

System zarządzania nauką online (LMS)

Projekt zespołowy

Szymon Świercz, nr albumu: 119274

Adam Waśko, nr albumu: 115986

Bartosz Surma, nr albumu: 120859

Mikołaj Noga, nr albumu: 119246

Ernest Milecki, nr albumu: 120807

Informatyka, rok III, stopień I

Semestr zimowy, rok akademicki 2023/2024

Streszczenie

Projekt LMS (Learning Management System) ma na celu stworzenie zaawansowanej platformy edukacyjnej, która umożliwi sprawną naukę online. Głównym celem projektu jest ułatwienie dostępu do materiałów edukacyjnych, poprawa jakości nauczania oraz personalizacja procesu nauki. Platforma opiera się na technologiach takich jak Java Spring, MySQL, React oraz Tailwind CSS, co zapewnia niezawodność, skalowalność i atrakcyjny wygląd interfejsu użytkownika. Projekt ma na celu zwiększenie efektywności edukacji online, monitorowanie postępów uczestników oraz ułatwienie komunikacji między użytkownikami. Ostatecznym celem jest stworzenie nowoczesnej platformy edukacyjnej, która spełni potrzeby uczestników nauki online i wspomogę zarządzanie procesem edukacyjnym.

Spis treści

Streszczenie	2
Wprowadzenie	5
Motywacja	5
Cele	5
Znaczenie projektu	5
Opis projektu	7
Szczegółowe wyjaśnienie zakresu i celu projektu	7
Wymagania i specyfikacje projektu	7
Metodologia	8
Baza danych	8
Budowa API	8
Budowa front-end	9
Projekt systemu	10
Architektura systemu	10
Diagramy UML	11
Diagram przypadków użycia	11
Diagramy sekwencji	12
Schemat bazy danych	13
Wdrożenie	14
Wdrożenie bazy danych:	14
Wdrożenie <i>back-end</i>	14
Wybrane technologie	14
Konfiguracja środowiska	14
Wdrożenie <i>front-end</i>	15
Testowanie i walidacja	16
Szczegóły procesu testowania:	16
Testy jednostkowe	16

Testy integracyjne	16
Przypadki testowe i scenariusze.....	16
Wyniki badań i procedury walidacyjne:	17
Rozwiązywanie problemów i błędów	17
Wyniki i dyskusja	19
Omówienie wszelkich ograniczeń i obszarów wymagających poprawy.....	25
Przyszła praca	25
Referencje	27
Załączniki	28

Wprowadzenie

W dzisiejszych czasach rozwój technologii stał się nieodłącznym elementem naszego życia. Świat coraz bardziej opiera się na dostępie do informacji i wiedzy, a jednym z głównych źródeł edukacji staje się Internet. W tym kontekście, projekt systemu zarządzania nauką online (Learning Management System, LMS) nabiera ogromnego znaczenia.

Motywacja

Motywacją do stworzenia kompleksowego systemu zarządzania nauką online jest dynamiczny rozwój technologiczny, który wpływa dotychczasowy proces edukacji, który staje się coraz bardziej hybrydowy, łącząc klasyczne metody nauczania z korzystaniem z technologii.

Cele

Celem projektu jest utworzenie kompleksowego systemu zarządzania nauką online. System ten ma służyć szkołom i uniwersytetom w sprawnym zarządzaniu kursami, zadaniami, ocenami, a także profilami użytkowników.

System będzie usprawniał proces nauczania i szkolenia poprzez zapewnienie uczniom łatwego dostępu do materiałów edukacyjnych (takich jak: prezentacje, podręczniki, nagrania wideo, itp.) oraz umożliwienie wykonywania zadań, testów i sprawdzianów utworzonych przez nauczycieli. Ponadto system umożliwi śledzenie postępów uczestników oraz dostarczy raporty i analizy dotyczących wyników i zaangażowania uczniów. System ma również pomóc w usprawnieniu komunikacji między uczestnikami i nauczycielami poprzez utworzenie forów dyskusyjnych oraz możliwości wysyłania wiadomości e-mail bezpośrednio z systemu.

Znaczenie projektu

System do zarządzania nauką online ma głęboki wpływ na sektor edukacyjny, przyczyniając się do transformacji procesów nauczania i uczenia się. Do jego kluczowych aspektów można zaliczyć:

1. **Dostępność i elastyczność edukacji:** LMS umożliwia uczniom i studentom dostęp do materiałów edukacyjnych z dowolnego miejsca i o dowolnym czasie. To znacząco zwiększa dostępność edukacji, zwłaszcza dla tych, którzy mają ograniczone możliwości uczestniczenia w tradycyjnych zajęciach.
2. **Dostosowanie do indywidualnych potrzeb:** System pozwala na personalizację procesu nauczania, co pozwala na dostosowanie treści i tempa nauki do indywidualnych potrzeb uczniów. Każdy może rozwijać się w własnym tempie.

3. **Interaktywne i zaawansowane narzędzia edukacyjne:** LMS dostarcza narzędzia umożliwiające tworzenie interaktywnych materiałów edukacyjnych, testów i zadań. To sprawia, że nauka staje się bardziej atrakcyjna i skuteczna.
4. **Śledzenie postępów i ocenianie:** System umożliwia nauczycielom monitorowanie postępów uczniów oraz dokładne ocenianie ich osiągnięć. To pozwala na dostosowanie procesu nauczania i zapewnienie odpowiedniego wsparcia.
5. **Dokumentacja i analizy:** LMS gromadzi dane na temat aktywności uczniów, co pozwala nauczycielom oraz szkołom tworzyć raporty i analizy dotyczące wyników oraz zaangażowania uczniów. To cenne narzędzie do oceny efektywności procesu nauczania.
6. **Wsparcie zdalnego nauczania:** LMS zapewnia ciągłość edukacji nawet w trudnych warunkach.
7. **Redukcja barier geograficznych:** Dzięki LMS, studenci mogą uczestniczyć w kursach i szkoleniach oferowanych przez instytucje z różnych części świata, co przyczynia się do globalizacji i różnorodności w edukacji.

Opis projektu

Szczegółowe wyjaśnienie zakresu i celu projektu

Projekt LMS ma na celu stworzenie zaawansowanej platformy edukacyjnej, która umożliwia zarządzanie procesem nauki online. Będzie to kompleksowy system, dostępny w formie aplikacji internetowej, który obsłuży różne typy użytkowników, takie jak uczniowie, nauczyciele i administratorzy.

Wymagania i specyfikacje projektu

Każdy użytkownik loguje się w systemie przy pomocy danych uwierzytelniających. Ma dostęp do strony głównej, na której znajduje się skrót do wszystkich istotnych informacji, z których może przejść do odpowiednich paneli, zakładek oraz funkcji systemu.

Korzystając z odpowiednich paneli i zakładek **uczeń** ma możliwość rejestracji na przedmioty, przeglądania przedmiotów, których jest uczestnikiem, wraz z materiałami do nich dołączonymi, a także brać udział na forum dyskusyjnym oraz wykonywać zadania przypisane do danego przedmiotu. Ponadto może przeglądać kartę ocen, a także kontaktować się z innymi uczniami, nauczycielami oraz pracownikami administracyjnymi poprzez wiadomości e-mail.

Nauczyciel, korzystając z systemu, ma możliwość dodawania oraz edytowania panelu przedmiotu oraz dodawać i edytować materiały, zadania oraz testy przypisane do danego przedmiotu. Nauczyciel może generować i wyświetlać raporty, ukazujące zaangażowanie uczniów w dany przedmiot, a także wystawiać oceny cząstkowe (za wykonane zadania) oraz oceny końcowe.

Administrator, za pośrednictwem systemu, ma możliwość tworzenia kont dla uczniów, nauczycieli oraz innych pracowników administracyjnych, a także przedłużania ważności kont uczniów. System umożliwia mu także pomoc w zarządzaniu przedmiotem poprzez jego rejestrację po utworzeniu przez nauczyciela, a także dodawania i usuwania poszczególnych uczniów. Ponadto administrator może generować i wyświetlać raporty, ukazujące zaangażowanie uczniów w dany przedmiot oraz raporty ukazujące zaangażowanie nauczycieli.

Metodologia

Metodologia tworzenia systemu opiera się na utworzeniu aplikacji API w obiektowym języku programowania Java przy użyciu framework'a Spring Boot. Będzie ona pełniła rolę serwera, dostarczającego dane do interfejsu użytkownika (*front-end*). Takie podejście pozwoli na efektywne przetwarzanie danych i komunikację między *front-endem* a bazą danych, zapewniając płynne działanie projektu.

W kontekście tworzenia warstwy *front-end* wykorzystano metodologię *Single Page Application* (SPA). Opiera się ona na tworzeniu interaktywnych stron internetowych, które ładują się raz i pozwalają na płynne przejście między różnymi widokami bez konieczności przeładowywania całej strony.

