

# sw-admin

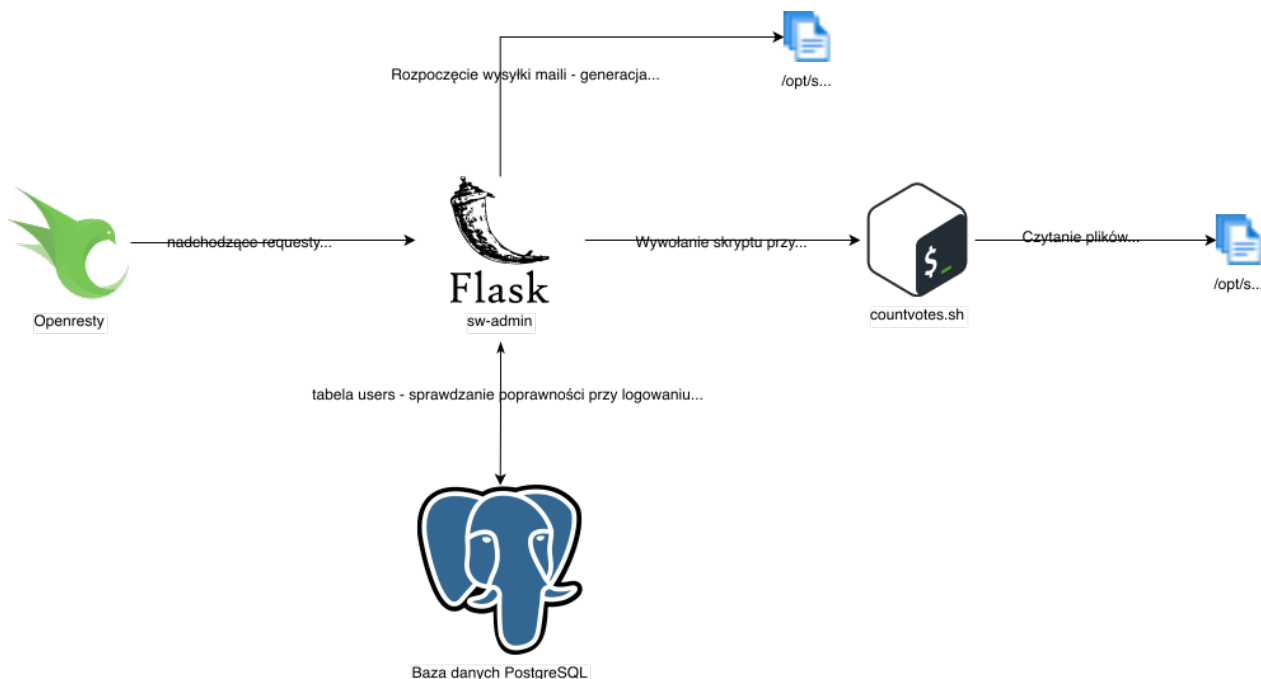


sw-admin jest backendem w napisanym w Pythonie z wykorzystaniem frameworka Flask, który odpowiada za podstrony /admin w SW.

Na serwerze produkcyjnym sw-admin jest uruchamiany za pomocą narzędzia [Gunicorn](#), co pozwala mu działać w kilku procesach jednocześnie oraz prosty sposobem być wywoływany z Openresty.

Katalog templates/ zawiera kod HTML dla każdej podstrony. Wykorzystane są domyślne templatki z Flask - biblioteka Jinja2 oraz Bootstrap.

## Interakcja z resztą systemu



## Zaimplementowane funkcjonalności

Logowanie się na różne konta administratorów - /admin/login



## Studencki System Wyborczy

POLITECHNIKI WROCŁAWSKIEJ

Login

Hasło

Zaloguj się

Logowanie jest zrealizowane z użyciem Flaskowego LoginManager. Każda funkcja ubrana w `@login_required` przekierowuje do `/admin/login` jeżeli użytkownik nie jest zalogowany. ID zalogowanego użytkownika odpowiadające ID w bazie danych można uzyskać wywołując `current_user.get_id()`.

