Raport

System przeprowadzania głosowania dla Koła Naukowego Project Management Group

Imię i nazwisko autora: Michał Fadrowski

Nr indeksu: 246666

Data ukończenia pracy: Styczeń 2021 r.

Prowadzący laboratorium: mgr inż. Natalia Piórkowska

Spis treści:

[Wymagania projektowe 3](#_Toc61335684)

[Podstawowe wymagania funkcjonalne: 3](#_Toc61335685)

[Opis architektury systemu 4](#_Toc61335686)

[Schemat bazy danych wykorzystanej przez zaimplementowany system: 4](#_Toc61335687)

[Opis implementacji i zastosowań rozwiązań 5](#_Toc61335688)

[Kluczowe funkcje systemu: 5](#_Toc61335689)

[Dodanie nowego członka do systemu 5](#_Toc61335690)

[Oddanie głosu na kandydata: 7](#_Toc61335691)

[Dodatkowe funkcje: 8](#_Toc61335692)

[Opis działanie aplikacji: 8](#_Toc61335693)

[Instalacja i sposób uruchomienia aplikacji: 8](#_Toc61335694)

[Migracja tabel do bazy danych: 8](#_Toc61335695)

[Sposób działania: 8](#_Toc61335696)

[Podsumowanie: 9](#_Toc61335697)

Wymagania projektowe

Podstawowe wymagania funkcjonalne:

* Projekt stworzenie aplikacji webowej, która umożliwi przeprowadzenie głosowania mającego na celu wybór nowego zarządu przez członków Koła Naukowego Project Management Group
* Cały projekt jest podzielony na mniejsze aplikacje, umożliwiające dodawanie kolejnych funkcjonalności bez wpływania na resztę systemu. W obecnym etapie projektu zaimplementowałem następujące aplikacje:
  + Members – aplikacja umożliwia zarządzanie danymi użytkowników przez administratora oraz członków koła PMG.
  + Voting – aplikacja umożliwiająca przeprowadzenia głosowania.
* Aplikacja Members umożliwia:
  + Dodanie nowego użytkownika przez admina, poprzez podanie adresu e-mail oraz imienia i nazwiska członka koła PMG
  + Zalogowanie się członka koła oraz admina do aplikacji
  + Wyświetlenie danych wszystkich członków koła
* Administrator posiada możliwość zarządzania danymi członków koła w tym między innymi: dodanie nowego członka, nadanie roli administratora.
* Dodanie nowego członka przeprowadzone jest poprzez podanie przez administratora imienia, nazwiska oraz adresu e-mail nowego członka. Na podany e-mail zostanie wysłany wygenerowane hasło, znane wyłącznie właścicielowi konta.
* Użytkownicy są rozpoznawani za pomocą unikalnego adres e-mail.
* Użytkownicy mogą zostać usunięci z systemu przez administratora.
* Aplikacja Members odpowiednio zabezpiecza funkcjonalności oraz wyświetlane strony na podstawie posiadanej roli zalogowanego użytkownika.
* Aplikacja Voting umożliwia:
  + Dodanie stanowiska, na które mogą kandydować członkowie
  + Oddanie jednego głosu w każdej kategorii (dostępne stanowiska) na jednego z kandydatów
  + Zgłoszenie kandydatury na wybrane stanowisko w kole
  + Podliczenie oddanych głosów oraz wyłonienie zwycięzców głosowania.
* Aplikacja Voting umożliwia weryfikacje, kto oddał głos na kandydata w każdej kategorii (dostępne stanowiska) oraz weryfikuje, czy została oddana prawidłowa liczba głosów.
* Wszelkie dane o użytkownikach zostają zapisane w bazie danych sqlite3
* Wszelkie dane o oddanych głosach, dostępnych stanowiskach oraz głosowaniach są zapisywane w bazie danych sqlite3

