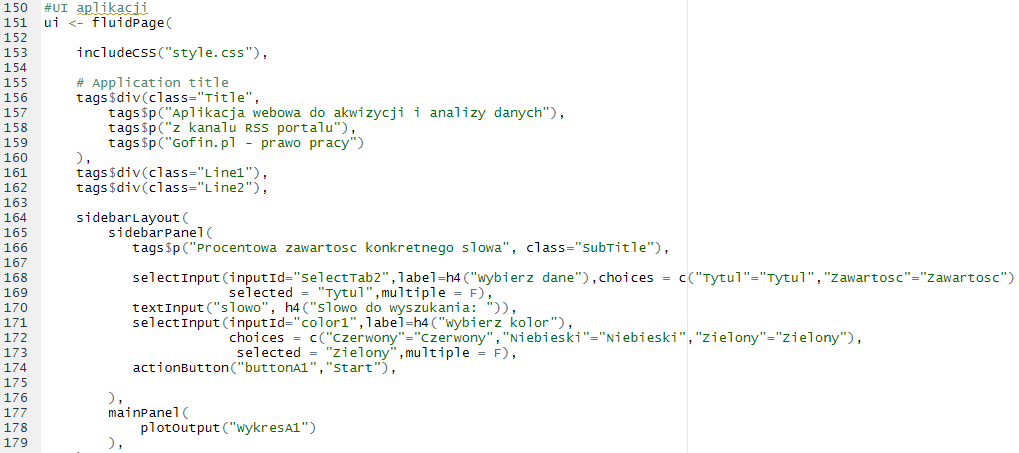
# Główne funkcjonalności

1. Pobieranie danych z kanału RSS.
2. Magazynowanie ich w Apache Solr.
3. Pobieranie danych z Apache Solr.
4. Lematyzacja tekstu.
5. Analiza występowania danego słowa w zadanym zakresie danych.
6. Tworzenie chmury słów.
7. Analiza częstości występowania słów.
8. Klasteryzacja danych.

# Szczegółowy opis funkcjonalności/wyglądu programu

# Graficzny interfejs użytkownika

Graficzny interfejs użytkownika został zrealizowany z wykorzystaniem pakietu Shiny. Jego szczegółowy wygląd został zaprezentowany poniżej:



