

# PassionTown

Anna Olszak, Dawid Zapolski, Jakub Wieliczko

Czerwiec 2024

## Streszczenie

W tym dokumencie przedstawiamy opis naszego projektu

## Spis treści

<b>1</b>	<b>Wprowadzenie</b>	<b>2</b>
<b>2</b>	<b>Oferowane funkcjonalności</b>	<b>2</b>
2.1	Kompleksowa społecznościowa platforma . . . . .	2
2.2	Narzędzia analizy danych . . . . .	3
2.3	System organizacji konferencji . . . . .	3
2.4	System archiwizacji danych . . . . .	3
<b>3</b>	<b>Opis diagramu</b>	<b>3</b>
<b>4</b>	<b>Napotkane problemy i ich rozwiązania</b>	<b>6</b>
4.1	Spójność danych . . . . .	6
4.2	Optymalizacja wyszukiwań . . . . .	6
4.3	Statystyki . . . . .	6
4.4	Harmonogram . . . . .	6
4.5	Przechowywanie i weryfikacja haseł . . . . .	7
4.6	System forum . . . . .	7
<b>5</b>	<b>Instalacja</b>	<b>7</b>
<b>6</b>	<b>Opis działania aplikacji</b>	<b>8</b>
6.1	Feed . . . . .	8
6.2	Wydarzenia . . . . .	8
6.3	Dołącz do społeczności / Dołącz do wydarzenia . . . . .	8
6.4	Panel administratora . . . . .	8
6.4.1	Dodaj wolontariuszy do wydarzenia . . . . .	9
6.4.2	Zarejestruj prelekcję . . . . .	9
6.4.3	Dodaj prelegenta do prelekcji . . . . .	9
6.4.4	Statystyki wydarzeń i społeczności . . . . .	9

## 1 Wprowadzenie

Głównym celem projektu jest stworzenie uniwersalnej bazy platformowej, która mogłaby służyć jako centrum komunikacyjne środowisk pasjonatów lub centrum zrzeszające absolwentów danych projektów naukowych, fundacji czy projektów. Obecnie możemy wyróżnić protoplastów tego projektu tj. Facebook czy fora tematyczne. Nie są one idealnie dostosowane do dzielenia się swoimi pasjami czy rozwiązaniami, przykładowo grupy na Facebooku jak i samoistne posty nie są przystosowane do wstawiania kodów czy wzorów matematycznych, a przy dużej ilości postów, ciężko optymalnie wyszukać interesujących nas podkategorii lub danego postu. W rozwiązaniu takiego problemu przydatne będzie archiwum, do którego (w odpowiednich kategoriach) dodawane mogą być posty. Dobrze zorganizowane i przystępne środowisko pozwala na zwiększenie kreatywności, możliwości dzielenia się swoimi pracami, jak i uzyskania pomocy i nowej wiedzy.

Jak już zostało wspomniane, istnieją protoplaści naszego projektu, ale każdy z nich boryka się z problemami lub ich archaiczne rozwiązania nie zachęcają do udziału w dyskusjach i innych interakcjach. Nowoczesny, intuicyjny wygląd, możliwość przeglądania strony na każdym urządzeniu z dostępem do internetu, to jeden z wielu atutów naszego rozwiązania. Równie ważnym aspektem będzie dostosowanie zgodnie z dyrektywą WCAG 2.0, dla osób z różnych społeczności takich jak personalizacji zaimków, zmiany imienia i nazwiska w każdym momencie, jak i możliwość wpisywania długich nazwisk i znaków specjalnych dla osób odmiennej narodowości.

Dodatkowy moduł organizacji wydarzeń, pomoże zorganizować małe spotkania, konferencje naukowe czy networkingi, a przy tym będzie w stanie działać w zakresie wybranych grup. Harmonogram prelekcji, system zarządzania wolontariuszami, generowanie identyfikatorów konferencyjnych, oraz wiele innych opcji usprawni organizację takich wydarzeń. Sprawia one również, że wydarzenia będą bardziej ustandaryzowane oraz mniej czasochłonne w aspektach organizacji. Poprzez dopasowanie tematyczne pomoże również promować dane wydarzenie dla innych użytkowników.

