**Obraz zawierający tekst

Opis wygenerowany automatycznie**

PRACA DYPLOMOWA

Adam Marciniak

Social network

Promotor: dr inż. Zbigniew Szpunar

Kalisz 2023

**Temat pracy dyplomowej  
numer INF/99/2023**

| Student | **Adam Marciniak** |
| --- | --- |
| Numer albumu | **30692** |
| Studia  (stacjonarne, niestacjonarne) | **Stacjonarne** |
| Tytuł pracy (PL) | Serwis społecznościowy |
| Tytuł pracy (EN) | Social networking service |
| Cel i zakres pracy (PL) | Celem projektu jest stworzenie serwisu społecznościowego. Pracę będzie obejmować analiza podobnych tego typu stron, potrzeb użytkowników, projektowanie aplikacji, tworzenie prototypów, implementacja projektu oraz testowe wprowadzenie na serwerze lokalnym |
| Promotor | **dr inż. Zbigniew Szpunar** |
| Termin oddania pracy | **29 lutego 2024 r**. |

Podpis promotora: …......................................

Zobowiązuję się samodzielnie wykonać pracę dyplomową na powyższy temat i nie zlecać innym osobom jej opracowania – w całości, ani w części. Wszystkie elementy – między innymi rysunki, tabele, programy komputerowe, urządzenia – które zostaną wykorzystane w pracy, a nie będą mojego autorstwa, zostaną w odpowiedni sposób oznaczone z podaniem źródła ich pochodzenia. Treść zawarta w elektronicznej wersji pracy będzie taka sama, jak w jej wersji papierowej. Przyjmuję do wiadomości, iż w przypadku niespełnienia powyższych warunków, Uczelnia może nie dopuścić mojej pracy dyplomowej do obrony lub wydać negatywną decyzję w sprawie nadania mi tytułu zawodowego.

|  |  |
| --- | --- |
| Kalisz, 30.09.2023 r. | Podpis studenta: ……….................................... |

Obraz zawierający tekst

Opis wygenerowany automatycznie

**Oświadczenie autora pracy dyplomowej  
numer INF/99/2023**

Ja, niżej podpisany/a ……………………………………………….., Student/ka Wydziału Politechnicznego Akademii Kaliskiej im. Prezydenta Stanisława Wojciechowskiego oświadczam, że przedkładana praca dyplomowa pod tytułem

Planszowo-karciana wieloosobowa gra sieciowa

1. została przygotowana przeze mnie samodzielnie\*, wspólnie z ….....................................................;
2. nie narusza praw autorskich w rozumieniu ustawy z 04 lutego 1994 roku o prawie autorskim i prawach pokrewnych (tekst jedn. Dz.U. z 2019 poz. 1231) oraz dóbr osobistych chronionych prawem;
3. nie zawiera danych i informacji, które uzyskałem/am w sposób niedozwolony;
4. nie była podstawą nadania dyplomu uczelni wyższej lub tytułu zawodowego ani mnie, ani innej osobie.

Ponadto oświadczam, że treść pracy przedstawionej przeze mnie do obrony, zawarta na przekazanym nośniku elektronicznym, jest identyczna z jej wersją drukowaną.

Miejsce, data i podpis dyplomanta: …..................................................................... .

\*) Uwzględniając merytoryczny wkład promotora (w ramach prowadzonego seminarium dyplomowego).

Wariant „wspólnie z ...” dotyczy prac dyplomowych realizowanych przez więcej niż jednego studenta.

Spis treści

[1. Cel i zakres pracy 5](#_Toc137725751)

[2. Wizja 6](#_Toc137725752)

[3. Model domenowy 9](#_Toc137725755)

[4. Model wymagań 11](#_Toc137725756)

[5. Model przypadków użycia 12](#_Toc137725757)

[6. Model architektury 14](#_Toc137725758)

[7. Projekt bazy danych 15](#_Toc137725759)

[8. Projekt interfejsu 16](#_Toc137725760)

[9. Implementacja 17](#_Toc137725761)

# **Cel i zakres pracy**

Cel projektu polega na stworzeniu serwisu społecznościowego, który ma za zadanie zrzeszać różne grupy ludzi, aby umożliwić im wymianę informacji, rozwiązań, problemów, zainteresowań oraz innych społecznych zagadnień. Strona ma łączyć w sobie najlepsze zalety innych takich serwisów oraz niwelować wady i problemy zauważone przez autora na podobnych stronach. Aplikacja głównie ma umożliwiać prowadzenie dyskusji na różne tematy opisane w postach użytkowników.

Serwis ma umożliwiać użytkownikom:

* Publikację własnych postów
* Komentowanie postów innych użytkowników
* Prowadzenie dyskusji w sekcji komentarzy
* Ocenianie postów i komentarzy
* Przeglądanie postów z różnych dziedzin
* Zakładanie konta i edytowanie swojego profilu.
* Zapisywanie ulubionych postów

W serwisie zostaną wprowadzone konta administratorów i moderatorów, które będą kontami użytkowników z dodatkowymi funkcjami i dostępem do wyodrębnionych podstron aby móc dostosowywać stronę.

