Contents

[1. Login 1](#_Toc160194114)

[1.1 Strona logowania w customowej templatece 1](#_Toc160194115)

[2. Formularz 7](#_Toc160194116)

[2.1 Jak sprawdzić jakie pola są podawane do formularza / customizacje wyglądu pól w formularzu 7](#_Toc160194117)

[Ikonki 11](#_Toc160194118)

[Error messages 12](#_Toc160194119)

[Jak ładnie wyświetlać error messages 12](#_Toc160194120)

[Class-based views 13](#_Toc160194121)

[Wysyłanie kontextu z class-based view do templatki 13](#_Toc160194122)

[Formularze 15](#_Toc160194123)

[Inline formset czyli formularz wbudowany w inny formularz 15](#_Toc160194124)

[Przekazywanie zmiennych z views do formularza 15](#_Toc160194125)

[Ukrywanie pola w formularzu 17](#_Toc160194126)

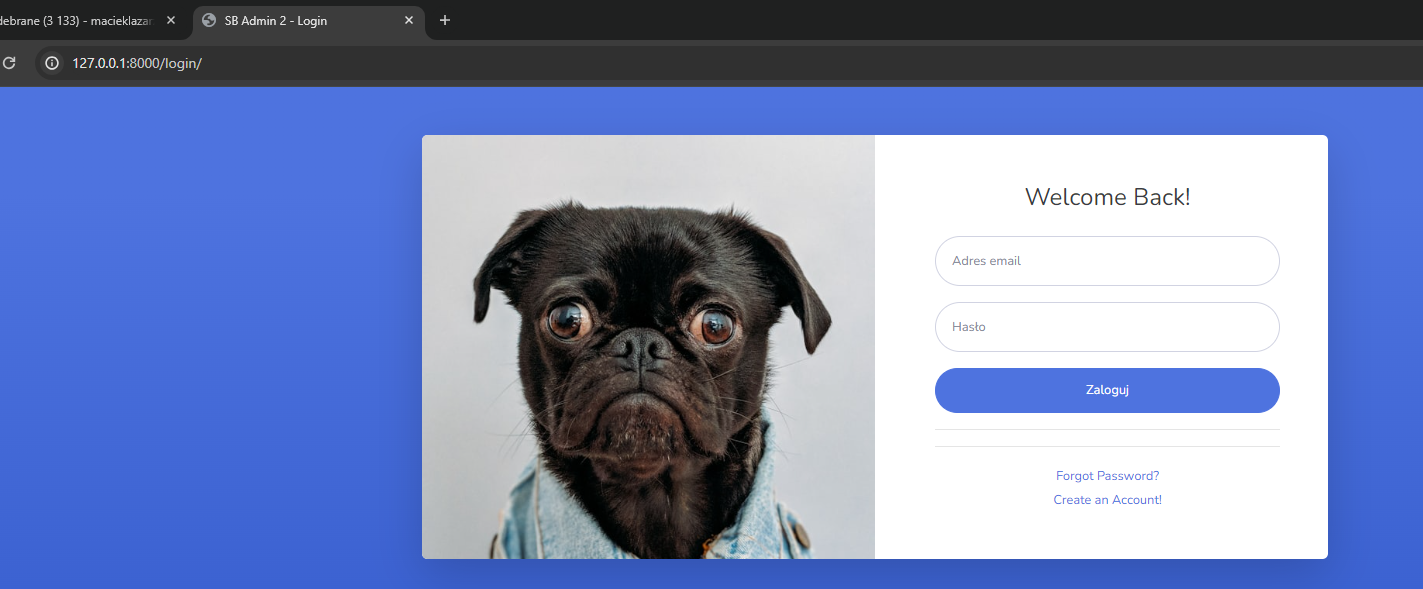
[Widoki 17](#_Toc160194127)

[Pobranie parametru z url do widoku 17](#_Toc160194128)

# Login

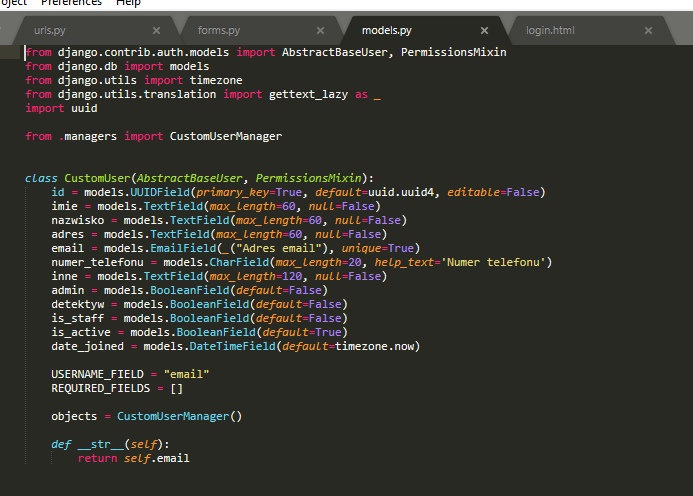
## Strona logowania w customowej templatece

Tak wygląda strona logowania:



W modelu mam ustawione, że loguję się mailem.

models.py:



from django.contrib.auth.models import AbstractBaseUser, PermissionsMixin

from django.db import models

from django.utils import timezone

from django.utils.translation import gettext\_lazy as \_

import uuid

from .managers import CustomUserManager

class CustomUser(AbstractBaseUser, PermissionsMixin):

id = models.UUIDField(primary\_key=True, default=uuid.uuid4, editable=False)

imie = models.TextField(max\_length=60, null=False)

nazwisko = models.TextField(max\_length=60, null=False)

adres = models.TextField(max\_length=60, null=False)

email = models.EmailField(\_("Adres email"), unique=True)

numer\_telefonu = models.CharField(max\_length=20, help\_text='Numer telefonu')

inne = models.TextField(max\_length=120, null=False)

admin = models.BooleanField(default=False)

detektyw = models.BooleanField(default=False)

is\_staff = models.BooleanField(default=False)

is\_active = models.BooleanField(default=True)

date\_joined = models.DateTimeField(default=timezone.now)

USERNAME\_FIELD = "email"

