I. Wymagania funkcjonalne i niefunkcjonalne

1. Wymagania funkcjonalne:
    - Rejestracja użytkownika: jest zaimplementowana przez SignUpView.
    - Logowanie użytkownika: funkcja login_view obsługuje logowanie użytkownika.
    - Nauka nowych słówek poprzez wyświetlanie ich na ekranie, możliwość przewijania oraz odznaczenia „umiem”: Widok nauka_view i funkcje w szablonie Nauka.html obsługują tę funkcjonalność.
    - Powtarzanie słówek które zostały oznaczone „umiem” poprzez wyświetlenie słówka oraz okna tekstowego do uzupełnienia tłumaczenia, a następnie sprawdzenie poprawności : Widok powtarzanie_view i funkcje w szablonie Powtarzanie.html obsługują tę funkcjonalność.
    - Administrowane aplikacji przez narzędzie „admin panel” pozwalające na zmianę danych użytkowników, dodawanie ich użytkowników, modyfikacje istniejącego słownika oraz odznaczanie znajomości słówka przez danego użytkownika
2. Wymagania niefunkcjonalne:
    - Bezpieczeństwo: Dane użytkowników muszą być bezpieczne i chronione przed nieautoryzowanym dostępem
    - Wydajność: System powinien działać płynnie nawet przy dużej liczbie użytkowników i słówek
    - Użyteczność: Interfejs użytkownika powinien być intuicyjny i łatwy w obsłudze
    - Skalowalność: Aplikacja powinna być łatwa do rozbudowy o nowe funkcje
    - Niezawodność: System powinien być odporny na błędy i awarie

II. Diagramy klas

User

Model **`User`** pochodzi z wbudowanego modelu użytkownika Django. Zawiera takie pola jak **`username`** i **`password`**.

Slowko

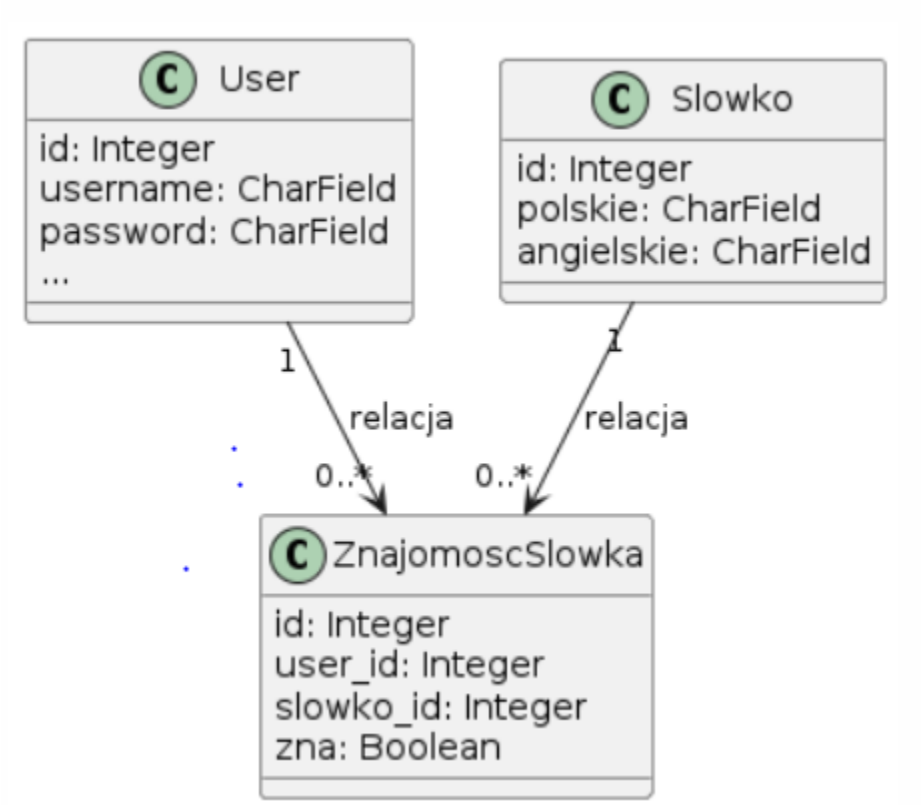
Model **`Slowko`** reprezentuje słówka do nauki. Zawiera pola:

- **`polskie`** - polskie słowo
- **`angielskie`** - odpowiednik angielski

ZnajomoscSlowka

Model **`ZnajomoscSlowka`** reprezentuje relację między użytkownikiem a słówkiem, informującą czy użytkownik zna dane słówko. Zawiera pola:

- **`user`** - odniesienie do modelu **`User`**
- **`slowko`** - odniesienie do modelu **`Slowko`**
- **`zna`** - boolean informujący, czy użytkownik zna dane słówko



III. Baza danych

Tabela `User`

Tabela **`User`** przechowuje dane dotyczące użytkowników aplikacji. Jest to standardowy model użytkownika Django z domyślnymi polami oraz ewentualnie innymi polami dostępnymi w modelu użytkownika.

Pola:

- **`id`** (Integer, Primary Key): Unikalny identyfikator użytkownika.
- **`username`** (CharField): Nazwa użytkownika.
- **`password`** (CharField): Hasło użytkownika.
- ... (Inne domyślne pola modelu użytkownika Django, np. **`email`**, **`first_name`**, **`last_name`**, itp.)

Tabela `Slowko`

Tabela **`Slowko`** przechowuje dane dotyczące słówek do nauki.

Pola:

- **`id`** (Integer, Primary Key): Unikalny identyfikator słówka.
- **`polskie`** (CharField): Polskie słowo.
- **`angielskie`** (CharField): Angielski odpowiednik słówka.

Metody:

- **`__str__`**: Zwraca reprezentację tekstową słówka w formacie "polskie - angielskie".

Tabela `ZnajomoscSlowka`

Tabela **`ZnajomoscSlowka`** przechowuje relacje między użytkownikami a słówkami, informując, czy dany użytkownik zna dane słówko.

Pola:

- **`id`** (Integer, Primary Key): Unikalny identyfikator rekordu.
- **`user`** (ForeignKey do **`User`**): Odniesienie do użytkownika.
- **`slowko`** (ForeignKey do **`Slowko`**): Odniesienie do słówka.
- **`zna`** (Boolean): Pole informujące, czy użytkownik zna dane słówko.

Relacje:

- **`user_id`** (ForeignKey): Relacja wiele-do-jednego z tabelą **`User`**. Każdy rekord w **`ZnajomoscSlowka`** odnosi się do jednego użytkownika, ale użytkownik może mieć wiele powiązanych rekordów w **`ZnajomoscSlowka`**.
- **`slowko_id`** (ForeignKey): Relacja wiele-do-jednego z tabelą **`Slowko`**. Każdy rekord w **`ZnajomoscSlowka`** odnosi się do jednego słówka, ale słówko może być powiązane z wieloma użytkownikami.

Klucz unikalny:

- **`unique_together = [['user', 'slowko']]`**: Wymusza unikalność kombinacji **`user`** i **`slowko`**, co oznacza, że dany użytkownik może mieć tylko jeden wpis dla danego słówka.

Metody:

- **`__str__`**: Zwraca reprezentację tekstową relacji w formacie "username - słówko - Known/Unknown".

Relacje między tabelami

- **Relacja `User` -> `ZnajomoscSlowka`:**
  - Jeden użytkownik może mieć wiele rekordów w tabeli **`ZnajomoscSlowka`**.
  - Każdy rekord w **`ZnajomoscSlowka`** odnosi się do jednego użytkownika.
- **Relacja `Slowko` -> `ZnajomoscSlowka`:**
  - Jedno słówko może być powiązane z wieloma rekordami w tabeli **`ZnajomoscSlowka`**.
  - Każdy rekord w **`ZnajomoscSlowka`** odnosi się do jednego słówka.