Baza danych

MySQL jest jednym z najpopularniejszych systemów zarządzania bazą danych, charakteryzującym się nie tylko wydajnością, ale również stabilnością i niezawodnością. MySQL został wybrany ze względu na swoją niezawodność i wydajność w obszarze zarządzania bazą danych. W projekcie LMS istotne jest skuteczne zarządzanie danymi uczestników, przedmiotów i ocen, co MySQL zapewnia.

Budowa API

1. Wybór języka programowania Java:

Java jest językiem programowania, który działa na wielu platformach, co oznacza, że kod napisany w Javie jest przenośny. To znaczy, że można go uruchamiać na różnych systemach operacyjnych bez konieczności dostosowywania kodu. Dla projektu LMS oznacza to większą elastyczność i możliwość obsługi różnych środowisk. Java jest znana z wysokiego poziomu bezpieczeństwa. Ma wbudowane mechanizmy bezpieczeństwa, które chronią przed atakami typu *SQL Injection* czy *Cross-Site Scripting* (XSS). Ponadto język ten oferuje szeroki zakres bibliotek i narzędzi, które znacznie przyspieszają rozwój aplikacji.

2. Wybór frameworka Spring:

Spring jest znany ze swoich zaawansowanych funkcji, które ułatwiają tworzenie aplikacji. Posiada narzędzia do zarządzania transakcjami, dostępu do baz danych, kontroli dostępu oraz wiele innych. Jest to framework modułowy, co oznacza, że można wybierać tylko te komponenty, które są potrzebne do danego projektu. To pozwala na elastyczność i unikanie nadmiaru zbędnych funkcji. Jednym z modułów tego frameworka jest Spring Security. Zapewnia on zaawansowane mechanizmy zabezpieczeń, co jest kluczowe w projekcie edukacyjnym, gdzie dane uczniów muszą być chronione. Spring jest często wybierany do budowy aplikacji dostarczających API.

Budowa front-end

Całość podejścia opiera się na efektywnym projektowaniu i implementacji interfejsu użytkownika. W tym celu wykorzystano technologię języka znaczników HTML, Tailwind CSS oraz framework React.

1. Wybór HTML

HTML jest niezbędny w tworzeniu struktury strony internetowej. W połączeniu z React, umożliwia tworzenie czytelnych i semantycznych interfejsów, które są przyjazne dla użytkownika i dostosowane do różnych urządzeń.

2. Wybór Tailwind CSS

Wybór Tailwind CSS jest uzasadniony jego prostotą użycia i możliwością szybkiego projektowania interfejsu użytkownika. Gotowe klasy stylów pozwalają na uniknięcie potrzeby pisania niestandardowych stylów CSS, co przyspiesza rozwój projektu. Dodatkowo, stosowanie jednolitych stylów jest ważne dla spójnego wyglądu i wrażenia użytkownika w projekcie LMS.

3. Wybór framework'u React

React jest wybierany ze względu na jego zdolność do tworzenia dynamicznych i interaktywnych interfejsów użytkownika. Jest to kluczowe, gdyż projekt LMS musi zapewniać użytkownikom płynne doświadczenie podczas nauki online. React pozwala na łatwe zarządzanie komponentami interfejsu i reakcję na akcje użytkowników, co przyczynia się do poprawy jakości projektu.

Projekt systemu

Architektura systemu

Architektura systemu do zarządzania nauką *online* opiera się na architekturze wielowarstwowej, która jest kluczowym elementem zapewnienia wydajności, skalowalności i utrzymania projektu.

Architektura wielowarstwowa to struktura, w której różne komponenty i funkcje systemu są podzielone na odrębne warstwy, z każdą z nich spełniającą określone zadania. W przypadku projektu LMS, architektura wielowarstwowa składa się z następujących elementów:

1. Warstwa prezentacji (*front-end*)

Warstwa prezentacji to pierwszy kontakt użytkownika z systemem. Znajduje się tutaj interfejs użytkownika, który jest tworzony przy użyciu technologii *React* oraz stylizowany za pomocą *Tailwind CSS*. Jest to warstwa, która jest widoczna dla użytkownika i odpowiada za wyświetlanie treści oraz interakcję z użytkownikiem. Pozwala to na niezależne zarządzanie wyglądem i zachowaniem interfejsu użytkownika. *React* jako narzędzie do tworzenia interfejsu pozwala na efektywne tworzenie komponentów UI, a *Tailwind CSS* ułatwia stylizację i zapewnienie spójnego wyglądu.

2. Warstwa serwerowa (*back-end*)

Warstwa serwerowa to serwer oparty na *Java Spring*, który dostarcza API (interfejs programowania aplikacji) do komunikacji między front-endem a bazą danych. Odpowiada za obsługę żądań klientów, przetwarzanie tych żądań i zwracanie odpowiednich danych. Jest to centralny element projektu, który pośredniczy w komunikacji między front-endem a bazą danych. Warstwa ta umożliwia efektywną obsługę różnorodnych operacji, takich jak logowanie, przeglądanie kursów czy zarządzanie danymi użytkowników.

3. Baza danych

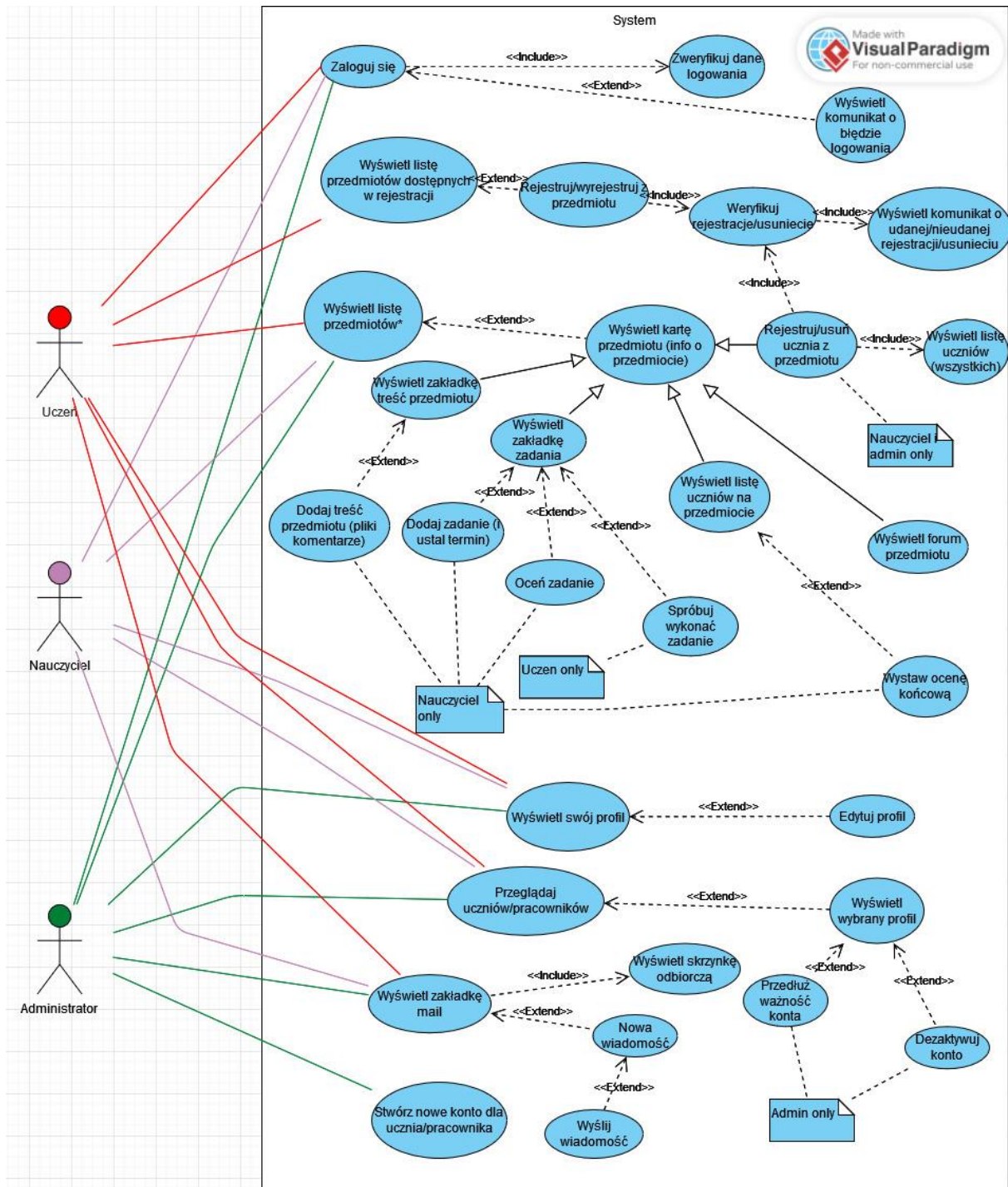
Baza danych MySQL jest wykorzystywana do przechowywania danych projektu, takich jak informacje o użytkownikach, przedmiotach, ocenach i innych istotnych informacjach. Baza danych zapewnia trwałe i bezpieczne przechowywanie danych.

