

**Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie**

Wydział Informatyki, Elektroniki i Telekomunikacji

KATEDRA INFORMATYKI



DOKUMENTACJA UŻYTKOWNIKA

TOMASZ KASPRZYK, DANIEL OGIELA, JAKUB STĘPAK

**SYSTEM OBLICZAJĄCY WYNIKI WYBORÓW DLA
UOGÓLNIENIA SYSTEMU K-BORDA**

PROMOTOR:

dr hab. inż. Piotr Faliszewski

Kraków 2016

Spis treści

1. Wstęp	3
2. Instrukcja uruchomienia systemu	4
2.1. Przygotowanie środowiska	4
2.1.1. Język	4
2.1.2. Repozytorium	4
2.1.3. Biblioteki	4
2.1.4. Baza danych	5
2.2. Uruchomienie serwera	5
3. Instrukcja obsługi systemu	6
3.1. Rejestracja	6
3.2. Logowanie	7
3.3. Strona główna	7
3.4. Lista wyborów	8
3.5. Tworzenie wyborów	8
3.6. Określenie kandydatów i wyborców	9
3.6.1. Wczytanie danych z pliku	9
3.6.2. Generacja danych z rozkładu normalnego	10
3.7. Szczegóły wyborów	11
3.7.1. TROLL LINE - EVERYONE BELOW WAS TROLLED	11
3.8. Szybkość i dokładność wykonywanych obliczeń	12
4. Opis modułów	13

1. Wstęp

Niniejszy podręcznik opisuje sposób użytkowania systemu Election Computing System, powstałego w ramach pracy inżynierskiej realizowanej na Akademii Górniczo-Hutniczej w Krakowie.

System umożliwia obliczenie wyników wyborów w uogólnionym systemie wyborczym k-Borda. Szczegóły na temat tego systemu wyborczego można przeczytać w Przewodniku po projekcie.

2. Instrukcja uruchomienia systemu

Działającą aplikację można przetestować na stronie:

<https://election-computing-system.herokuapp.com>.

Ze względu na ograniczenia na ilość rekordów w bazie danych narzucone przez darmową wersję Heroku, nie będzie można dodać tam zbyt „dużych” wyborów (ograniczona jest liczba głosujących i kandydatów). Aby korzystać z pełnych możliwości aplikacji, należy uruchomić ją na własnym komputerze. Dalsza część tego rozdziału opisuje jak to wykonać.

2.1. Przygotowanie środowiska

2.1.1. Język

System jest aplikacją internetową opartą o framework Django napisaną w języku Python 2.7. Aby uruchomić aplikację, należy uprzednio zainstalować na swoim komputerze interpreter Pythona.

2.1.2. Repozytorium

Repozytorium projektu znajduje się pod adresem:

<https://github.com/jakubste/election-computing-system>. Projekt należy sklonować używając programu Git

```
$ git clone git@github.com:jakubste/election-computing-system.git
```

lub ściągnąć jako ZIP bezpośrednio z GitHuba i rozpakować w wybranym miejscu.

2.1.3. Biblioteki

Do instalacji bibliotek zaleca się używanie mechanizmu `virtualenv`, który separuje środowiska uruchomieniowe dla poszczególnych projektów. Autorzy projektu zalecają też dla wygody wykorzystanie `virtualenvwrapper`'a. Do instalacji bibliotek można użyć programu `pip`. Informacje o wymaganych bibliotekach są zawarte w pliku `requirements.txt`.

```
$ mkvirtualenv inz
```

```
$ pip install -r requirements.txt
```

2.1.4. Baza danych

Aplikacja domyślnie jest skonfigurowana do użycia z bazą danych dostarczaną przez Heroku (PostgreSQL). Dla ułatwienia zostanie przedstawiony sposób konfiguracji z użyciem SQLite3. O konfiguracji dostępu do innych baz danych można przeczytać w dokumentacji Django.