Administratorskie funkcje mają pozwalać na:

* usuwanie postów niezgodnych z daną kategorią/naruszające normy kulturalne i etyczne
* tworzenie moderatorów i nadawać im uprawnienia
* wysyłanie powiadomień do użytkownika
* ograniczanie uprawnień użytkownika

Moderatorzy

* ukrywanie postów niezgodnych z daną kategorią/naruszające normy kulturalne i etyczne
* wysyłanie wiadomości o ograniczenie uprawnień dla danego użytkownika z określeniem jego przyczyny

Zakres pracy będzie obejmować etapy:

1. analiza wiedzy dziedzinowej
2. stworzenie jednolitej wizji
3. określenie głównych elementów
4. wybranie technologii
5. stworzenie prototypów interfejsów graficznych
6. implementacja na serwerze lokalnym
7. testowanie
8. implementacja na serwerze produkcyjnym w celu lepszego przetestowania
9. podsumowanie
10. prezentacja

# 2. **Wizja**

Serwis ma być alternatywą dla innych tego typu stron jednocześnie korzystając z najlepszych konceptów oraz rozwiązań jakie można na nich zaimplementować. Będzie to aplikacja webowa o budowie SPA z trzema podstronami.

Struktura serwisu zostanie podzielona na trzy główne podstrony, z których każda będzie pełniła określone funkcje:

Główna strona:

* To główna strona serwisu, na której użytkownicy będą przeglądać posty i komentarze od innych użytkowników.

Strona konta:

* Na tej stronie użytkownicy będą mieli możliwość:
* Edycji swojego profilu i danych konta.
* Przeglądania swoich własnych postów i komentarzy.
* Przeglądania postów, które dodali do ulubionych.

(Opcjonalnie) - Korzystania z narzędzia do tworzenia memów.

(Opcjonalnie) - Dostępu do generatora baniek informacyjnych, który analizuje zachowanie użytkownika, aby dostarczyć spersonalizowane rekomendacje.

(Opcjonalnie) - Generator memów:

(Opcjonalnie) – Generator baniek Informacyjnych:

To dodatkowa funkcja analizująca zachowanie użytkownika, takie jak klikane treści i komentarze, aby zrozumieć ich zainteresowania.

Na podstawie tych danych zostaną utworzone spersonalizowane profile (bańki) użytkowników.

Na podstawie tych profili serwis może wyświetlać posty z określonych kategorii jako pierwsze, dostosowując zawartość do preferencji użytkownika.

# 3. Model domenowy

Głównym elementem tej struktury będzie post, którego właścicielem będzie użytkownik, który go dodaje. Każdy post ma swoje oznaczenie w formie kategorii i tagów, które informują innych użytkowników o rodzaju posta. Każdy post może posiadać polubienia tzw. like, którymi społeczność będzie głosować czy dany post lub komentarz jest ważny/zawiera poprawne informacje/rozwiązuję problem lub wyraża opinie, z którą zgadza się większość społeczności. Użytkownik może zapisać również interesujące go posty do listy ulubionych. Każdy komentarz oraz odpowiedź na komentarz użytkownika generuje powiadomienie, które jest wysyłane do określonego użytkownika.

Konto zwyczajnego może zostać kontem moderatora lub administratora po nadaniu mu uprawnień przez administratora.

## 3.1 Słownik pojęciowy

* mem - gatunek humorystycznej wypowiedzi internetowej, komunikat obrazkowy zbudowany w oparciu o schemat konstrukcyjny wykorzystujący skonwencjonalizowane elementy związane z kulturą i historią
* post - [wiadomość](https://pl.wikipedia.org/wiki/Wiadomo%C5%9B%C4%87), [informacja](https://pl.wikipedia.org/wiki/Informacja) wysłana na [grupę](https://pl.wikipedia.org/wiki/Grupa_dyskusyjna) lub [forum dyskusyjne](https://pl.wikipedia.org/wiki/Forum_dyskusyjne), napisana w określonym temacie [wypowiedź](https://pl.wikipedia.org/wiki/Wypowied%C5%BA). Może to być pierwsza wiadomość w nowym wątku, wyznaczająca kierunek dyskusji, bądź też będąca odpowiedzią na inny. Obok posta widoczna jest zawsze nazwa użytkownika internetowego, który ją napisał, zazwyczaj także informacje o nim oraz data i godzina napisania. Post najczęściej składa się tytułu, tekstu i/lub obrazka. Posty powinny mieć zawsze wprowadzony tytuł, tekst lub obrazek oraz zaznaczoną kategorię
* kategoria – informacja, który oznacza do jakiej dziedziny należy post lub jego treść. Służy również do filtrowania postów
* tag – podkategoria wprowadzone aby określić post w który zawiera elementy innej kategorii
* komentarz – wypowiedź użytkownika strony na temat posta lub treści innego komentarza
* like – liczba oznaczająca ile osób się zgadza z danym postem lub uważa go za pomocny lub użyteczny
* rest api – element architektury systemu określający w jaki sposób będą komunikowały się dane elementy systemu
* backend – część systemu zajmująca się bazą danych, przetwarzaniem danych i odpowiedzią na żądania systemu
* frontend – część systemu zajmująca się wyświetlaniem interfejsu użytkownikowi, reagowanie na akcje użytkownika i wysyłanie odpowiednich żądań i odbieranie ich aby wykonywać określone akcji
* SPA(single page aplication) –czyli aplikacja która składa się z jednej strony, ale dzięki zastosowaniu odpowiednich technologii pozwala na wiele operacji i funkcja bez jej przeładowywania
* Ulubione – spis zapisanych przez użytkownika postów, każdy użytkownik posiada własny spis
* Powiadomienie – wiadomość dla użytkownika informująca usera o nowym komentarzu do posta, odpowiedzi na komentarz lub wiadomość od administratora albo systemowa
* Administrator – osoba lub osoby kontrolujące całą stronę poprzez tworzenie moderatorów z odpowiednimi uprawnieniami, filtrująca zawartość podstron i postów,
* Moderator – osoba kontrolująca jakiej treści pojawiają się w postach i komentarza. Usuwająca lub ukrywająca niepożądane treści za pomocą odpowiednich narzędzi

