**Obraz zawierający tekst

Opis wygenerowany automatycznie**

PRACA DYPLOMOWA

Adam Marciniak

Social networking service

**Entertainment Service**

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)

[2.1 Słownik pojęciowy 7](#_Toc137725753)

[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)

# 1. **Cel i zakres pracy**

Celem projektu jest stworzenie serwisu społecznościowego, który będzie zrzeszał różne grupy ludzi aby mogli wymieniać się informacjami, rozwiązaniami, problemami, zainteresowaniami i innymi tego typu społecznymi zagadnieniami. Aplikacja ma głównie zapewniać możliwość prowadzenia dyskusje na różne tematy podane poście użytkownika.

Serwis ma umożliwiać użytkownikom:

* wstawiać swoje posty z zaznaczeniem kategorii i tagów
* komentować posty
* prowadzić dyskusje w komentarzach
* oceniać posty oraz komentarze poprzez like
* przeglądać posty z danej kategorii
* założyć konto i edytować je
* zapisywać ulubione posty

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

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

Zakres pracy będzie obejmować etapy:

* analiza wiedzy dziedzinowej
* stworzenie jednolitej wizji
* określenie głównych elementów
* wybranie technologii
* stworzenie prototypów interfejsów graficznych
* implementacja na serwerze lokalnym
* testowanie
* implementacja na serwerze produkcyjnym
* testowanie

# 2. **Wizja**

Serwis będzie aplikacją webową o budowie SPA z trzema podstronami.

* Główna strona:

Będzie to najważniejsza strona, na której będą wyświetlane posty i komentarze

* Strona konta:

Ta strona ma posiadać możliwość:

* + zmianę danych konta
  + przegląd swoich postów i komentarzy
  + przegląd ulubionych postów

(opcjonalne)

* Generator memów:

Ta podstrona będzie oferowała proste narzędzie umożliwiające stworzenie mema

(opcjonalne)

* Generator bańiek informacyjnych:

Będzie to dodatkowa funkcja, która będzie analizowała zachowanie użytkownika(treści jakie najczęściej klika/komentuje) aby sprawdzić co go interesuje. Na podstawie tych danych zostaną utworzone profile (bańki) na podstawie, których będą wyświetlane posty danych kategorii jako pierwsze

## 2.1 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 skałda się tytułu, tekstu i/lub obrazka. Posty powinny mieć zawsze wprowadzony tytuł, tekst lub obrazek oraz zaznaczoną kategorię
* Tag – podkategoria wprowadzone aby określić post w który zawiera elementy innej kategorii
* RESTAPI – 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

# 3. Model domenowy

Serwis zostanie podzielony na dwa główne elementy backend i fronted

Obraz zawierający diagram, linia, Plan, Prostokąt

Opis wygenerowany automatycznie

*Rys 3.1 Diagram domenowy(opracowanie własne)*

# 4. Model wymagań

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

* umożliwienie użytkownikom wstawianie postów
* umożliwienie użytkownikom komentować posty oraz komentarze
* umożliwienie użytkownikom dawanie postom like
* umożliwienie użytkownikom przeglądanie postów z danej kategorii
* umożliwienie użytkownikom założenie konta i jego edycji
* umożliwienie użytkownikom zapisywać ulubione posty
* umożliwienie administratorom usuwanie postów
* umożliwienie administratorom wysyłać powiadomienia do użytkowników
* umożliwienie administratorom ograniczać uprawnienia użytkownikom

Rejestracja i Logowanie:

* Użytkownicy mogą tworzyć konta i logować się za pomocą adresu e-mail, loginu lub kont społecznościowych.

Tworzenie Profilu:

* Użytkownicy mogą tworzyć i zarządzać swoimi profilami, dodając informacje o sobie, zdjęcia profilowe itp.

Dodawanie Treści:

* Użytkownicy mogą tworzyć i udostępniać różne rodzaje treści, takie jak posty, zdjęcia, filmy itp.

Zarządzanie Treściami:

* Użytkownicy mogą edytować, usuwać i zarządzać swoimi opublikowanymi treściami.

Interakcja z Treściami:

* Użytkownicy mogą komentować, polubić i udostępniać treści innych użytkowników.

Powiadomienia:

* Użytkownicy otrzymują powiadomienia o aktywności związaną z ich kontem, takie jak nowe komentarze lub polubienia.