Przekierowanie niezalogowanych użytkowników jest zrealizowane w taki sposób:

```
@login_manager.unauthorized_handler
def unauthorized():
    return redirect('/admin/login', code=303)
```

Uwaga! Adnotacja `@login_required` musi występować poniżej adnotacji `@app.route('...')` - jeżeli wystąpi linijkę wyżej nie ma ona żadnego efektu. Przykład:

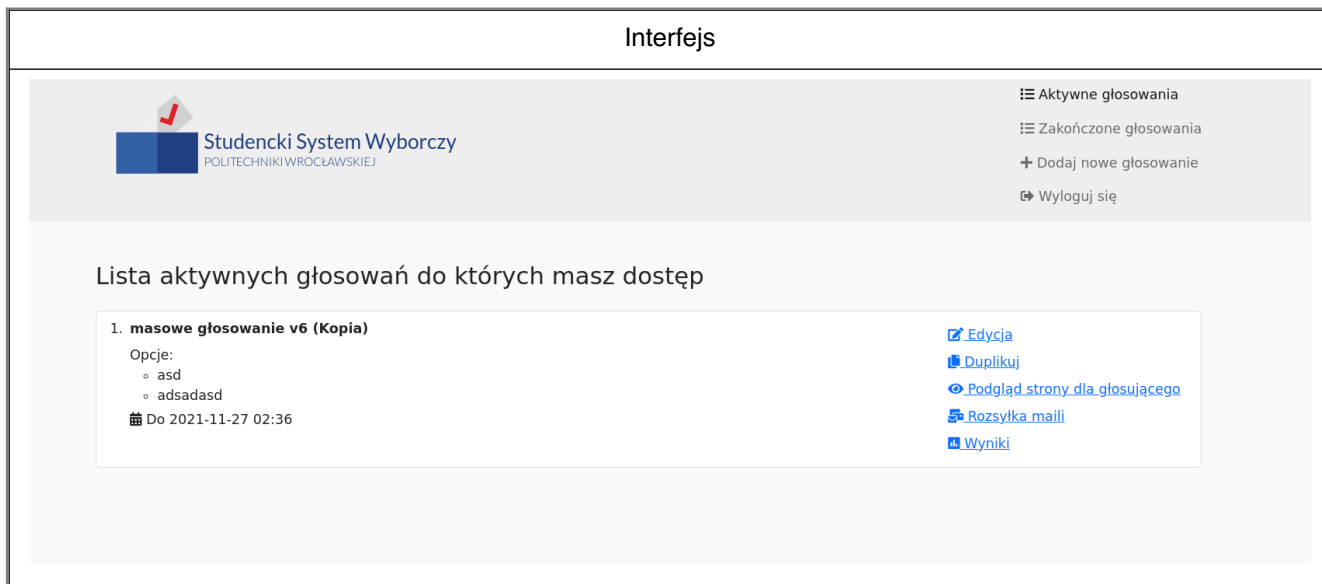
```
# bad code - everyone can see this regardless of whether they are logged in or not
@login_required
@app.route('/admin/addpoll', methods=['GET'])
def admin_addpoll():
    return render_template('addpoll.html')

# good code - only logged in users can see this
@app.route('/admin/addpoll', methods=['GET'])
@login_required
def admin_addpoll():
    return render_template('addpoll.html')
```

Otwarcie strony `/admin/login` wywołuje funkcję `admin_login()` w Pythonie. Naciśnięcie na guzik "Zaloguj się" wywołuje `admin_login_post()`.

Templatka HTML dla logowania: [templates/login.html](#). Rozszerza templatkę [templates/base.html](#).

## Wyświetlanie aktywnych głosowań - `/admin/polls`



Strona wyświetla listę głosowań z bazy danych (tabela `polls`), które:


- nie są zakończone (`closed=false` w bazie danych)
- są stworzone przez użytkownika który jest zalogowany (`owner_user` w bazie danych jest równe ID użytkownika zalogowanego)

Funkcja w pythonie reagująca na otwarciu strony: `admin_polls()`

Templatka HTML: [templates/polls.html](#) (z parametrem `closed=False`) - rozszerza [templates/baseloggedin.html](#) która rozszerza [templates/base.html](#).