Rys. 1



Rys. 2

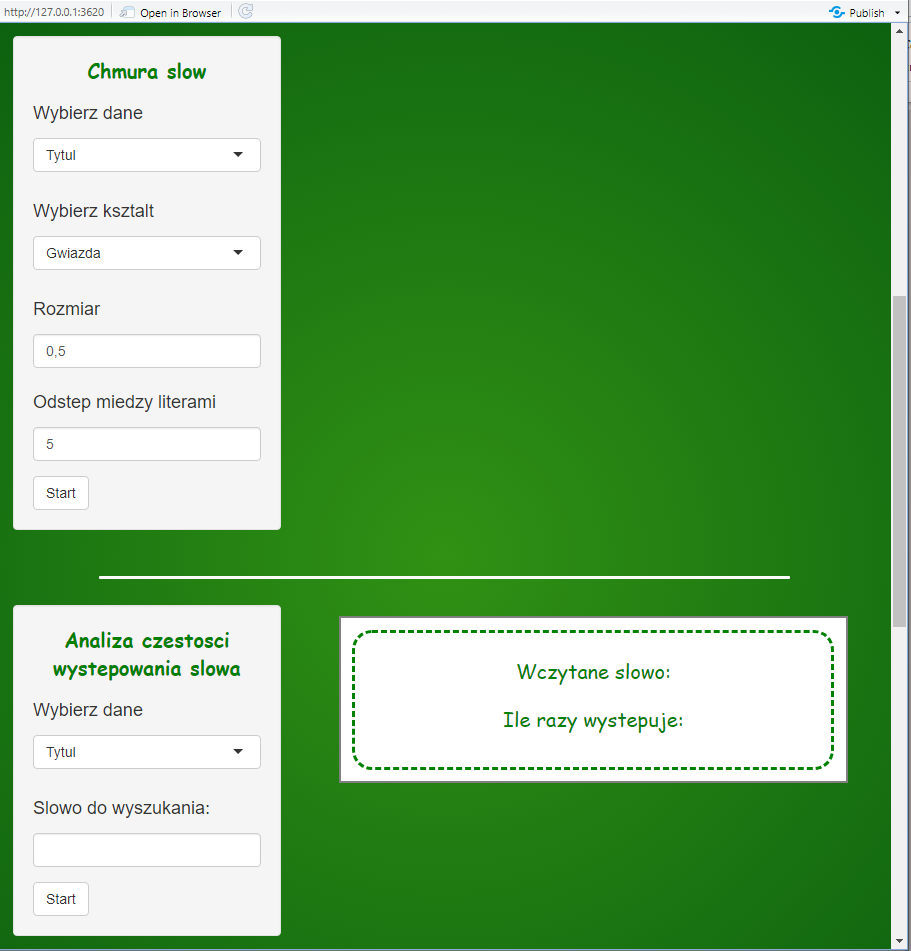
Rys. 1 przedstawia początek kodu UI. W linii 153 znajduje się załączony plik zawierający CSS, który w stanowi o wyglądzie aplikacji. W liniach od 156 do 160 znajduje się tytuł projektu, a poniżej tego są dwie linie służące do separacji między blokami w projekcie. Poniżej znajduje się pierwszy panel wchodzący w skład aplikacji. W aplikacji znajdują się cztery takie panele umożliwiające użytkownikowi analizę danych. Składają się z takich elementów jak:

* selectInput – służy do wybierania danych (168 linia Rys. 1)
* textInput – służy do wczytywania danych tekstowych (170 linia Rys. 1)
* actionButton – przycisk, który służy do uruchomienia funkcji (174 linia Rys. 1)
* numericInput – służy do wczytywania liczb z zakresu (194 linia Rys. 2)

Wygląd UI został przedstawiony na Rys. 3, Rys. 4, Rys. 5 poniżej:



Rys. 3



Rys. 4



Rys. 5

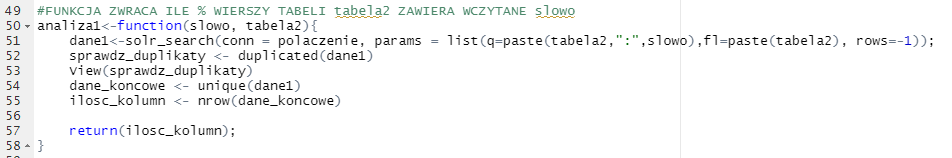
# Apache Solr



Rys. 6

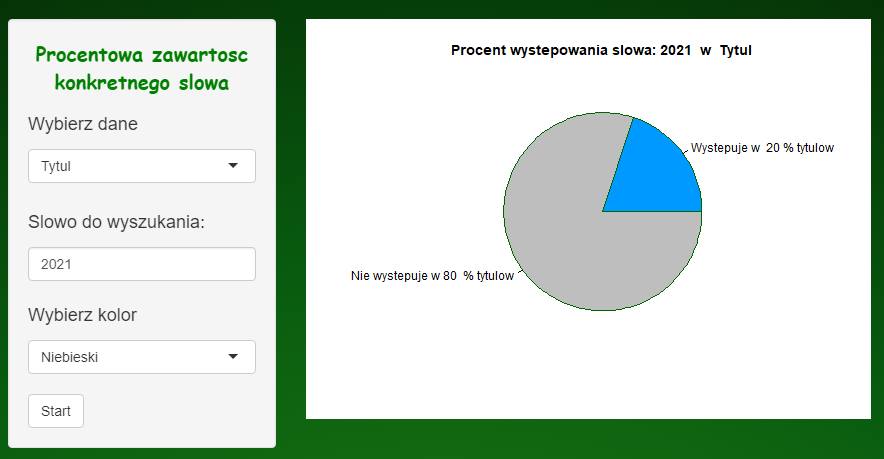
Na Rys. 6 została przedstawiona obsługa Apache Solr. W linii 21 wczytuje dane z kanału RSS za pomocą funkcji tidyfeed, następnie w liniach od 24-26 przypisuje obiektom (w celu ułatwienia sobie pracy) dane wczytane z kanału RSS, następnie w linii 34 tworzę połączenie z węzłem ProjektWZWI, tworzę obiekt do przechowywania danych i w linii 44 dodaję do Apache Solr moje dane. Pobieranie danych z Solr-a zostało opisane przy funkcjach, które to wykorzystują.