Opis architektury systemu

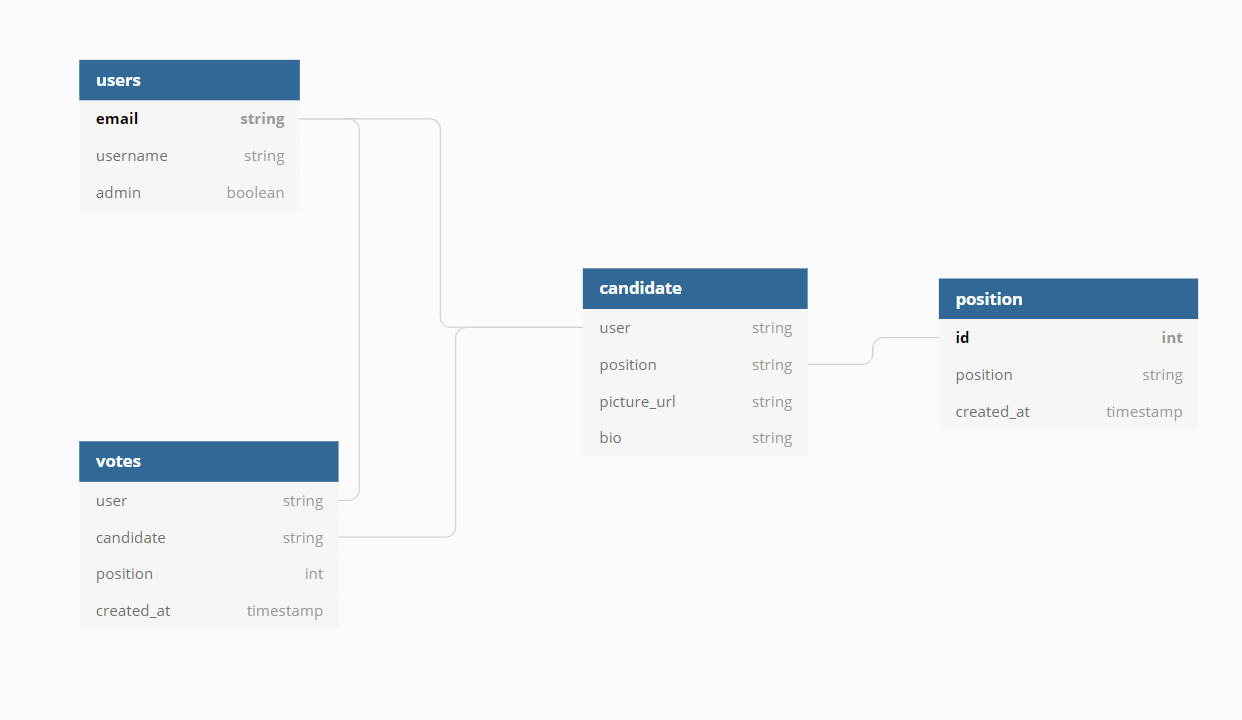
System został napisany w języku Python przy pomocy framework’u: Django. Architektura systemu polega na podziale jej na pomniejsze aplikacje, które się ze sobą współzależnie działają. Każda zaimplementowana aplikacja została napisana z wykorzystaniem wzorca architektonicznego MTV (model -template-view). Zaimplementowane aplikacje to Members oraz Voting

Aplikacja Members odpowiada za zarządzanie danymi użytkowników. Szczególnie istotną kwestią w tej aplikacji jest wykorzystanie modelu User. Implementacja modelu User została dostarczona w paczce oferowanej w pakiecie Django: django.contrib.auth.models. Model został ten zmodyfikowany w celu dostarczenia odpowiedniego rozwiązania dla koła naukowego.