Społeczność i Sieć Społeczna:

* Użytkownicy mogą łączyć się ze znajomymi, obserwować innych użytkowników i budować swoją sieć społeczną.

Przeszukiwanie i Sortowanie Treści:

* Użytkownicy mogą przeszukiwać i sortować treści według różnych kryteriów, takich jak data, popularność lub kategorie.

Grupy i Społeczności:

* Użytkownicy mogą tworzyć i dołączać do różnych grup i społeczności o określonych zainteresowaniach.

Prywatność i Ustawienia Konta:

* Użytkownicy mają możliwość zarządzania ustawieniami prywatności i bezpieczeństwa swojego konta.

Wiadomości Prywatne:

* Użytkownicy mogą wysyłać wiadomości prywatne do innych użytkowników.

Raportowanie i Moderacja:

* Użytkownicy mogą zgłaszać treści nieodpowiednie lub naruszające zasady serwisu, a moderatorzy mogą podejmować odpowiednie działania.

Integracja z Mediami Społecznościowymi:

* Użytkownicy mogą udostępniać swoje treści na innych platformach społecznościowych.

Statystyki i Analizy:

* Administratorzy mogą uzyskiwać dostęp do statystyk dotyczących aktywności użytkowników i użyteczności serwisu.

Zarządzanie Kontem:

* Użytkownicy mogą zarządzać swoimi kontami, resetować hasła i przywracać dostęp do konta w razie potrzeby.

Mobilna Aplikacja:

* Serwis może mieć zintegrowaną aplikację mobilną, dostępną na platformy iOS i Android.

Reklamy i Monetyzacja:

* Administracja może zarabiać na serwisie za pomocą reklam lub innych strategii monetyzacji.

Usługi Dla Biznesów:

* Serwis może oferować usługi dla firm i marek, takie jak tworzenie stron i kampanie reklamowe.

Zarządzanie Zdjęciami i Multimediami:

* Użytkownicy mogą dodawać i zarządzać swoimi zdjęciami i multimediami.

Współdzielenie Treści i Integracje:

* Użytkownicy mogą udostępniać treści z zewnętrznych źródeł i integrować serwis z innymi aplikacjami.

Wsparcie Techniczne i Obsługa Użytkowników:

* Dostępne jest wsparcie techniczne i obsługa użytkowników w razie pytań, problemów lub zgłoszeń.

Wersje Językowe:

* Serwis może obsługiwać wiele wersji językowych w celu przyciągnięcia użytkowników z różnych krajów.

Integracja z Zewnętrznymi Systemami:

* Serwis może integrować się z zewnętrznymi systemami, takimi jak płatności online czy dostawcy usług autoryzacji.

Kontrola Dostępu i Uprawnienia:

* Zarządzanie uprawnieniami dostępu do treści i funkcji serwisu, w tym role użytkowników.

Opcje Tworzenia Zawartości:

* Użytkownicy mają różne opcje tworzenia treści, takie jak

**Wymagania niefunkcjonalne:**

Wydajność:

* 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.

Dostępność:

* Serwis musi być dostępny 24/7, bez większych przerw technicznych.
* Wymagane jest monitorowanie dostępności i zdolność szybkiego reagowania na awarie.

Bezpieczeństwo:

* Dane użytkowników muszą być chronione przed dostępem osób nieuprawnionych.
* Wymagane jest zabezpieczenie przed atakami typu SQL Injection, Cross-Site Scripting (XSS) i inne zagrożenia.

Skalowalność:

* Serwis musi być skalowalny, aby obsłużyć rosnącą liczbę użytkowników i danych.
* Wymagane jest równoważenie obciążenia, aby rozproszyć ruch między serwerami.

Zgodność z przepisami prawymi:

* Serwis musi być zgodny z przepisami dotyczącymi prywatności, ochrony danych i praw autorskich.
* Wymagane jest przestrzeganie regulacji dotyczących przechowywania danych użytkowników.

Zarządzanie błędami i monitorowanie:

* Wymagane jest monitorowanie aplikacji w celu wykrywania błędów i problemów wydajnościowych.
* Serwis musi gromadzić dane diagnostyczne, które pomogą w identyfikacji problemów.