# Występowanie danego słowa w wybranej tabeli



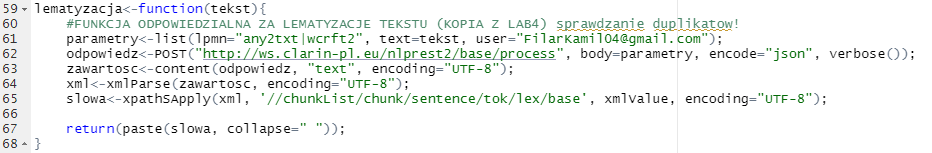
Rys. 7

Funkcja analiza1 służy do zliczenia ilości kolumn, w jakich zawiera się dane słowo. Funkcja przyjmuje dwa prametry: *slowo*  oraz  *tabela2*. Parametr słowo jest wczytywany w UI przez użytkownika i za pomocą logiki serwera zostaje przekazane do funkcji jako parametr przy wywołaniu, podobnie jest z drugim parametrem. Funkcja sprawdza czy są duplikaty (linia 52-54) w wczytanych danych z Apache Solr (samo wczytywanie jest zrealizowane za pomocą funkcji solr\_search w linii 51) i wybiera tylko unikatowe wartości. Zwrócone wartości to ilość kolumn. Na Rys. 8 (poniżej) został zaprezentowany wygląd wyniku działania funkcji:



Rys. 8

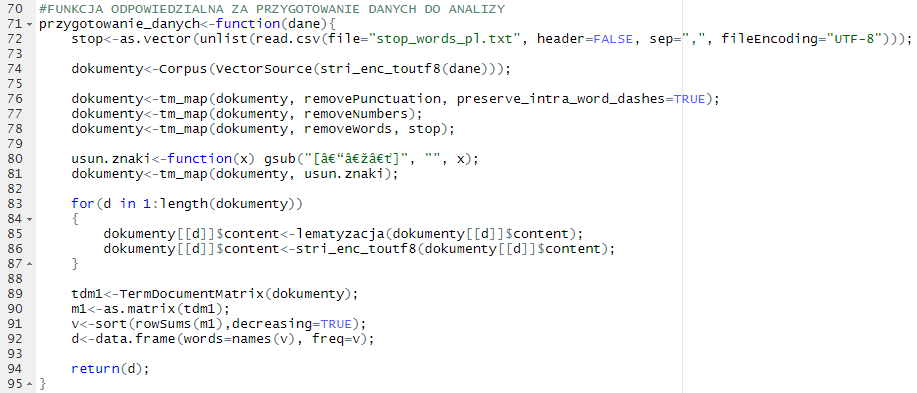
# Lematyzacja tekstu



Rys. 9

Funkcja służąca do lematyzacji tekstu została przedstawiona na Rys. 9. Jest to funkcja, która została przedstawiona w laboratorium 4 podczas zajęć.