Procedury składowane są wykorzystywane do ułatwienia wykonywania operacji CRUD (*Create, Read, Update, Delete*) na tabelach bazy danych. Dla każdej tabeli, z wyjątkiem tych zawierających stałe dane, utworzono procedury składowane, które pozwalają na bezpieczne zarządzanie danymi. Procedury te mogą być wywoływane z poziomu aplikacji, co zapewnia kontrolę nad dostępem do danych i ich integralnością.

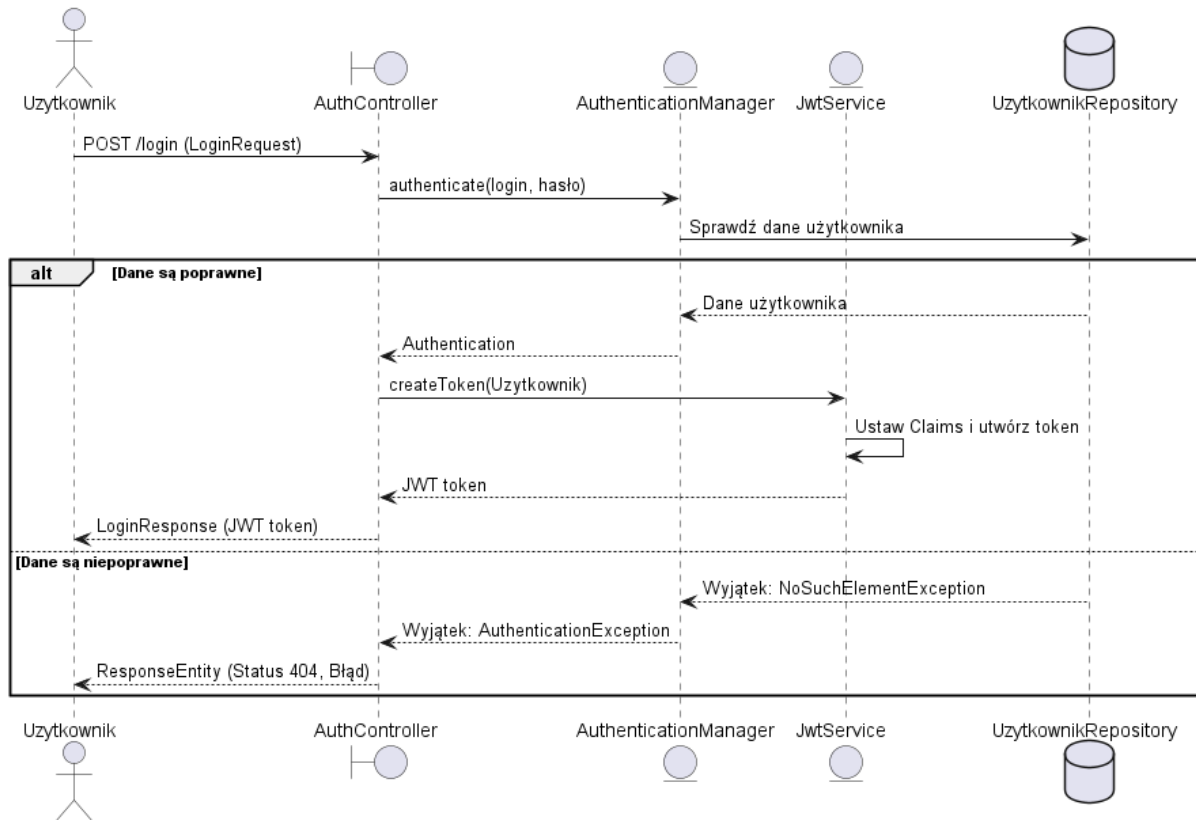
W celu walidacji i utrzymaniu spójności danych utworzono odpowiednie klucze unikalne, a także wykorzystano check constraints, które pozwalają na określenie warunków, które dane muszą spełnić. W tym celu również wykorzystywane są triggerzy, które także pełnią rolę generowania powiadomień.