## Wyświetlanie zakończonych głosowań - `/admin/closedpolls`

## Interfejs



**Studencki System Wyborczy**  
POLITECHNIKI WROCŁAWSKIEJ

Aktywne głosowania

Zakończone głosowania

+ Dodaj nowe głosowanie

Wyloguj się

### Lista zakończonych głosowań do których masz dostęp

<b>1. Głosowanie testowe</b> Opcje: <ul style="list-style-type: none"><li>Jan Kowalski (W1)</li><li>Adam Nowak (W2)</li></ul> <div>📅 Do 2021-11-11 15:00</div>	<a href="#">Edycja</a> <a href="#">Duplikuj</a> <a href="#">Podgląd strony dla głosującego</a> <a href="#">Rozsyłka maili</a> <a href="#">Wyniki</a>
<b>2. Masywne głosowanie testowe</b> Opcje: <ul style="list-style-type: none"><li>Opcja 1 (W1)</li><li>Opcja 2 (W2)</li><li>Opcja 3 (W3)</li></ul> <div>📅 Do 2021-11-10 03:43</div>	<a href="#">Edycja</a> <a href="#">Duplikuj</a> <a href="#">Podgląd strony dla głosującego</a> <a href="#">Rozsyłka maili</a> <a href="#">Wyniki</a>
<b>3. wybory na 🎮🎮🎮🎮🎮🎮🎮</b> Opcje: <ul style="list-style-type: none"><li>🎮🎮🎮🎮 (wydział gier i zabaw)</li><li>koks (wydział wciągania koksu)</li></ul> <div>📅 Do 2021-11-14 17:03</div>	<a href="#">Edycja</a> <a href="#">Duplikuj</a> <a href="#">Podgląd strony dla głosującego</a> <a href="#">Rozsyłka maili</a> <a href="#">Wyniki</a>

Strona wyświetla listę głosowań z bazy danych (tabela polls ), które:


- są zakończone ( closed=true w bazie danych)
- są stworzone przez użytkownika który jest zalogowany (owner\_user w bazie danych jest równe ID użytkownika zalogowanego)

Funkcja w pythonie reagująca na otwarciu strony: admin\_closedpolls()

Templatka HTML: [templates/polls.html](#) (z parametrem closed=True ) - rozszerza [templates/baseloggedin.html](#) która rozszerza [templates/base.html](#).

Tworzenie głosowań - /admin/addpoll

## Interfejs



**Studencki System Wyborczy**  
POLITECHNIKI WROCŁAWSKIEJ

Aktywne głosowania

Zakończone głosowania

+ Dodaj nowe głosowanie

Wyloguj się

### Dodawanie nowego głosowania

Nazwa głosowania:

Wybory na prezydenta Rzeczypospolitej Polskiej

Opcje do wyboru:

Wybór	Opis
<a href="#">+ Dodaj nową opcję</a>	

Na ile opcji jedna osoba może maksymalnie zagłosować: 1

Głosujący (adresy email w nowych liniach):

1@student.pwr.edu.pl  
2@student.pwr.edu.pl  
...

Dodaje nowe głosowanie z `owner_user` ustawionym jako ID aktualnie zalogowanego użytkownika.

Funkcja w pythonie reagująca na otwarciu strony: `admin_addpoll()`

Funkcja w pythonie reagująca na naciśnięciu przycisku "Dodaj głosowanie": `admin_addpoll_post()`

Templatka HTML: [templates/addpoll.html](#) - rozszerza [templates/baseloggedin.html](#) która rozszerza [templates/base.html](#).

Edytowanie głosowań - `/admin/editpoll?id={poll_id}`

## Interfejs

The screenshot shows the 'Studencki System Wyborczy' (Student Election System) interface. At the top, there is a header with the system logo and name, and a sidebar with navigation links: 'Aktywne głosowania', 'Zakończone głosowania', 'Dodaj nowe głosowanie', and 'Wyloguj się'. A yellow warning banner at the top states: 'Nie można edytować głosowania ponieważ wiadomości zostały już wysłane lub zakolejnowane do wysłania do 1 osób'. The main section is titled 'Edycja głosowania "Głosowanie testowe"'. It contains a form for editing the poll. The 'Nazwa głosowania:' field is filled with 'Głosowanie testowe'. Below it, the 'Opcje do wyboru:' section contains a table with two columns: 'Wybór' and 'Opis'. The table has two rows: one with 'Jan Kowalski' and 'W1', and another with 'Adam Nowak' and 'W2'. Each row has a red trash icon to its right. Below the table is a blue button labeled '+ Dodaj nową opcję'. At the bottom, there is a field for 'Na ile opcji jedna osoba może maksymalnie zagłosować:' with the value '1'.