## 2 Oferowane funkcjonalności

### 2.1 Kompleksowa społecznościowa platforma

- (a) Interaktywna platforma internetowa, która nie tylko umożliwi użytkownikom publikowanie treści, komentowanie i udostępnianie, ale także będzie oferować możliwość tworzenia grup dyskusyjnych

- (b) Harmonogram wydarzeń związanych z grupami zainteresowań, wraz ze wszystkimi informacjami o prelegentach i prelekcjach na nich się odbywających
- (c) Przystosowanie strony dla osób z niepełnosprawnością zgodnie ze standardem WCAG 2.0.
- (d) Możliwość wyboru zaimków oraz informacji o nich w profilu użytkownika, a także przy pseudonimie lub imieniu i nazwisku.
- (e) Uznawanie znaków specjalnych różnych języków w imionach i nazwiskach.

## 2.2 Narzędzia analizy danych

- (a) Wdrożenie statystyk ilości osób w grupach i ich aktywności

## 2.3 System organizacji konferencji

- (a) Opracowanie systemu zarządzania konferencjami, który będzie integrowany z kalendarzem do planowania salami, wykładami i warsztatami.
- (b) Środowiska do planowania i zarządzania wolontariuszami.
- (c) System generujący imienne identyfikatory
- (d) Newsletter konferencyjny

## 2.4 System archiwizacji danych

# 3 Opis diagramu

1. Naszą podstawową jednostką organizacyjną jest grupa zainteresowań - inaczej społeczność - została u nas zainicjalizowana jako osobna tablica. Przechowuje ona informacje o aktualnie utworzonych społecznościach w serwisie. Tabela zawiera pola: id\_spolecznosci (klucz główny) oraz nazwa (pole obowiązkowe).
2. W tabeli członkowie przechowywane są informacje o wszystkich zarejestrowanych członkach. Do rejestracji podawane są obowiązkowo: nazwa użytkownika (która jest unikatowa), e-mail (który jest unikatowy), imię i nazwisko (później można wykorzystać je do organizacji wydarzeń oraz dodatkowo ustalana jest data dołączenia. Ponadto użytkownik musi wpisać swoje imię i nazwisko i może określić swoje zaimki, czy zapisać się na newsletter. Klucz główny: id\_uzytkownika