**3.2. Reguły biznesowe**

Rejestracja i Konta Użytkowników:

* Każdy użytkownik musi zarejestrować się, aby korzystać z serwisu.
* Użytkownicy mogą zakładać i zarządzać swoimi kontami.
* Hasła użytkowników powinny być zabezpieczone i poufne.

Posty i Komentarze:

* Użytkownicy mogą dodawać posty, określając kategorię i dodając tagi.
* Posty mogą być komentowane przez innych użytkowników.
* Użytkownicy mogą udzielać "lajków" na posty i komentarze.
* Administracja serwisu monitoruje posty pod kątem niezgodności z kategorią lub naruszeń norm kulturalnych i etycznych.
* Moderatorzy mają możliwość ukrywania postów niezgodnych z kategorią lub normami.

Profil Użytkownika:

* Użytkownicy mogą edytować swoje profile, w tym zmieniać dane osobowe, zdjęcia profilowe itp.
* Strona profilu zawiera informacje o aktywności użytkownika, takie jak posty, komentarze i ulubione posty.
* Użytkownicy mogą przeglądać swoje ulubione posty na swoim profilu.

Ocena i Rekomendacje:

* Ocenianie postów i komentarzy przez użytkowników ma na celu dostarczanie feedbacku na jakość treści.
* Generowanie bańek informacyjnych w oparciu o zachowanie użytkownika, aby dostarczać spersonalizowane rekomendacje, jest opcjonalne.

Administratorzy i Moderatorzy:

* Administracja serwisu ma możliwość usuwania postów niezgodnych z kategorią lub naruszających normy.
* Administratorzy mogą tworzyć i nadawać uprawnienia moderatorom.
* Moderatorzy mogą ukrywać posty i informować o ograniczeniach uprawnień użytkownikom.

Zasady Etyczne:

* Serwis ma zdefiniowane zasady etyczne i normy społeczne, które muszą być przestrzegane przez użytkowników.
* Użytkownicy nie mogą promować treści obraźliwych, nielegalnych lub szkodliwych.

Narzędzia Opcjonalne:

* Opcjonalne narzędzia, takie jak generator memów i generator bańek informacyjnych, mogą być dostępne dla użytkowników, jeśli zostaną wdrożone.

Ochrona Danych i Prywatności:

* Dane użytkowników muszą być odpowiednio chronione i przetwarzane zgodnie z przepisami dotyczącymi ochrony danych.

Wsparcie Techniczne i Pomoc Dla Użytkowników:

* Serwis powinien dostarczać wsparcie techniczne i pomoc dla użytkowników w razie problemów lub pytań.

Rozwój i Aktualizacje:

* Serwis musi być regularnie aktualizowany i rozwijany, aby dostarczać nowe funkcje i poprawiać jakość usług.

Reklama i Monetyzacja:

* Jeśli planowana jest reklama lub monetyzacja, to muszą być zdefiniowane zasady dotyczące reklamodawców i sposobu wykorzystania reklam na serwisie.

Zarządzanie Społecznością:

* Administracja serwisu ma obowiązek dbać o kulturę i atmosferę w społeczności, usuwając treści obraźliwe lub niewłaściwe.

Zgłaszanie Naruszeń:

* Użytkownicy powinni mieć możliwość zgłaszania naruszeń zasad i nieodpowiednich treści.

**3.3 Diagram Klas**

Obraz zawierający tekst, diagram, linia, zrzut ekranu

Opis wygenerowany automatycznie

* *Rys 3.1 Domenowy diagram klas (opracowanie własne)*

**użytkownik** - może być rozszerzony do moderatora lub administratora przez innego administratora w razie potrzeb, odbywa się to poprzez nadanie odpowiednich ról i uprawnień

**użytkownik – post** – użytkownik może posiadać wiele postów natomiast post ma tylko jednego użytkownika (czyli jego autora)

**użytkownik – like** – użytkownik może posiadać wiele like, ale like ma tylko jednego użytkownika

**użytkownik – powiadomienie** - użytkownik może posiadać wiele powiadomień, ale powiadomienie ma tylko jednego użytkownika