Przepustowość:

* Serwis musi obsłużyć określoną liczbę jednoczesnych użytkowników i żądań.
* Wymagane jest zapewnienie odpowiedniej przepustowości serwera i bazy danych.

Czas odpowiedzi:

* Serwis musi zapewnić akceptowalny czas odpowiedzi na żądania użytkowników.
* Wymagana jest optymalizacja aplikacji w celu skrócenia czasu odpowiedzi.

Interfejs użytkownika i użyteczność:

* Interfejs użytkownika musi być intuicyjny i przyjazny dla użytkownika.
* Wymagane jest zapewnienie dostępności dla użytkowników niepełnosprawnych.

Zarządzanie sesjami i uwierzytelnienie:

* System uwierzytelniania musi być bezpieczny i zapewniać poufność danych użytkowników.
* Wymagane jest zarządzanie sesjami, w tym możliwość wylogowania.

Zgodność z przeglądarkami i urządzeniami:

* Serwis musi działać poprawnie na różnych przeglądarkach internetowych i urządzeniach (komputery, smartfony, tablety).

Jakość kodu i utrzymanie:

* Wymagana jest dobra jakość kodu źródłowego, umożliwiająca łatwe wprowadzanie zmian i utrzymanie systemu.

Wydajność bazy danych:

* Baza danych musi być zoptymalizowana pod kątem wydajności, aby zapewnić szybkie zapytania i manipulację danymi.

Zasoby serwera:

* Określenie, jakie zasoby serwera są wymagane do działania serwisu, takie jak moc obliczeniowa, przestrzeń dyskowa itp.

# 5. Model przypadków użycia

Poniższy diagram [Rysunek 5.1] obrazuje funkcjonalności, które będzie posiadać użytkownik oraz dodatkowe funkcjonalności administratora

*Obraz zawierający tekst, zrzut ekranu, diagram, design

Opis wygenerowany automatycznieRys 5.1 Diagram przypadków użycia*

# 6. Model architektury

Architektura systemu [Rysunek 6.1] 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 6.1 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, Równolegle

Opis wygenerowany automatycznie

*Rys 6.1 Szczegółowa architektura systemu*

# 7. Projekt bazy danych

Poniższy rysunek [Rysunek 7.1] obrazuje planowaną bazę danych.

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

Opis wygenerowany automatycznie

*Rys. 7.1 projekt bazy danych*

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

Inne tabele:

* 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

# 8. Projekt interfejsu

Na stronie głównej dostępne będzie :

