

## OPIS ZADANIA:

W ramach tego zadania należy stworzyć aplikację desktopową umożliwiającą zliczanie głosów oddanych na poszczególnych kandydatów startujących w wyborach.

## WYMAGANIA BIZNESOWE:

### ▶ Użytkownicy:

- Aby oddać głos należy się wylegitymować: podać imię, nazwisko (polskie znaki są dozwolone) oraz PESEL.
- Każdy głosujący może zagłosować tylko raz.
- Przed ostatecznym oddaniem głosu użytkownik powinien zatwierdzić swój wybór.
- Aplikacja powinna umożliwić wylogowanie się osoby aktualnie głosującej (przeniesienie na ekran logowania) w dowolnym momencie (przed lub po oddaniu głosu).
- Osoba, która wyloguje się przed oddaniem głosu ma prawo oddać głos później.

### ▶ Kalkulator:

- Lista kandydatów jest stała i zawiera imię i nazwisko każdego kandydata oraz partię, do jakiej kandydat należy.
- Lista osób z odebranymi prawami wyborczymi (lista numerów PESEL) może zmieniać się w czasie.
- Podczas legitymowania głosujących należy upewnić się, że mają oni prawa wyborcze - ich nr PESEL nie znajduje się na czarnej liście oraz w dniu głosowania mają ukończone 18 lat.
- Głos pusty lub oddany na więcej niż jednego kandydata jest uznawany jako nieważny.
- Głosujący może oddać głos ważny lub nieważny.
- Po każdym oddanym głosie aplikacja powinna prezentować w postaci liczbowej aktualny rozkład głosów na poszczególnych kandydatów, na poszczególne partie oraz liczbę głosów nieważnych. Dodatkowo te same dane na żądanie użytkownika powinny być zaprezentowane w postaci wykresu słupkowego.
- Aplikacja powinna umożliwić eksport aktualnego stanu głosów (głosy ważne/nieważne), rozkład głosów na poszczególnych kandydatów, na poszczególne partie oraz ilości prób oddania głosów przez osoby bez praw wyborczych do pliku CSV oraz PDF.
- Aplikacja może być uruchamiana na jednym komputerze w kilku instancjach jednocześnie. Wszystkie instancje korzystają ze wspólnej bazy danych.

## WYMAGANIA TECHNICZNE:

- ▶ Listę kandydatów program powinien pobierać komunikując się z udostępnionym przez Future Processing serwerem: <http://webtask.future-processing.com:8069/candidates>.
- ▶ Listę osób nieuprawnionych do głosowania program powinien pobierać komunikując się z udostępnionym przez Future Processing serwerem: <http://webtask.future-processing.com:8069/blocked>.
- ▶ Domyślnym formatem zwracanych danych jest JSON. Aby uzyskać dane w postaci XML należy dodać do żądania nagłówki Accept o wartości application/xml.
- ▶ Nr PESEL powinien być zgodny ze standardem (tutaj można przetestować za pomocą generatora: [www.bogus.ovh.org/generatory/all.html](http://www.bogus.ovh.org/generatory/all.html)).
- ▶ Wszystkie dane osób wylegitymowanych powinny być przechowywane w bazie danych w sposób bezpieczny. W szczególności nr PESEL nie powinien być przechowywany w jawnej postaci.
- ▶ Aplikacja powinna być zabezpieczona przed atakami z zewnątrz.
- ▶ W ramach zadania powinna zostać stworzona baza danych w taki sposób, aby można ją było łatwo odtworzyć podczas sprawdzania zadania. Należy też dostarczyć kod źródłowy aplikacji, jej wersję wykonywalną oraz instrukcję jak skonfigurować środowisko developerskie.
- ▶ Całość rozwiązania (tzn. zarówno kod jak i interfejs użytkownika) powinna być w języku angielskim.
- ▶ Do rozwiązania zadania można użyć dowolnej technologii, dowolnego języka programowania i dowolnych narzędzi.

## UWAGI KOŃCOWE:

Program powinien być najwyższej jakości i charakteryzować się zamkniętą funkcjonalnością, czyli wszystkie zaimplementowane funkcjonalności muszą działać od początku do końca.

Program powinien demonstrować rozsądne użycie możliwości języka w zakresie abstrakcji, interfejsów, dziedziczenia, dobrą strukturę klas oraz pokazywać dobre praktyki programistyczne. Zwracamy uwagę na jakość kodu i projektu.

**Nieprzekraczalny czas na wykonanie zadania: 3 dni.**