W celu skonfigurowania swojej bazy danych należy w katalogu `ecs` utworzyć plik `local_settings.py` i umieścić tam następujący kod:

```
from settings import *
```

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.sqlite3',  
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),  
    }  
}
```

Następnie w katalogu głównym programu należy wykonać polecenie:

```
$ ./manage.py migrate
```

2.2. Uruchomienie serwera

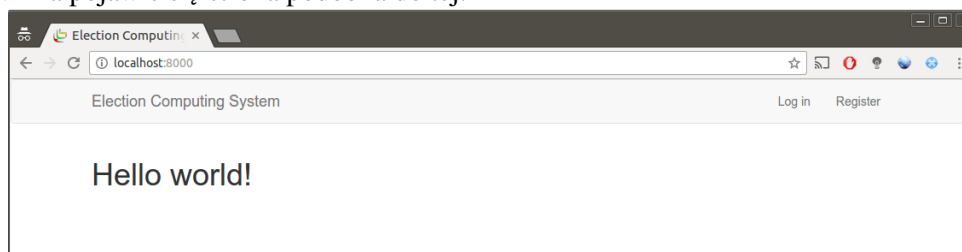
W tym momencie aplikacja powinna być gotowa do działania. W katalogu głównym programu należy wykonać polecenie:

```
$ ./manage.py run_server
```

Po otwarciu w przeglądarce internetowej adresu:

```
http://localhost:8000/
```

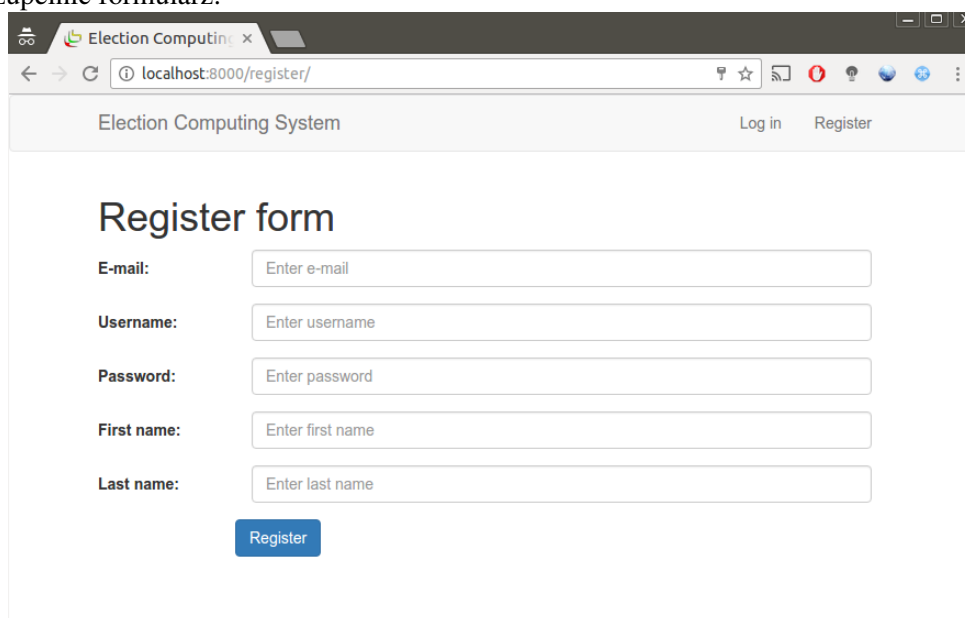
powinna pojawić się strona podobna do tej:



3. Instrukcja obsługi systemu

3.1. Rejestracja

Aby korzystać z systemu, należy utworzyć konto. W tym celu należy z menu wybrać opcję rejestracji i uzupełnić formularz:



The screenshot shows a web browser window with the title "Election Computing System". The address bar shows "localhost:8000/register/". The page has a header with "Election Computing System" on the left and "Log in" and "Register" links on the right. The main content area is titled "Register form" and contains five input fields with labels: "E-mail:", "Username:", "Password:", "First name:", and "Last name:". Each field has a placeholder text: "Enter e-mail", "Enter username", "Enter password", "Enter first name", and "Enter last name". Below the fields is a blue "Register" button.

Election Computing System

Log in Register

Register form

E-mail:

Username:

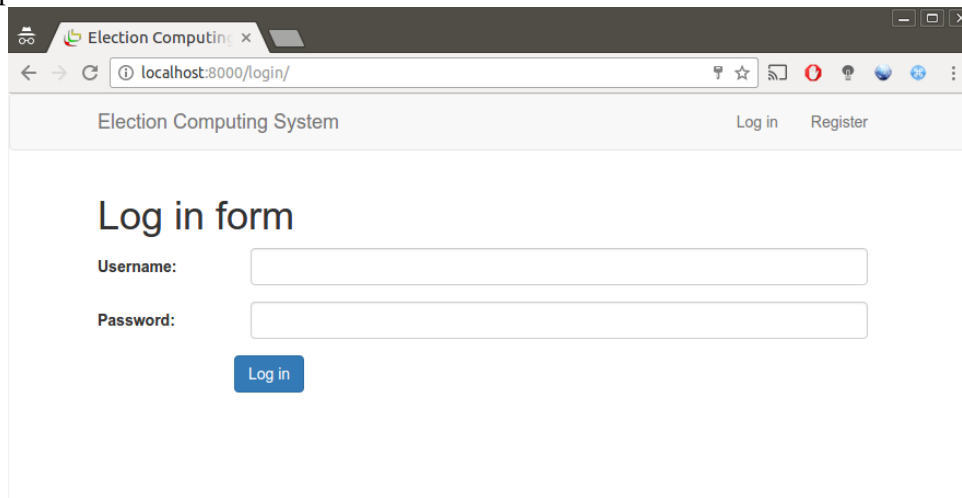
Password:

First name:

Last name:

3.2. Logowanie

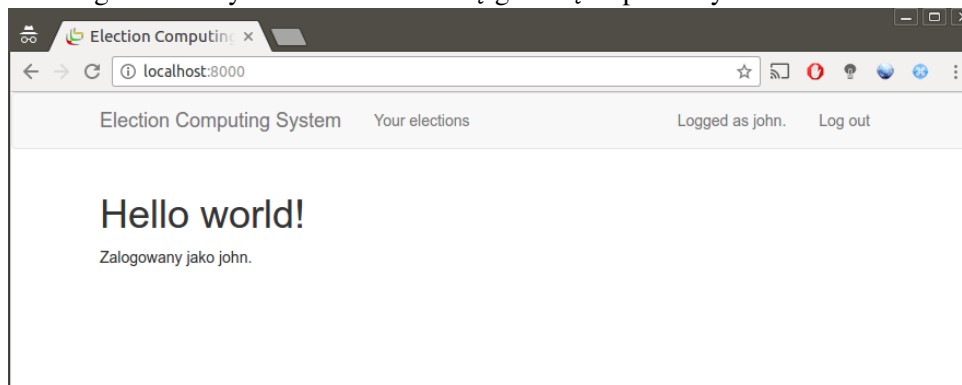
W celu zalogowania się na wcześniej utworzone konto należy wybrać z menu opcję logowania i wypełnić formularz:



The screenshot shows a web browser window with the title "Election Computing System". The address bar shows "localhost:8000/login/". The page has a header with "Election Computing System" on the left and "Log in" and "Register" links on the right. The main content area is titled "Log in form" and contains two input fields: "Username:" and "Password:". Below the password field is a blue "Log in" button.

3.3. Strona główna

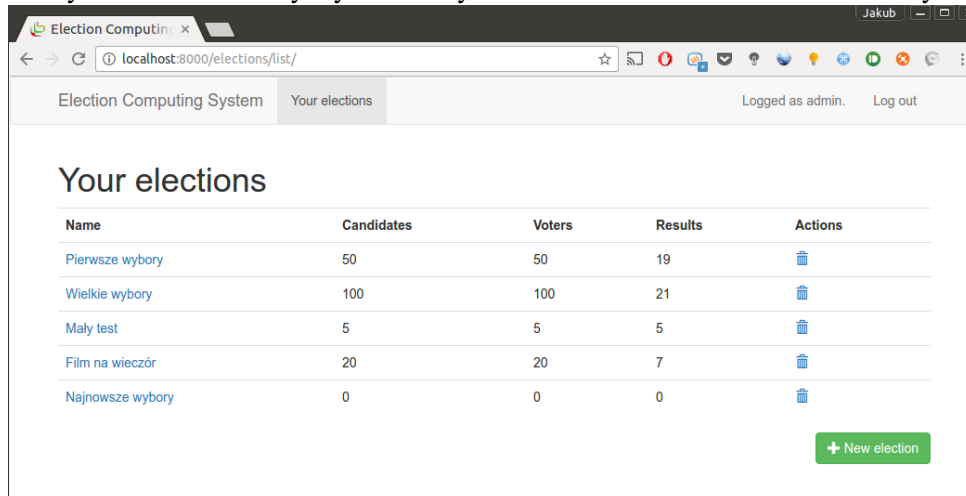
Po zalogowaniu użytkownik widzi stronę główną z opisem systemu:



The screenshot shows the same web browser window after login. The address bar now shows "localhost:8000". The header now includes "Your elections" and "Logged as john." next to "Log out". The main content area displays "Hello world!" in a large font, with "Zalogowany jako john." in a smaller font below it.

3.4. Lista wyborów

Po wybraniu z menu listy wyborów użytkownik ma możliwość zobaczenia listy swoich wyborów:



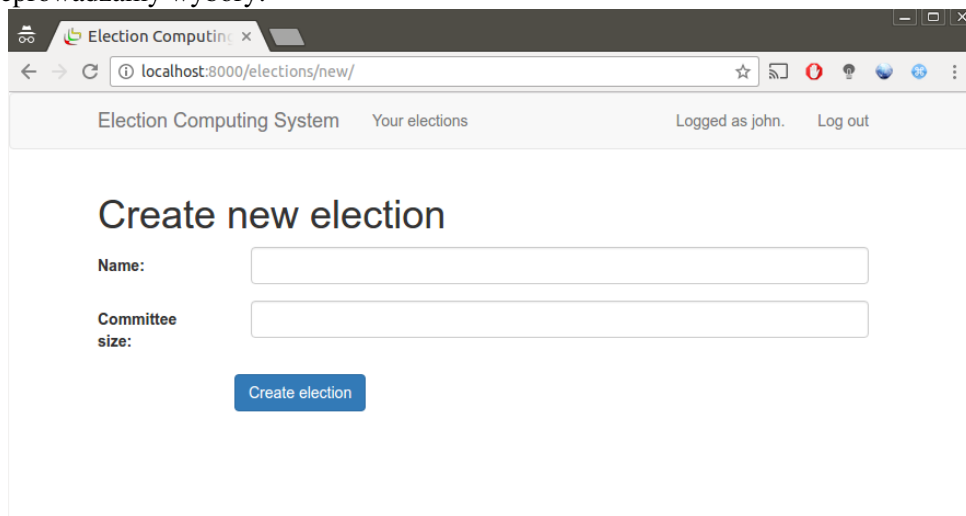
Name	Candidates	Voters	Results	Actions
Pierwsze wybory	50	50	19	
Wielkie wybory	100	100	21	
Mały test	5	5	5	
Film na wieczór	20	20	7	
Najnowsze wybory	0	0	0	

[+ New election](#)

Stąd użytkownik ma możliwość przejścia do tworzenia nowych wyborów.

3.5. Tworzenie wyborów