**użytkownik – ulubione –** użytkownik może posiadać wiele ulubionych postów, ale rekord zapisujący to może mieć tylko jednego użytkownika

**post – komentarz –** komentarz jest przypisany do jednego posta, ale jeden post może posiadać wiele komentarzy

**post – tag –** post może posiadać wiele tagów, a tag może być używany przez wiele postów

**post – kategoria –** post posiada tylko jedna kategorię, ale jedna kategoria jest używana przez wiele postów

**post – like –** like jest przypisany do jednego posta, ale post może mieć wiele like

**post – ulubione –** jeden rekord zapisujący ulubiony post może mieć zapis tylko do jednego posta na raz, ale jeden post może się pojawić w wielu rekordach, czyli kilku użytkowników może zapisać sobie jeden post jako ulubiony i każdy będzie go widział w swojej liście ulubionych

**komentarz – like -** like jest przypisany do jednego komentarza, ale komentarz może mieć wiele like

# 4. Model wymagań

**Wymagania funkcjonaln**e**:**

* Użytkownicy mogą tworzyć konta i logować się za pomocą adresu e-mail, loginu lub kont społecznościowych.
* Użytkownicy mogą tworzyć i zarządzać swoimi profilami, dodając informacje o sobie, zdjęcia profilowe itp.
* Użytkownicy mogą tworzyć i udostępniać różne rodzaje treści, takie jak posty, zdjęcia, filmy itp.
* Użytkownicy mogą edytować, usuwać i zarządzać swoimi opublikowanymi treściami.
* Użytkownicy mogą komentować, polubić i udostępniać treści innych użytkowników.
* Użytkownicy otrzymują powiadomienia o aktywności związaną z ich kontem, takie jak nowe komentarze.
* Użytkownicy mogą obserwować innych użytkowników
* Użytkownicy mogą przeszukiwać i sortować treści według różnych kryteriów, takich jak data, popularność lub kategorie.
* Użytkownicy mogą zgłaszać treści nieodpowiednie lub naruszające zasady serwisu, a moderatorzy mogą podejmować odpowiednie działania.
* Administratorzy mogą uzyskiwać dostęp do statystyk dotyczących aktywności użytkowników i użyteczności serwisu.
* Użytkownicy mogą zarządzać swoimi kontami, resetować hasła i przywracać dostęp do konta w razie potrzeby.
* Administracja może zarabiać na serwisie za pomocą reklam lub innych strategii monetyzacji.
* Dostępne jest wsparcie techniczne i obsługa użytkowników w razie pytań, problemów lub zgłoszeń.
* Serwis może obsługiwać wiele wersji językowych w celu przyciągnięcia użytkowników z różnych krajów.
* Zarządzanie uprawnieniami dostępu do treści i funkcji serwisu, w tym role użytkowników.

**Wymagania niefunkcjonalne:**

* Serwis musi być responsywny i szybki, z minimalnymi opóźnieniami w interakcji użytkownika.
* (opcjonalne) Czas wczytywania stron internetowych i interfejsu użytkownika musi być krótki, zwłaszcza na urządzeniach mobilnych.
* Serwis musi być dostępny 24/7, bez większych przerw technicznych.
* Wymagane jest monitorowanie dostępności i zdolność szybkiego reagowania na awarie.
* Dane użytkowników muszą być chronione przed dostępem osób nieuprawnionych.
* Wymagane jest zabezpieczenie przed atakami typu SQL Injection, Cross-Site Scripting (XSS) i inne zagrożenia.
* Serwis musi być skalowalny, aby obsłużyć rosnącą liczbę użytkowników i danych.
* Wymagane jest równoważenie obciążenia, aby rozproszyć ruch między serwerami.
* Serwis musi być zgodny z przepisami dotyczącymi prywatności, ochrony danych i praw autorskich.
* Wymagane jest przestrzeganie regulacji dotyczących przechowywania danych użytkowników.
* Wymagane jest monitorowanie aplikacji w celu wykrywania błędów i problemów wydajnościowych.
* Serwis musi gromadzić dane diagnostyczne, które pomogą w identyfikacji problemów.
* Serwis musi obsłużyć określoną liczbę jednoczesnych użytkowników i żądań.
* Wymagane jest zapewnienie odpowiedniej przepustowości serwera i bazy danych.
* Serwis musi zapewnić akceptowalny czas odpowiedzi na żądania użytkowników.
* Wymagana jest optymalizacja aplikacji w celu skrócenia czasu odpowiedzi.
* Interfejs użytkownika musi być intuicyjny i przyjazny dla użytkownika.
* Wymagane jest zapewnienie dostępności dla użytkowników niepełnosprawnych.
* System uwierzytelniania musi być bezpieczny i zapewniać poufność danych użytkowników.
* Wymagane jest zarządzanie sesjami, w tym możliwość wylogowania.
* Serwis musi działać poprawnie na różnych przeglądarkach internetowych i urządzeniach (komputery, smartfony, tablety).
* Wymagana jest dobra jakość kodu źródłowego, umożliwiająca łatwe wprowadzanie zmian i utrzymanie systemu.
* Baza danych musi być zoptymalizowana pod kątem wydajności, aby zapewnić szybkie zapytania i manipulację danymi.
* Określenie, jakie zasoby serwera są wymagane do działania serwisu, takie jak moc obliczeniowa, przestrzeń dyskowa itp.