3. Tabela `czlonkowie.spolecznosci` służy do rejestracji członków do danych społeczności. Przechowuje ona obowiązkowe pola: `id_czlonka` (obowiązkowy klucz obcy z tabeli `czlonkowie`), `id_spolecznosci` (obowiązkowy klucz obcy z tabeli `spolecznosci`), `id_rol` (obowiązkowy klucz obcy z tabeli `role`), `haslo` (które jest zaszyfrowane). Klucz główny składa się z właśnie tych trzech pól. Każdy członek może mieć różne role w różnych społecznościach, w tym w danej społeczności może mieć on wiele roli.
4. Tabela `role` przechowuje informacje o rolach, które może pełnić dany użytkownik danej społeczności. Przechowywane pola: `id_rol` (klucz główny), `nazwa` (obowiązkowa).
5. Tabela `zaimki` przechowuje spis zaimków spośród których może wybrać użytkownik podczas rejestracji konta. Przechowywane pola: `id_zaimka` (klucz główny), `nazwa` (obowiązkowa).
6. Tabela `posty` przechowuje informacje o postach, które są publikowane przez użytkowników w danej społeczności. Przechowywane są następujące pola: `id_posta` (klucz główny), `id_czlonka` (obowiązkowy klucz obcy z tabeli `czlonkowie`, informacja o autorze posta), `id_posta_nad` (klucz obcy z tabeli `posty`, jeżeli niepusty - informacja o tym że post jest odpowiedzią na inny post), `data_dodania`, `tytul` i `tresc` (obowiązkowe)
7. Tabele `posty`, `czlonkowie` mają utworzone odpowiadające im tabele `posty_archiwalne`, `czlonkowie_archiwalne` w celu archiwizacji. Tabele te pozbawione są kluczy obcych, ze względu na konflikty gdy np przechowywany archiwalny post będzie napisany przez użytkownika którego id znajduje się jedynie w archiwum. Klucze główne: takie jak w tabelach oryginalnych
8. Tabela `wydarzenia` przechowuje informacje o organizowanych wydarzeniach. Przechowywane pola: `id_wydarzenia` (klucz główny), `nazwa` (nazwa wydarzenia, obowiązkowa), `typ_wydarzenia` (obowiązkowy, klucz obcy na tabelę `rodzaje_wydarzen`).
9. Tabela `rodzaje_wydarzen` przechowuje informacje o typach organizowanych wydarzeń. Przechowywane pola: `typ_wydarzenia` (klucz główny), `nazwa` (obowiązkowa).
10. Tabela `edycje` przechowuje informacje o kolejnych odbywających się edycjach zapowiedzianych wydarzeń. Przechowywane pola: `id_edycji` (klucz główny), `nr_edycji` (pole obowiązkowe - numer odbywającej się edycji,  $\text{nr\_edycji} > 0$ ) `id_wydarzenia` (obowiązkowy klucz obcy na tabelę `wydarzenia`), `data_rozpoczecia` (obowiązkowa), `data_zakonczenia` (obowiązkowa), `miejsce` (obowiązkowe, klucz obcy na tabelę `adresy`), `podtytul`. Ograniczenie -  $\text{data\_rozpoczecia} < \text{data\_zakonczenia}$
11. Tabela `czlonkowie_edycje` przechowuje informacje o członkach, którzy rejestrują się na dane edycje wydarzeń. Tabela przechowuje pola: `id_czlonka`

- (obowiązkowe, klucz obcy na tabelę członkowie), id\_edycji (obowiązkowe, klucz obcy na tabelę edycje). Ograniczenia: unique(id\_czlonka, id\_edycji).
12. Tabela wolontariusze przechowuje informacje o członkach, którzy rejestrują się na dane edycje wydarzeń żeby zostać w nich wolontariuszami. Tabela przechowuje pola: id\_czlonka (obowiązkowe, klucz obcy na tabelę członkowie), id\_edycji (obowiązkowe, klucz obcy na tabelę edycje). Ograniczenia: unique(id\_czlonka, id\_edycji).
  13. Tabela organizatorzy przechowuje informacje o organizatorach danych edycji wydarzeń. Tabela przechowuje pola: id\_czlonka (obowiązkowe, klucz obcy na tabelę członkowie), id\_edycji (obowiązkowe, klucz obcy na tabelę edycje). Ograniczenia: unique(id\_czlonka, id\_edycji).
  14. Tabela sale przechowuje informacje o salach, które są udostępnione dla danych edycji danych wydarzeń. Przechowywane są następujące pola: id\_sali (klucz główny), adres (obowiązkowy, klucz obcy na tabelę adresy), nazwa\_sali (obowiązkowa), pojemnosc\_sali (obowiązkowa, pojemnosc\_sali > 0).
  15. Tabela edycje\_sale przechowuje informacje o tym, które sale są udostępnione dla których edycji naszych wydarzeń. Przechowywane pola: id\_edycji (obowiązkowe, klucz obcy na tabelę edycje), id\_sali (obowiązkowe, klucz obcy na tabelę sale). Ograniczenia: unique(id\_sali, id\_edycji)
  16. Tabela wydarzenia\_spolecznosci przechowuje informacje o tym, dla jakich społeczności organizowane są dane wydarzenia. Przechowywane są następujące pola: id\_wydarzenia (obowiązkowe, klucz obcy na tabelę wydarzenia), id\_spolecznosci (obowiązkowe, klucz obcy na tabelę spolecznosci). Ograniczenia: unique(id\_wydarzenia, id\_spolecznosci)
  17. Tabela prelekcje przechowuje informacje o prelekcjach organizowanych podczas danej edycji. Przechowywane pola: id\_prelekcji (klucz główny), id\_edycji (obowiązkowy klucz obcy na tabelę edycje), id\_sali (klucz obcy na tabelę sale), data\_prelekcji (pole obowiązkowe), dlugosc\_prelekcji (obowiązkowy klucz obcy na tabelę dlugosci), temat (pole obowiązkowe), opis.
  18. Tabela dlugosci przechowuje pulę możliwych długości danych prelekcji. Przechowywane pola: dlugosc\_prelekcji (klucz główny), dlugosc (pole obowiązkowe, unikatowe, dlugosc > 0)
  19. Tabela prelegenci przechowuje informacje o zarejestrowanych prelegentach, którzy będą prowadzić prelekcje. Przechowywane pola: id\_prelegenta (klucz główny), id\_czlonka (unikatowe, klucz obcy na tabelę członkowie), imie (pole obowiązkowe), nazwisko (pole obowiązkowe).
  20. Tabela prelekcje\_prelegenci przechowuje informacje o doborze odpowiednich prelegentów do prowadzenia danych prelekcji. Tabela przechowuje dwa pola: id\_prelegenta (obowiązkowe, klucz obcy na tabelę prelegenci),

