

# **Aplikacja przeglądarkowa we Flasku - HSAB – ocenianie wpisów**

## 1. Czym jest Flask?

Flask to popularny framework aplikacji internetowych napisany w języku Python. Jest to minimalistyczny i łatwy w użyciu framework, który umożliwia tworzenie skalowalnych aplikacji internetowych i API.

Flask oparty jest na języku Python i korzysta z biblioteki Werkzeug do obsługi routingu i obsługi żądań HTTP oraz Jinja2 do generowania dynamicznych szablonów HTML. Dzięki swojej prostocie, Flask jest często wybierany przez programistów do tworzenia prototypów, aplikacji internetowych o niskim stopniu złożoności oraz mikroservisów.

Główne cechy Flaska to:

- **Routing:** Flask umożliwia definiowanie różnych adresów URL (routing), które odpowiadają na żądania HTTP. Na przykład, możemy zdefiniować funkcję, która obsługuje żądanie GET na adresie `"/hello"` i zwraca `"Hello, World!"`.
- **Obsługa żądań HTTP:** Flask dostarcza wiele narzędzi do obsługi różnych rodzajów żądań HTTP, takich jak GET, POST, PUT, DELETE itp. Możemy łatwo pobierać dane przesłane przez użytkownika, przekierowywać użytkownika na inne strony, obsługiwać pliki statyczne i wiele więcej.
- **Szablony:** Flask korzysta z silnika szablonów Jinja2, który umożliwia generowanie dynamicznych stron HTML. Możemy tworzyć szablony zawierające zmienne i pętle, co ułatwia generowanie złożonych stron internetowych.
- **Rozszerzalność:** Flask jest bardzo elastyczny i można go rozszerzać za pomocą różnych rozszerzeń. Istnieje wiele dostępnych rozszerzeń, które ułatwiają integrację z bazami danych, uwierzytelnianie użytkowników, obsługę formularzy i wiele innych funkcji.

Dzięki prostocie i elastyczności, Flask cieszy się dużą popularnością w społeczności programistów Pythona i jest szeroko stosowany do tworzenia różnego rodzaju aplikacji internetowych.

## 2. Opis zadania

Celem zadania jest stworzenie dynamicznej strony internetowej opartej na bibliotece Flask. Wymagania funkcjonalne:

- formularz z danymi zapisywanymi do bazy danych,
- osadzone elementy multimedialne,
- przynajmniej dwie podstrony

## 3. Aplikacja – główne założenia

Projekt HSAB zakłada stworzenie prostego systemu udostępniania krótkich wpisów typu: cytaty, sentencja, złota myśl, kawał oraz możliwość ocenienia danego wpisu.

System będzie pozwalał na:

- Dodaniu wpisu z limitem znaków ( 128 znaków )
- Edycji wpisu jako administrator
- Usunięcie wpisu jako administrator
- Ocenienie wpisu - Poziom fajności
- Resetowanie Poziomu fajności jako administrator

#### 4. Model logowania administratora

W projekcie występuje jeden model (User).

Implementacja w kodzie aplikacji wygląda następująco:

```
43 class User(UserMixin):
44     def __init__(self, id, username, password):
45         self.id = id
46         self.username = username
47         self.password = password
48         self.authenticated = False
49
50     def is_active(self):
51         return self.is_active()
52
53     def is_anonymous(self):
54         return False
55
56     def is_authenticated(self):
57         return self.authenticated
58
59     def is_active(self):
60         return True
61
62     def get_id(self):
63         return self.id
64
```

UserMixin to klasa dostarczana przez bibliotekę Flask-Login, która zawiera gotowe metody i właściwości pomocne przy implementacji systemu uwierzytelniania użytkowników w aplikacjach internetowych opartych na Flask.

Korzystanie z UserMixin pozwala na łatwe rozszerzenie naszej klasy użytkownika (np. modelu bazy danych) o niezbędne funkcje związane z uwierzytelnianiem. Obejmuje ona kilka gotowych metod i właściwości, takich jak:

1. `is_authenticated`: Właściwość logiczna (property), która zwraca `True`, jeśli użytkownik jest uwierzytelniony, tj. jeśli sesja użytkownika jest aktywna.
2. `is_active`: Właściwość logiczna, która zwraca `True`, jeśli konto użytkownika jest aktywne. Może być przydatne, jeśli chcemy implementować funkcjonalność blokady konta lub dezaktywacji użytkownika.
3. `is_anonymous`: Właściwość logiczna, która zwraca `True`, jeśli użytkownik jest anonimowy, tj. jeśli nie został uwierzytelniony.
4. `get_id()`: Metoda, która zwraca unikalny identyfikator użytkownika. Może być używana przez Flask-Login do przechowywania identyfikatora użytkownika w sesji.