# 5. Model przypadków użycia

*Obraz zawierający tekst, zrzut ekranu, diagram, design

Opis wygenerowany automatycznieRys 6.1 Diagram przypadków użycia*

**Przeglądanie postów i komentarzy** - użytkownik przegląda strumień postów i odpowiadających im komentarzy, śledząc aktywność społecznościową w witrynie.

**Dodawanie postów** - użytkownik dodaje nowe treści lub informacje poprzez opublikowanie nowego posta, dzieląc się swoimi myślami lub doświadczeniem.

**Komentowanie postów** - użytkownik komentuje posty, wyrażając swoje zdanie, udzielając odpowiedzi lub inicjując dyskusję z innymi użytkownikami.

**Dawanie postom like lub dislike** - użytkownik wyraża swoje pozytywne lub negatywne uczucia względem postów, korzystając z opcji "like" lub "dislike".

**Zapisywanie ulubionych postów do listy ulubionych** - użytkownik zapisuje wybrane posty do osobistej listy ulubionych, aby łatwo móc do nich wracać w przyszłości.

**Przeglądanie swojego konta** - użytkownik przegląda swoje własne konto, obejmuje to informacje o profilu, aktywnościach, postach i komentarzach.

**Edycja danych konta** - użytkownik modyfikuje informacje w swoim profilu, takie jak zdjęcie profilowe, opis, czy dane kontaktowe.

**Przeglądanie swoich postów** - użytkownik przegląda listę swoich własnych opublikowanych postów, analizując ich popularność i interakcje.

**Przeglądanie komentarzy** - użytkownik przegląda komentarze pod swoimi postami lub komentarze innych użytkowników, uczestnicząc w dyskusji.

**Przeglądanie postów w jednej kategorii** - użytkownik przegląda posty związane z konkretną kategorią tematyczną, dostosowując swoje doświadczenie do konkretnych zainteresowań.

**Przeglądanie kont innych użytkowników** - użytkownik przegląda profile innych użytkowników, aby dowiedzieć się więcej o ich aktywności i zainteresowaniach.

**Ukrywanie postów** -moderatorzy i administratorzy ukrywają nieregulaminowe treści aby utrzymać odpowiednie standardy.

**Wysyłanie powiadomień** - użytkownik otrzymuje powiadomienia o nowych komentarzach, polubieniach, odpowiedziach lub innych aktywnościach związanych z jego postami.

**Ograniczenia uprawnień użytkownika** – administrator ogranicza dostęp do określonych funkcji lub treści w zależności od zachowania danego użytkownika na stronie

**5.1 Specyfikacja przypadków użycia**

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa | Dodanie nowego posta | | |
| Aktor | Zalogowany użytkownik | | |
| Opis | Przypadek użycia pozwala użytkownikowi dodać nowy post | | |
| Warunki wstępne | Użytkownik posiada konto w systemie, jest zalogowany oraz ma odpowiednie uprawnienia | | |
| Przepływ zdarzeń | Lp. | akcja aktora | Odpowiedź systemu |
| 1 | Użytkownik wybiera opcje „Dodaj nowy post” | System wyświetla formularz do dodania nowego posta |
| 2 | Użytkownik wprowadza poprawne dane | System dodaje nowy post i potwierdza dodanie |
|  |  |  |
| Warunki końcowe | Nowy post został dodany | | |
| Przepływ alternatywny 1 | 2a | Użytkownik wprowadza błędne dane | System wyświetla komunikat o podaniu złych danych |
| 3 | Użytkownik poprawia wprowadzone dane | System potwierdza dodaje nowy post i potwierdza dodanie |
| Przepływ alternatywny 2 | 2a | Użytkownik wprowadza niekompletne dane | System wyświetla komunikat o braku danych |
| 3 | Użytkownik uzupełnia wprowadzone dane | System potwierdza dodaje nowy post i potwierdza dodanie |