**Studencki System Wyborczy**  
POLITECHNIKI WROCŁAWSKIEJ

Aktywne głosowania  
Zakończone głosowania  
Dodaj nowe głosowanie  
Wyloguj się

⚠ Nie można edytować głosowania ponieważ wiadomości zostały już wysłane lub zakolejnowane do wysłania do 1 osób

### Edycja głosowania "Głosowanie testowe"

Nazwa głosowania:  
Głosowanie testowe

Opcje do wyboru:

Wybór	Opis	
Jan Kowalski	W1	
Adam Nowak	W2	

+ Dodaj nową opcję

Na ile opcji jedna osoba może maksymalnie zagłosować: 1

Pozwala na edytowanie głosowania z id `poll_id`. Przed wykonaniem czegokolwiek sprawdza czy głosowanie należy do aktualnie zalogowanego użytkownika ( `owner_user` ).

Logika jest bardzo podobna co przy dodawaniu głosowania (`/admin/addpoll`) - z takimi różnicami że wartości w interfejsie są od razu załadowane z tabeli `polls`, a naciśnięcie przycisku "Zapisz głosowanie" nie tworzy nowego głosowania ale modyfikuje już istniejące.


Funkcja w pythonie reagująca na otwarcie strony: `admin_editpoll()`

Funkcja w pythonie reagująca na naciśnięcie przycisku "Zapisz głosowanie": `admin_editpoll_post()`

Templatka HTML: [templates/editpoll.html](#) - rozszerza [templates/baseloggedin.html](#) która rozszerza [templates/base.html](#).

Duplikowanie głosowań - `/admin/copypoll?id={poll_id}`

## Interfejs



**Studencki System Wyborczy**  
POLITECHNIKI WROCŁAWSKIEJ

Aktywne głosowania

Zakończone głosowania

+ Dodaj nowe głosowanie



Wyloguj się

### Tworzenie nowego głosowania z głosowania "masowe głosowanie v6 (Kopia)"

Nazwa głosowania:

masowe głosowanie v6 (Kopia) (Kopia)

Opcje do wyboru:

Wybór	Opis	
asd	Wydział Architektury	
adsadasd	Wydział Architektury	
<a href="#">+ Dodaj nową opcję</a>		

Na ile opcji jedna osoba może maksymalnie zagłosować: 1

Głosujący (adresy email w nowych liniach):

informatyzacja@samorzad.pwr.edu.pl###1  
informatyzacja@samorzad.pwr.edu.pl###2

Duplikowanie głosowania wypełnia wartości w interfejsie tak jak edytowanie głosowania, jednak naciśnięcie przycisku "Dodaj głosowanie" na dole strony powoduje dodanie nowego głosowania zamiast edycji istniejącego.

Funkcja w pythonie reagująca na otwarciu strony: `admin_copypoll()`

Reakcja na przycisk wykorzystuje logikę dodawania głosowania - jest to POST do `/admin/addpoll` wywołujący `admin_addpoll_post()`.

Templatka HTML: [templates/copypoll.html](#) - rozszerza [templates/editpoll.html](#) która rozszerza [templates/baseloggedin.html](#) która rozszerza [templates/base.html](#).

Podgląd strony dla głosującego - `/admin/peek?id={poll_id}`



## Studencki System Wyborczy

POLITECHNIKI WROCŁAWSKIEJ

### masowe głosowanie v6 (Kopia)

☐ asd  
☐ adsadasd

Pozostałe głosy: 1

Głosuj

Podgląd strony wykorzystuje tę samą logikę jak generacja pliku `/opt/sw/poll/{poll_id}/index.html` przy starcie wysyłki maili, jedynie ze zmienioną ścieżką. Dlatego przy podglądzie generowany jest plik `/opt/sw/poll/{poll_id}/index-peek.html`, po czym jest on czytany i zwracany do przeglądarki.

Funkcja w pythonie: `look_at_poll()`

Funkcja w pythonie nie robiąca nic oprócz przekierowania użytkownika do listy głosowań, ale reagująca na przycisk "Głosuj" naciśnięty przy podglądzie: `admin_peek_vote()`.

Templatka HTML: [templates/vote.html](#) - rozszerza [templates/base.html](#).