Aplikacja Voting wykorzystuje klasy dziedziczące po klasie django.db.models.Model. Klasy zostały napisane w celu uproszczenia komunikacji systemu z bazą danych, uproszczenia implementacji oraz modyfikacji całego systemu oraz by zwiększyć czytelność kodu. Zaimplementowane Modele to:

* Vote – reprezentuje oddany przez użytkownika głos
* Position – reprezentuje dostępne stanowisko, na które może kandydować członek koła
* Candidate – reprezentacja kandydatury członka koła na możliwe stanowisko

Schemat bazy danych wykorzystanej przez zaimplementowany system:

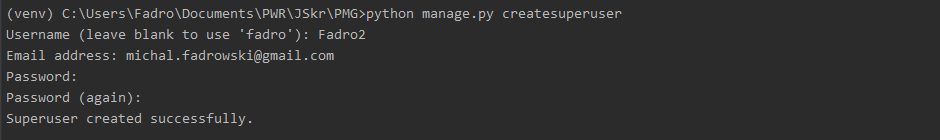


Opis implementacji i zastosowań rozwiązań

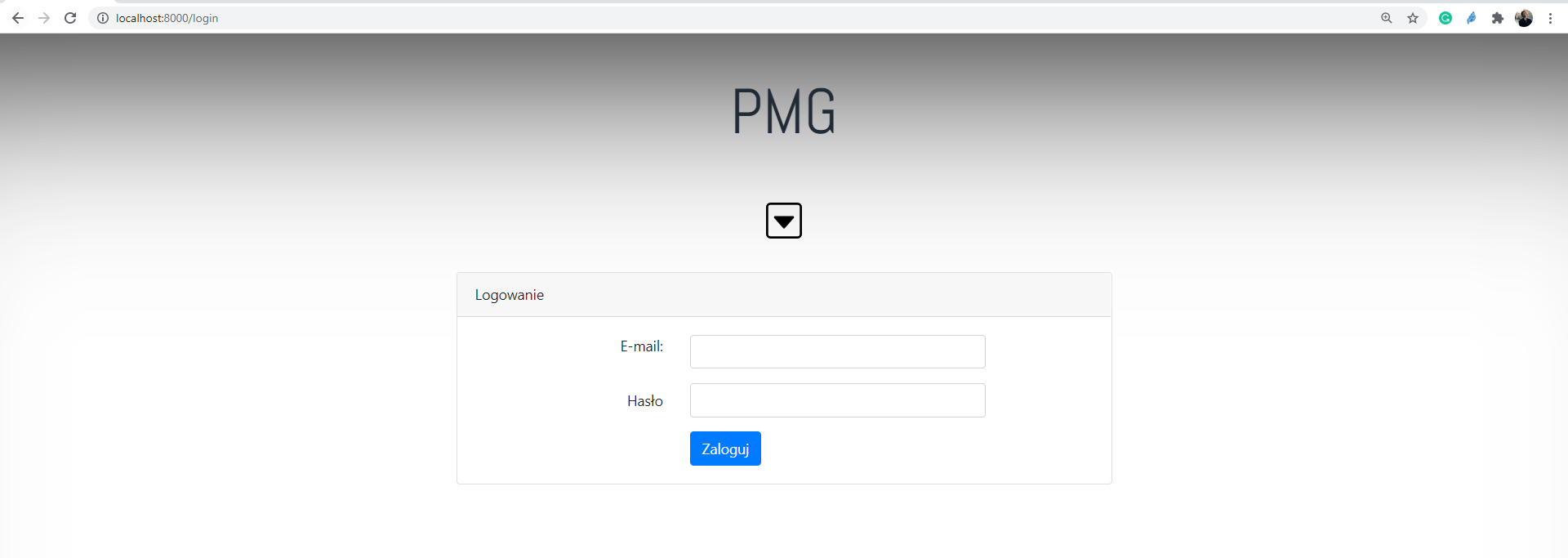
Kluczowe funkcje systemu:

Dodanie nowego członka do systemu

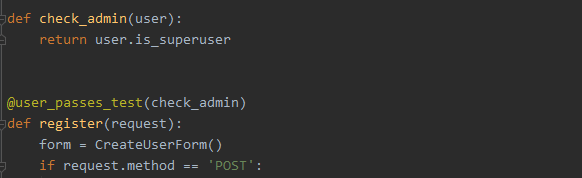
Jedną z podstawowych funkcjonalności zapewnionej przez system jest dodanie nowego członka do systemu przez administratora. Jeśli system nie posiada jeszcze żadnych danych, na początku należy utworzyć użytkownika z uprawnieniami administratora a poziomu konsoli:

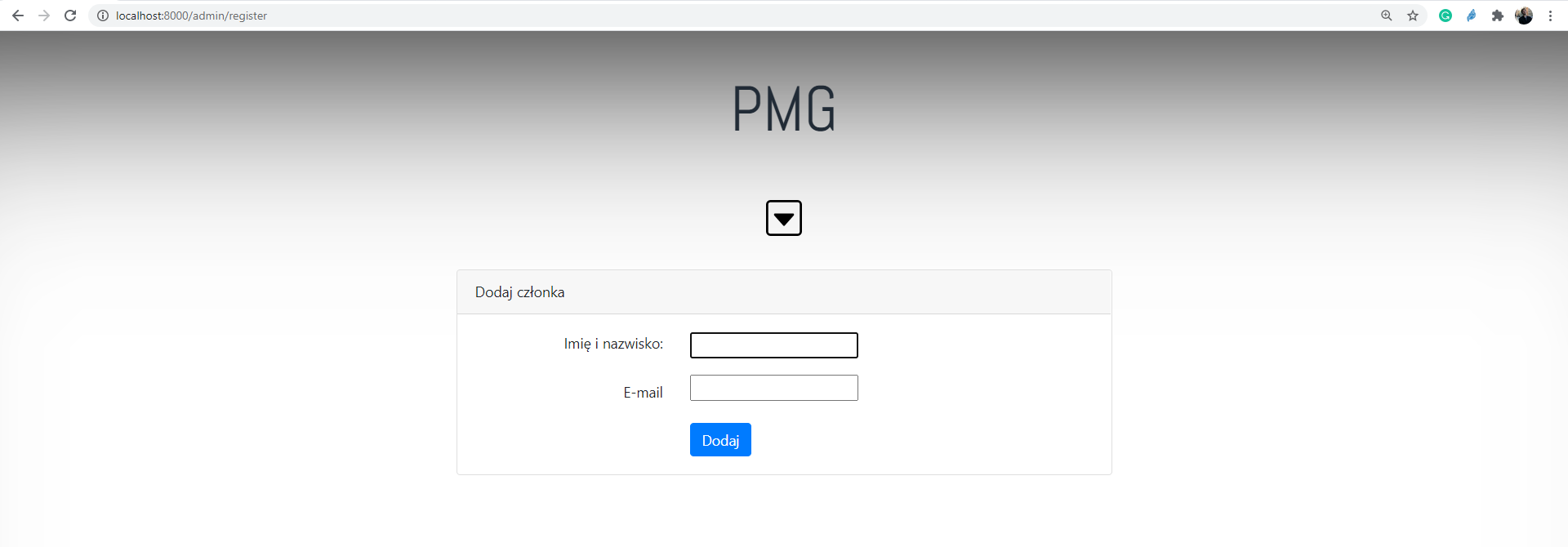


Umożliwi to zalogowanie się do systemu przez panel logowania z wykorzystaniem podanych podczas rejestracji danych:

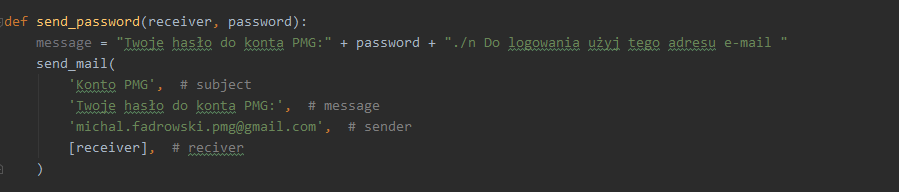


Następnie użytkownik może przejść do panelu dodania nowego członka koła. Panel ten jest zabezpieczony odpowiednią adnotacją ‘user\_passes\_test’ zapewnioną przez paczkę django.contrib.auth.decorators. Uniemożliwia to wykonanie funkcji dodania nowego użytkownika przez użytkownika bez uprawnień administratora:



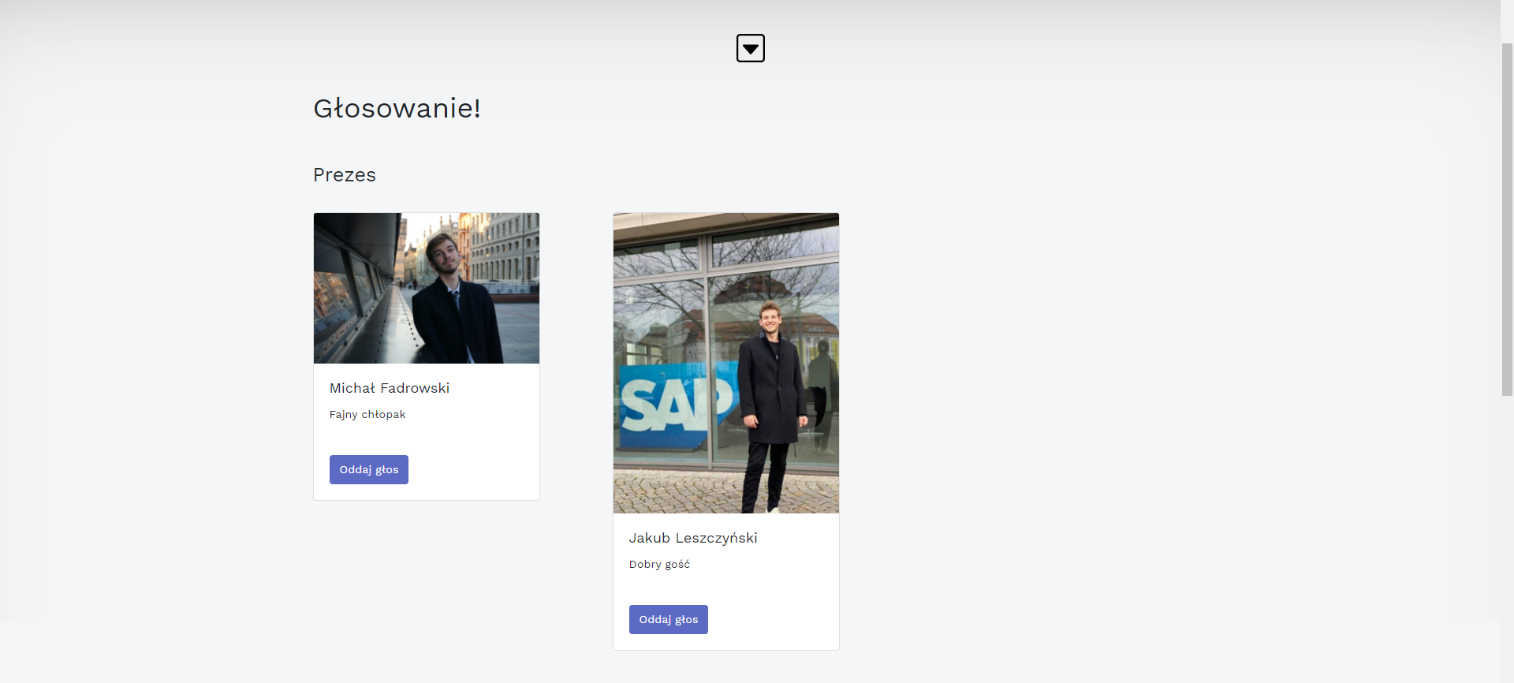


Po podaniu danych w ekranie wyświetlonym panelu oraz po naciśnięciu przycisku ‘Dodaj’ zostanie wykonana funkcja, która utworzy w systemie nowego użytkownika. Hasło dla nowego użytkownika zostanie wygenerowane przez system i przesłane drogą e-mailową na wskazany przy rejestracji adres. Dzięki temu, administrator nie ma możliwości sprawdzenia jakie hasło zostało przypisane konkretnemu użytkownikowi. Wysyłanie wiadomości e-mail zostało zaimplementowane za pomocą wykorzystania paczki django.core.mail.