IV. Implementacja:

Modele: models.py

```
from django.db import models

from django.contrib.auth.models import User


from django.dispatch import receiver
from django.db.models.signals import post_save


class Slowko(models.Model):
 polskie = models.CharField(max_length=100)
 angielskie = models.CharField(max_length=100)

 def __str__(self):
 return f"{self.polskie} - {self.angielskie}"

class ZnajomoscSlowka(models.Model):
 user = models.ForeignKey(User, on_delete=models.CASCADE)
```

```
 slowko = models.ForeignKey(Slowko, on_delete=models.CASCADE)
 zna = models.BooleanField(default=False)

 class Meta:
 unique_together = [['user', 'slowko']]

@receiver(post_save, sender=User, dispatch_uid="new user")
def Funkcja(sender, instance, created, raw, using, **kwargs):
 if created:
 for i in Slowko.objects.all():
 ZnajomoscSlowka.objects.create(slowko=i, user = instance)
```

Widoki: views.py

```
from django.shortcuts import render, redirect
from django.contrib.auth import authenticate, login, logout
from django.contrib.auth.forms import AuthenticationForm, UserCreationForm
from django.urls import reverse_lazy, reverse
from django.http import HttpResponse
from django.views import generic
from django.contrib.auth.decorators import login_required
from .models import Slowko, ZnajomoscSlowka
from django.views.decorators.http import require_POST
from django.forms.models import model_to_dict
from django.http import JsonResponse
import json
from django.core.serializers.json import DjangoJSONEncoder



def start(request):
 return render(request, 'start.html')
def login_view(request): #Logowanie
 if request.method == 'POST':
```

```
 form = AuthenticationForm(request, data=request.POST)
 if form.is_valid():
 username = form.cleaned_data.get('username')
 password = form.cleaned_data.get('password')
 user = authenticate(request, username=username, password=password)
 if user is not None:
 login(request, user)
 return redirect('home') # Przekieruj do strony głównej po zalogowaniu.
 else:
 return HttpResponse("Nieudane logowanie. Spróbuj ponownie.", status=401)
 else:
 form = AuthenticationForm()

 return render(request, 'login.html', {'form': form})

@login_required
@require_POST
def logout_request(request):
 logout(request)
 return redirect(reverse('login'))

class SignUpView(generic.CreateView): #Przekierowuje do URL logowania po pomyślnej
rejestracji
 form_class = UserCreationForm
 success_url = reverse_lazy('login') # Przekierowuje do URL logowania po pomyślnej
rejestracji
 template_name = 'signup.html'

@login_required # Dekorator wymusza, aby tylko zalogowani użytkownicy mieli dostęp do
tego widoku
def home(request):
 return render(request, 'home.html')

@login_required
def powtarzanie_view(request):
 return render(request, 'powtarzanie.html')
```

```
@login_required
def nauka_view(request):
 slowka_ktorych_uzytkownik_nie_zna = ZnajomoscSlowka.objects.filter(
 user=request.user,
 zna=False
 ).select_related('slowko')

 slowka_do_nauki = [
 {
 'id': znajomosc.slowko.id,
 'polskie': znajomosc.slowko.polskie,
 'angielskie': znajomosc.slowko.angielskie
 } for znajomosc in slowka_ktorych_uzytkownik_nie_zna
 ]

 # Używamy DjangoJSONEncoder do bezpiecznego konwertowania na JSON
 slowka_do_nauki_json = json.dumps(slowka_do_nauki, cls=DjangoJSONEncoder)

 return render(request, 'nauka.html', {'slowka_do_nauki_json': slowka_do_nauki_json})

@login_required
@require_POST
def update_znajomosc_slowka(request):
 slowko_id = request.POST.get('slowko_id')
 if slowko_id:
 znajomosc, created = ZnajomoscSlowka.objects.get_or_create(user=request.user,
slowko_id=slowko_id, defaults={'zna': True})
 if not created:
 znajomosc.zna = True
 znajomosc.save()
 return JsonResponse({'success': True})
 return JsonResponse({'success': False})


@login_required
def powtarzanie_view(request):
 slowka_do_powtorki = ZnajomoscSlowka.objects.filter(user=request.user,
zna=True).select_related('slowko')
```

```
slowka_do_powtorki_json = json.dumps(
[model_to_dict(znajomosc.slowko, fields=['id', 'angielskie', 'polskie']) for znajomosc in
slowka_do_powtorki],
 cls=DjangoJSONEncoder
)
 return render(request, 'powtarzanie.html', {'slowka_do_powtorki_json':
slowka_do_powtorki_json})
```