Aby utworzyć nowe wybory należy podać ich nazwę oraz określić rozmiar komitetu do którego przeprowadzamy wybory:



Create new election

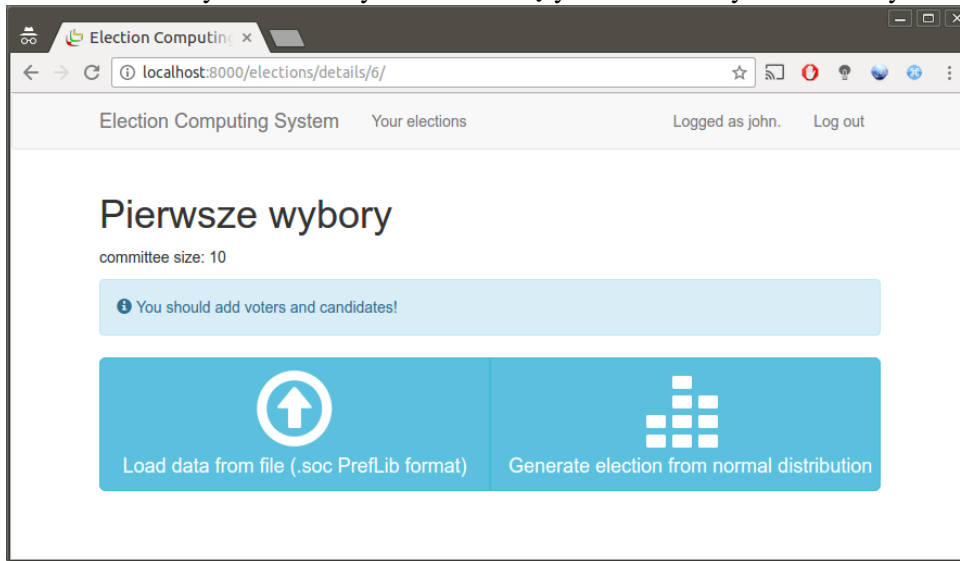
Name:

Committee size:

[Create election](#)

3.6. Określenie kandydatów i wyborców

Po utworzeniu wyborów należy określić biorących udział w wyborach kandydatów oraz wyborców.

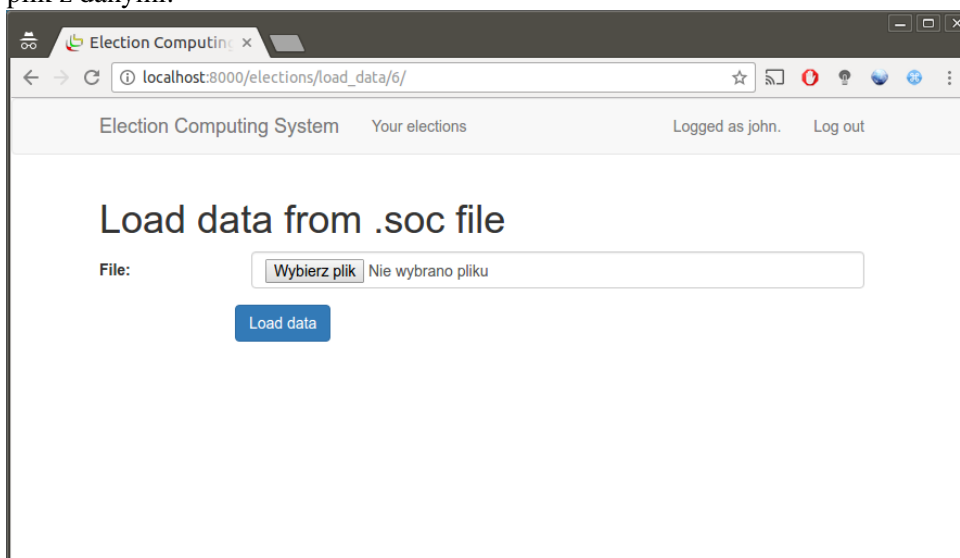


Użytkownik ma do wyboru dwie opcje:

- wczytać dane z pliku w formacie .soc określonym przez PrefLib,
- wygenerować losowy układ wyborców w oparciu o rozkład naturalny.

3.6.1. Wczytanie danych z pliku

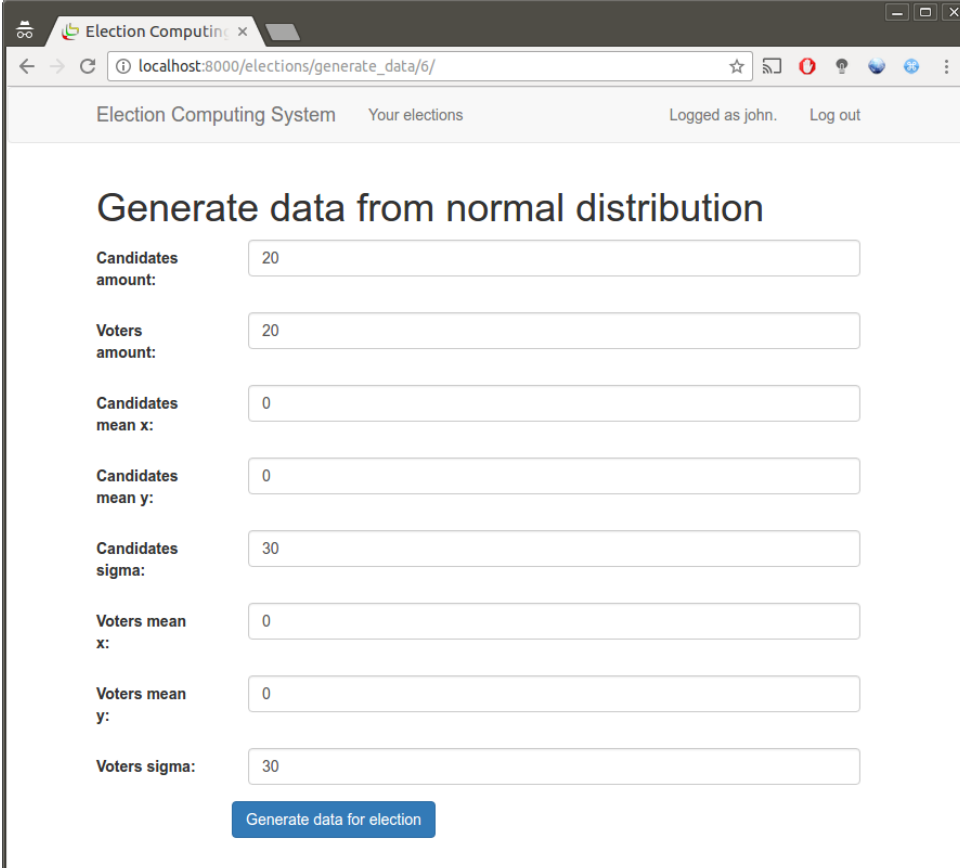
Aby wczytać dane dotyczące wyborów z pliku należy wybrać odpowiednią opcję a następnie wskazać plik z danymi:



Przykładowe pliki z danymi można pobrać ze strony projektu PrefLib. Pośród tych plików można m.in. znaleźć rzeczywiste preferencje studentów naszego kierunku w wyborach przedmiotów obieralnych.

3.6.2. Generacja danych z rozkładu normalnego

Aby wygenerować dane losowe na podstawie rozkładu normalnego należy wybrać odpowiednią opcję, a następnie uzupełnić formularz:



The screenshot shows a web browser window with the URL `localhost:8000/elections/generate_data/6/`. The page title is "Election Computing System" and it shows the user is logged in as "john". The main heading is "Generate data from normal distribution". The form contains the following fields:

Field Label	Value
Candidates amount:	20
Voters amount:	20
Candidates mean x:	0
Candidates mean y:	0
Candidates sigma:	30
Voters mean x:	0
Voters mean y:	0
Voters sigma:	30

At the bottom of the form is a blue button labeled "Generate data for election".

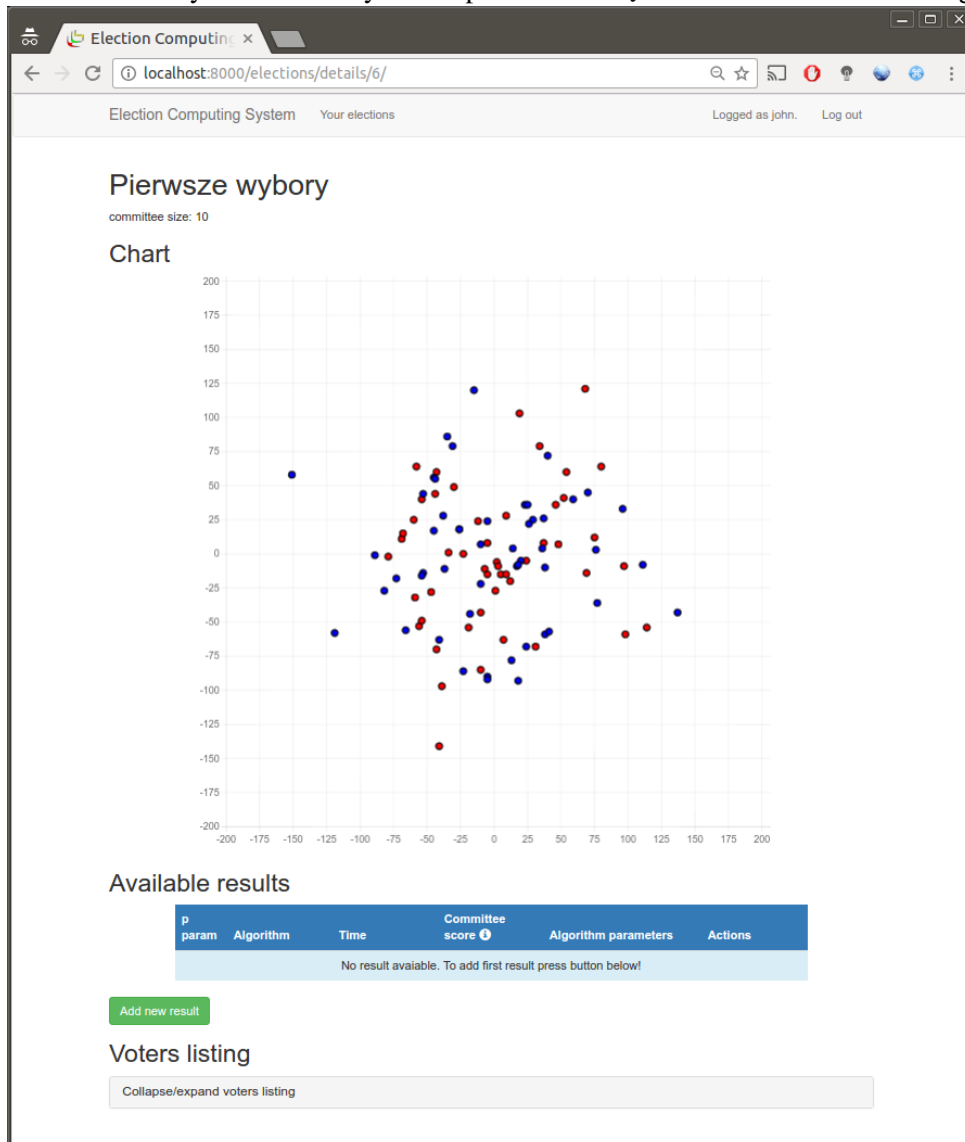
W formularzu należy określić liczbę kandydatów i głosujących, punkt centralny, wokół którego będą skupione ich „poglądy” oraz sigmę, czyli średni „rozstrzał” poglądów.

Koncepcyjnie ten sposób generacji danych odzwierciedla mapę poglądów politycznych. Porządek preferencji kandydatów wśród wyborców określa ich odległość euklidesowa w układzie współrzędnych.

W dalszej części instrukcji będziemy posługiwali się częściej przykładem z wyborami wygenerowanymi w ten sposób.

3.7. Szczegóły wyborów

Po wczytaniu danych pokaże się ekran z szczegółami wyborów:



Wykres będzie widoczny tylko w przypadku wyborów wygenerowanych z rozkładu normalnego.

Poniżej można zobaczyć spis dostępnych rozstrzygnięć wyborów oraz odnośnik do formularza dodawania nowego wyniku.