id\_prelekcji (obowiązkowe, klucz obcy na tabelę prelekcje). Klucz główny (id\_prelegenta, id\_prelekcji)

21. W tabeli wolontariusze\_prelekcje przechowujemy informacje o tym, jaki wolontariusz pilnuje daną prelekcję. Klucz główny (id\_czlonka, id\_prelekcji)
22. W tabeli adresy przechowujemy nasz zbiór adresów, do którego odwołujemy się z innych tabel. Klucz główny: id\_adresu, ograniczenia: adres obowiązkowy

## 4 Napotkane problemy i ich rozwiązania

### 4.1 Spójność danych

Przede wszystkim, gdy rejestrujemy daną prelekcję to musi być tak że zarówno sala jest wolna jak i prelegent jest wolny. W tym celu przy wywołaniu funkcji `register_talk` sprawdzane są wszystkie te rzeczy. Podobnie ma się rzecz z dodawaniem drugiego prowadzącego do naszej prelekcji - funkcja `add_to_talk` robi to wszystko za nas!

Utworzone zostało wiele wyzwalaczy, mających na celu gwarantować spójność danych: Jeżeli organizator postanowi wyrzucić członka z wydarzenia, automatycznie przestaje również on być wolontariuszem i odłączany jest od prelekcji. Jeżeli aktualizujemy dane użytkownika, to dla prelegenta który ma podłączone konto również dane się zmieniają - i działa to również w drugą stronę. Gdy usuwamy prelekcję lub prelegenta, aktualizowana jest tabela `prelekcje_prelegenci` żeby zapobiec ewentualnym konfliktom. W przypadku archiwizacji gdy zdecydujemy się zarchiwizować danego użytkownika, to archiwizujemy również jego komentarze i przegrupowujemy do nich odpowiedzi. Przykładów jest więcej, nad wszystkim czuwają nasze wyzwalacze.

### 4.2 Optymalizacja wyszukiwań

Wraz z rozrostem naszej platformy niektóre tabele zaczęły puchnąć, z tego względu wprowadziliśmy indeksy na tabele `czlonkowie`, `posty`, `posty_archiwum`, `czlonkowie_archiwum` i `prelekcje`.

### 4.3 Statystyki

W celu implementacji statystyk wprowadziliśmy wiele funkcji umożliwiających zliczanie konkretnych danych, podobnie jak przy generowaniu identyfikatorów czy wyświetlaniu harmonogramu, o tym jednak następny paragraf. Powstały widoki umożliwiające pobranie statystyk zarówno o społecznościach, jak i wydarzeniach.