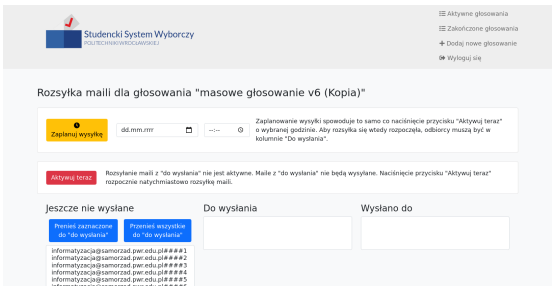
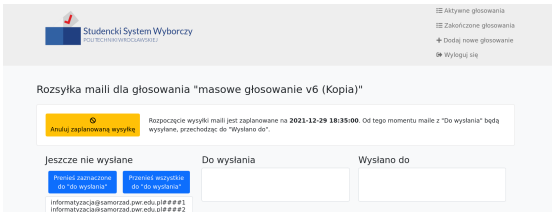
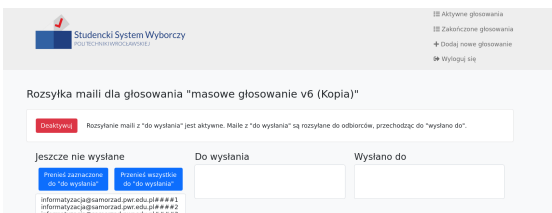
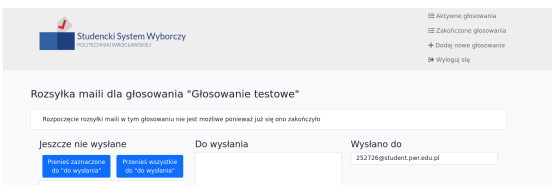
Rozsyłka maili - `/admin/sendout?id={poll_id}`

Interfejs rozsyłki maili zależy od stanu w jakim znajduje się głosowanie.

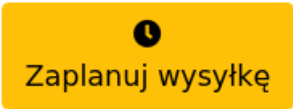

Funkcja w pythonie reagująca na wejście na stronę: `admin_sendout()`

Interfejs w zależności od stanu głosowania w bazie danych:



closed	mailing_active	planned_start_sending	Interfejs
false	false	NULL	
false	false	not NULL	
false	true	cokolwiek	
true	cokolwiek	cokolwiek	

Przyciski w widoku rozsyłki maili:

Przycisk Akcja POST i funkcja ją obsługująca	Co dzieje się po naciśnięciu
 <p>Akcja: /admin/sendout/plan?id={poll_id} Funkcja: admin_sendout_plan_post()</p>	<ul style="list-style-type: none"> <li>planned_start_sending głosowania jest ustawione na podaną datę/godzinę (w strefie czasowej Europe/Warsaw )</li> </ul> <p>Gdy minie zaplanowana godzina, sw_start_planned_mailing.py automatycznie zaimportuje sw_admin.py i wywoła activate_mailing() symulując naciśnięcie na guzik "Aktywuj teraz", rozpoczynając głosowanie.</p>
 <p>Akcja: /admin/sendout/unplan?id={poll_id} Funkcja: admin_sendout_unplan_post()</p>	<ul style="list-style-type: none"> <li>planned_start_sending głosowania jest ustawione na NULL</li> </ul>
	<ul style="list-style-type: none"> <li>Katalogi /opt/sw/poll/{poll_id}/ oraz /opt/sw/poll/{poll_id}/results/ są tworzone jeżeli nie istnieją</li> <li>Plik /opt/sw/poll/{poll_id}/index.html jest</li> </ul>

