

# Dokumentacja projektu na Technologie Internetowe

---

Łukasz Stachnik, Informatyka z specjalnością Algorytmika i Programowanie

Prowadzący ćwiczenia: mgr Szymon Głownia

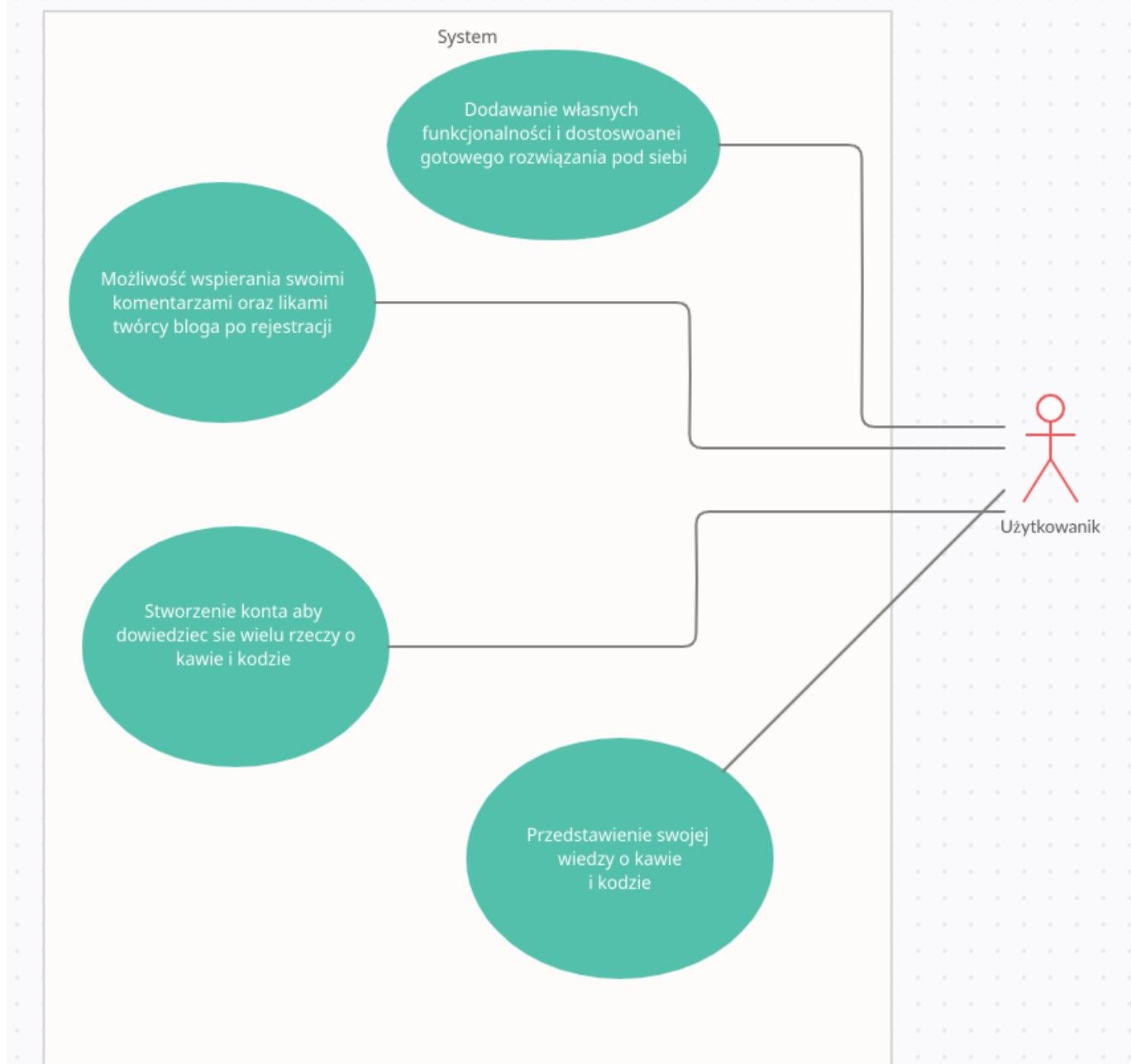
Temat projektu: Platforma blogowa pasjonatów kawy i kodu

[Repozytorium na GitHub](#)