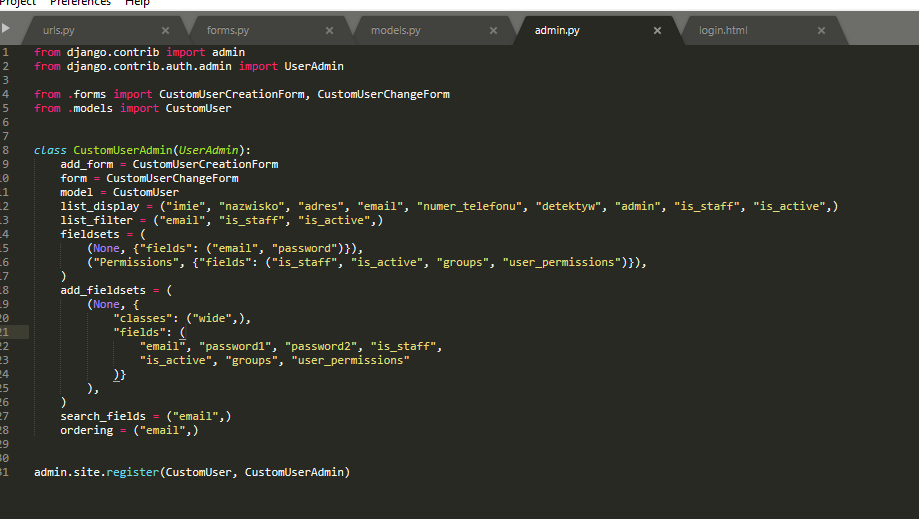
REQUIRED\_FIELDS = []

objects = CustomUserManager()

def \_\_str\_\_(self):

return self.email

admin.py:



from django.contrib import admin

from django.contrib.auth.admin import UserAdmin

from .forms import CustomUserCreationForm, CustomUserChangeForm

from .models import CustomUser

class CustomUserAdmin(UserAdmin):

add\_form = CustomUserCreationForm

form = CustomUserChangeForm

model = CustomUser

list\_display = ("imie", "nazwisko", "adres", "email", "numer\_telefonu", "detektyw", "admin", "is\_staff", "is\_active",)

list\_filter = ("email", "is\_staff", "is\_active",)

fieldsets = (

(None, {"fields": ("email", "password")}),

("Permissions", {"fields": ("is\_staff", "is\_active", "groups", "user\_permissions")}),

)

add\_fieldsets = (

(None, {

"classes": ("wide",),

"fields": (

"email", "password1", "password2", "is\_staff",

"is\_active", "groups", "user\_permissions"

)}

),

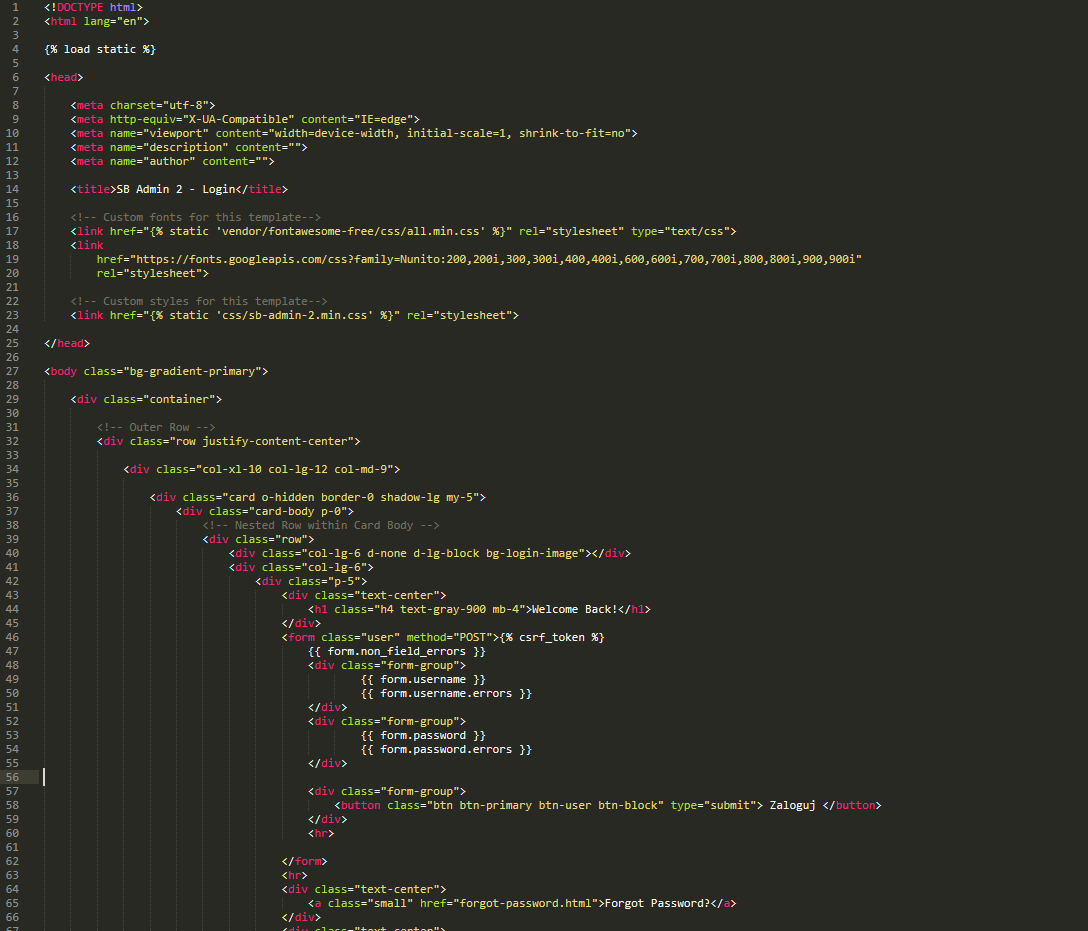
)

search\_fields = ("email",)

ordering = ("email",)

admin.site.register(CustomUser, CustomUserAdmin)

login.html



<!DOCTYPE html>

<html lang="en">

{% load static %}

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<meta name="description" content="">

<meta name="author" content="">

<title>SB Admin 2 - Login</title>

<!-- Custom fonts for this template-->

<link href="{% static 'vendor/fontawesome-free/css/all.min.css' %}" rel="stylesheet" type="text/css">