### 4.4 Harmonogram

Potrzebowaliśmy do każdej edycji zaimplementować funkcję wyświetlającą nam informacje o wszystkich prelekcjach z podziałem na dni, które się podczas niej

odbywają. Dodatkowo funkcja przechwytytuje informacje o prowadzących dany talk, czy o wolontariuszach go pilnujących. Funkcje odpowiedzialne za spójność przy zapisywaniu i odczytywanie informacji o prelekcjach to największe z naszych zaimplementowanych funkcji.

## 4.5 Przechowywanie i weryfikacja haseł

W celu bezpiecznego przechowywania haseł zastosowaliśmy "solenie", gdzie przy haszowaniu hasła jest dodawana do niego losowa sól. W celu poprawnej weryfikacji przy logowaniu sól jest zapisywana razem z hasłem. Z tego powodu, w celu weryfikacji poprawności hasła, konieczne okazało się uzyskiwanie zaszyfrowanej postaci hasła z bazy oraz jego weryfikacja na serwerze, a nie porównywanie w funkcji PostgreSQL-owej, jak początkowo planowaliśmy.

## 4.6 System forum

Głównym problemem w tej sekcji był sposób na przechowywanie struktury postów i ich odpowiedzi. Zdecydowaliśmy się na pole `id_posta_nad`, które trzyma informacje o tym czy odpowiada na jakiś post i jeżeli tak, to na jaki. Przyjeliśmy, że post może być odpowiedzią na dokładnie jeden inny post - jak na facebooku. Dzięki temu w naszej aplikacji możemy wyświetlić dany post i jego bezpośrednie odpowiedzi, a potem zagłębiać się w odpowiedzi do tych odpowiedzi itd.

# 5 Instalacja

Do uruchomienia aplikacji jest wymagany Python, prawdopodobnie w wersji  $\geq 3.8$  (używaliśmy 3.12, ale powinien działać na starszych).

W celu utworzenia środowiska wirtualnego oraz instalacji wymaganych bibliotek proszę, będąc w folderze, gdzie rozpakowano projekt, wykonać komendy:

```
python -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
```

Następnie w celu uruchomienia, będąc wciąż w tym samym folderze:

```
flask --app "basic" run
```

Uruchomi to serwer, do którego link pokaże się w konsoli. Po otwarciu linku w przeglądarce powinno nastąpić przekierowanie do strony głównej aplikacji. Aplikacja zakłada, że ma dostęp do bazy danych przez lokalny socket. Domyślnie aplikacja zakłada, że łączy się z bazą o tej samej nazwie, co nazwa użytkownika systemowego (w pliku `basic.py` ustawione `database=os.getlogin()`, `user=os.getlogin()` ).

## 6 Opis działania aplikacji

Nasza aplikacja działa lokalnie, także po uruchomieniu Flaska należy załadować ją wchodząc na `http://127.0.0.1:5000/home`. Wyświetla się strona główna, w której mamy możliwość rejestracji konta lub zalogowania się na istniejące konto. Po zalogowaniu wyświetla nam się panel użytkownika. Możemy tu dołączyć do społeczności lub wydarzenia a gdy już dołączymy to przeglądać społeczności w których jesteśmy.

Omówię pokrótce dostępne zakładki (dla zwykłego użytkownika):

### 6.1 Feed

Po wejściu wyświetla nam się panel społeczności, w których jesteśmy oraz 25 ostatnich postów wrzuconych na te wszystkie społeczności. Możemy z tej pozycji wybrać interesujący nas post i zobaczyć społeczność, która nas zainteresuje. Po wybraniu naszej społeczności wyświetlane są informacje o wszystkich w niej opublikowanych wątkach. Mamy możliwość dodania własnego wątku albo zobaczenia innego wątku i napisania do niego odpowiedzi.

### 6.2 Wydarzenia

Po wejściu wyświetla się nam lista wydarzeń, w których braliśmy lub bierzemy udział. Po wybraniu wydarzenia wyświetlają nam się informacje o czasie trwania wydarzenia oraz o prelekcjach, które podczas niego się odbywają. W momencie, gdy jesteśmy wolontariuszem w danej edycji, wyświetlają nam się również informacje o tym kto pilnuje danej prelekcji oraz możliwość rejestracji i anulowania w pilnowaniu wybranej prelekcji odbywającej się podczas tej edycji. Funkcjonalność ta ma na celu usprawnić organizację wolontariatu.