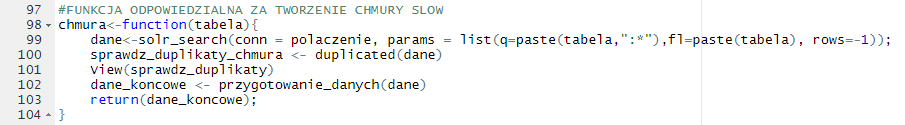
# Przygotowanie danych do analizy

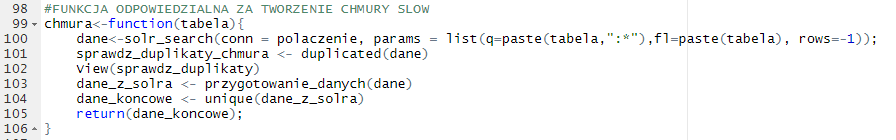


Rys. 10

Na Rys. 10 została przedstawiona funkcja (wykorzystana z laboratorium 4), która przyjmuje parametr wejściowy *dane*, które są poddawane działaniom wewnątrz funkcji. Funkcja zwraca tabele z uporządkowanymi danymi w kolumnie słowa oraz częstotliwość ich występowania.

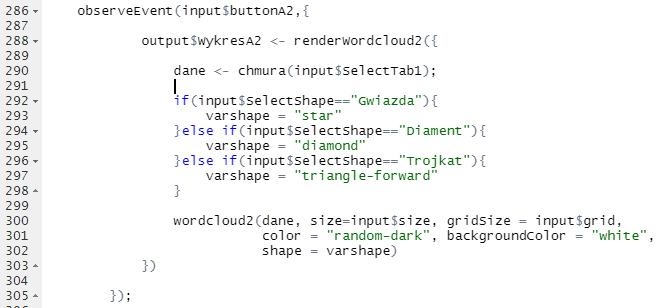
# Chmura słów





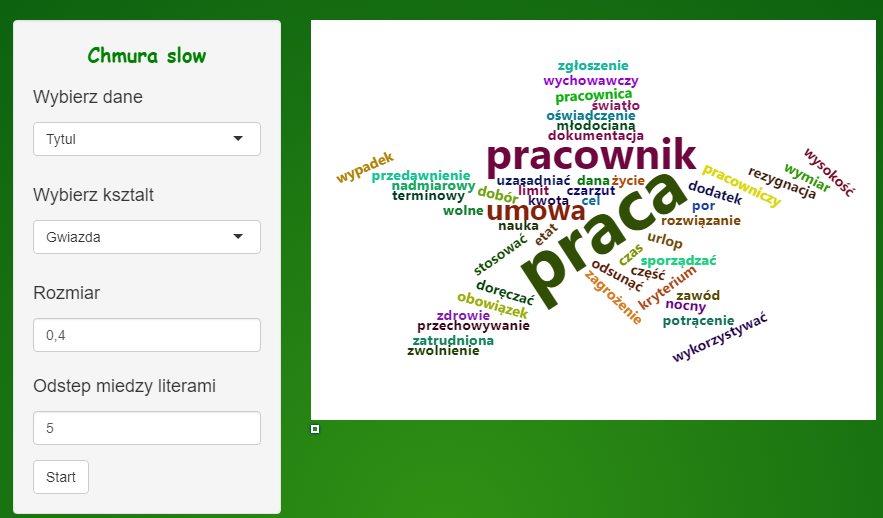
Rys. 11

Funkcja *chmura* przyjmuje parametr tabela wczytany z UI przez użytkownika. Parametr wskazuje na tabelę z której solr pobiera dane do dalszego przetworzenia (linia 100 Rys.11). Następnie dane są sprawdzane pod kątem występowania duplikatów, a kolejnym krokiem jest zwrócenie unikatowych danych końcowych, które trafiają do przetworzenia w logice serwera (Rys. 12) gdzie w linii 300 została zastosowana funkcja tworząca chmurę słów (worldcloud2).