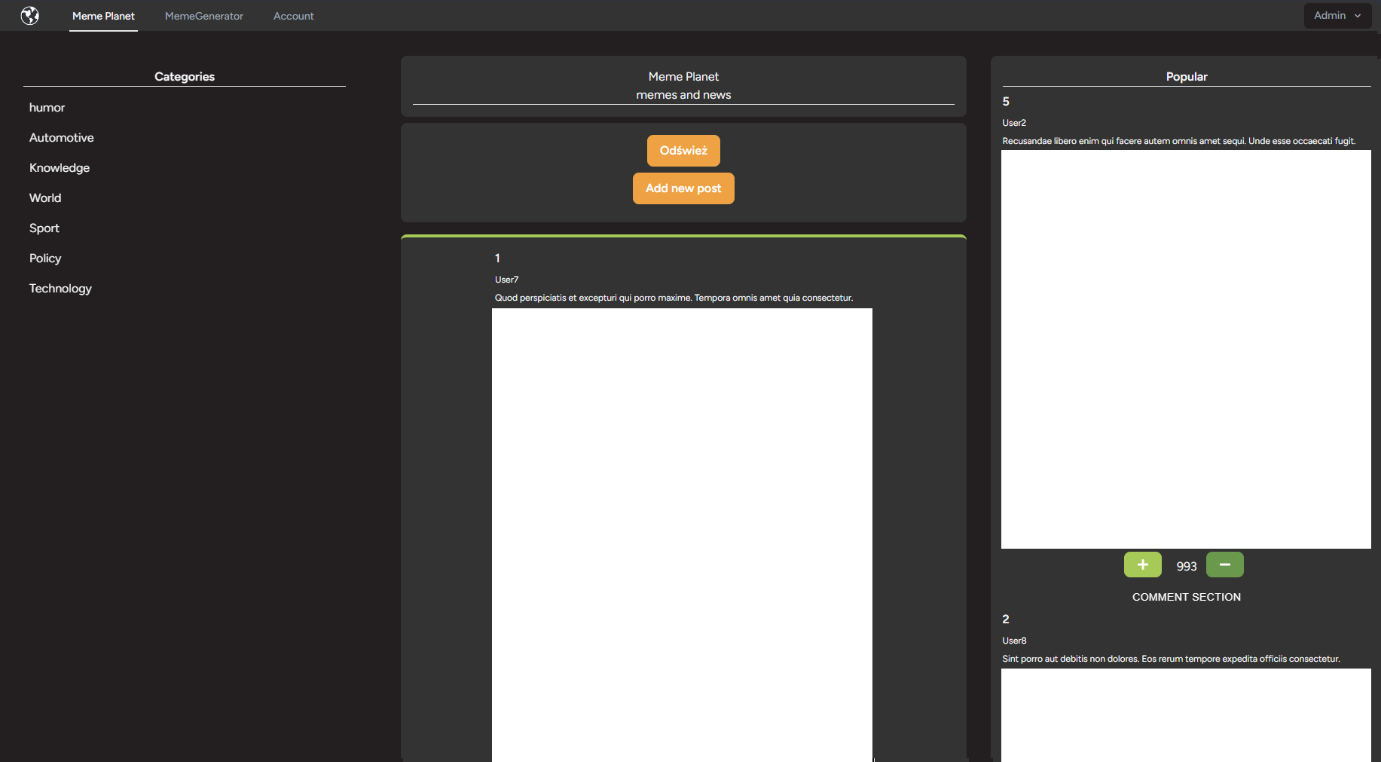
* przeglądanie postów i komentarzy wraz opcją zaznaczenia jakiej kategorii posty chce się przeglądać
* dodanie własnego posta
* komentowanie postów
* ocenianie ich za pomocą likeów
* zapisywanie ulubionych
* przeglądanie kont innych użytkowników

W podstronie konta:

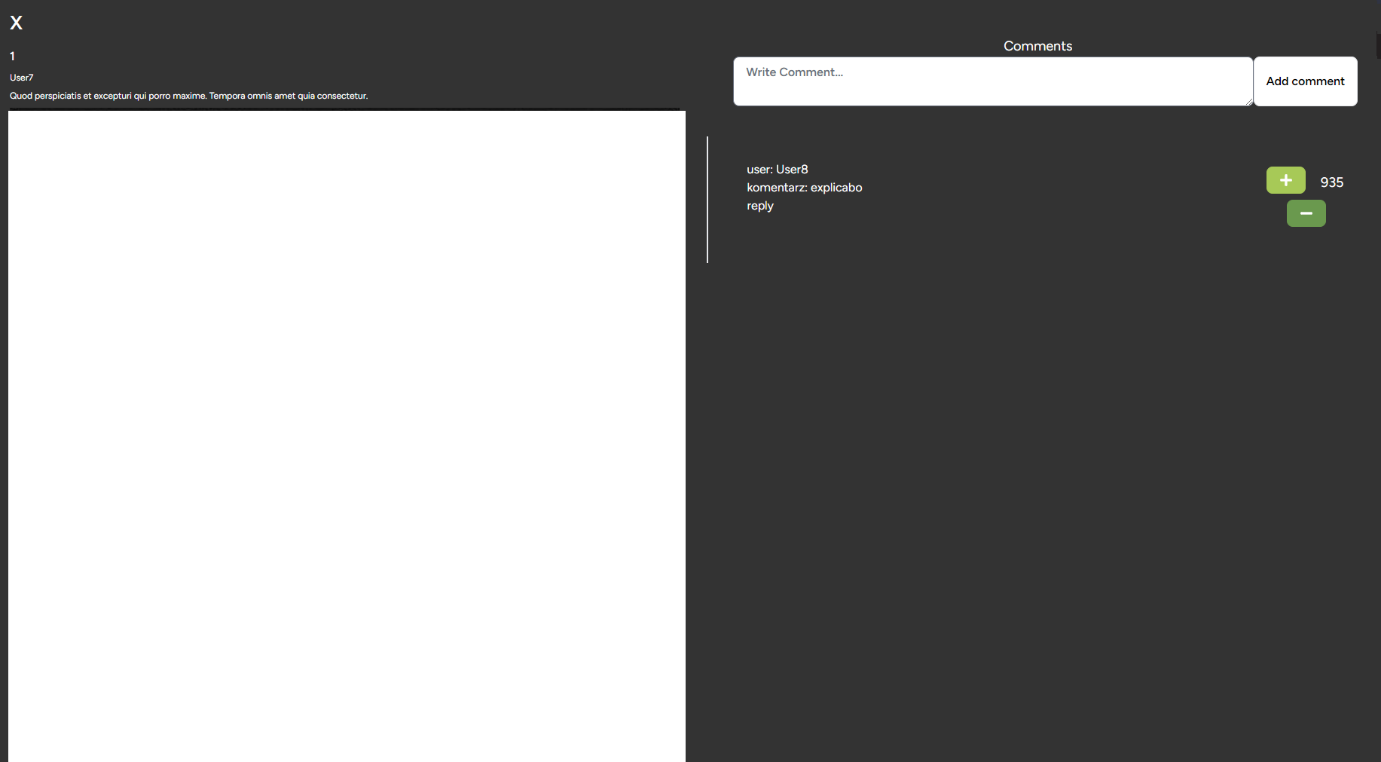
* personalizacja konta poprzez zmianę danych
* przeglądanie własnych postów i komentarzy
* przeglądanie ulubionych postów