Dzięki dziedziczeniu z UserMixin i implementacji wymaganych metod, możemy łatwo zintegrować naszą klasę użytkownika z Flask-Login. Flask-Login zapewnia dodatkowe funkcje, takie jak uwierzytelnianie, przechowywanie sesji użytkownika i ochrona tras przed nieuwierzytelnionym dostępem.

## 5. Baza danych

Schemat bazy danych ( znajduje się w pliku `schema.sql` w katalogu głównym aplikacji )

```
1 DROP TABLE IF EXISTS posts;
2
3 CREATE TABLE posts
4 (
5     id      INTEGER PRIMARY KEY AUTOINCREMENT,
6     created TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
7     content TEXT      NOT NULL,
8     vote    INTEGER   NOT NULL DEFAULT '0'
9 );
10
11 DROP TABLE IF EXISTS login;
12
13 create table login
14 (
15     id  INTEGER PRIMARY KEY AUTOINCREMENT,
16     username TEXT NOT NULL,
17     password TEXT NOT NULL,
18     access INTEGER NOT NULL DEFAULT '0'
19 );
```

W aplikacji wykorzystywana jest baza danych SQLite3.

SQLite3 to lekka, samodzielna i bezserwerowa baza danych, która jest wbudowana w język Python. Jest to popularny wybór jako lokalna baza danych do aplikacji desktopowych i aplikacji internetowych o niskim poziomie złożoności.

Oto kilka cech i cech charakterystycznych dla SQLite3:

- Bezserwerowa: SQLite3 jest bezserwerową bazą danych, co oznacza, że nie wymaga osobnego procesu serwera do obsługi zapytań. Baza danych jest przechowywana w jednym pliku, który można łatwo przenosić między różnymi środowiskami.
- Wbudowany w Pythona: SQLite3 jest wbudowany w język Python, co oznacza, że nie jest konieczna instalacja dodatkowego oprogramowania lub sterowników. Moduł `sqlite3` jest częścią biblioteki standardowej Pythona i jest dostępny "od ręki".
- Kompaktowy rozmiar: Bazy danych SQLite3 są zapisywane jako pojedynczy plik na dysku, co czyni je łatwymi do przenoszenia i udostępniania. Plik bazy danych może mieć rozmiar od kilku kilobajtów do kilkuset megabajtów, w zależności od ilości danych.
- Transakcje i wsparcie ACID: SQLite3 zapewnia transakcyjność i wsparcie dla ACID (Atomicity, Consistency, Isolation, Durability). Można używać transakcji, aby wykonywać zbiór operacji jako jedną jednostkę, zapewniając integralność danych i unikając utraty danych w przypadku awarii.
- Składnia SQL: SQLite3 obsługuje standardową składnię SQL, dzięki czemu można wykonywać różne operacje na danych, takie jak tworzenie tabel, wstawianie, aktualizacja, usuwanie danych i wiele więcej. Moduł `sqlite3` w Pythonie udostępnia interfejs do wykonywania zapytań SQL na bazie danych SQLite3.

SQLite3 jest szczególnie przydatne w przypadku prostych aplikacji, które wymagają lokalnej bazy danych do przechowywania danych, takich jak aplikacje notatników, aplikacje desktopowe, prototypy, testowanie i wiele innych. Jego prostota, niezawodność i niewymagający charakter czynią go popularnym wyborem w wielu projektach.

Inicjalizację bazy danych wykonujemy skryptem `init_db.py` znajdujący się w katalogu głównym.

## 6. Możliwości aplikacji

### 6.1 Strona główna

Strona główna aplikacji prezentuje się następująco. Wyświetlane są wpisy od najnowszego.