Poniżej, jeśli liczba wyborców nie jest na tyle duża, że utrudniałoby to używanie strony przez zbytne obciążenie przeglądarki, widać listing wyborców. Szczególnie przydatny do sprawdzenia czy poprawnie zostały wczytane dane z plików .soc. W przypadku danych z rozkładu normalnego jego przydatność jest ograniczona.

3.7.1. TROLL LINE - EVERYONE BELOW WAS TROLLED

Zanim wprowadzone zostanie pojęcie uogólnionego systemu k-Borda warto przypomnieć wzór na normę ℓ_p

Norma ℓ_p

Wówczas, w uogólnionej wersji metody k-Borda, funkcja zadowolenia f_{kB} zostaje uzależniona również od parametru p z powyższego wzoru. Norma liczona jest z wyników według Bordy, $\beta(i)$. Wzór uogólniony funkcji zadowolenia przyjmuje zatem postać:

$$f_{\ell_p B}(p, (i_1, \dots, i_k)) = p[(i_1)]^p + [(i_2)]^p + \dots + [(i_k)]^p$$

Systemy k-Borda i Cahmberlin'a-Courant'a są szczególnymi przypadkami zdefiniowanego powyżej systemu ℓ_p -Borda:

Dla $p = 1, l_1$

$$f_{\ell_p B}(1, (i_1, \dots, i_k)) = \beta(i_1) + \beta(i_2) + \dots + \beta(i_k) = f_{kB}(i_1, \dots, i_k)$$

Dla $p = \infty, l_\infty = \max$

$$f_{\ell_p B}(\infty, (i_1, \dots, i_k)) = \lim_{p \rightarrow \infty} \sqrt[p]{\beta[(i_1)]^p + \beta[(i_2)]^p + \dots + [\beta(i_k)]^p} = \max \beta(i_1), \beta(i_2), \dots, \beta(i_k) = \beta(i_1) = f_{CC}$$

3.8. Szybkość i dokładność wykonywanych obliczeń

4. Opis modułów