Szablony HTML:

Base.html

```
<html lang="en">

<head>
 <!-- Nagłówek -->
 <meta charset="UTF-8">

 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet">
 <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></sc
ript>

 <title>Aplikacja Do Nauki</title>
 <style>


 body {
 background-color: #f9f6e1;
 }

 .clock-container {
 text-align: center;
```

```
}

.clock {
font-size: 2em;
color: #333;
}

.date {
font-size: 1.2em;
color: #666;
}

.animate-top {
animation: animateTop 1.5s linear;
}

/* Animacja schodzenia napisu */

@keyframes animateTop {
from {
transform: translateY(-100px);
}



to {
transform: translateY(0);
}
}

/* Animacja schodzenia napisu */

</style>

<div style="background:
url(https://media.istockphoto.com/id/547214792/pl/zdj%C4%99cie/bia%C5%82ym-tle-%C5%9Bciany-z-ceg%C5%82y.jpg?s=2048x2048&w=is&k=20&c=fTrsFjum98b3GPIpWhGW3luhXlxsqizHsyUh50TezZ0=)" class="page-holder bg-cover">
```

```
<div class="container py-5">
<header class="text-center text-white py-5 animate-top">
<h1 class="display-4 font-weight-bold mb-4">Program do nauki języka </h1>

</div>

</div>
</div>
</head><!-- Nagłówek -->

<body>
<!-- body -->
<div class="container text-center">



<section class="clock-container">
<!-- sekcja zegara -->
<br>
<div class="clock" id="clock"></div>
<div class="date" id="date"></div>
</section> <!-- sekcja zegara -->

<script>
// skrypt zegara
function updateClock() { // update zegara
const now = new Date();
const options = {
weekday: 'long',
year: 'numeric',
month: 'long',
day: 'numeric'
};
const dateStr = now.toLocaleDateString('pl-PL', options);
const timeStr = now.toLocaleTimeString([], {
hour: '2-digit',
minute: '2-digit'
```

```
});

document.getElementById('date').textContent = dateStr;
document.getElementById('clock').textContent = timeStr;
}



setInterval(updateClock, 1000);



updateClock(); // update zegara

</script><!-- skrypt do godziny -->
</div> <!-- Wszystka zawartość umieszczona w tym kontenerze zostanie wyśrodkowana na stronie -->
</body><!-- body -->
{% block content %}{% endblock %}
<script>
function openFlashModal(message, messageType) {
var modal = document.getElementById("flashModal");
var messageParagraph = document.getElementById("flashMessage");
messageParagraph.innerHTML = message;
if (messageType === "error") {
messageParagraph.style.color = "red";
} else {
messageParagraph.style.color = "green";
}
modal.style.display = "block";
}

function closeFlashModal() {
var modal = document.getElementById("flashModal");
modal.style.display = "none";
}
</script>

</body>
</html>
```