The screenshot shows the main page of the HSAB application. At the top is a dark navigation bar with the text "HSAB" and links for "Strona główna", "Nowy wpis", and "Kontakt". A "Zaloguj" button is on the right. Below the navigation bar is a "Sortuj wpisy" dropdown menu. The main content area displays four posts, each with a title, a row of buttons for "Fajne", "Poziom fajności" (with a counter), and "Słabe", and a timestamp. The posts are:

- Kto nie ma szczęścia w kartach, ten nie ma pieniędzy na miłość!** (2023-06-08 13:55:33)
- Najwięcej blizn zostawiają bliźni.** (2023-06-08 13:55:30)
- Boję się, że Reni Jusis kiedyś mnie znajdzie...** (2023-06-08 13:55:27)
- Te upały przypominają człowiekowi, w jakiej części składa się z wody.** (2023-06-08 13:55:24)

### 6.2 Sortowanie

W prawym górnym rogu po kliknięciu „Sortuj wpisy” mamy możliwość posortowania wpisów.

This screenshot shows the same main page as before, but with the "Sortuj wpisy" dropdown menu open. A red arrow points to the dropdown button. The menu lists four sorting options: "Data (Najnowsze)", "Data (Najstarsze)", "Poziom fajności (Najlepsze)", and "Poziom fajności (Najgorsze)". The first post is visible below the menu.

Standardowo sortowanie ustawione jest po Id wpisu -> Najnowszy

### 6.3 Ocena wpisu

Każdy wpis ma możliwość oceny. Kliknięcie Fajne (+1) lub Słabe (-1) ustawia licznik Poziomu fajności. ( Dodatkowo wyświetlany jest komunikat o poprawnym kliknięciu przycisku)

Ustawiono poziom fajności 🍌 !

Sortuj wpisy ▾

Kto nie ma szczęścia w kartach, ten nie ma pieniędzy na miłość!

👍 Fajne

🍌 Poziom fajności 1 🍌

👎 Słabe

2023-06-08 13:55:33

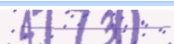
## 6.4 Nowy wpis

Aby dodać nowy wpis należy kliknąć w menu głównym „Nowy wpis” i wpisać odpowiednie dane.

### Utwórz nowy wpis

Coś ciekawego

Limit znaków  
do wpisania:  
115



Przepisz kod z obrazka

Wyślij

AW @ 2023

Każdy może dodać nowy wpis lecz są pewne obostrzenia.

- Długość wpisu nie może być dłuższa niż 128 znaków ( przy polu tekstowym jest licznik który pokazuje ilość znaków do wpisania.
- Dodawanie wpisu jest zabezpieczone kodem CAPTCHA który trzeba przepisać z obrazka w odpowiednie pole.

Te dwa pola są sprawdzane i w przypadku błędu wyświetlany jest komunikat.

Po poprawnym dodaniu wpisu zostajemy przekierowani na stronę główną oraz zostaje wyświetlony komunikat.

Dodano nowy wpis.

Sortuj wpisy ▾

Cos ciekawego2

👍 Fajne

🍌 Poziom fajności 0 🍌

👎 Słabe

2023-06-08 22:58:43

## 6.5 Logowanie jako administrator

Aby zalogować się jako administrator należy kliknąć przycisk znajdujący się w prawym górnym rogu -> Zaloguj.

HSAB

[Strona główna](#)

[Nowy wpis](#)

[Kontakt](#)

Zaloguj

Zaloguj się jako administrator

Login

Hasło

☐ Zapamiętaj mnie

Login

AW @ 2023

Wpisujemy Login oraz Hasło, opcjonalnie możemy zaznaczyć checkbox „Zapamiętaj mnie” aby zapamiętać dane logowania.

Podstawowe dane: Login to **admin**, a Hasło to **zxc123**

Po poprawnym wprowadzeniu danych zostajemy przekierowani na stronę główną wraz z komunikatem o poprawnym zalogowaniu się w przeciwnym przypadku zostanie wyświetlony komunikat o błędnych danych.

HSAB

[Strona główna](#)

[Nowy wpis](#)

[Kontakt](#)

Wyloguj (admin)

Zalogowano poprawnie.