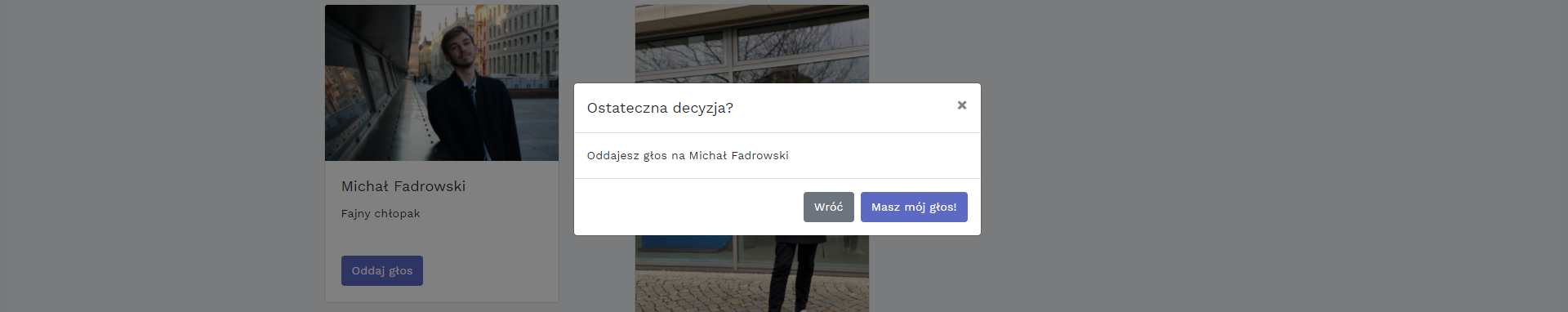


Oddanie głosu na kandydata:

Kolejną kluczową funkcjonalnością jest oddanie głosu przez zalogowanego użytkownika na wybranego kandydata. Odbywa się to przez przejście do panelu ‘Oddaj głos’. Zostanie wyświetlony panel z kandydatami w odpowiednich kategoriach:



Po naciśnięciu przycisku ‘Zagłosuj’ wyświetli się komunikat, który upewni użytkownika o oddaniu głosu na danego kandydata:



Po zagłosowaniu możliwość oddania głosu w danej kategorii zostaje zablokowana, a użytkownik zostaje przekierowany do panelu z aktualnymi wynikami wyborów. Wyniki wyborów zostają wyświetlone wyłącznie w kategoriach, w których użytkownik oddał już głos. Następnie może wrócić do panelu głosowania i oddać głos w innej kategorii. Użytkownik nie ma możliwości zmiany głosu w kategorii, w której już zagłosował.

Dodatkowe funkcje:

Innymi ważnymi dla systemu funkcjami są:

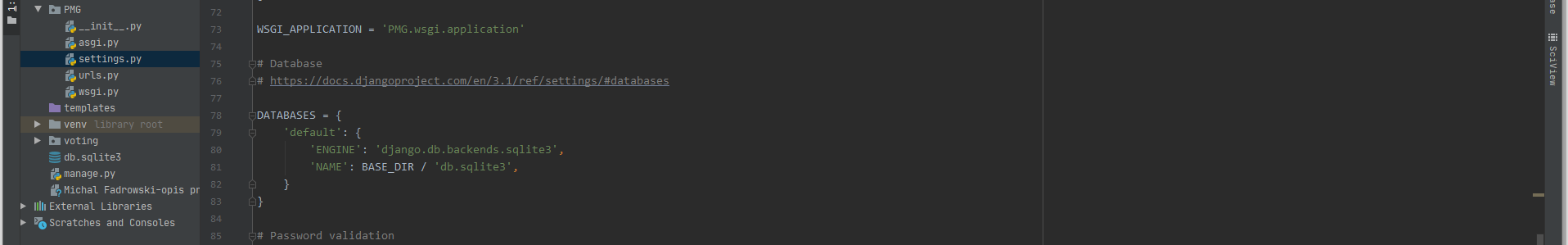
* Wyświetlanie przez administratora w jakich kategoriach, kto oddał głos. Nie jest możliwe sprawdzenie z poziomu aplikacji, na kogo dany użytkownik oddał głos.
* Zgłoszenie kandydatury przez użytkownika na wybrane stanowisko. Użytkownik może zgłosić kandydaturę na wyłącznie jedno stanowisko.