Home.html

```
{% extends "base.html" %}

{% block title %}Strona Główna - Moja Aplikacja{% endblock %}

{% block content %}
 <br>
 <div class="container text-center">
 <h1 style="color: #007bff; font-size: 40px;">STRONA GŁÓWNA</h1>
 <br>
 <a href="{% url 'nauka' %}" style="background-color: #28a745; color: white; border: none; border-radius: 5px; padding: 10px 15px; margin-right: 10px; box-shadow: 2px 2px 5px rgba(0,0,0,0.2); text-decoration: none; display: inline-block;">Nauka</a>
 <a href="{% url 'powtarzanie' %}" style="background-color: #007bff; color: white; border: none; border-radius: 5px; padding: 10px 15px; margin-right: 10px; box-shadow: 2px 2px 5px rgba(0,0,0,0.2); text-decoration: none; display: inline-block;">Powtarzanie</a>
 <br>
 <br>

 <form action="{% url 'logout' %}" method="post">
 {% csrf_token %}
 <button type="submit" style="background-color: #dc3545; color: white; border: none; border-radius: 5px; padding: 10px 15px; box-shadow: 2px 2px 5px rgba(0,0,0,0.2);">Wyloguj się</button>
 </form>
 </div>
{% endblock %}
```

Login.html

```
{% extends "base.html" %}

{% block title %}Strona Główna - Moja Aplikacja{% endblock %}
```

```
{% block content %}
 <div class="container text-center">
 <h1 style="color: #007bff; font-size: 40px;">STRONA LOGOWANIA</h1>
 <form method="post">
 {% csrf_token %}
 {{ form.as_p }}
 <button type="submit" style="background-color: #28a745; color: white; border: none; border-radius:
5px; padding: 10px 15px; box-shadow: 2px 2px 5px rgba(0,0,0,0.2);">Zaloguj się</button>
 <a href="{% url 'start' %}" class="button" style="background-color: #ffc107; color: white; border: none;
border-radius: 5px; padding: 10px 15px; text-decoration: none; box-shadow: 2px 2px 5px
rgba(0,0,0,0.2); display: inline-block;">Strona startowa</a>
 </form>
</div>
{% endblock %}
```