Diagramy UML

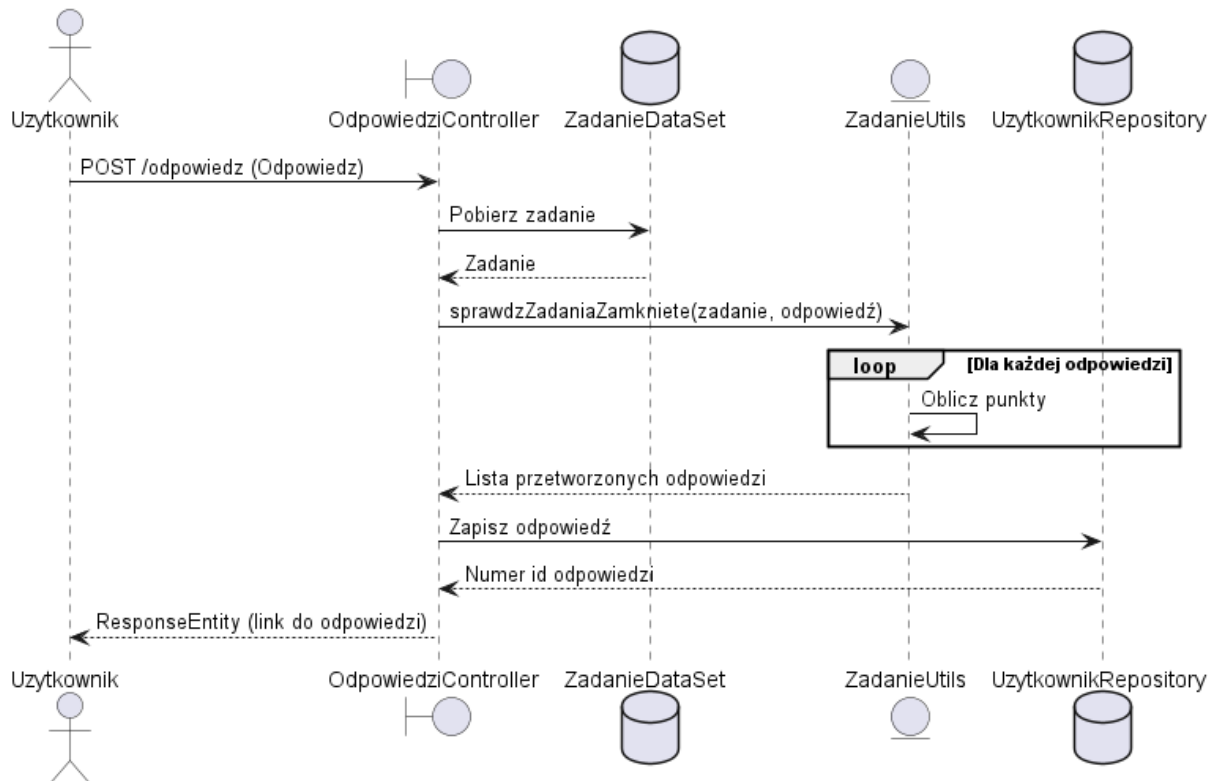
Diagram przypadków użycia



Diagramy sekwencji

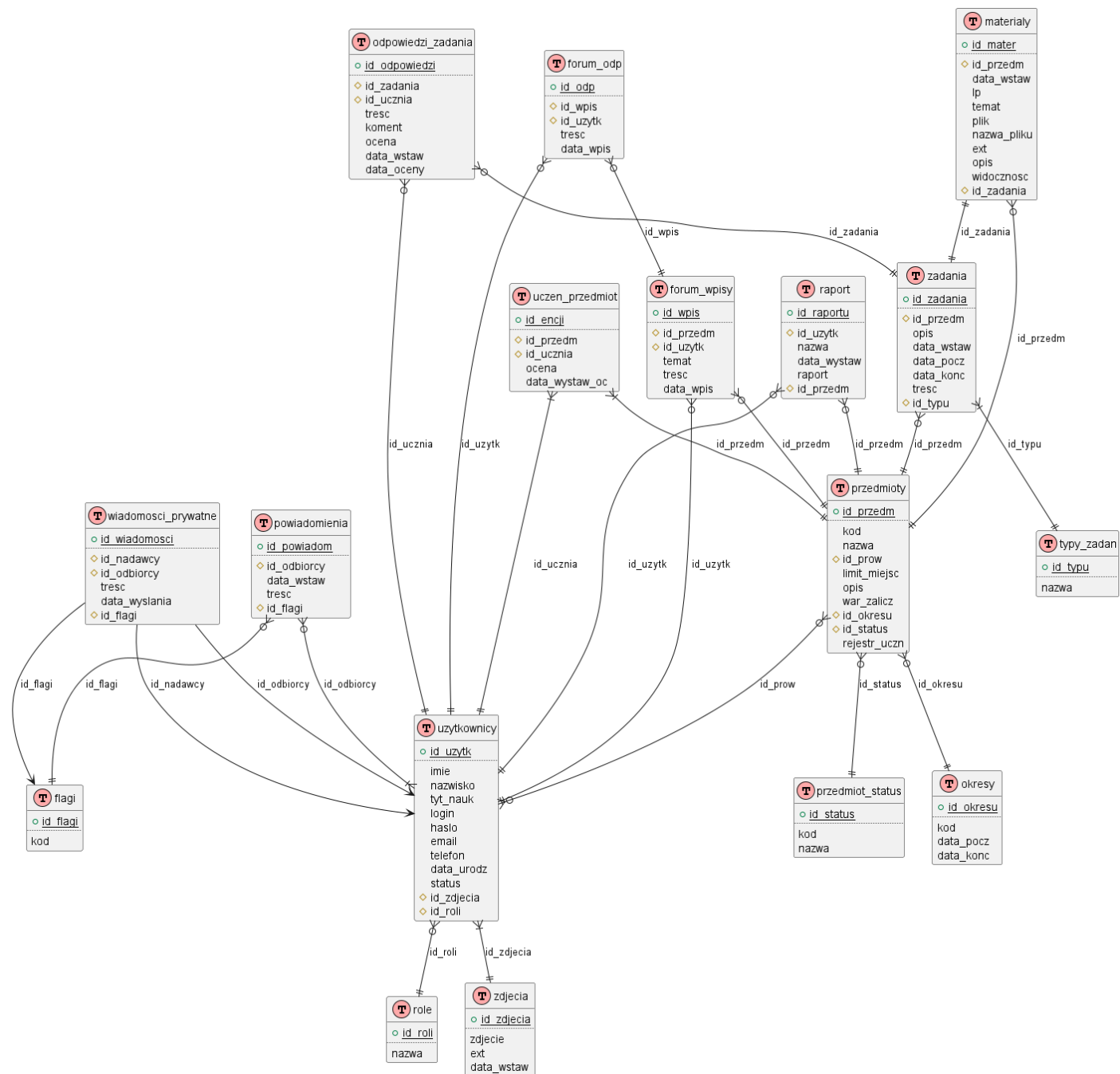


Rysunek 1 Logowanie użytkownika



Rysunek 2 Automatyczne sprawdzanie zadań

Schemat bazy danych



Wdrożenie

Wdrożenie projektu LMS może zostać podzielony na dwie to kluczowy etap procesu tworzenia oprogramowania, który obejmuje implementację, testowanie oraz dostosowanie systemu do środowiska produkcyjnego.

Wdrożenie bazy danych:

1. Instalacja systemu zarządzania bazą danych: Na serwerze produkcyjnym instalujemy system zarządzania bazą danych MySQL oraz narzędzie do zarządzania bazą danych.
2. Importowanie struktury i danych: Na serwerze produkcyjnym importujemy strukturę bazy danych oraz dane początkowe, jeśli są dostępne. Możemy to zrobić za pomocą narzędzi eksportu i importu dostępnych w systemie zarządzania bazą danych.
3. Konfiguracja dostępu: Upewniamy się, że baza danych jest dostępna dla aplikacji, a dane uwierzytelniające są skonfigurowane w aplikacji.
4. Testowanie i monitorowanie: Po wdrożeniu bazy danych, przeprowadzamy testy, aby upewnić się, że wszystko działa poprawnie. Regularnie monitorujemy działanie bazy danych, aby reagować na ewentualne problemy.

Wdrożenie *back-end*

Wdrożenie *back-endu* wymaga starannego przygotowania i konfiguracji środowiska produkcyjnego, aby zapewnić niezawodność i wydajność działania aplikacji.

Wybrane technologie

Projekt LMS wykorzystuje technologie Java 17 i Spring Boot 3.2.0. Java 17 oferuje najnowsze ulepszenia języka, zapewniając lepszą wydajność i bezpieczeństwo. Spring Boot 3.2.0 natomiast ułatwia tworzenie aplikacji poprzez automatyczną konfigurację i wsparcie dla mikro serwisów. Dzięki wykorzystaniu narzędzia Apache Maven możliwe jest automatyczne budowanie oprogramowania wraz z automatyczną aktualizacją zależności i bibliotek.

Konfiguracja środowiska

1. Instalacja Java Runtime Environment (JRE): Na serwerze produkcyjnym należy zainstalować odpowiednią wersję JRE zgodną z Java 17. JRE zapewnia jednolite środowisko uruchomieniowe, które jest niezbędne do uruchomienia aplikacji Java.
2. Konfiguracja pliku `application.properties`: W pliku konfiguracyjnym `application.properties` określamy wszystkie niezbędne ustawienia, takie jak konfiguracja bazy danych, zabezpieczeń i

inne parametry specyficzne dla środowiska produkcyjnego. To pozwala na elastyczne dostosowanie aplikacji do konkretnych wymagań i warunków produkcji.

Wdrożenie *front-end*

W procesie wdrożenia front-endu aplikacji React do projektu LMS skoncentrowaliśmy się na kilku kluczowych aspektach. Rozpoczęliśmy od instalacji środowiska Node.js i narzędzia npm na serwerze, aby umożliwić uruchomienie aplikacji React. Następnie pobraliśmy kod źródłowy front-endu i zainstalowaliśmy wszystkie zależności zdefiniowane w pliku package.json.

Kluczowym etapem było wygenerowanie zoptymalizowanej wersji aplikacji React za pomocą polecenia npm run build. Utworzona w ten sposób wersja była gotowa do dostarczenia na serwerze HTTP. Skonfigurowaliśmy serwer HTTP tak, aby obsługiwał żądania kierując je do zbudowanej aplikacji React oraz dostarczyliśmy zawartość katalogu build do katalogu publicznego na serwerze.

Warto zaznaczyć, że dokładnie monitorowaliśmy działanie aplikacji po wdrożeniu, aby reagować na ewentualne problemy. Jako część procesu wdrożenia, zapewniliśmy również odpowiednie zabezpieczenia, takie jak nagłówki bezpieczeństwa HTTP, aby chronić aplikację przed potencjalnymi zagrożeniami.

Testowanie i walidacja

Szczegóły procesu testowania:

Testy jednostkowe

Podczas rozwoju oprogramowania były przeprowadzane testy jednostkowe, które sprawdzają, czy poszczególne komponenty i moduły działają poprawnie w izolacji od reszty systemu.

Testy integracyjne

Testy integracyjne były wykonywane w miarę rozwijania oprogramowania, zwłaszcza w momencie łączenia front-endu i back-endu. Ich celem było upewnienie się, że różne części systemu współpracują ze sobą poprawnie.

Przypadki testowe i scenariusze

Przypadki testowe zostały stworzone, aby przetestować różne funkcje i aspekty projektu LMS. Były one adekwatne do istniejących punktów końcowych w aplikacji API. Scenariusze testowe obejmują m.in.:

- Rejestrację nowego użytkownika.
- Logowanie do systemu.
- Tworzenie i zarządzanie kursami.
- Wprowadzanie danych uczniów i nauczycieli.
- Wykonywanie testów i zadań.
- Komunikację między uczniami i nauczycielami.

Przykłady przypadków testowych i scenariuszy dla trzech różnych żądań HTTP:

Przypadek testowy 1: Logowanie użytkownika

Scenariusz:

1. Wysłanie żądania POST do `/api/v1/auth/login` z następującym ciałem:

```
{
  "login": "j.doe",
  "haslo": "haslo"
}
```
2. Spodziewany wynik: Użytkownik zostaje uwierzytelniony i otrzymuje token dostępu.

Przypadek testowy 2: Dodawanie przedmiotu

Scenariusz:

1. Wysłanie żądania POST do `/api/przedmiot` z następującym ciałem i ważnym tokenem autoryzacyjnym:


```
{
  "nazwa": "Matematyka",
  "idProwadzacego": 123,
  "limit": 30,
  "opis": "Przedmiot z zakresu matematyki",
  "warunkiZaliczenia": "Egzamin końcowy",
  "idOkresu": 1,
  "status": "DO_ZATWIERDZENIA",
  "czyRejestrUczn": true
}
```

2. Spodziewany wynik: Przedmiot zostaje dodany do systemu.

Przypadek testowy 3: Dodawanie materiału do przedmiotu

Scenariusz:

1. Wysłanie żądania POST do `/api/przedmiot/material` z następującym ciałem i ważnym tokenem autoryzacyjnym:

```
{
  "idPrzedmiotu": 1,
  "temat": "Nowy temat",
  "plik": "<Zawartość pliku zakodowana w base64>",
  "nazwaPliku": "nazwa_pliku ",
  "ext": "txt",
  "opis": "Opis materiału",
  "idZadania": 2,
  "widocznosc": 1
}
```

2. Spodziewany wynik: Materiał zostaje dodany do przedmiotu.

Wyniki badań i procedury walidacyjne:

Wyniki testów były dokładnie dokumentowane. Podczas testowania wykryto różne rodzaje błędów, w tym:

- **Błędy wprowadzanych danych:** Błędy wynikające z nieprawidłowych danych wejściowych, które były weryfikowane i poprawiane w procesie walidacji.
- **Błędy istniejących danych:** Wykrywanie niezgodności w istniejących danych i procedurach ich naprawiania.
- **Błędy mechanizmów po stronie *back-end*:** Błędy w logice biznesowej, które były identyfikowane, debugowane i poprawiane przez zespół deweloperów.
- **Błędy po stronie mechanizmów *front-end*:** Problemy z interfejsem użytkownika, wyglądem i funkcjonalnością, które były rozwiązywane w trakcie testowania.