### 6.3 Dołącz do społeczności / Dołącz do wydarzenia

Opcje pozwalające na dołączanie do społeczności i wydarzeń. Przy dołączaniu do społeczności użytkownik dostaje domyślną rolę 'członek'. Uwaga: Możemy dołączyć do wydarzeń organizowanych jedynie przez społeczności, w których się obecnie znajdujemy.

### 6.4 Panel administratora

Żeby móc z niego korzystać, trzeba być administratorem na co najmniej jednej społeczności. Zakładamy w celach bezpieczeństwa, że role administratorów będą przydzielane poza naszą aplikacją, zatem żeby dostać się do panelu administratora należy w PSQl-u wykonać `insert into czlonkowie_spolesznosci values (id_uzytkownika, id_spolesznosci, 1)`; dla naszego użytkownika oraz pewnej wybranej społeczności. Ewentualnie na potrzeby naszego projektu zrobiliśmy dla



Pani konto o loginie ‘admin1’ i hasło ‘pass’ które ma uprawnienia administratora.

Panel pozwala nam na kilka dodatkowych operacji:

#### **6.4.1 Dodaj wolontariuszy do wydarzenia**

Strona umożliwiająca rejestrację wolontariuszy do konkretnych edycji naszych wydarzeń. Dodanie zakończy się sukcesem jedynie w przypadku gdy osoba którą dodajemy jest aktualnie członkiem naszego wydarzenia.

#### **6.4.2 Zarejestruj prelekcję**

Strona umożliwiająca rejestrację prelekcji dla naszej wybranej edycji. Założenia na prelekcje – musi odbywać się podczas trwania edycji, podczas całej długości rejestrowanej prelekcji sala i prowadzący muszą być do dyspozycji – nie może być innej prelekcji odbywającej się w tym samym czasie w tej samej sali, albo prowadzonej przez tego samego prowadzącego

#### **6.4.3 Dodaj prelegenta do prelekcji**

Z racji, że prelekcje mogą mieć więcej niż jednego prowadzącego, istnieje możliwość dodania prelegenta do aktualnie prowadzonej prelekcji. Dodawany prelegent nie może w tym czasie prowadzić innej prelekcji.

#### **6.4.4 Statystyki wydarzeń i społeczności**

Strona wyświetlająca rozmaite statystyki na temat naszych wydarzeń i społeczności.

Dodatkowo mamy opcję wygenerowania identyfikatorów dla naszych członków, wolontariuszy, organizatorów i prelegentów danej edycji.

## **7 Uwagi**

Poniżej wyróżniamy nasze odpowiedzi do kolejno ponumerowanych uwag

1. Ze względu na obejście „patologicznych” długości prelekcji, w dodatku ułatwia to wybór długości poprzez listę, zamiast jej ręcznego wpisywania
2. Poprawiliśmy to – usunęliśmy tabelę z hasłami i zamiast tego użytkownik przechowuje swoje zaszyfrowane hasło w tabeli członkowie
3. Poprawiliśmy to
- 4.
5. Członek traci rolę gdy jest usunięty ze społeczności, zaś może pełnić jedną rolę będąc w danej społeczności

6. Pole `id_edycji` to identyfikator w tabeli `edycje`, zaś `nr_edycji` mówi o tym, który raz dane wydarzenie jest organizowane
7. Utworzyliśmy tabelę `adresy` z listą wszystkich adresów i we wszystkich odnośnikach podajemy po prostu `id_adresu`
- 8.
9. Członek sam nie może opuścić społeczności. Jednak w przypadku usunięcia, członek trafia do tabeli `czlonkowie_archiwum`, a wszystkie jego posty zostają oznaczone jako usunięte
10. W przypadku usunięcia prelegenta/prelekcji modyfikowana jest tabela `prelekcje_prelegenci`, żeby zachować spójność danych.