Nauka.html

```
{% extends "base.html" %}

{% block title %}Strona Główna - Moja Aplikacja{% endblock %}

{% block content %}
 <div class="container text-center">
 <h1 style="color: #007bff; font-size: 40px;">NAUKA NOWYCH SŁÓWEK</h1>
 <div id="slowkoContainer">
 </div>
 <br>
 <button id="prevButton" onclick="prevSlowko()" style="background-color: #6c757d; color: white;
border: none; border-radius: 5px; padding: 10px 15px; margin-right: 10px; box-shadow: 2px 2px 5px
rgba(0,0,0,0.2);">Poprzednie</button>
<button id="nextButton" onclick="nextSlowko()" style="background-color: #28a745; color: white;
border: none; border-radius: 5px; padding: 10px 15px; margin-right: 10px; box-shadow: 2px 2px 5px
rgba(0,0,0,0.2);">Następne</button>
<button id="knowButton" onclick="znamSlowko()" style="background-color: #007bff; color: white;
```

```
border: none; border-radius: 5px; padding: 10px 15px; margin-right: 10px; box-shadow: 2px 2px 5px
rgba(0,0,0,0.2);">Znam</button>
<a href="{% url 'home' %}" class="button" style="background-color: #ffc107; color: white; border: none;
border-radius: 5px; padding: 10px 15px; text-decoration: none; box-shadow: 2px 2px 5px
rgba(0,0,0,0.2); display: inline-block;">Strona Główna</a>

<form action="{% url 'logout' %}" method="post" style="margin-top: 10px;">
 {% csrf_token %}
 <button type="submit" style="background-color: #dc3545; color: white; border: none; border-radius:
5px; padding: 10px 15px; box-shadow: 2px 2px 5px rgba(0,0,0,0.2);">Wyloguj się</button>
</form>

 <script id="slowka-data" type="application/json">
 {{ slowka_do_nauki_json|safe }}
 </script>
 <script>
 var slowkaDataElement = document.getElementById('slowka-data');
 var slowka = JSON.parse(slowkaDataElement.textContent || slowkaDataElement.innerText);
 var aktualneSlowkoIndex = 0;
 var slowkoContainer = document.getElementById('slowkoContainer');

 function updateView() {
 if (slowka.length > 0) {
 slowkoContainer.innerHTML = slowka[aktualneSlowkoIndex].polskie + ' - ' +
slowka[aktualneSlowkoIndex].angielskie;
 } else {
 slowkoContainer.innerHTML = 'Brak słówek do nauki.';
 document.getElementById('prevButton').style.visibility = 'hidden';
 document.getElementById('nextButton').style.visibility = 'hidden';
 document.getElementById('knowButton').style.visibility = 'hidden';
 }
 }

 function nextSlowko() {
 if (aktualneSlowkoIndex < slowka.length - 1) {
 aktualneSlowkoIndex++;
 updateView();
 }
 }

 function prevSlowko() {
```

```
if (aktualneSlowkoIndex > 0) {
aktualneSlowkoIndex--;
updateView();
}
}

function znamSlowko() {
var csrfToken = document.querySelector('[name=csrfmiddlewaretoken]').value;
var slowkoId = slowka[aktualneSlowkoIndex].id;

fetch('{% url 'update_znajomosc_slowka' %}', {
method: 'POST',
headers: {
'Content-Type': 'application/x-www-form-urlencoded',
'X-CSRFToken': csrfToken
},
body: 'slowko_id=' + encodeURIComponent(slowkoId)
})
.then(response => response.json())
.then(data => {
if (data.success) {
slowka.splice(aktualneSlowkoIndex, 1);
if (aktualneSlowkoIndex >= slowka.length) {
aktualneSlowkoIndex = slowka.length - 1;
}
updateView();
}
});
}

// Inicjalizacja pierwszego słówka
updateView();
</script>
</div>
{% endblock %}
```