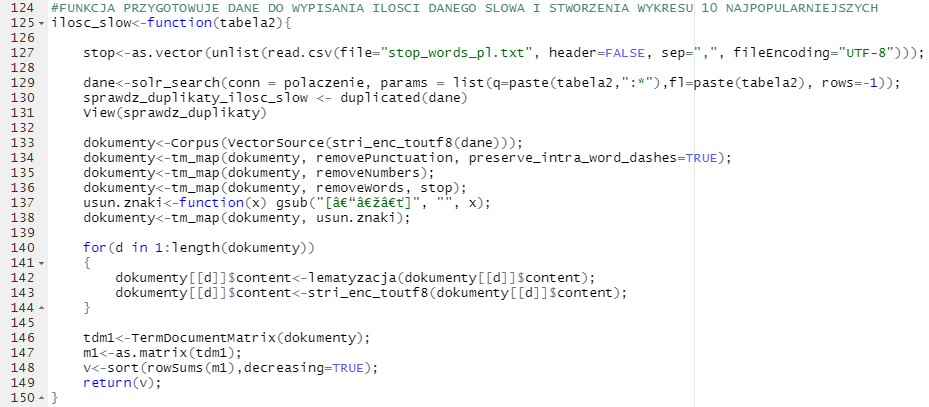
Rys. 12

Wynik działania tej funkcji można zobaczyć na Rys. 13.