<link

href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600,600i,700,700i,800,800i,900,900i"

rel="stylesheet">

<!-- Custom styles for this template-->

<link href="{% static 'css/sb-admin-2.min.css' %}" rel="stylesheet">

</head>

<body class="bg-gradient-primary">

<div class="container">

<!-- Outer Row -->

<div class="row justify-content-center">

<div class="col-xl-10 col-lg-12 col-md-9">

<div class="card o-hidden border-0 shadow-lg my-5">

<div class="card-body p-0">

<!-- Nested Row within Card Body -->

<div class="row">

<div class="col-lg-6 d-none d-lg-block bg-login-image"></div>

<div class="col-lg-6">

<div class="p-5">

<div class="text-center">

<h1 class="h4 text-gray-900 mb-4">Welcome Back!</h1>

</div>

<form class="user" method="POST">{% csrf\_token %}

{{ form.non\_field\_errors }}

<div class="form-group">

{{ form.username }}

{{ form.username.errors }}

</div>

<div class="form-group">

{{ form.password }}

{{ form.password.errors }}

</div>

<div class="form-group">

<button class="btn btn-primary btn-user btn-block" type="submit"> Zaloguj </button>

</div>

<hr>

</form>

<hr>

<div class="text-center">

<a class="small" href="forgot-password.html">Forgot Password?</a>

</div>

<div class="text-center">

<a class="small" href="register.html">Create an Account!</a>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- Bootstrap core JavaScript-->

<script src="{% static 'vendor/jquery/jquery.min.js' %}"></script>

<script src="{% static 'vendor/bootstrap/js/bootstrap.bundle.min.js' %}"></script>

<!-- Core plugin JavaScript-->

<script src="{% static 'vendor/jquery-easing/jquery.easing.min.js' %}"></script>

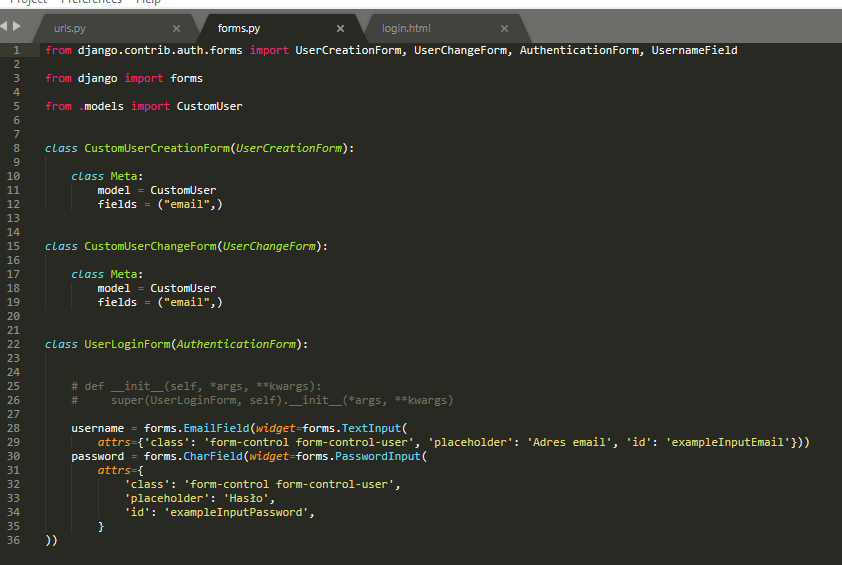
<!-- Custom scripts for all pages-->

<script src="{% static 'js/sb-admin-2.min.js' %}"></script>

</body>

</html>

forms.py



from django.contrib.auth.forms import UserCreationForm, UserChangeForm, AuthenticationForm, UsernameField

class UserLoginForm(AuthenticationForm):

# def \_\_init\_\_(self, \*args, \*\*kwargs):

# super(UserLoginForm, self).\_\_init\_\_(\*args, \*\*kwargs)

username = forms.EmailField(widget=forms.TextInput(

attrs={'class': 'form-control form-control-user', 'placeholder': 'Adres email', 'id': 'exampleInputEmail'}))

password = forms.CharField(widget=forms.PasswordInput(

attrs={

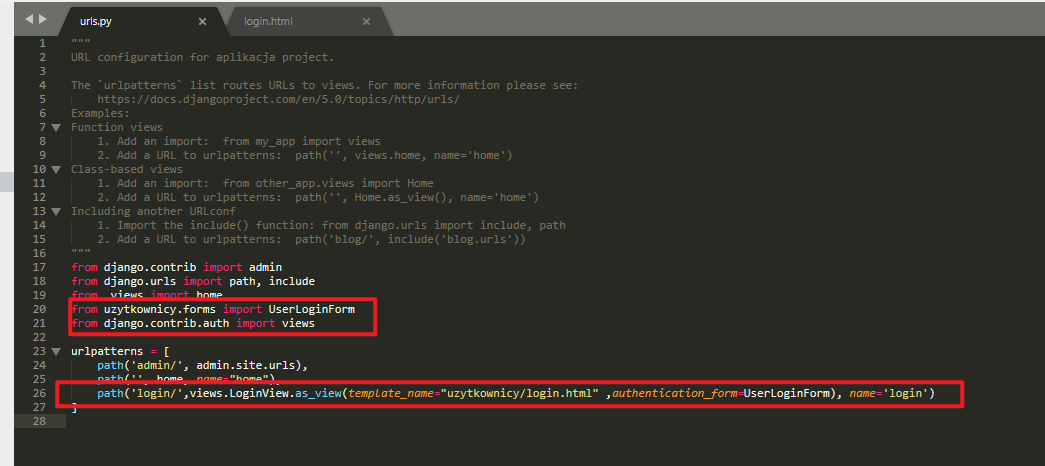
'class': 'form-control form-control-user',