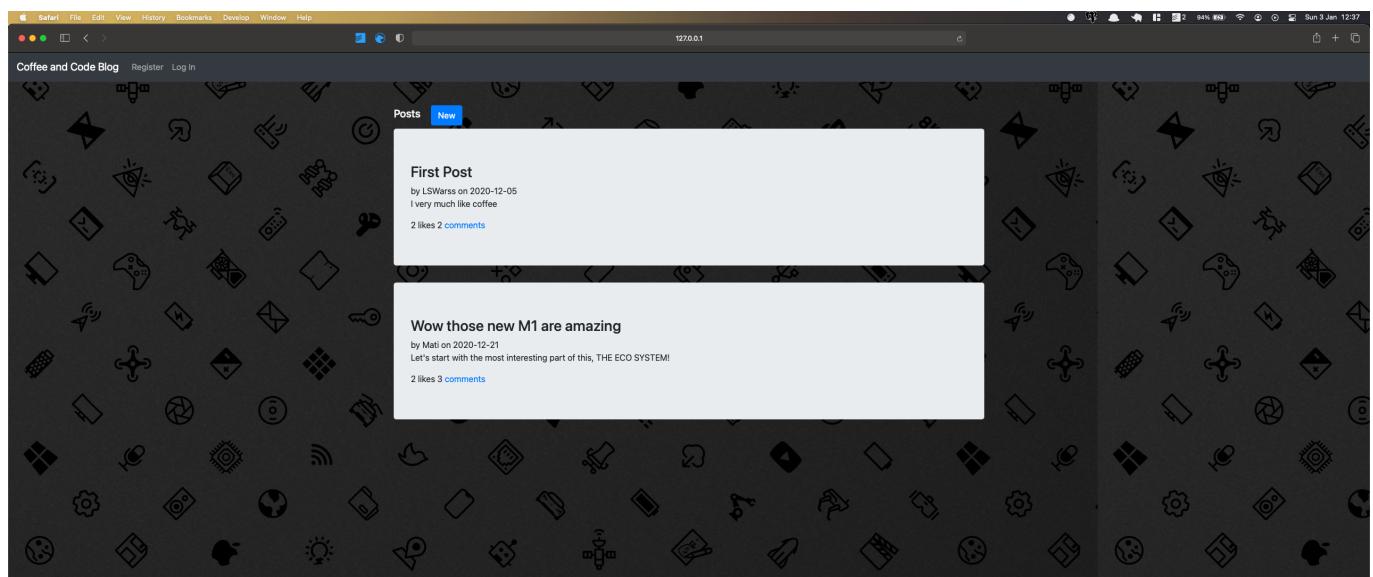
## 1. Opis programu

Program to aplikacja webowa pozwalająca na postawienie własnego bloga o kawie i kodzie. Z autentykacją, zabezpieczeniami zapytań do bazy oraz jeśli chemy też gotowymi ustawieniami pod deployment na Heroku. W skład aplikacji wchodzi strona główna gdzie widnieją wszystkie posty, do postów można dodawać komentarze, likować czy je edytować jeśli jesteśmy użytkownikiem który stworzył dany post. Wszystko jest podpięte pod bazę PostgreSQL, przy użyciu SQL Alchemy dla pythona, co pozwala nam na operowanie na modelach i tworzeniu instancji w bazie danych z poziomu kodu oraz łatwą kontrolę nad migracjami.