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa | Report posta | | |
| Aktor | Zalogowany użytkownik | | |
| Opis | Przypadek użycia pozwala użytkownikowi zgłosić moderatorom i administratorom potencjalnie nieregulaminowe treści | | |
| Warunki wstępne | Użytkownik posiada konto w systemie, jest zalogowany oraz ma odpowiednie uprawnienia | | |
| Przepływ zdarzeń | Lp. | akcja aktora | Odpowiedź systemu |
| 1 | Użytkownik wybiera opcje reportuj (przycisk z znakiem wykrzyknika) pod postem | System wyświetla formularz do wybrania powodu zgłoszenia |
| 2 | Użytkownik wybiera dany powód | System dodaje nowy zgłoszenie potwierdza je |
| 3 | Użytkownik próbuje dodać drugie zgłoszenie | System wyświetla powiadomienie o już istniejącym zgłoszeniu |
| Warunki końcowe | Nowy zgłoszenie zostało dodane | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa | Banowanie użytkownika | | |
| Aktor | Zalogowany administrator z odpowiednimi uprawnieniami | | |
| Opis | Przypadek użycia pozwala administratorowi ukarać użytkownika za niepoprawne zachowanie poprzez ograniczenie dostępnych funkcji | | |
| Warunki wstępne | Administrator posiada konto w systemie, jest zalogowany oraz ma odpowiednie uprawnienia | | |
| Przepływ zdarzeń | Lp. | akcja aktora | Odpowiedź systemu |
| 1 | administrator wybiera opcje „Ban” | System wyświetla formularz do wybrani okresu i powodu bana |
| 2 |  | System ogranicza uprawnienia wybranego użytkownika i potwierdza wykonanie operacji |
|  |  |  |
| Warunki końcowe | Wybrany użytkownik stracił część uprawnień | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa | Kontrola zgłoszonych postów w panelu administratora | | |
| Aktor | Zalogowany administrator | | |
| Opis | Przypadek użycia pozwala administratorowi usuwać nieregulaminowe posty | | |
| Warunki wstępne | Administrator posiada konto w systemie, jest zalogowany oraz ma odpowiednie uprawnienia | | |
| Przepływ zdarzeń | Lp. | akcja aktora | Odpowiedź systemu |
| 1 | Administrator przechodzi na podstronę „panel administratora” | System wyświetla podstronę z postami, które mają zgłoszenia o możliwym naruszeniu regulaminu |
| 2 | Administrator wybiera dany post i klikają „pokaż raporty” | System otwiera okienka i wyświetla rodzaje zgłoszeń w raz z ich liczbą |
| 3 | Administrator przegląda zgłoszenia i usuwa post klikając „Usuń post” | System usuwa post i wyświetla potwierdzenie |
| Warunki końcowe | Wybrany post został usunięty lub przywrócony | | |
| Przepływ alternatywny 1 | 3a | Administrator określa status posta jako „checked” | System wyświetla komunikat o zmianie statusu |

# 6. Projekt interfejsu

Na stronie głównej dostępne będzie :

* przeglądanie postów i komentarzy wraz opcją zaznaczenia jakiej kategorii posty chce się przeglądać
* dodanie własnego posta
* komentowanie postów
* ocenianie ich za pomocą like
* zapisywanie ulubionych
* przeglądanie kont innych użytkowników

W podstronie konta:

* personalizacja konta poprzez zmianę danych
* przeglądanie własnych postów i komentarzy
* przeglądanie ulubionych postów

Administratorzy będą mogli dodatkowo

* Na stronie głównej
* Ukrywać posty nie mieszczące się w normach
* Wysyłać powiadomienia użytkownikowi, aby go poinformować o złym zachowaniu
* Przeglądając konta innych
* Ograniczyć uprawnienia użytkownika

