

Tydzień pracy zdalnej warsztaty

v3.1

Cel warsztatów

Celem warsztatów jest napisanie prostej skrzynki kontaktowej.

Aplikacja powinna być napisana w Django.

Funkcje aplikacji (User Stories)

1. Jako użytkownik chcę po wejściu na stronę mieć wypisane alfabetycznie wszystkie osoby z mojej książki adresowej. Obok każdej osoby ma być guzik do usunięcia i edycji tej osoby.
2. Jako użytkownik po kliknięciu w imię osoby wyświetlonej w mojej skrzynce chcę być przeniesiony do strony wyświetlającej jej dane.
3. Po wejściu na stronę z danymi nowej osoby chcę widzieć jej wszystkie dane. Chcę też widzieć link do zmiany tych danych.

Funkcje aplikacji (User Stories)

4. Jako użytkownik po kliknięciu na link służący dodaniu nowej osoby chcę być przeniesiony do nowej strony z formularzem do wypełnienia podstawowych danych (na razie bez adresu, numeru telefonu i emaili).
Po wypełnieniu tych danych i kliknięciu **submit** chcę być przeniesiony do strony wyświetlającej dane nowej osoby.
5. Na stronie edycji osoby chcę mieć możliwość zmiany jej danych, dodania lub usunięcia nowego adresu, maila lub telefonu dla niej.

Funkcje aplikacji (User Stories)

Jako użytkownik chcę mieć następujące funkcje:

1. Możliwość tworzenia nowych grup w systemie.
2. Możliwość dodania użytkownika do wielu grup.
3. Mieć stronę, gdzie wyświetlają mi się listy grup.

Modele w aplikacji

Osoba

Jest to reprezentacja jednej osoby w naszej skrzynce adresowej. Powinna mieć możliwość trzymania następujących danych osobowych: imię, nazwisko, opis.

Adres

Ma być adresem przypisanym do osoby, a trzymać następujące dane: miasto, ulicę, numer domu, numer mieszkania. Ma być połączony z osobą relacją jeden do wielu.

Telefon

Ma być telefonem przypisanym do osoby, a trzymać numer telefonu, typ (napis np. domowy, służbowy itp.). Ma być połączony z osobą relacją jeden do wielu.

Email

Ma być emailem przypisanym do osoby. Ma trzymać adres email, typ (napis, np. domowy, służbowy itp.). Ma być połączony z osobą relacją jeden do wielu.

Modele w aplikacji

Grupy

Grupy będą dodatkowym modelem, który ma być połączony relacją wiele do wielu z osobami (relacja dwukierunkowa). Czyli osoba może być przypisana do wielu grup, a grupa może mieć wiele osób.

Zadania

1. Przygotowanie projektu

Najpierw przygotuj nowy projekt, skonfiguruj projekt (połączenie z bazą danych, itp.), stwórz nową aplikację.

2. Utwórz bazowy HTML

Nie musi być wyrafinowany i skomplikowany. Zaplanuj jednak bloki, które chcesz wypełniać w widokach.

3. Model Osoby

Stwórz nowy model dla osoby. Jeżeli chcesz przetrzymywać dodatkowe informacje, to zaplanuj to teraz. Jeżeli chcesz przetrzymywać np. zdjęcie przedstawiające, jak radzić sobie z plikami, możesz przeczytać w dokumentacji Django.

<http://djangoproject.com>

Zadania

4. URL-e do Osoby

Stwórz URL-e do zarządzania osobami, który powinien mieć następujące funkcjonalności:

1. Tworzenie formularza do stworzenia nowego modelu (**/new**).
2. Tworzenie nowej osoby (formularz wysyłany **POST**-em na adres **/new**).
3. Tworzenie formularza do modyfikacji osoby (**/modify/{id}**).
4. Modyfikacja osoby (formularz wysyłany **POST**-em na adres **/modify/{id}**).
5. Usunięcie podanej osoby (**/delete/{id}**).
6. Pokazanie jednej osoby (**/show/{id}**).
7. Pokazanie wszystkich osób (adres **/**).

5. Osoba

Stwórz widoki do Osoby.

(na końcu prezentacji znajdziesz, jak zbudować poprawnego URL-a)

Zadania

6. Adres

Stwórz nowy model dla Adresu. Zmodyfikuj model Osoby w taki sposób, żeby trzymał relacje.

7. Osoba

Zmodyfikuj widok modyfikacji osoby, aby umożliwiał dopisanie tej osobie nowego adresu. Widok zmień następująco:

1. Dodaj do widoku formularz (przypisany do adresu), który będzie odsyłał do strony **POST /{id}/addAddress**.
2. Dodaj widok, który obsłuży formularz.

8. Widoki dla modelu Osoby

Powtórz punkt 6. i 7. dla maili i telefonów.

Zadania

8. Grupy

Stwórz nowy model dla Grup. Zmodyfikuj model Osoby w taki sposób, żeby trzymał relacje.

Następnie stwórz wszystkie potrzebne URL-e (w pliku `urls.py`) i widoki.

9. Grupy

Dodaj do wyświetlania możliwość wyszukiwania po imieniu lub nazwisku. W tym celu stwórz nowy widok (np. **POST** formularza na adres `/group-search` i przeszukaj model Grupy przy pomocy Pythona).

Podpowiedź

Jak zbudować poprawnego URL-a?

W pliku `urls.py` znajdziesz wpisy definiujące url-e.

Jeśli chcesz dodać np. URL-a, który kieruje pod adres `/person/{id}/`, gdzie `id`, to identyfikator osoby (liczba naturalna), napisz:

```
url(r'^person/(?P<id>[0-9]+)/$', person_view, name="person"),
```

`person_view`, to funkcja (lub klasa) widoku, a `name="person"`, to etykieta, pod którą widok będzie widoczny w aplikacji.