Rozwiązywanie problemów i błędów

W przypadku wykrycia błędów, procedura ich rozwiązywania obejmowała:

- Dokładną identyfikację i dokumentację błędów.

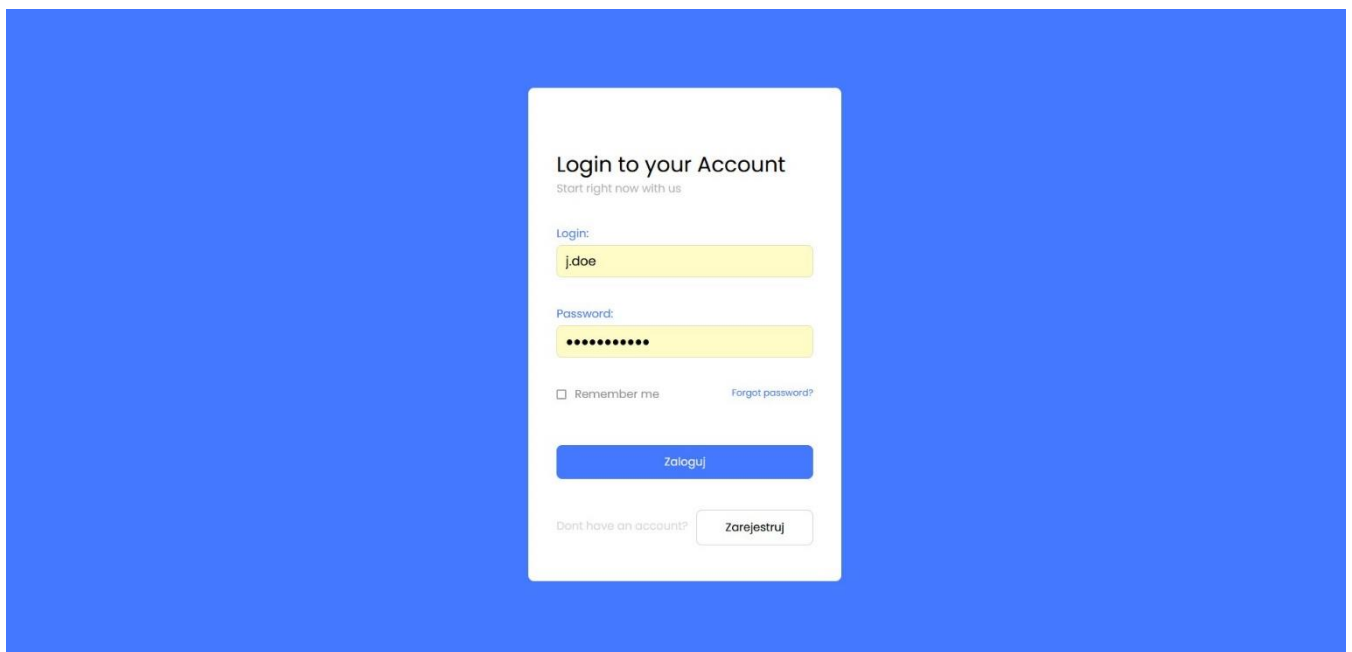
- Przydzielanie odpowiednich priorytetów naprawy w zależności od ich znaczenia i wpływu na działanie systemu.
- Debugowanie i refaktoryzację kodu w celu usunięcia błędów.
- Weryfikację poprawności napraw i powtarzanie testów, aby upewnić się, że błędy zostały skorygowane.

Wyniki i dyskusja

Projekt LMS wprowadza szereg funkcjonalności, które wspierają zarządzanie procesem nauki online.

Poniżej opisano kluczowe komponenty systemu:

Panel logowania:

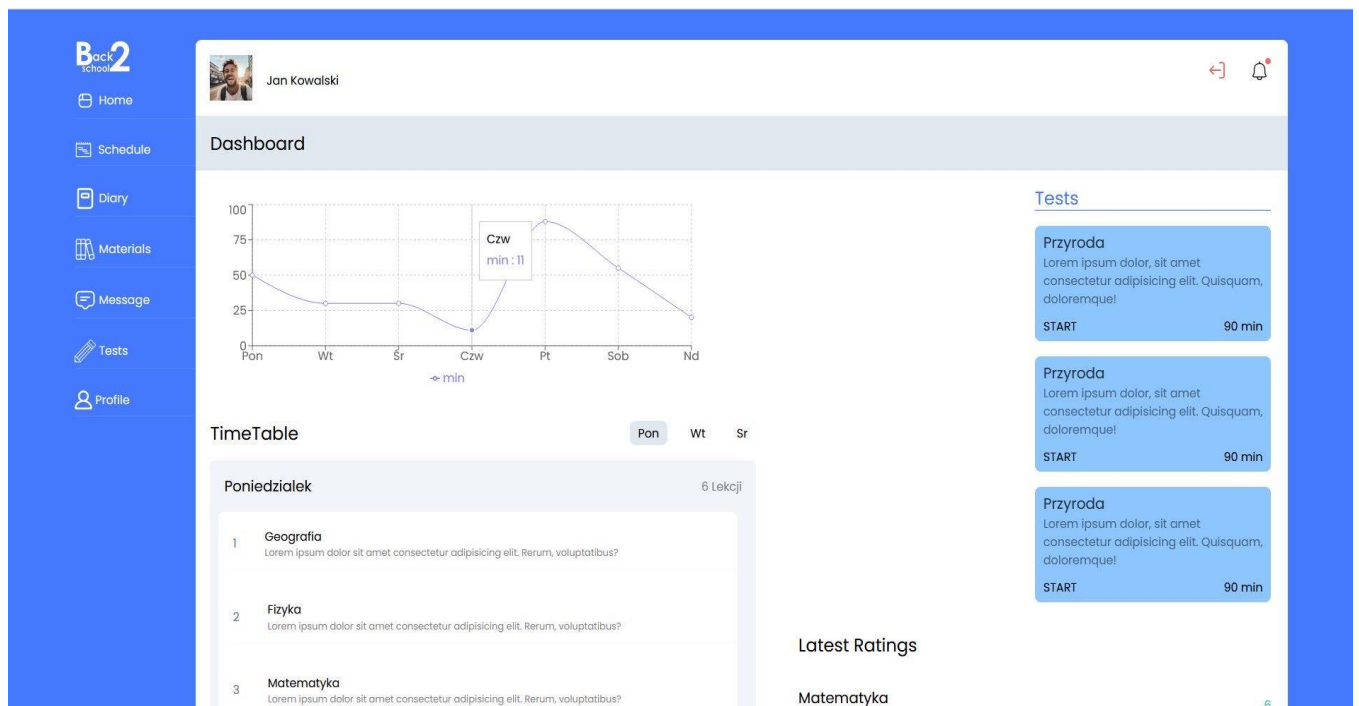


The screenshot shows a login interface with a white card centered on a solid blue background. The card contains the following elements:

- Title:** "Login to your Account" in bold black text, followed by the subtitle "Start right now with us" in a smaller, lighter font.
- Login Field:** A label "Login:" above a yellow input field containing the text "j.doe".
- Password Field:** A label "Password:" above a yellow input field filled with black dots.
- Remember Me:** A checkbox labeled "Remember me".
- Forgot Password:** A link labeled "Forgot password?" in blue text.
- Login Button:** A blue button with the text "Zaloguj" in white.
- Registration Link:** A link labeled "Dont have an account?" in gray text, followed by a button labeled "Zarejestruj" in gray text.