'placeholder': 'Hasło',

'id': 'exampleInputPassword',

}

)) urls.py



from django.contrib import admin

from django.urls import path, include

from .views import home

from uzytkownicy.forms import UserLoginForm

from django.contrib.auth import views

urlpatterns = [

path('admin/', admin.site.urls),

path('', home, name="home"),

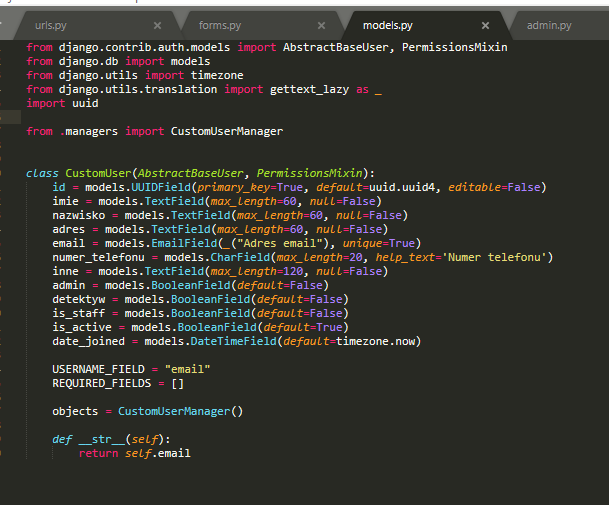
path('login/',views.LoginView.as\_view(template\_name="uzytkownicy/login.html" ,authentication\_form=UserLoginForm), name='login')

]

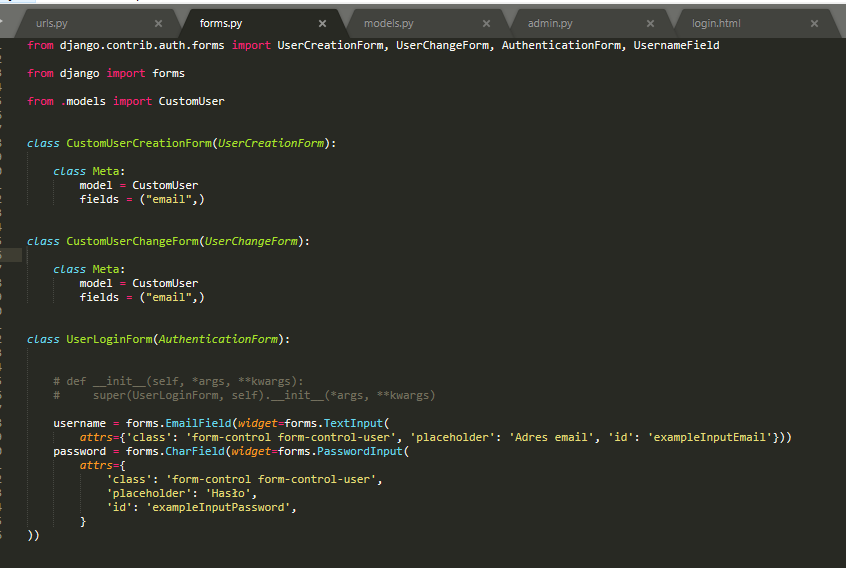
# Formularz

## Jak sprawdzić jakie pola są podawane do formularza / customizacje wyglądu pól w formularzu

Mamy taki model:

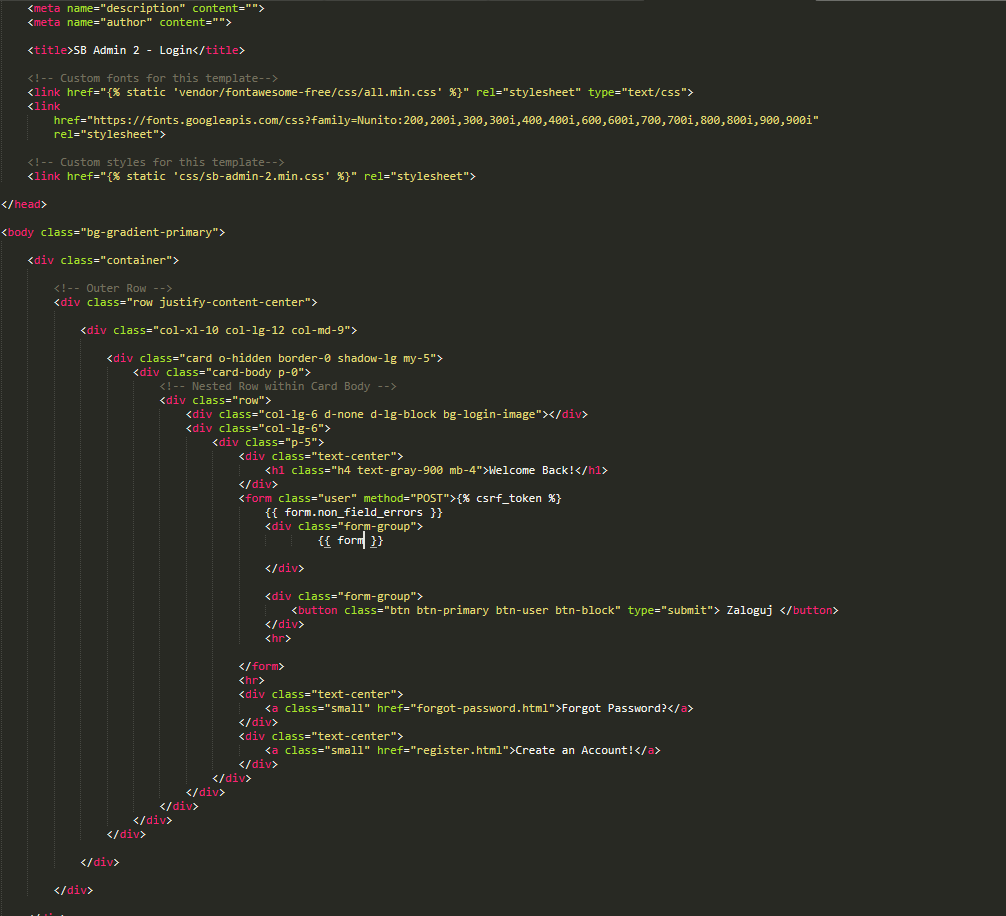


Formularz dla tego modelu:



Żeby sprawdzić sobie jakie pola są podawane do templatki robimy tak:

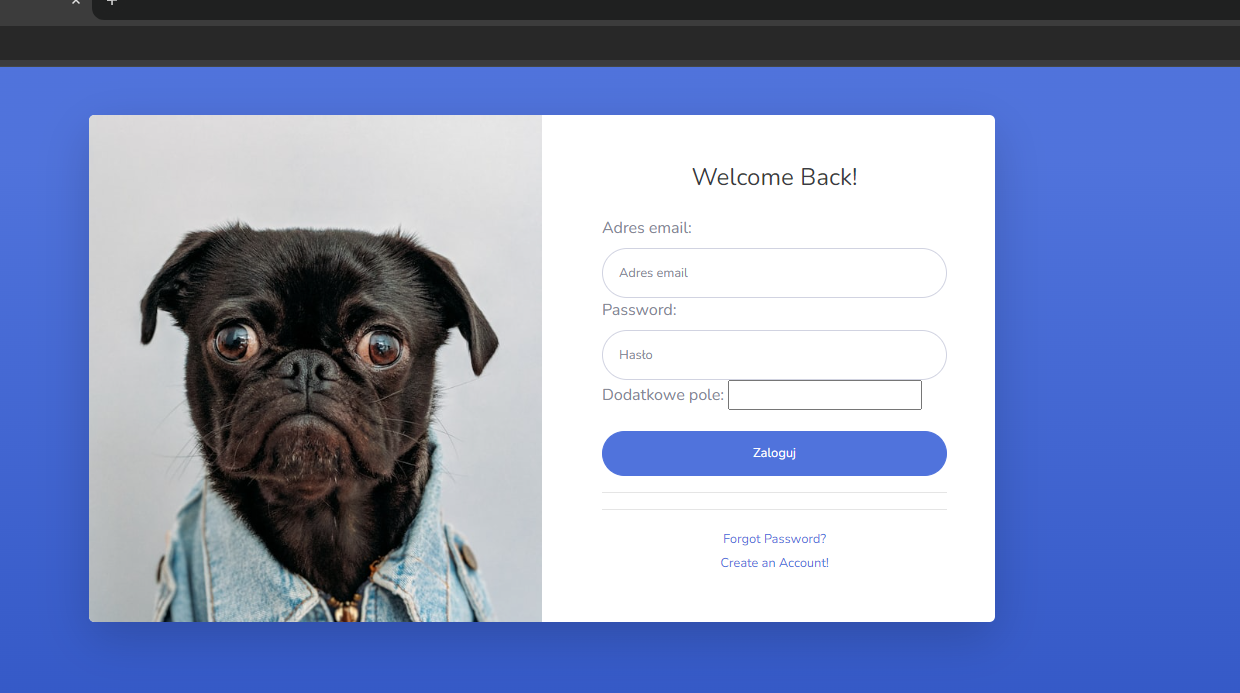
Template.html:



Dodaję jakieś pole:



I taki jest efekt:

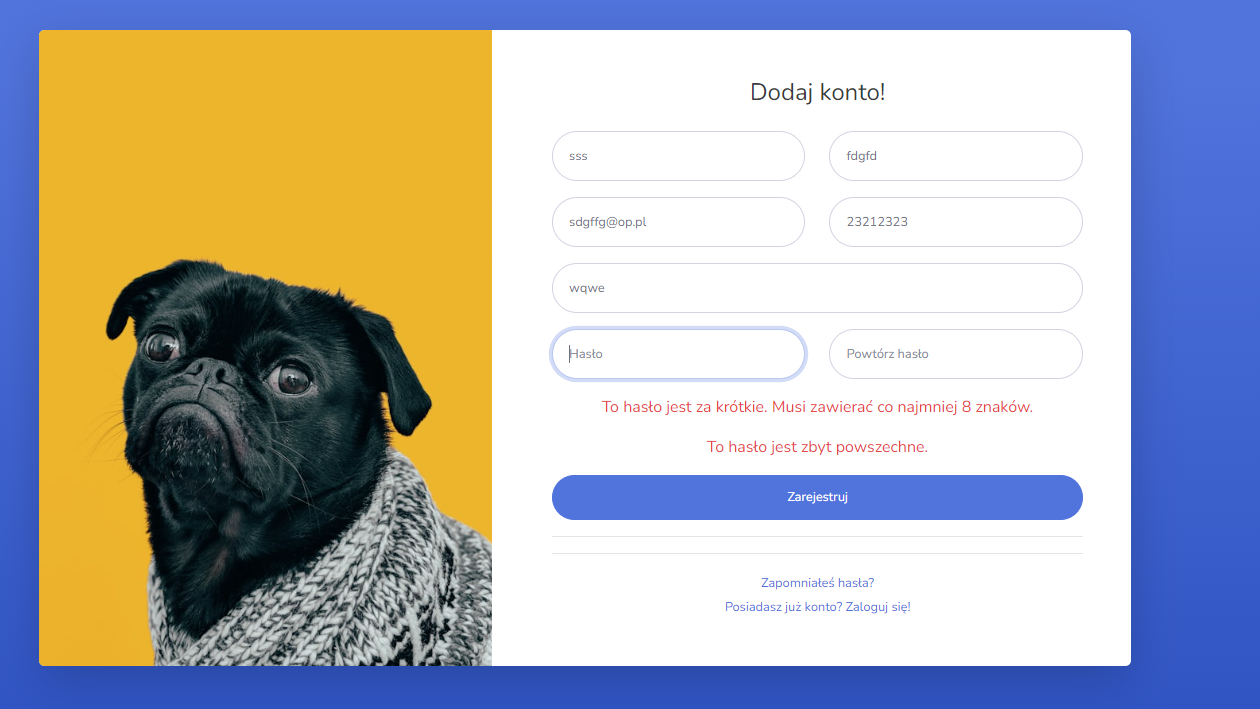


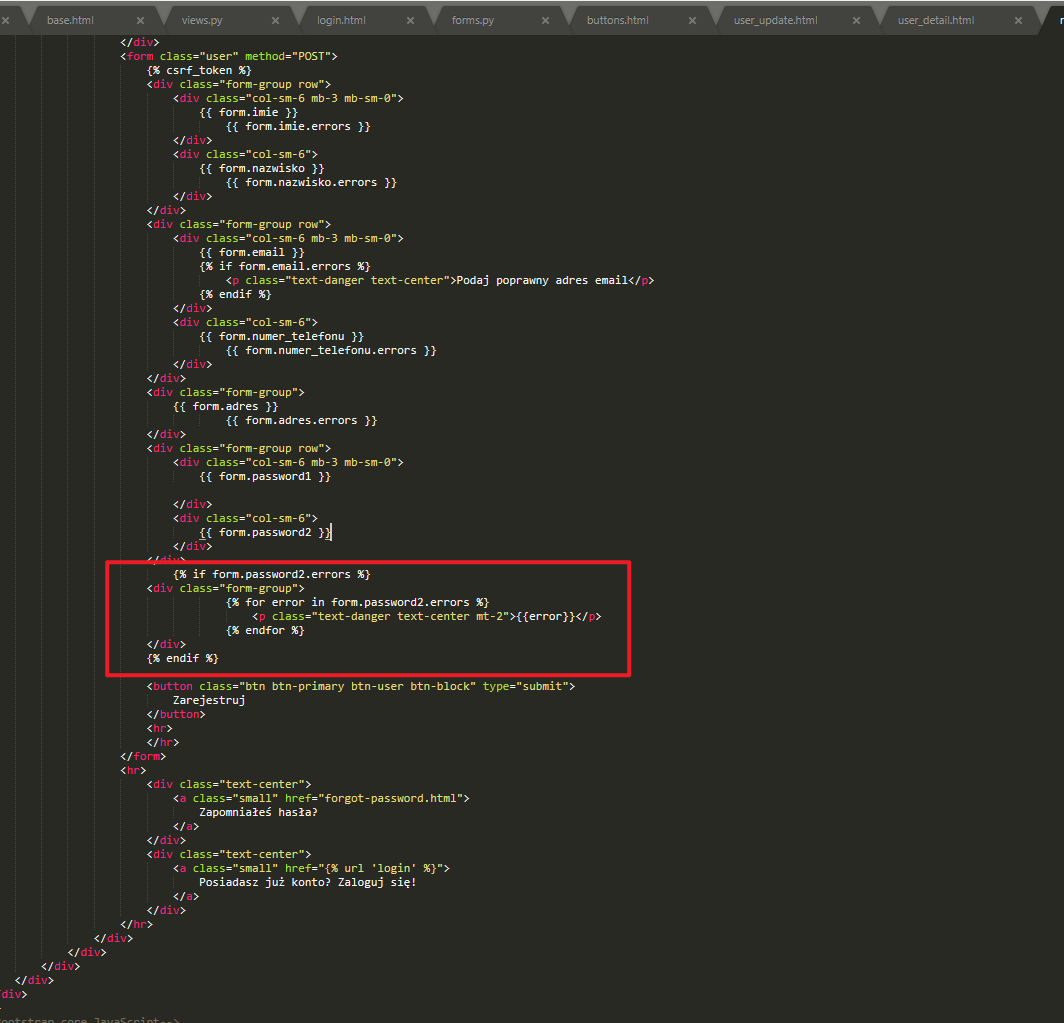
# Ikonki

<https://fontawesome.com/v5/search?q=people&o=r>

# Error messages

## Jak ładnie wyświetlać error messages





{% if form.password2.errors %}

<div class="form-group">

{% for error in form.password2.errors %}

<p class="text-danger text-center mt-2">{{error}}</p>

{% endfor %}

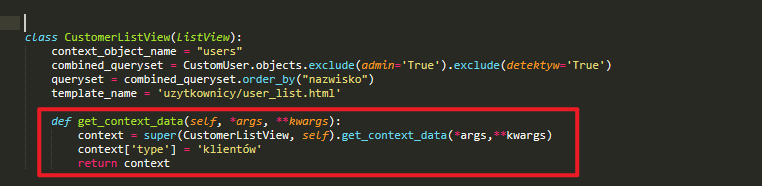
</div>

{% endif %}

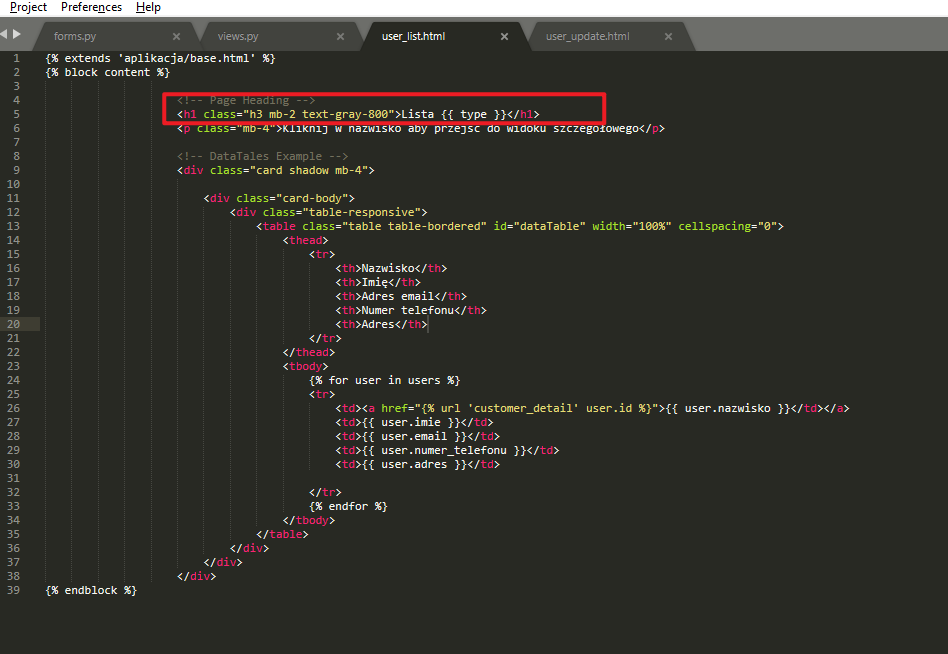