**Obraz zawierający tekst, zrzut ekranu, Oprogramowanie multimedialne, Oprogramowanie graficzne

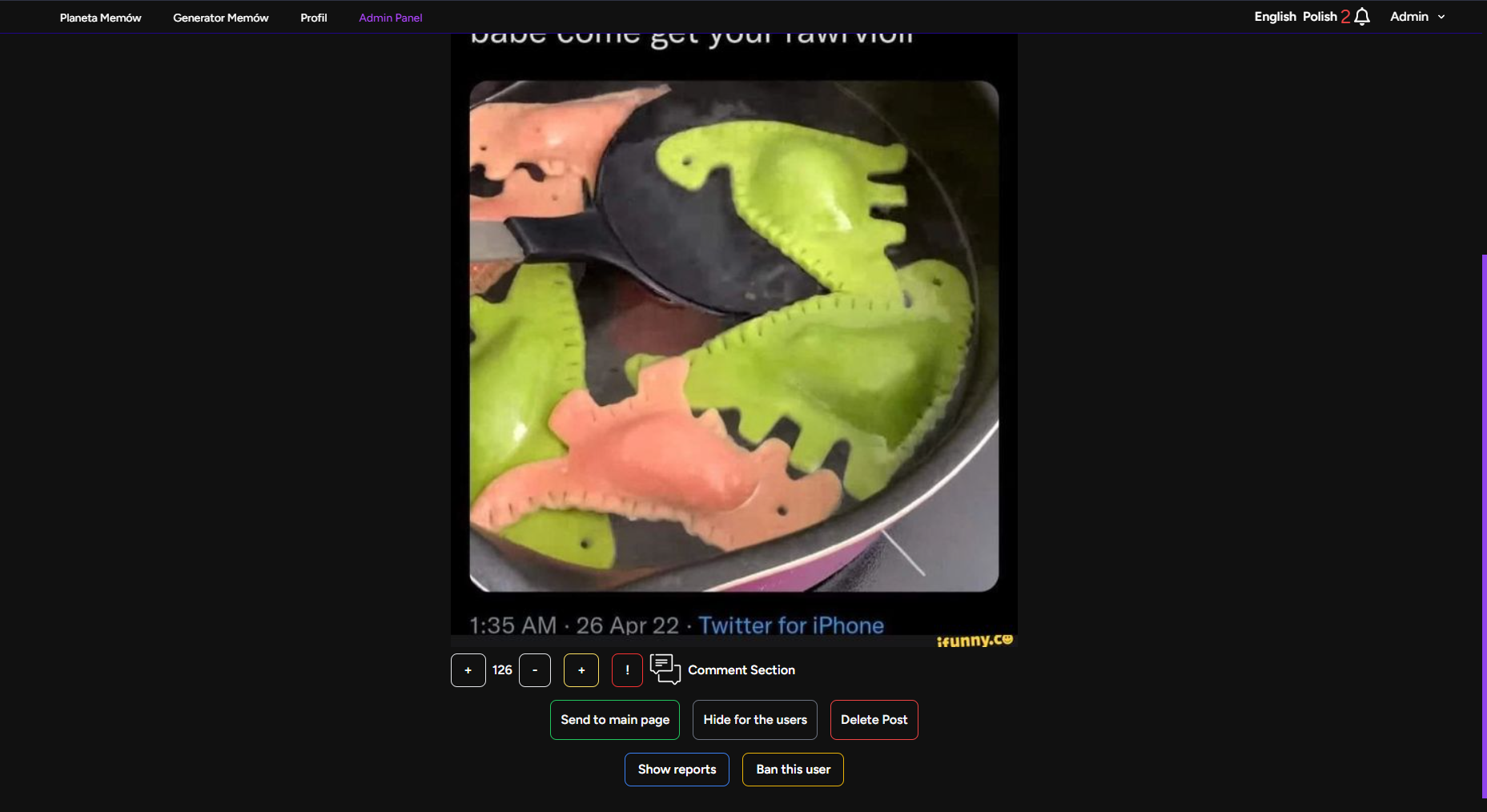
Opis wygenerowany automatycznie**

*Rys. 6.1 widok głównej strony*

**Obraz zawierający tekst, zrzut ekranu, Oprogramowanie multimedialne, Oprogramowanie graficzne

Opis wygenerowany automatycznie**

*Rys. 6.2 widok profilu*

**

*Rys. 6.3 widok panelu administratora*

# 7. Model architektury

Architektura systemu [Rysunek 7.1] składa się z dwóch warstw. backendu zajmującego się obsługiwanie żądań i operacjami na bazie danych i frontendu wyświetlającego dane w odpowiednim formacie i reagującego na akcje użytkownika.

Obraz zawierający zrzut ekranu, tekst, diagram, Prostokąt

Opis wygenerowany automatycznie

*Rys 7.1 architektura systemu*

Do tworzenia projektu zostanie wykorzystany Xampp, na którym będzie działać baza danych. Hostingiem zajmie się wbudowany serwer Laravel’a, a przetwarzaniem Frontendu Node.js

Obraz zawierający diagram, tekst, zrzut ekranu, Plan

Opis wygenerowany automatycznie

*Rys 7.1 Szczegółowa architektura systemu*

# 8. Projekt bazy danych

Tabela Users będzie główny elementem w całej strukturze, łącząca wszystkie tabele wspólną całość

* Users – dane użytkowników
* Posts –podstawowe dane o danym poście
* Comments –komentarze do danego posta i komentarzy
* Favorites – zapisuje ulubione posty użytkownika
* Categories – przechowuje kategorię działów
* TagsList – zapisuje tagi nadane danemu postowi
* Tags – przechowuje tagi, które można nadać postowi
* Notifications – przechowuje dane powiadomień

Users-posts – jeden użytkownik wiele postów

Users-comments – jeden do wielu

Users-likes- – jeden do wielu

Users-favourites – jeden do wielu

Users-notifications– jeden do wielu

Post-comments– jeden do wielu

Post-like– jeden do wielu

Post-favourite– jeden do wielu

Category-post-jeden do wielu

TagsList - post– zapis tagów do posta bo post może mieć wiele tagów z tabeli Tags

Tags-TagsList – słownik tagów

Obraz zawierający tekst, zrzut ekranu, Oprogramowanie multimedialne, Oprogramowanie graficzne

Opis wygenerowany automatycznie

*Rys. 8.1 projekt bazy danych*

# 10. Implementacja

# 10.1 Opis wybranych technologii i narzędzi

Dom implementacji wybrano laravel gdyż jest on mocno zagnieżdżony na rynku dzięki czemu istnieje mnóstwo gotowych rozwiązań i funkcji. Skróci to tworzenie backendowej części projektu. Użyto także react oraz do stylizacji tailwind. Obie technologie są ze sobą kompatybilne i łatwo tworzy się za ich pomocą interaktywny interfejs. Backend i frontend będzie komunikował się za pomocą axios, który będzie przesyłał odpowiednie dane.

Do implementacji jako IDE wykorzystane zostanie visual studio code. Jest to popularne narzędzie z masą rozszerzeń, w pełni kompatybilne z githubem i większością języków programowania. Do implementacji w fazie tworzenia zostanie także wykorzystany xampp w roli bazy danych.