<div data-bbox="296 98 536 163">Aktywuj teraz</div> <p>Akcja: /admin/sendout/activatemailing? id={poll_id}</p> <p>Funkcja: admin_sendout_activatemailing_post()</p>	<p>tworzony, generowany na podstawie templatki <a href="templates/vote.html">templates/vote.html</a></p> <ul style="list-style-type: none"> <li>mailing_active głosowania jest ustawione na true</li> <li>planned_start_sending głosowania jest ustawione na NULL</li> </ul> <p>Ta funkcjonalność ( activate_mailing() z sw_admin.py) jest też wywoływana przez sw_start_planned_mailing.py gdy planned_start_sending jest ustawiony na datę/godzinę która nadeszła</p>
<div data-bbox="317 533 515 598">Deaktywuj</div> <p>Akcja: /admin/sendout/deactivatemailing?id= {poll_id}</p> <p>Funkcja: admin_sendout_deactivatemailing_post()</p>	<ul style="list-style-type: none"> <li>mailing_active głosowania jest ustawione na false</li> </ul>
<div data-bbox="253 835 577 945">Przenieś wszystkie do "do wysłania"</div> <p>Akcja: /admin/sendout/queueall?id= {poll_id}</p> <p>Funkcja: admin_sendout_queueall_post()</p>	<ul style="list-style-type: none"> <li>Tablica adresów email sending_out_to dla głosowania zostaje ustawiona na zawartość tablicy possible_recipients</li> </ul>
<div data-bbox="253 1102 582 1211">Przenieś zaznaczone do "do wysłania"</div> <p>Akcja: /admin/sendout/queueselect?id= {poll_id}</p> <p>Funkcja: admin_sendout_queueselect_post()</p>	<ul style="list-style-type: none"> <li>Wybrane adresy email zostają dodany do tablicy adresów possible_recipients</li> </ul>

Jakie adresy email znajdują się na listach w interfejsie:

Lista	Adres znajduje się w possible_recipients	Adres znajduje się w sending_out_to	Adres znajduje się w sent_to
<p>Jeszcze nie wysłane</p> <div> <div>Przenieś zaznaczone do "do wysłania"</div> <div>Przenieś wszystkie do "do wysłania"</div> </div> <div></div>	nie	tak	tak
<p>Do wysłania</p> <div></div>	nie	nie	tak
<p>Wysłano do</p> <div>252726@student.pwr.edu.pl</div> <div></div>	nie	nie	nie

Dla klasyfikacji tabelki: w "Wysłano do" wyświetlają się adresy które są we wszystkich trzech tablicach, a w "Jeszcze nie wysłane" adresy które są jedynie w `possible_recipients`.

## Wyniki głosowania - /admin/results?id={poll\_id}

Interfejs											
<div> <div>  <div> <b>Studencki System Wyborczy</b>  <small>POLITECHNIKI WROCŁAWSKIEJ</small> </div> </div> <div> <ul style="list-style-type: none"> <li>Aktywne głosowania</li> <li>Zakończone głosowania</li> <li>+ Dodaj nowe głosowanie</li> <li>Wyloguj się</li> </ul> </div> </div> <div> <p>Wyniki głosowania "masowe głosowanie v6 (Kopia)"</p> <div> <p>Uprawnionych do głosowania: 1000 osób</p> <p>Dodanych do "do wysłania": 0 osób</p> <p>Wysłano mail do: 0 osób</p> <p>Zagłosowało: 0 osób</p> </div> <table> <tr> <th>Wybór</th><th>Opis</th><th>Ilość głosów</th></tr> <tr> <td>asd</td><td></td><td>0</td></tr> <tr> <td>adsadasd</td><td></td><td>0</td></tr> </table> </div>			Wybór	Opis	Ilość głosów	asd		0	adsadasd		0
Wybór	Opis	Ilość głosów									
asd		0									
adsadasd		0									

Wyniki głosowania są liczone przez skrypt `countvotes.sh` przy otwarciu strony z wynikami. Głosy są zapisane w osobnych plikach w katalogu `/opt/sw/polls/{poll_id}/results/`. Skrypt czyta każdy plik w tym katalogu i podaje wyniki, które są pokazywane na stronie.

Format plików z głosami

Przykład pliku z głosem gdzie głosujący wybrał opcje 0 ("Jan Kowalski"), 2 ("Jan Nowak"), oraz 13 ("Adam

Nowak") (pierwsza linijka pusta):

```
option_0=Jan%20Kowalski&option_2=Jan%20Nowak&option_13=Adam%20Nowak
```

Przykład pliku gdzie głosujący wybrał jedynie opcję 1 ("Jan Nowakowski"):

```
option_1=Jan%20Nowakowski
```

Przykład pliku gdzie głosujący oddał pusty głos (pusty plik):

Funkcja w pythonie reagująca na otwarciu strony: `admin_results()`

Templatka HTML: [templates/results.html](#) - rozszerza [templates/baseloggedin.html](#) która rozszerza [templates/base.html](#).

---