# Class-based views

## Wysyłanie kontextu z class-based view do templatki

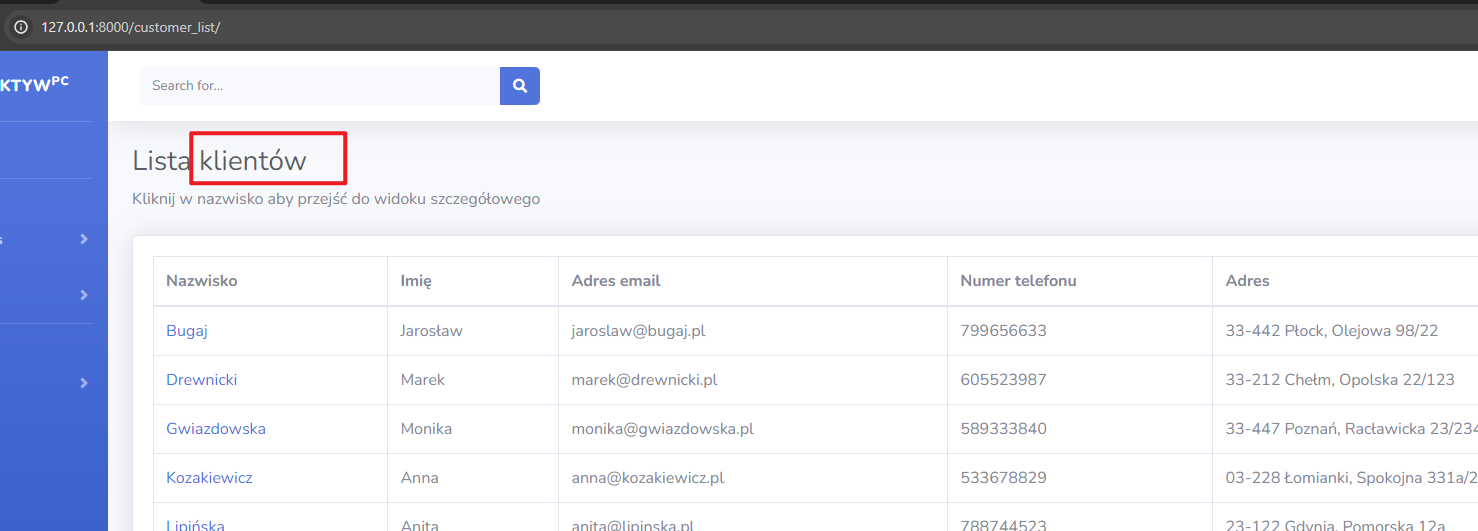
views.py

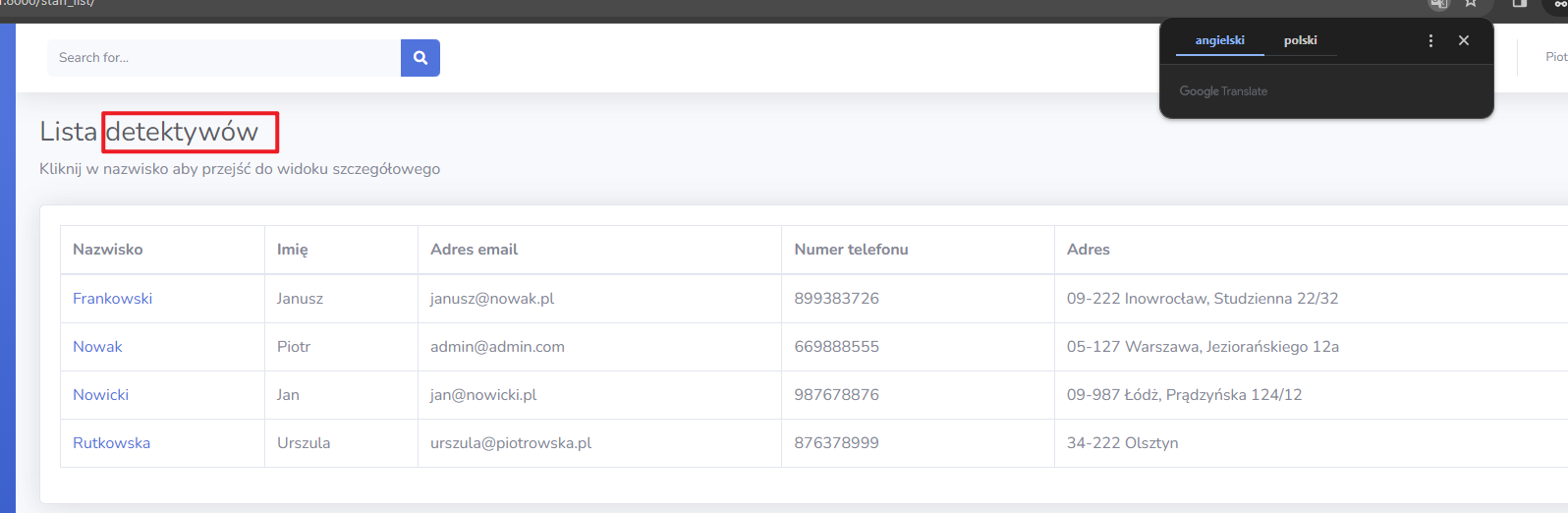


Plik html:



I tak to wygląda:





# Formularze

## Inline formset czyli formularz wbudowany w inny formularz

<https://www.youtube.com/watch?v=o45AqEtX5jU>

<https://www.letscodemore.com/blog/django-inline-formset-factory-with-examples/>

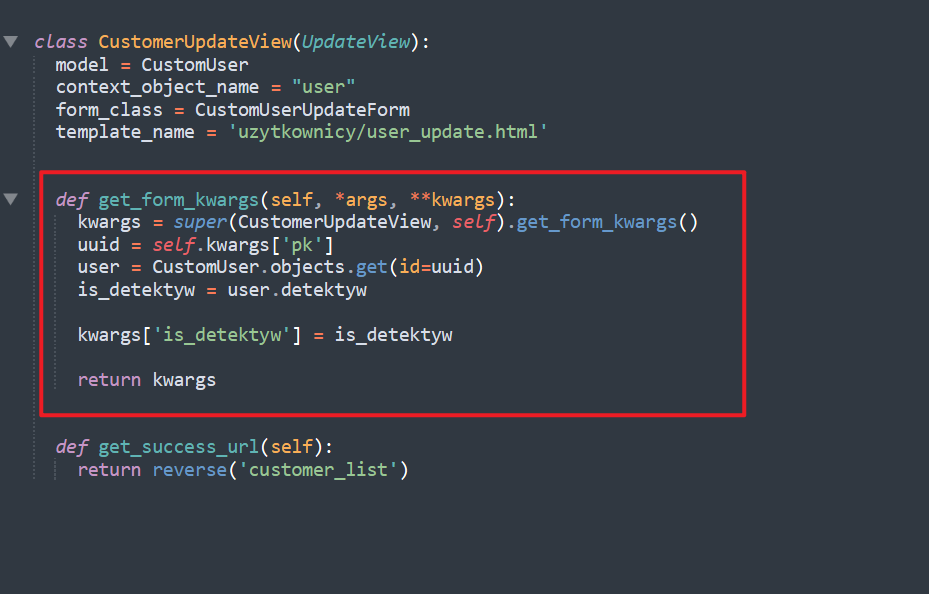
<https://swapps.com/blog/working-with-nested-forms-with-django/>