**10.2 Opis implementacji wybranych funkcji aplikacji**

**10.2.1 Niekończący się potok postów**

System ten został zaimplementowany za pomocą gotowego komponentu react InfiniteScroll, komponent ten pozwala wyświetlanie niekończącej się listy postów w formie jednej długiej kolumny. Przewija się to jak normalną stronę, lecz przy każdej próbie dojścia do końca ładowane są nowe posty.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

*Rys. 10.2.1.1 kod użycia komponentu InfiniteScroll*

**10.2.2System wyświetlania komentarzy**

Komentarze są wyświetlane w oddzielnej sekcji, która jest wysuwana od dołu ekranu. Komentarze posiadają linie po lewej stronie aby zaznaczyć zagnieźdzenie danej konwersacji. Dodatkowo jest oznaczenie kto i do komu odpowiada aby użytkownicy czytając to mieli punkt odnieśnia. Zagnieżdzone komentarze renderują się rekurencyjnie wyszukując czy posiadaja jakieś komentarze.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

*Rys. 10.2.2.1 Kod inicjujący wyświetlanie komentarzy*

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

*Rys. 10.2.2.2 Kod wyszukujący zagnieżdżone komentarze dla danego komentarza*

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

*Rys. 10.2.2.3 Kod renderujący zagnieżdzone komentarze poprzez rekurencyjne wyświetlanie komponenetu Comment*

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

*Rys. 10.2.2.4 Widok zagnieżdzonych komentarzy*

|  |
| --- |
| **11.Testy**    **11.1 Testy jednostkowe**    11.1.1 Dodaj post  11.1.2 Dodaj komentarz  11.1.3 Dodawanie komentarza  **11.2 Testy funkcjonalne**  11.2.1  11. jednostkowe, funkcjonalne, ciekawe przypadki testowe dla niektórych funckcjonalności  12. podsumowanie potwórzenie celu pracy, opisanie co zostało zrealzowane prae zdań, co nie zostało zrealizowane, w jakim kierunku bedize rozwijana aplikacja  13. prezentacja aplikacji  06.paź  przygotowanie harmonogramu + |
| dokończenie przebudowy struktury 50% |
| dokończenie podstrony konta użytkownika +  13.paź |
| powiadomienia systemowe odnośnie komentarzy i postów + |
| prywatne wiadomości od innych użytkowników (opcjonalne)  funkcja pokazująca, że dany komponent się ładuje  poprawki dokumentacji  27.paź |
| funkcje administratorskie (edycja strony, postów, banowanie itp.):  +przegląd ukrytych, zreporotwany postów i komentarzy oraz przyczyn ich reportów  +banowanie użytkowników  +możlwiosć dodawania moderatorów i adminów oraz określania ich uprawnień  +możliwość tworzenia kategorii i wynoszenia tagów do rangi kategorii  +przy dodawaniu posta możliwość dodania własnego taga przez użytkownika  +system reportowania postów  +tłumaczenia  poprawki dokumentacji |
| 17.lis  +zabezpieczenia systemowe oraz pól w których użytkownik wprowadza dane  +funkcje administratorskie (edycja strony, postów, banowanie itp.):  +przegląd ukrytych, zreporotwany postów i komentarzy oraz przyczyn ich reportów  +banowanie użytkowników  +możlwiosć dodawania moderatorów i adminów oraz określania ich uprawnień  +możliwość tworzenia kategorii i wynoszenia tagów do rangi kategorii  +rzy dodawaniu posta możliwość dodania własnego taga przez użytkownika  +system reportowania postów  +tłumaczenia  poprawki dokumentacji |
| 24.lis  Porządkowanie kodu  (czas na dodatkowe mechanizmy: mechanizm baniek informacyjnych, generator postów) |
| poprawki dokumentacji  01.gru |
| tłumaczenia |
| testy |
| Poprawki  poprawki dokumentacji |
| 15.gru  zakończenie pracy |
| 12.sty  poprawki + przygotowanie na obronę |
| 19.sty  poprawki + przygotowanie na obronę |
| 26.sty  Poprawki + przygotowanie na obronę |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| role\Permissions | add post | add comment | add post to fav | report post | like post | send post ot main | hide post | show report | delete post | ban user |
| Admin | + | + | + | + | + | + | + | + | + | + |
| Moderator | + | + | + | + | + | + | + | + |  |  |
| User | + | + | + | + | + |  |  |  |  |  |
| Observer |  |  |  |  |  |  |  |  |  |  |

NOTATKI

Banki informacyjne/profile

* 1. Zabierz dane o użytkowniku jakie posty wyswietla, gdzie pisze komentarz, jakie zapisuje do ulubionych jakie likuje w pod jakimi daje like do komentarzy
  2. Przelicz je i podczas kolejnego pobranai postów lub wysłania posta
  3. Na backendzie zapisz zmiany w profilu user, dopasuj profil do niego lub zapisz draft
  4. Jak będzie dużo draftów lub 10000 draftów. Pzelicz profile usuń nie używane wzmocnij te najcześciej uzywane utwórz nowe z najczesciej pojawiajacych się draftów

System w formie globalnego algorytmu genetycznego/mrowkowego + dodawanie specyficznych cech profilu user od niego coś jak pokaż userowi to co jego grupa/profil/bańka lubi aby go zainsteresować, a jak nie polubi to pokaż mu to co nakazuje jego prywatny profil

Przy kolejnym przeładowaniu profili powtórz

Testy jednostkowe i funkcyjne wystarczą