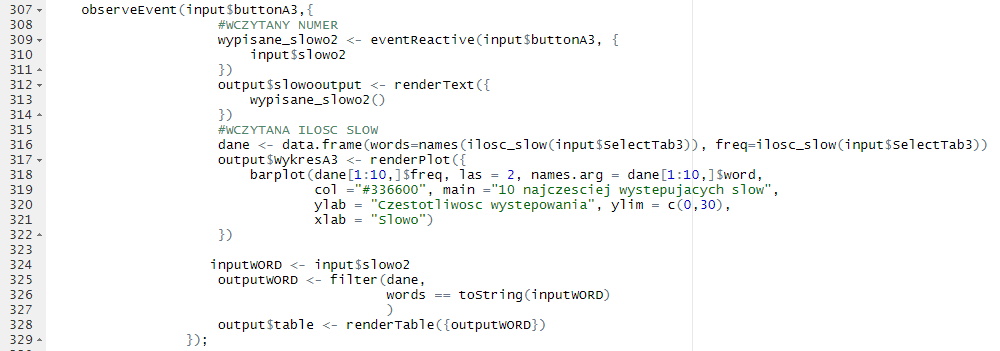
Rys. 14

# Częstotliwość występowania danego słowa



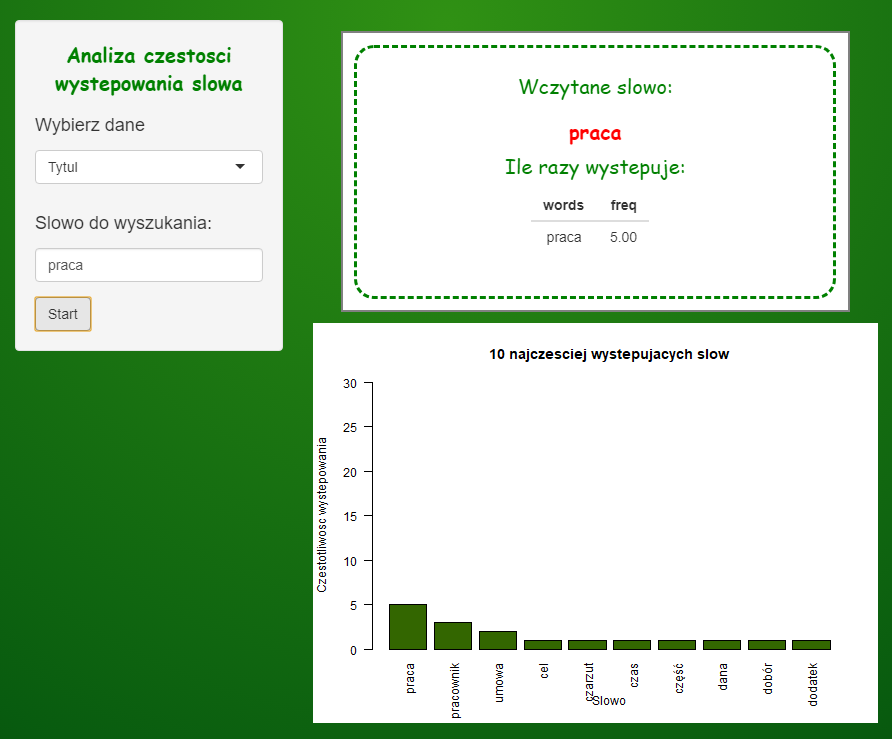
Rys. 15

Funkcja która przygotowuje dane do wyświetlenia danego słowa została przedstawiona na Rys. 15. Wykorzystuje funkcje lematyzującą tekst i zwraca posortowana i zliczona ilość słów. Następnie trafia do logiki serwera (Rys. 16) gdzie dane są przetwarzane. W liniach od 317 do 322 jest zawarty wykres, który wyświetla 10 najbardziej popularnych słów (wg. Występowania). W linii 328 Rys. 16 znajduje się tabela, która jest wyświetlana w UI.



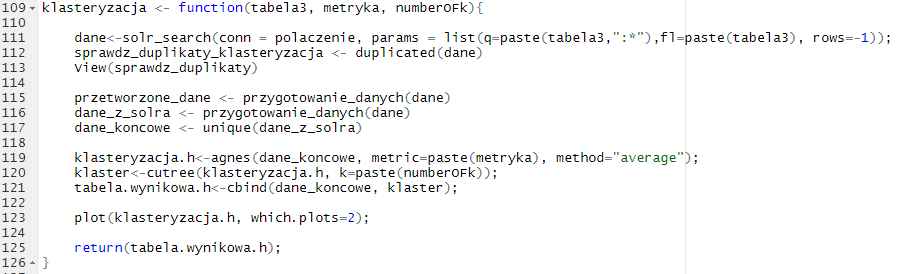
Rys. 16

Wygląd panelu z analizą częstotliwości występowania słowa został przedstawiony na Rys. 17.



Rys. 17

# Klasteryzacja danych

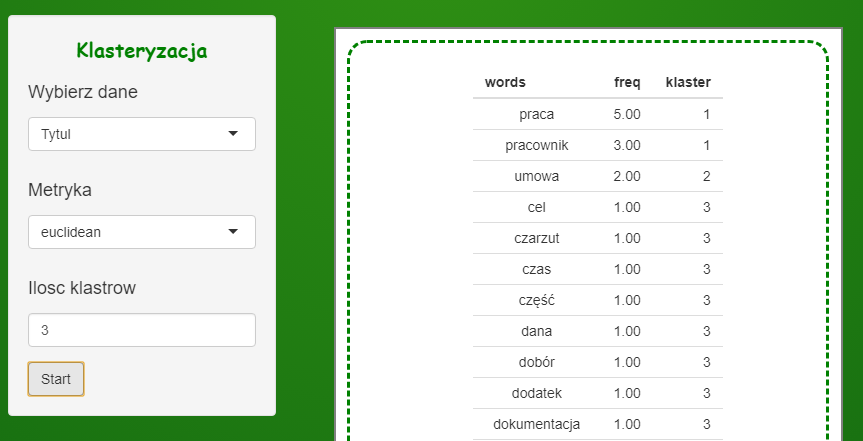


Rys. 18

W projekcie wykorzystałem klasteryzację hierarchiczną,. Funkcja klasteryzacja przyjmuje trzy parametry *tabela3*, *metryka*, *numberOFk*. Te parametry są wczytywane w UI przez użytkownika. Funkcja zwraca tabele z wynikami, która jest przetwarzana (Rys. 19) i wyświetlana (Rys. 20).



Rys. 19



Rys. 20