



Praca dyplomowa inżynierska

na kierunku Informatyka

Aplikacja do nauki języków z napisów do
filmów udostępnianych na platformie YouTube

Paweł Niziołek

numer albumu 102488

promotor

dr hab. inż. Szczepan Paszkiel

Opole, 2024

Aplikacja do nauki języków z napisów do filmów udostępnianych na platformie YouTube Streszczenie

Praca dyplomowa skupia się na projektowaniu i realizacji szablonu do tworzenia pracy dyplomowej przy użyciu składu tekstu \LaTeX . W pracy zostały przedstawione korzyści korzystania z \LaTeX a w porównaniu do innych edytorów tekstu skupiając się na takich rzeczach jak automatyczne formatowanie, obsługa bibliografii, profesjonalny wygląd i łatwa edycja struktury dokumentu. Teoretyczna część pracy jest poświęcona przedstawieniu krótkiej historii powstawania \LaTeX a oraz korzyści płynących z korzystania z \LaTeX a przy tworzeniu profesjonalnie wyglądających i automatycznie formatujących się prac.

W praktycznej części pracy przedstawiono stworzony szablon z przedstawieniem jego funkcji oraz możliwością spersonalizowania go do indywidualnych potrzeb użytkownika. Opisano zostało również jego proces instalacji i konfiguracji tak aby ułatwić korzystanie z niego innym użytkownikom.

Słowa kluczowe: Skład tekstu \LaTeX , Praca Dyplomowa, Narzędzia i środowiska pracy

Aplikacja do nauki języków z napisów do filmów udostępnianych na platformie YouTube Abstract

The thesis focuses on the design and implementation of a template for creating a thesis using \LaTeX text composition. The thesis presents the advantages of using \LaTeX a in addition to other text editors focusing on such things as automatic formatting, bibliography support, professional appearance and easy editing of the document structure. The theoretical part of the work is devoted to presenting a brief history of the creation of \LaTeX a and the benefits of using \LaTeX a to create professional-looking and automatically formatting works.

The practical part of the work presents a ready-made template with a demonstration of its functions and the possibility of personalizing it to individual user needs. Its installation and configuration process is also described so as to facilitate its use by other users.

Keywords: Text Composition \LaTeX , Dissertation, Tools and working environments.

Spis treści

1	Wprowadzenie	5
1.1	Wstęp	5
1.2	Przegląd aktualnych rozwiązań	7
1.2.1	Anki	7
1.2.2	Language Reactor	8
1.2.3	Trancy	9
2	Cel i zakres pracy	11
2.1	Cel pracy	11
2.2	Zakres pracy	12
3	Projekt	13
3.1	Wprowadzenie	13
3.2	interfejs użytkownika	13
3.2.1	Widok startowy	13
3.2.2	Strona Nauki	13
3.2.3	Strona Profilu	14
3.2.4	Strona Słownika	14
3.2.5	Strona Statystyk	14
3.2.6	Strona logowania oraz rejestracji	14
3.2.7	Strona oglądania filmów	14
4	Implementacja	15
4.1	Wprowadzenie	15
4.2	Środowisko i narzędzia programistyczne	15
4.2.1	Środowisko programistyczne	15
4.2.2	Wybór technologii	15
4.2.3	Opis technologii	16
4.3	Baza danych	17
4.4	Backend	17
4.5	Frontend	17
4.6	Testy	17

4.7	Problemy implementacyjne	17
4.7.1	Wirtualizacja List	17
4.7.2	Przetwarzanie języka naturalnego (NLP)	19
5	Podsumowanie	20
5.1	Wnioski	20
5.2	Kierunki dalszego rozwoju	20
	Bibliografia	20
	Spis rysunków	22
	Spis tabel	23

Rozdział 1

Wprowadzenie

1.1 Wstęp

W dzisiejszym globalnym świecie znajomość języków obcych jest nieocenioną umiejętnością. Tradycyjne metody nauki, takie jak lekcje w klasach, samouczki i aplikacje do nauki słownictwa, często są czasochłonne i nie zawsze dostosowane do indywidualnych potrzeb ucznia. Dodatkowo, oglądanie filmów i seriali w obcym języku jest powszechnie uznawane za skuteczny sposób na poprawę umiejętności językowych, ale brakuje narzędzi, które integrują te aktywności z formalnym procesem nauki. Niniejsza praca inżynierska koncentruje się na stworzeniu innowacyjnej aplikacji webowej, która połączy te dwa aspekty, oferując użytkownikom skuteczniejsze i przyjemniejsze doświadczenie edukacyjne.

Współczesny świat, w którym żyjemy, jest coraz bardziej zglobalizowany i wymaga od nas umiejętności komunikacji w różnych językach, a przede wszystkim w języku angielskim, który stał się międzynarodowym językiem na świecie. Wraz z rozwojem technologii, nauka języków obcych stała się bardziej dostępna i atrakcyjna. . Znajomość języków obcych nie tylko otwiera drzwi do nowych możliwości zawodowych, ale także umożliwia pełniejsze zrozumienie innych kultur i poszerza horyzonty. Wraz z dynamicznym rozwojem technologii, nauka języków obcych stała się bardziej dostępna, a tradycyjne metody nauczania ewoluowały, oferując nowe, bardziej interaktywne formy edukacji. Jednym z najpopularniejszych sposobów nauki języków jest korzystanie z platform internetowych, takich jak duoLingo, gdzie użytkownicy mogą uczyć się od podstaw słów i zdań które zostały wcześniej przygotowane. Jednakże, nauka języka obcego w ten sposób ogranicza nas w kwestii wyboru czego chcielibyśmy się dokładnie uczyć.