Powtarzanie.html

```
{% extends "base.html" %}

{% block title %}Strona Główna - Moja Aplikacja{% endblock %}
```

```
{% block content %}
 <div class="container text-center">
 <h1 style="color: #007bff; font-size: 40px;">POWTARZANIE SŁÓWEK</h1>
 <div id="slowkoContainer">
 <span id="angielskieSlowko"></span>
 <input type="text" id="tłumaczenieInput" placeholder="Wpisz tłumaczenie" style="margin-top: 10px;
padding: 5px; border-radius: 5px; border: 1px solid #ced4da;">
 <button onclick="sprawdzTlumaczenie()" style="background-color: #17a2b8; color: white; border:
none; border-radius: 5px; padding: 10px 15px; box-shadow: 2px 2px 5px
rgba(0,0,0,0.2);">Sprawdź</button>
 </div>
 <br>
 <button id="prevButton" onclick="prevSlowko()" style="background-color: #6c757d; color: white;
border: none; border-radius: 5px; padding: 10px 15px; margin-right: 10px; box-shadow: 2px 2px 5px
rgba(0,0,0,0.2);">Poprzednie</button>
 <button id="nextButton" onclick="nextSlowko()" style="background-color: #28a745; color: white;
border: none; border-radius: 5px; padding: 10px 15px; margin-right: 10px; box-shadow: 2px 2px 5px
rgba(0,0,0,0.2);">Następne</button>
 <a href="{% url 'home' %}" class="button" style="background-color: #ffc107; color: white; border: none;
border-radius: 5px; padding: 10px 15px; text-decoration: none; box-shadow: 2px 2px 5px
rgba(0,0,0,0.2); display: inline-block;">Strona Główna</a>

 <form action="{% url 'logout' %}" method="post" style="margin-top: 10px;">
 {% csrf_token %}
 <button type="submit" style="background-color: #dc3545; color: white; border: none; border-radius:
5px; padding: 10px 15px; box-shadow: 2px 2px 5px rgba(0,0,0,0.2);">Wyloguj się</button>
 </form>

 <script id="slowka-data" type="application/json">
 {{ slowka_do_powtorki_json|safe }}
 </script>
 <script>
 var slowkaDataElement = document.getElementById('slowka-data');
 var slowka = JSON.parse(slowkaDataElement.textContent);
 var aktualneSlowkoIndex = 0;

 function updateView() {
 if (slowka.length > 0) {
 document.getElementById('angielskieSlowko').textContent = slowka[aktualneSlowkoIndex].angielskie;
 document.getElementById('tłumaczenieInput').value = '';
 } else {
```

```
document.getElementById('angielskieSlowko').textContent = 'Brak słówek do powtórzenia.';
document.getElementById('tłumaczenieInput').style.visibility = 'hidden';
document.getElementById('prevButton').style.visibility = 'hidden';
document.getElementById('nextButton').style.visibility = 'hidden';
}
}

function nextSlowko() {
if (aktualneSlowkoIndex < slowka.length - 1) {
aktualneSlowkoIndex++;
updateView();
}
}

function prevSlowko() {
if (aktualneSlowkoIndex > 0) {
aktualneSlowkoIndex--;
updateView();
}
}

function sprawdzTlumaczenie() {
var userTranslation = document.getElementById('tłumaczenieInput').value;
var correctTranslation = slowka[aktualneSlowkoIndex].polskie;
if (userTranslation.trim().toLowerCase() === correctTranslation.toLowerCase()) {
alert("Poprawne tłumaczenie!");
if (aktualneSlowkoIndex < slowka.length - 1) {
aktualneSlowkoIndex++;
updateView();
} else {
alert("To było ostatnie słówko!");
}
} else {
alert("Spróbuj jeszcze raz!");
}
}

updateView();
</script>
</div>
```

```
{% endblock %}
```

Signup.html

```
{% extends "base.html" %}

{% block title %}Strona Główna - Moja Aplikacja{% endblock %}

{% block content %}
 <div class="container text-center">
 <h2 style="color: #007bff; font-size: 30px;">ZAREJESTRUJ SIĘ</h2>
 <form method="post">
 {% csrf_token %}
 {{ form.as_p }}
 <button type="submit" style="background-color: #28a745; color: white; border: none; border-radius:
5px; padding: 10px 15px; box-shadow: 2px 2px 5px rgba(0,0,0,0.2); margin-top: 10px;">Zarejestruj
się</button>
 <a href="{% url 'start' %}" class="button" style="background-color: #ffc107; color: white; border: none;
border-radius: 5px; padding: 10px 15px; text-decoration: none; box-shadow: 2px 2px 5px
rgba(0,0,0,0.2); display: inline-block;">Strona startowa</a>
 </form>
 </div>
{% endblock %}
```

Start.html

```
{% extends "base.html" %}

{% block title %}Strona Główna - Moja Aplikacja{% endblock %}

{% block content %}
 <div class="container text-center">
 <h1 style="color: #007bff; font-size: 40px;">STRONA STARTOWA</h1>
 <form method="post">
```

```
 {% csrf_token %}
 {{ form.as_p }}
 <a href="{% url 'login' %}" style="background-color: #007bff; color: white; border: none; border-radius: 5px; padding: 10px 15px; margin-right: 10px; box-shadow: 2px 2px 5px rgba(0,0,0,0.2); text-decoration: none; display: inline-block;">Zaloguj się</a>
 <a href="{% url 'signup' %}" style="background-color: #28a745; color: white; border: none; border-radius: 5px; padding: 10px 15px; margin-right: 10px; box-shadow: 2px 2px 5px rgba(0,0,0,0.2); text-decoration: none; display: inline-block;">Zarejestruj się</a>


 </form>
</div>
{% endblock %}
```