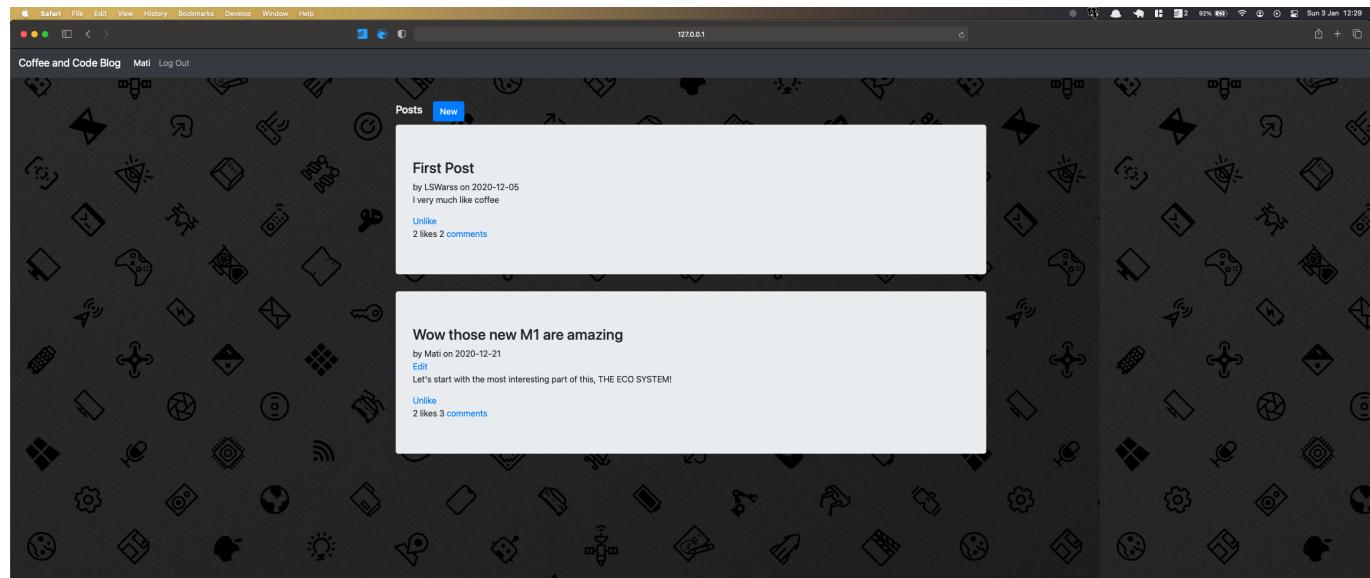
## 2. Diagram przykadów użycia



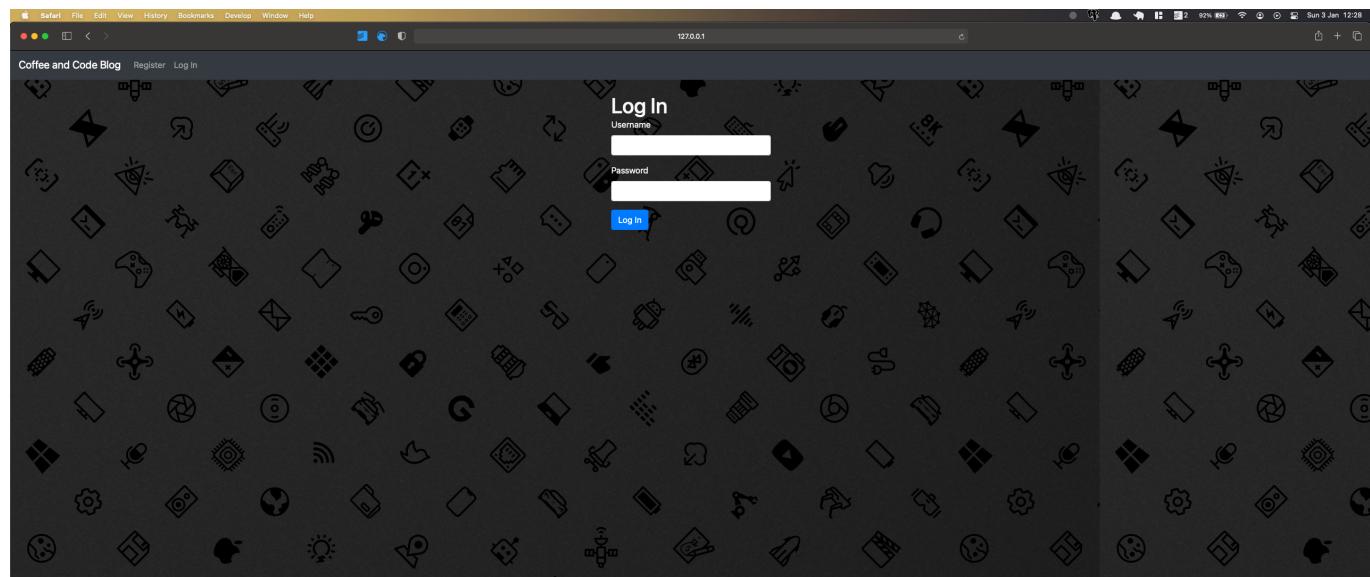
### 3. PrtScr projektu z opisami



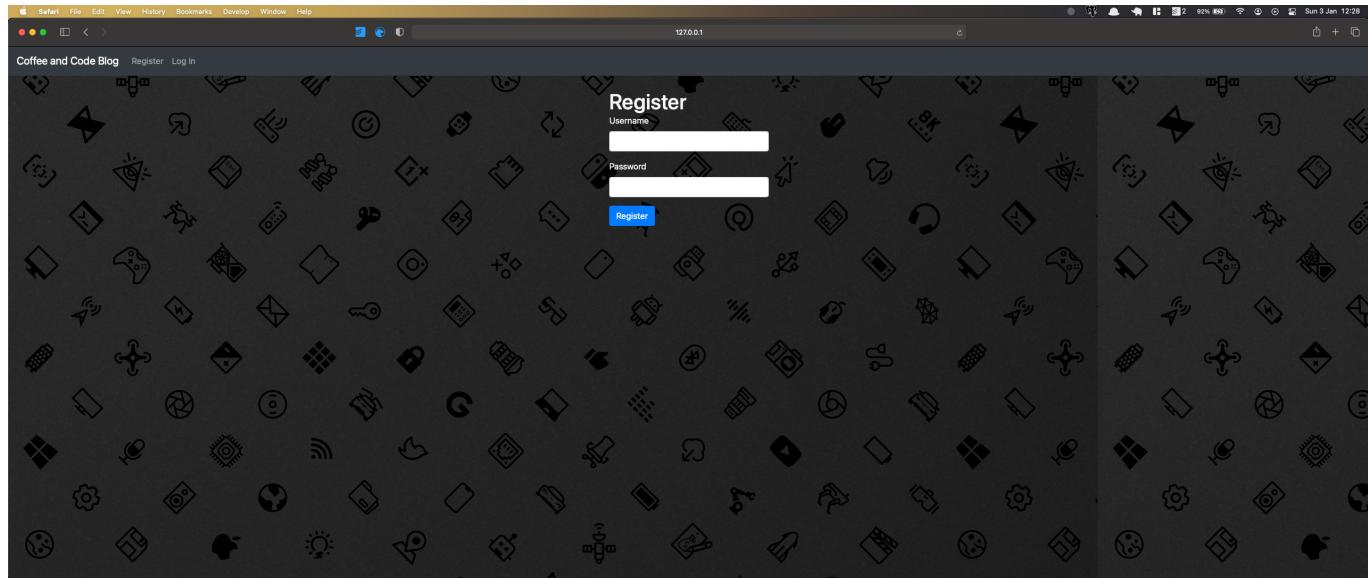
To jest główna strona aplikacji jeszcze przed zalogowaniem, Flask daje nam opcje łatwego ukrywania elementów przed zalogowaniem przy użyciu templatów i rozszerzania