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

****

*Rys. 8.1 widok głównej strony*

****

*Rys. 8.2 widok pojedyńczego posta*

# 9. Implementacja

Jako Backend wybrano framework laravel gdyż jest on mocno zagnieżdżony na rynku dzięki czemu istnieje mnóstwo gotowych rozwiązań i funkcji. Skróci to tworzenie tej części projektu. Do Frontend wybrano React, a do stylizacji Tailwind. Obie technologie są ze sobą kompatybilne i łatwo tworzy się za ich pomocą interaktywny interfejs strony. Postawiono na React gdyż jest on stworzony przez facebook co również pozwala skrócić czas włożony w implementację. Dodatkowo jest duże prawdopodobieństwo jego długoletniego wsparcia i łatwiejsze rozszerzanie projektu w miarę wymagań konsumentów.

Aplikacja będzie działać na zasadzie tworzenia danych w oparciu o modele i kontrolery laravel, które będą przesyłane na front w formacie json. Na frontend interfejs będzie wyświetlany w formie SPA. System będzie wsparty o technologie inertia.js aby implementować przechodzenie na podstrony bez przeładowywania strony imitując tym jednolity interfejs.

**9.1Niekończący się potok postów**

System ten został zaimplementowany za pomocą gotowego komponentu react InfiniteScroll, komponent ten pozwala wyświetlanie niekończącej się listy podanych elementów oraz dodawania do ich listy nowych elementów w miarę potrzeb

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

*Rys. 9.1.1 kod użycia komponentu InfiniteScroll*

**9.2System 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. 9.2.1 Kod inicjujący wyświetlanie komentarzy*

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

*Rys. 9.2.2 Kod wyszukujący zagnieżdżone komentarze dla danego komentarza*

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

*Rys. 9.2.3 Kod renderujący zagnieżdzone komentarze poprzez rekurencyjne wyświetlanie komponenetu Comment*

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

*Rys. 9.2.1 Widok zagnieżdzonych komentarzy*

|  |
| --- |
| 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.)  poprawki dokumentacji |
| 17.lis  zabezpieczenia systemowe oraz pól w których użytkownik wprowadza dane  poprawki dokumentacji |
| 24.lis  generator postów |
| mechanizm baniek informacyjnych  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  Co uprawnienia mogą być nadawane elstycznie przez admina to jest tabela uprawnień do seedera   |  |  |  | | --- | --- | --- | | funkcja | Admin wszystko może | Moderator | | Add post to main page | + | + | | Hide post | + | + | | Send message/notfication to user and select reason from list | + | + | | Hide comment | + | + | | Hard delete post | + |  | | Create kategories by set tag as category | + |  | | Add things to change log | + |  | | Add socjal-system message under website navigationbar | + |  | | Create moderator | + |  | | Create admin | + |  | | Ban user |  |  | |