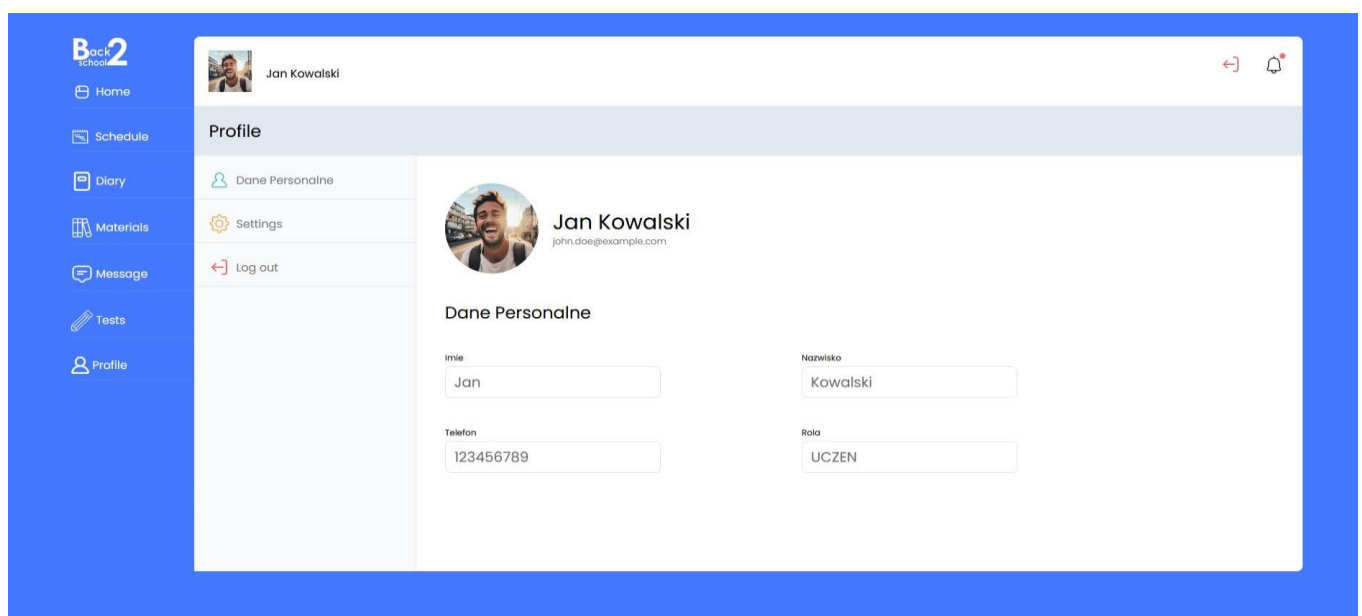
Zaimplementowany panel logowania umożliwia użytkownikom rejestrację oraz logowanie się do systemu.

Strona główna:



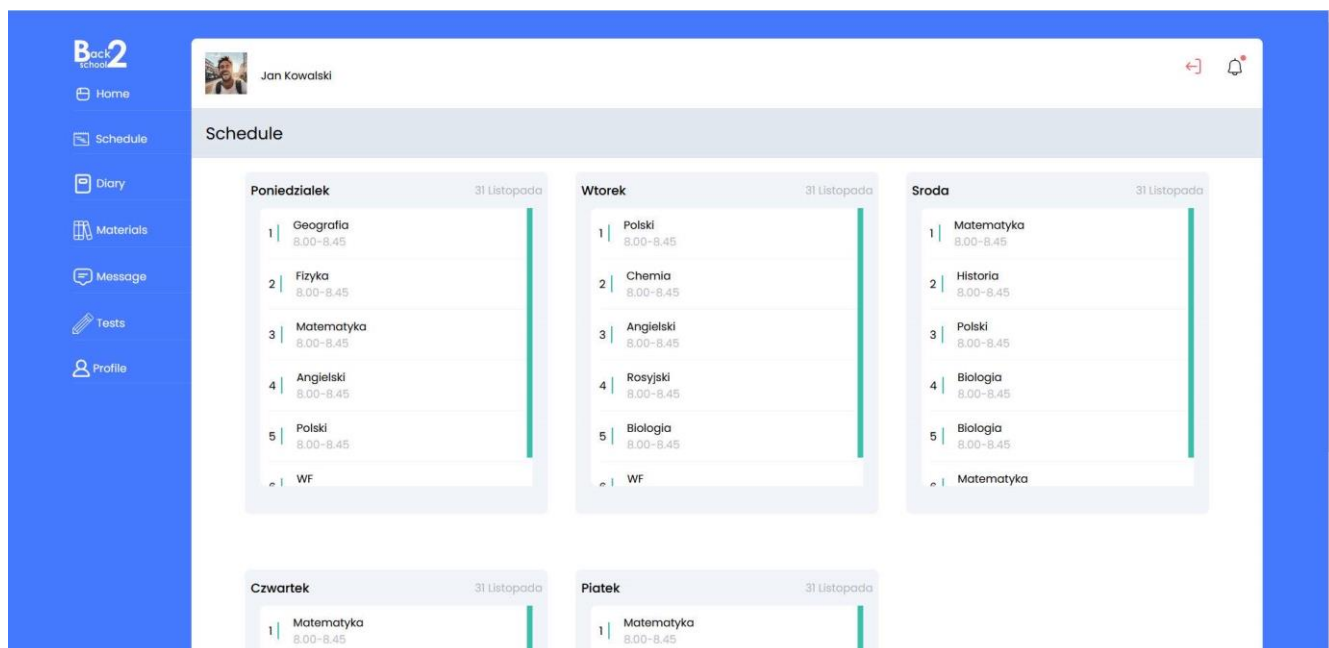
Centralny punkt nawigacyjny systemu, prezentujący użytkownikom dostępne funkcje oraz aktualności związane z kursami.

Panel profilu użytkownika:



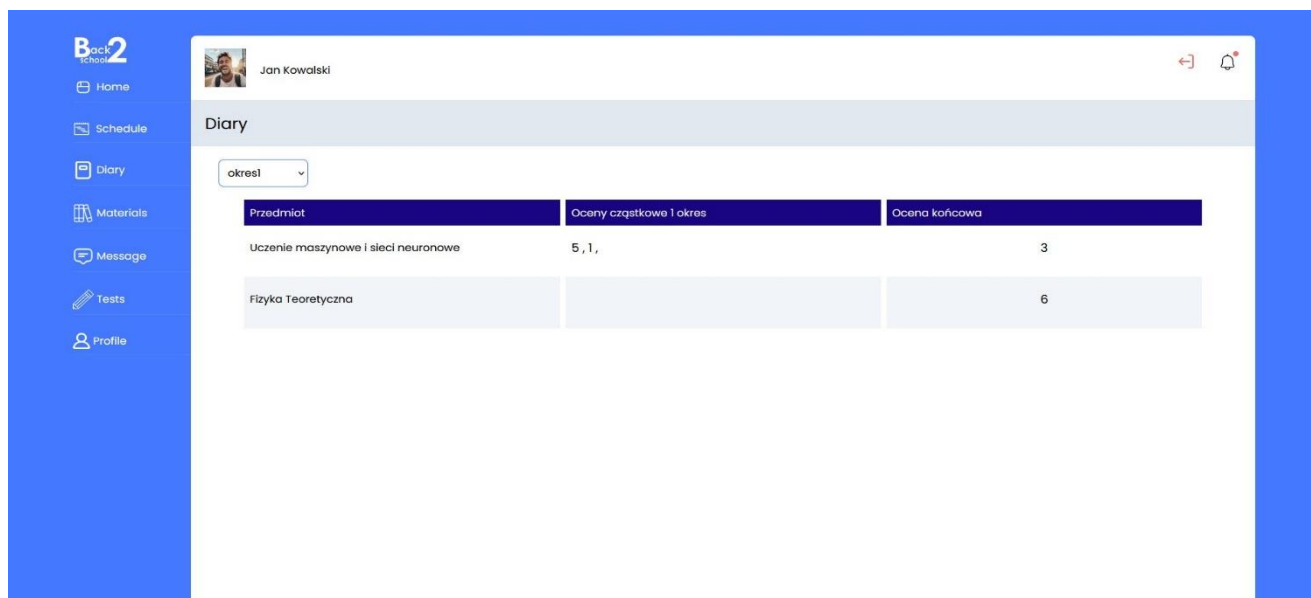
Umożliwia edycję danych osobowych użytkowników, takich jak numer telefonu, adres email i zdjęcie profilowe.

Kalendarz:



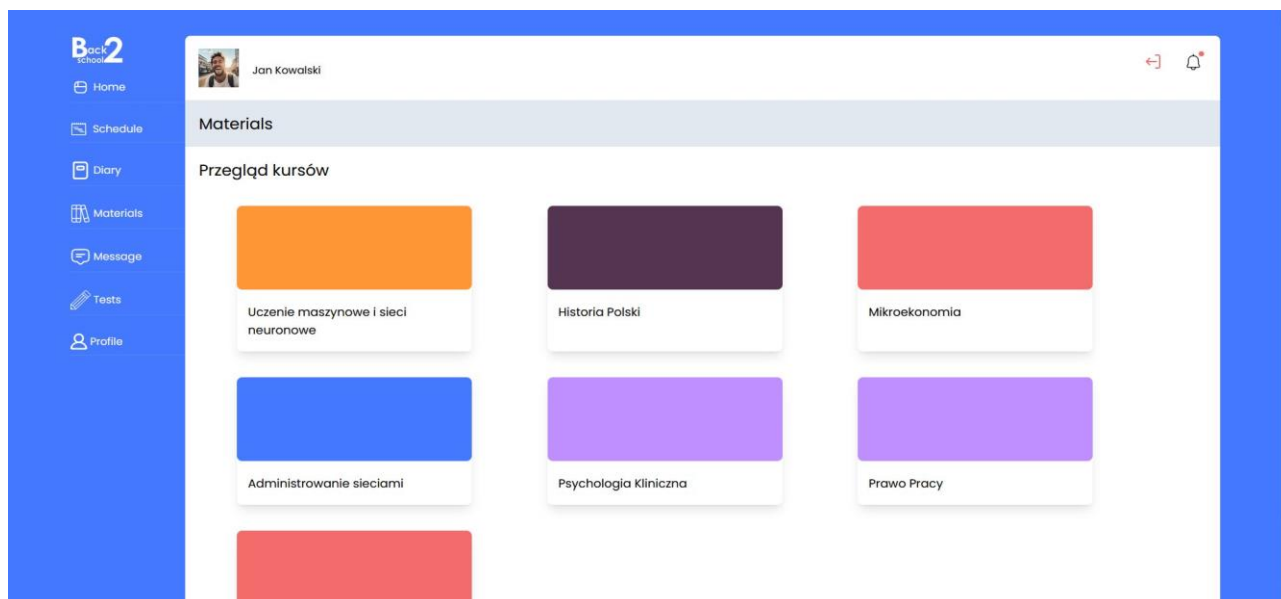
Zawiera informacje o nadchodzących zajęciach.