plików html o elementy z innych plików html co widać na powyższym obrazku gdyż baza posiada tylko navbar (stylizowany z pomocą boostrapa) a reszta zmienia się zależnie od tego jaką akcję wykonamy.



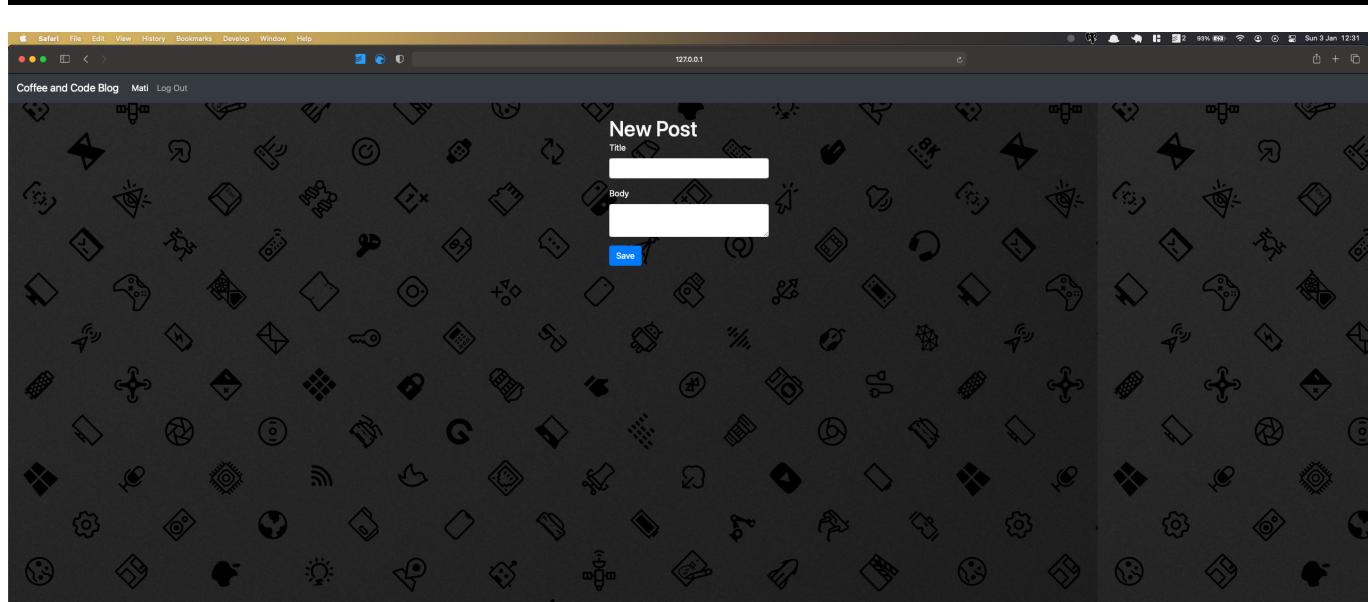
Tutaj widzimy już główną stronę po zalogowaniu, użytkownik może edytować oraz usuwać tylko swoje posty.