Opis działanie aplikacji:

Instalacja i sposób uruchomienia aplikacji:

**Migracja tabel do bazy danych:**

Aby aplikacja mogła działać po umieszczeniu jej na serwerze, należy dokonać migracji tabel zdefiniowanych w modelach. Na początku należy zdefiniować bazę danych oraz dane potrzebne do wykonania poprawnego połączenia. Zdefiniowane jest to w pliku PMG/settings.py :



Następnie z poziomu konsoli należy wykonać komendę python manage.py migrate



**Sposób działania:**

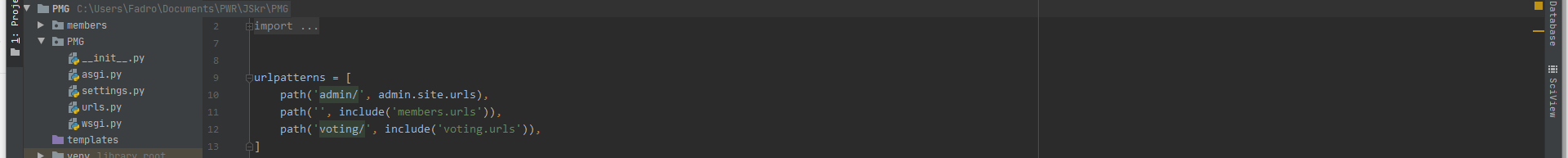
Jeśli aplikację na lokalnym środowisku uruchamia się poprzez wykonanie komendy python manage.py runserver.

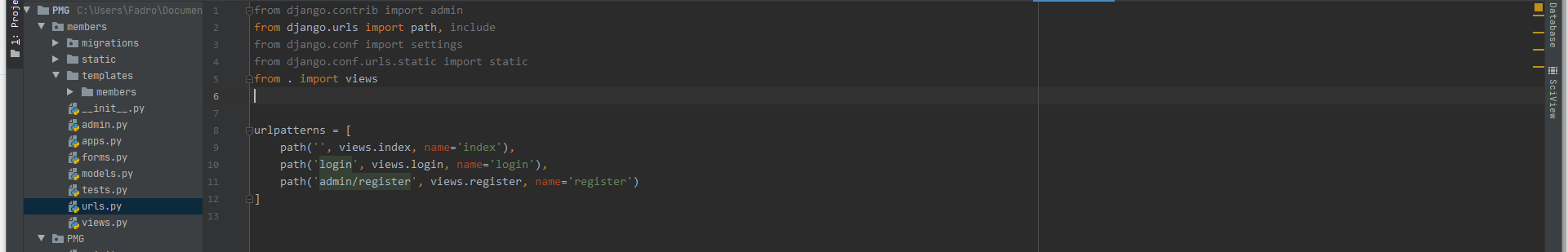


Stworzy to wirtualne środowisko pozwalające na korzystanie z aplikacji na komputerze. Aplikacja będzie działać na lokalnym serwerze na porcie 8000. Po wpisaniu w przeglądarce adresu: <http://127.0.0.1:8000> lub localhost:8000 zostaniemy przekierowani do głównej strony aplikacji:



Wszystkie dostępne podstrony aplikacji są dostępne dzięki za pomocą protokołu http. Wszystkie dostępne żądania są umieszczone w plikach urls.py:





Podsumowanie:

Aplikacja implementuje wszystkie potrzebne na obecnym etapie funkcjonalności. Ułatwia ona zarządzanie członkami oraz przeprowadzenie zdalnego głosowanie, co szczególnie w czasie pandemii jest wielce pomocne. Dzięki wykorzystaniu popularnego języka Python, Framwork’u Django oraz rozbicia na mniejsze aplikacje, cały projekt świetnie nadaję się do dalszego rozwoju.