Panel ocen:



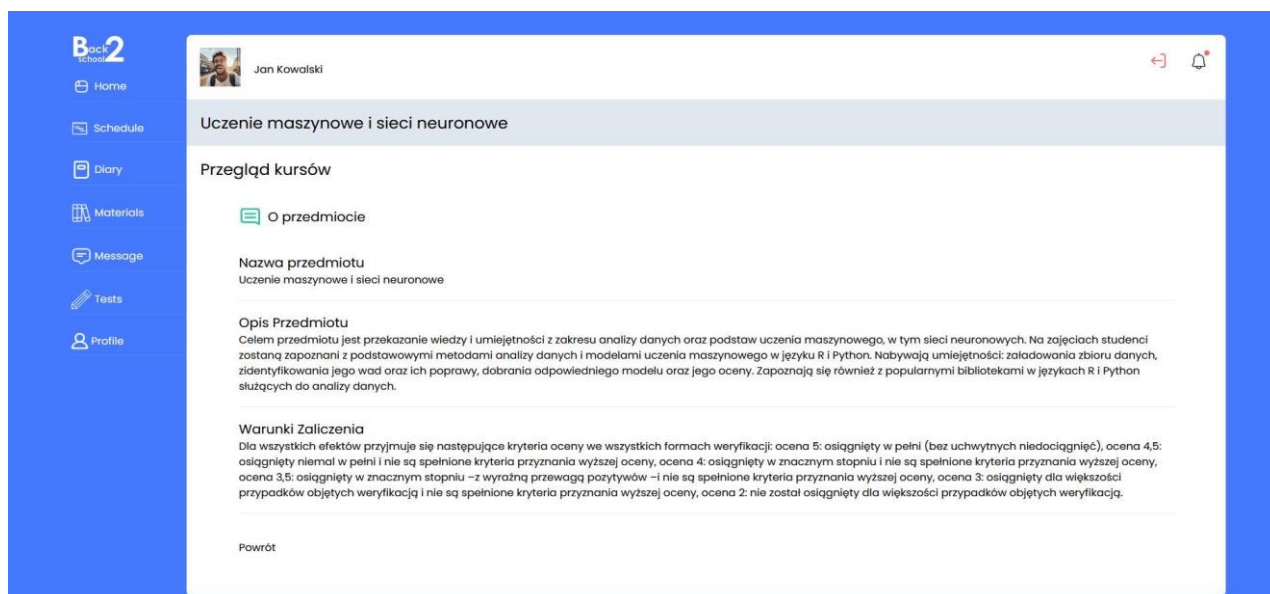
Pozwala uczniom na przegląd ocen, zapewniając dostęp do informacji o wynikach ich pracy.

Przegląd kursów:



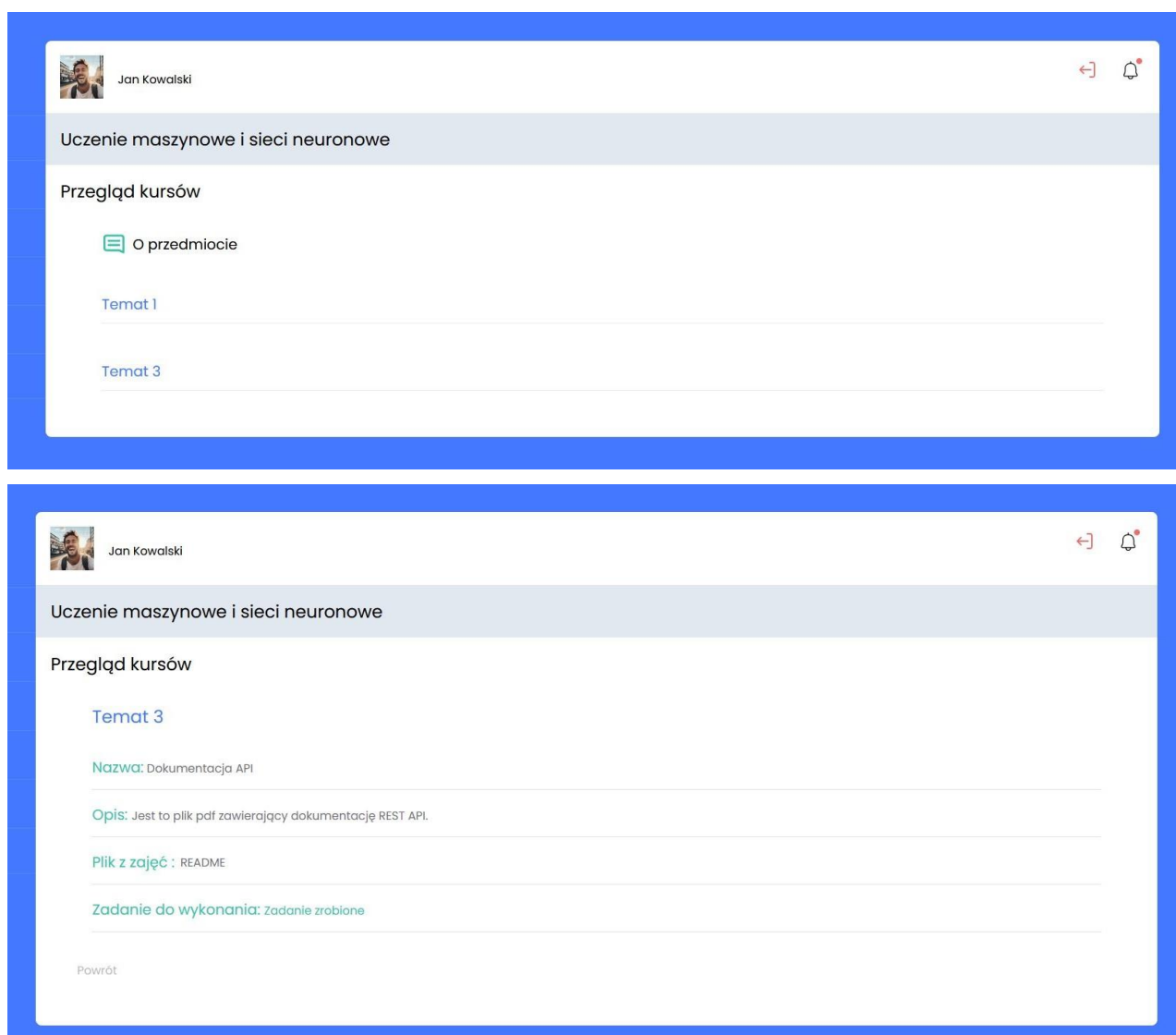
Sekcja ta przedstawia przegląd dostępnych kursów, umożliwiając wyszukiwanie i zapisywanie się na wybrane przedmioty.