To jest strona logowania, przy złym wpisaniu hasła czy nicku pokaże nam się wiadomość że są niepoprawne i mamy je sprawdzić. Dodatkowo hasło jest hashowane w bazie danych także w platformie zadbałem o bezpieczeństwo oraz prywatność użytkownika, po za tym przy wysyłaniu requestów post, używany jest token csrf zabezpieczający jest przed man in the middle attacks z strony hackera.



Tutaj bardzo podobny form do rejestracji oczywiście z tymi samymi zabezpieczeniami co w wypadku logowania. Oraz tutaj następuje hasowanie hasła do bazy danych.



Ekran tworzenia nowego posta, dostepny dla uzytkownika z poziomu strony glownej tylko po zalogowaniu.

# New Post

Title

Body

Please fill out this field.

asdasd

Save

Wszystkie pola w formularzach na stronie posiadają warningi o tym że mamy wpisać dane by przejść dalej zabezpieczające bazę przed problemami z typem czy też nullami.

---

```
Terminal
Terminal x Terminal
=====
===== FAILURES =====
===== test_create_update_validate[/create] =====
client = <FlaskClient <Flask 'flaskr'>>, auth = <conftest.AuthActions object at 0x7f6e17a13e80>, path = '/create'

@pytest.mark.parametrize('path', (
    '/create',
    '/1/update',
))
def test_create_update_validate(client, auth, path):
    auth.login()
    response = client.post(path, data={'title': '', 'body': ''})
E   assert b'Title is required.' in b'<!DOCTYPE html><title>New Post - Flaskr </title><link rel="stylesheet" href="/static/style.css"><nav>
<h1... <textarea name="body" id="body"></body></textarea><n         <input type="submit" value="Save"><n         </form><n><section>
E   + where b'<!DOCTYPE html><n><title>New Post - Flaskr </title><n><link rel="stylesheet" href="/static/style.css"><n><nav>
E   me="body" id="body"></textarea><n         <input type="submit" value="Save"><n         </form><n><section>' = <Response 703 bytes [200 OK]>.data
tests/test_blog.py:80: AssertionError
=====
===== short test summary info =====
FAILED tests/test_blog.py::test_create_update_validate[/create] - assert b'Title is required.' in b'<!DOCTYPE html><title>New Post - Flaskr </title>...
===== 1 failed, 21 passed in 1.94s =====
env > warss > Milijon > ~ > Workspace > flask-tutorial > pytest
=====
===== test session starts =====
platform linux -- Python 3.8.5, pytest-6.1.2, py-1.9.0, pluggy-0.13.1
rootdir: /home/warss/Workspace/flask-tutorial, configfile: setup.cfg, testpaths: tests
collected 22 items

tests/test_auth.py .....
tests/test_blog.py .....
tests/test_db.py ...

===== 22 passed in 1.48s =====
env > warss > Milijon > ~ > Workspace > flask-tutorial > pytest
[ 36%]
[ 90%]
[100%]
```

Aplikacja ma podstawowe testy jednostkowe z użyciem PyTest.