Sortuj wpisy ▾

Cos ciekawego2

Fajne

Poziom fajności 0 🔔

Słabe

2023-06-08 22:58:43

Edytuj

Usuń

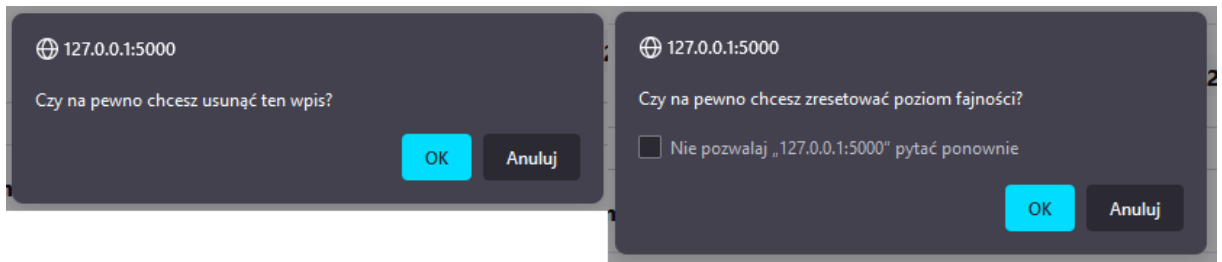
Resetuj PF

O poprawnym zalogowaniu informuje nas też przycisk „Wyloguj” wraz z nazwą zalogowanego użytkownika. Oczywiście kliknięcie w przycisk wyloguje nas z konta.

Poprawne zalogowanie udostępnia nam dodatkowe funkcję danego wpisu:

- Edytuj – edytuje wpis i aktualizuje datę dodania.
- Usuń – Usuwa wpis ( Kliknięcie uruchamia monit Czy na pewno chcemy usunąć wpis?)
- Resetuj PF – Ustawia licznik Poziomu fajności na 0. ( Kliknięcie uruchamia monit Czy na pewno chcemy to zrobić?)





## 6.6 Zakładka kontakt

Wyświetla wskazane dane oraz daje możliwość pobrania tego raportu.

## 7. Podsumowanie

- Z sukcesem udało mi się stworzyć kompletną aplikację internetową, która umożliwia dodawanie fascynujących wpisów.
- Za pomocą relatywnie niewielkiego wysiłku, ten projekt można rozwijać i przekształcić w narzędzie o powszechnym zastosowaniu.
  - Dodając nową funkcję np. upload obrazków wraz z opisem.
  - Bardziej rozbudowany panel administratora. (Klasy użytkowników)
- Zdobyłem umiejętność obsługi narzędzia Flask.
- Główny kod aplikacji znajduje się w pliku app.py.
- W katalogu flask\_session znajdują się tymczasowe pliki sesji utworzone przez pakiet flask\_session.
- W katalogu static znajdują się statyczne pliki takie jak obrazy, ikony, style strony, pliki do pobrania.
- W katalogu templates znajdują się pliki stron zakodowane w języku HTML wraz z dynamicznymi zmiennymi.
- Szablon stron wspomagany jest przez bibliotekę Bootstrap oraz jQuery popularną bibliotekę JavaScript
- Forms.py odpowiada za Walidację pól logowania.
- Database.db zawiera plik bazy danych sqlite3
- init\_db.py - inicjalizacja bazy danych
- schema.sql – schemat bazy danych

## 8. Bibliografia

Dokumentacja Flask - <https://flask.palletsprojects.com/en/2.3.x/>

Dokumentacja Bootstrap 5.2.x - <https://getbootstrap.com/docs/5.2/getting-started/introduction/>

Dokumentacja jQuery 3.7.0 - <https://api.jquery.com/category/version/3.7/>

Dokumentacja sqlite3 - <https://docs.python.org/3/library/sqlite3.html>

Dokumentacja Jinja 3.1.x - <https://jinja.palletsprojects.com/en/3.1.x/>

Dokumentacja werkzeug - <https://werkzeug.palletsprojects.com/en/2.3.x/>

Dokumentacja Flask-login - <https://flask-login.readthedocs.io/en/latest/>

Dokumentacja Flask-session-captcha - <https://pypi.org/project/flask-session-captcha/>

Dokumentacja Flask-WTF - <https://flask-wtf.readthedocs.io/en/1.0.x/>

Dokumentacja Flask\_Session - <https://flasksession.readthedocs.io/en/latest/>