<https://www.letscodemore.com/blog/django-inline-formset-factory-with-examples/>

z tego linku letscodemore mi się udało

## Przekazywanie zmiennych z views do formularza

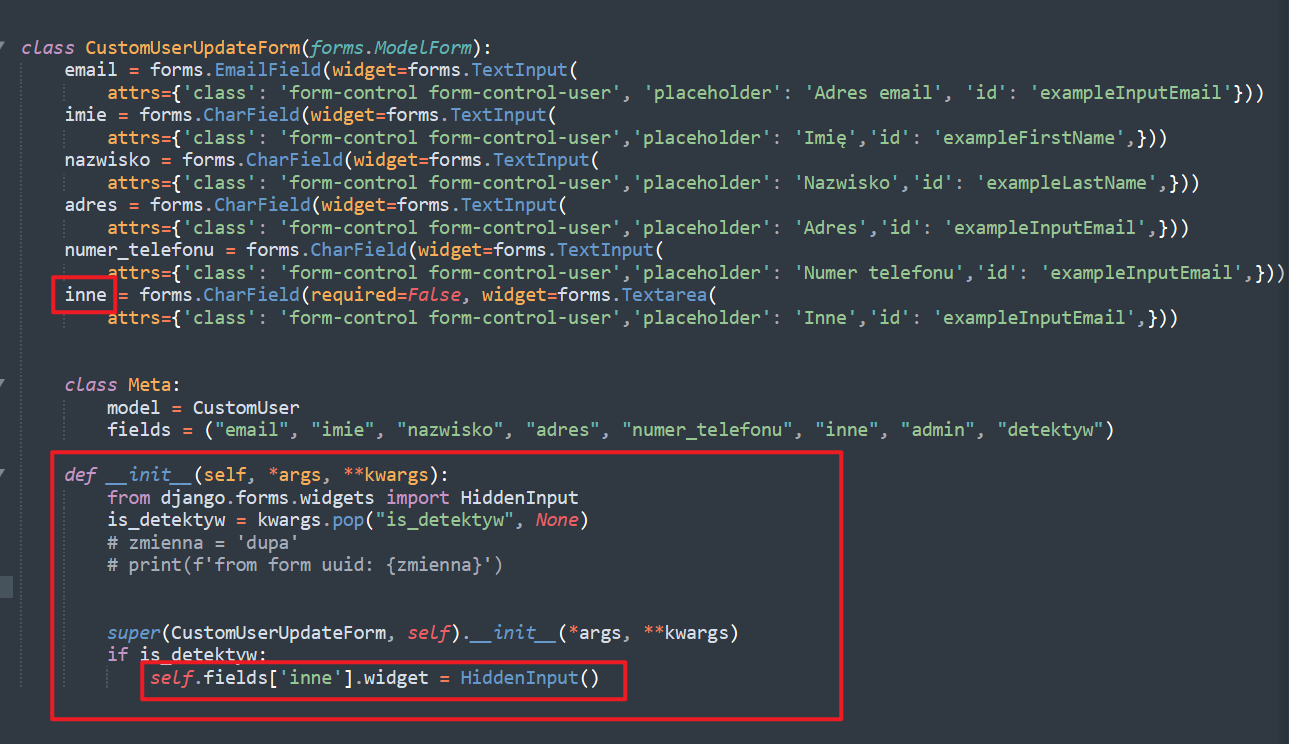
views.py:



forms.py:



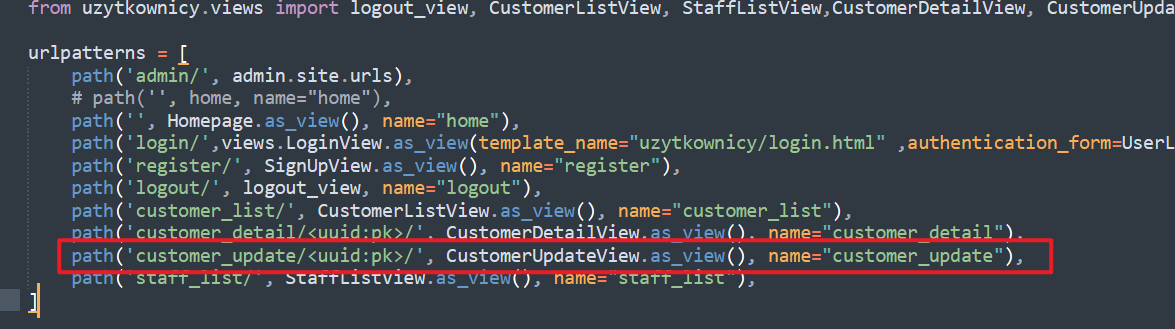
## Ukrywanie pola w formularzu



# Widoki

## Pobranie parametru z url do widoku

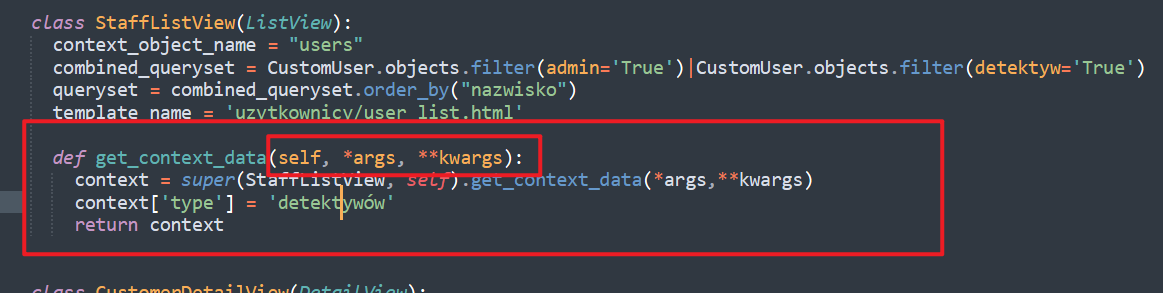
urls.py



views.py:



To nie musi być w funkcji get\_form\_kwargs, może być w innych jak na przykład get\_context\_data bo ta funkcja też przyjmuje self w argumentach:



# Stylizowanie z crispy form

https://simpleisbetterthancomplex.com/tutorial/2018/11/28/advanced-form-rendering-with-django-crispy-forms.html