Panel kursu:



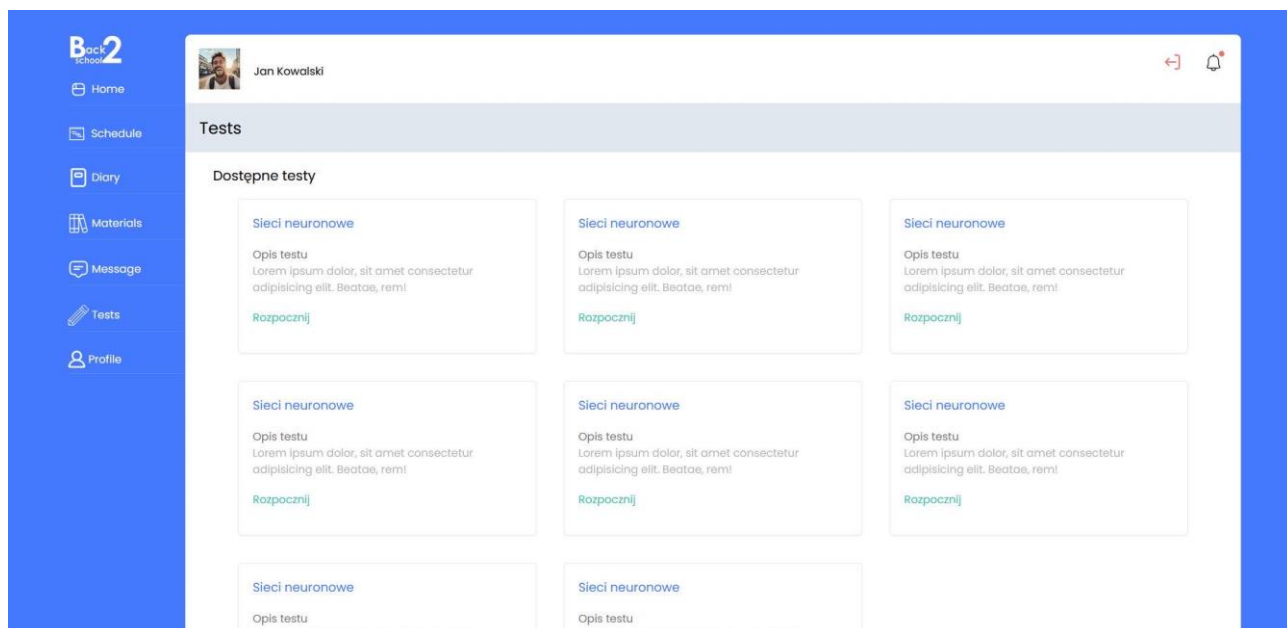
Zawiera szczegółowe informacje o kursie, w tym opis i warunki zaliczenia.

Panele tematów z materiałami:



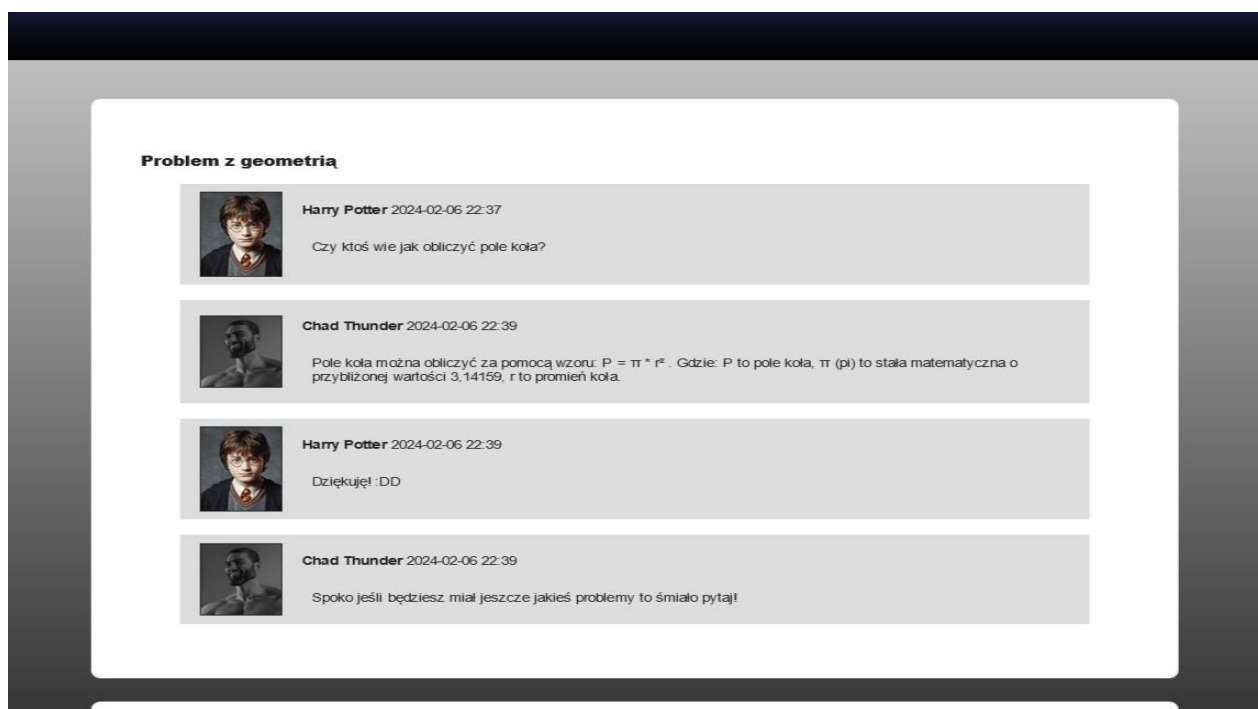
Umożliwiają organizację i przeglądanie materiałów dydaktycznych, upraszczając dodawanie nowych treści przez nauczycieli i dostęp do nich dla uczniów.

Panel zadań z kursu:



Nauczyciele mogą tworzyć zadania różnego typu, a uczniowie – rozwiązywać je. Umożliwia ocenianie wykonanych prac przez nauczycieli.

Fora dyskusyjne:



Dla każdego kursu dostępny jest forum, na którym nauczyciele i uczniowie mogą prowadzić dyskusje.

Analiza wyników w kontekście celów projektu

W ramach realizacji projektu systemu do zarządzania nauką online (LMS) udało się osiągnąć szereg zaplanowanych celów:

- **Stworzenie interfejsów użytkowników:** Opracowaliśmy interfejsy dla użytkowników, co ułatwia nawigację i korzystanie z systemu.
- **Wdrożenie rejestracji użytkowników:** Implementacja funkcji rejestracji umożliwia łatwe i bezpieczne tworzenie nowych kont.
- **Stworzenie systemu zarządzania przedmiotami:** Opracowaliśmy system zarządzania kursami, umożliwiający nauczycielom tworzenie i edytowanie kursów, a uczniom zapisywanie się na interesujące ich przedmioty oraz wykonywanie zadań w ramach przedmiotów.
- **Wprowadzenie for dla danych przedmiotów:** Jako uzupełnienie systemu, wprowadziliśmy fora dyskusyjne dla poszczególnych przedmiotów, umożliwiające wymianę wiedzy, dyskusje oraz współpracę między uczniami i nauczycielami na tematy związane z kursem.
- **Stworzenie generowania raportów:** Funkcjonalność generowania raportów pozwala na bieżąco monitorować postępy uczniów.
- **Częściowe stworzenie systemu przesyłania wiadomości:** System przesyłania wiadomości ułatwia komunikację między wszystkimi użytkownikami systemu.
Uwagi: brak zrealizowania systemu na front-endzie.

Omówienie wszelkich ograniczeń i obszarów wymagających poprawy

W projekcie zidentyfikowano następujące obszary wymagające dalszych prac i usprawnień:

- **Brak realizacji niektórych funkcji na front-endzie:** niektóre planowane funkcjonalności, takie jak wiadomości prywatne oraz dedykowane panele dla nauczycieli i administratorów, nie zostały jeszcze w pełni zaimplementowane.
- **Brak wdrożenia LMS na serwerze internetowym:** Aby system był w pełni funkcjonalny i dostępny dla użytkowników, konieczne jest jego wdrożenie na serwerze internetowym.

Przyszła praca

W oparciu o zidentyfikowane obszary wymagające usprawnień, proponujemy następujące kierunki przyszłych prac nad systemem LMS:

- **Rozwój funkcjonalności front-endu:** Priorytetem jest dokończenie implementacji brakujących funkcji, w szczególności wiadomości prywatnych, systemu powiadomień, oraz rozbudowa paneli dla nauczycieli i administratorów.
- **Wdrożenie systemu na serwerze internetowym:** Aby umożliwić szeroki dostęp do systemu, konieczne jest jego wdrożenie na serwerze internetowym, co również umożliwi przeprowadzenie testów w realnych warunkach użytkowania.
- **Stworzenie możliwości komunikacji poprzez wysyłanie wiadomości i powiadomień na skrzynkę pocztową:** Aby usprawnić komunikację w systemie, można rozwinąć funkcjonalność, która pozwala na wysyłanie wiadomości i powiadomień bezpośrednio do skrzynki pocztowej użytkownika.
Dalsze prace nad projektem powinny skupić się na rozwiązywaniu tych problemów, aby zapewnić użytkownikom bezpieczny, efektywny i kompleksowy system nauki online

Referencje

https://en.wikipedia.org/wiki/Learning_management_system

<https://en.wikipedia.org/wiki/REST>

https://pl.wikipedia.org/wiki/Spring_Framework

<https://spring.io/>

[https://en.wikipedia.org/wiki/React_\(software\)](https://en.wikipedia.org/wiki/React_(software))

<https://react.dev/>

https://en.wikipedia.org/wiki/Tailwind_CSS

<https://tailwindcss.com>

<https://en.wikipedia.org/wiki/MySQL>

https://en.wikipedia.org/wiki/Create,_read,_update_and_delete

Załączniki

1. Repozytorium: <https://github.com/amrusb/lms-pz>
2. Dokumentacja API: <https://github.com/amrusb/lms-pz/blob/master/lms-api/README.md>