Coraz więcej osób szuka alternatywnych metod nauki, które są bardziej angażujące i interaktywne. Filmy i seriale oferują naturalny kontekst, w którym używane są różne zwroty i słownictwo, co czyni je doskonałym narzędziem do nauki języka. Oglądanie treści w języku obcym nie tylko pomaga w nauce nowych słów i zwrotów, ale także w poprawie umiejętności słuchania i rozumienia języka w różnych akcentach i dialektach.

Aplikacja ta ma umożliwić użytkownikom aktywne uczestnictwo w procesie nauki, poprzez interaktywne narzędzia i funkcje, które wspomagają naukę słownictwa i gramatyki. Wśród nich znajdują się m.in. możliwość zapisu słów z listy napisów, które są wyświetlane pod lub obok odtwarzacza video, a także możliwość dodania ich do bazy danych, aby uniknąć powtórzeń baza nie przyjmie drugiego takiego samego słowa użytkownikowi. Użytkownik będzie miał dostęp do panelu nauki, słownika wszystkich słów, możliwości logowania z różnych urządzeń obsługujących przeglądarkę, a także do różnych sposobów prezentacji danych, takich jak słownik, flashcards czy moduł do edycji napisów.

Aplikacja ta ma również uwzględnić elementy gamifikacji, aby zachęcić użytkowników do nauki i śledzenia postępów. Na profilu użytkownika będą widoczne wszystkie nauczone słowa, a także osobna podstrona z wykresami i informacjami o postępach. Dzięki tej aplikacji, użytkownicy będą mogli efektywnie i atrakcyjnie uczyć się języka obcego, korzystając z platformy YouTube i własnych filmów z napisami z dysku własnego komputera. Napisy których użytkownik może użyć będą w różnych formatach, więc w aplikacji będzie można wybrać rodzaj pliku i przekopiować całą zawartość lub wrzucić plik w odpowiednie miejsce, napisy muszą zostać zapisane w systemie ponieważ nie ma możliwości zapisaniu ścieżki do żadnego pliku ze względów bezpieczeństwa w internecie.

Wybór technologii do tworzenia aplikacji webowej jest kluczowy dla jej stabilności, skalowalności i wydajności. W projekcie tej aplikacji językowej zdecydowano się na framework Next.js, który oparty jest na React i oferuje wiele korzyści. Jedną z głównych zalet Next.js jest możliwość elastycznego renderowania treści, zarówno po stronie serwera (SSR), jak i klienta (CSR). Dzięki SSR, aplikacja może szybko ładować wstępnie załadowane strony, co znacząco poprawia widoczność w wyszukiwarkach (SEO - Search Engine Optimization) i przyspiesza czas ładowania, co jest szczególnie istotne dla użytkowników korzystających z platformy edukacyjnej. CSR z kolei umożliwia dynamiczne i płynne aktualizacje interfejsu bez konieczności przeładowywania całej strony, co poprawia doświadczenie użytkownika.

Kolejną istotną zaletą Next.js jest intuicyjny i wydajny system routingu oparty na strukturze plików. Ułatwia to organizację aplikacji i nawigację po niej, co jest kluczowe dla zachowania przejrzystości i spójności struktury. Next.js oferuje również uproszczone pobieranie danych oraz wsparcie dla różnych metod stylizacji, takich jak moduły CSS i Tailwind CSS, co pozwala na tworzenie estetycznego i responsywnego interfejsu użytkownika szybciej i łatwiej. Dodatkowo, framework zapewnia wsparcie dla TypeScript, co umożliwia tworzenie bezpiecznego i stabilnego kodu przy użyciu typów które nam podkreślą jeśli będziemy próbowali błędnie używać naszych funkcji lub zmiennych. Wszystkie te cechy czynią Next.js idealnym wyborem do stworzenia nowoczesnej i wydajnej aplikacji językowej.

Wybór bazy danych do aplikacji webowej ma ogromne znaczenie dla jej wydajności i skalowalności. W tym projekcie zdecydowano się na MongoDB Atlas, która jest obiektową bazą danych typu NoSQL. MongoDB charakteryzuje się elastyczną strukturą danych, co pozwala na szybkie i efektywne przechowywanie oraz zarządzanie różnorodnymi danymi. Dzięki obiektowemu modelowi danych,

MongoDB doskonale nadaje się do aplikacji, które wymagają pracy z dynamicznie zmieniającymi się strukturami danych. Jest to istotne w kontekście naszej aplikacji językowej, ponieważ umożliwia łatwe przechowywanie słów i fraz w różnych formatach i językach, co jest kluczowe dla elastyczności i funkcjonalności aplikacji.

Jedną z kluczowych zalet MongoDB jest jej skalowalność. Baza ta umożliwia łatwe skalowanie poziome, co oznacza, że możemy dodawać nowe serwery do naszego klastra bazodanowego w miarę wzrostu ilości danych i liczby użytkowników. Jest to szczególnie ważne dla aplikacji edukacyjnych, które mogą szybko rosnąć w popularność i wymagać zwiększonej mocy obliczeniowej. Dzięki temu, nasza aplikacja będzie mogła obsługiwać rosnącą liczbę użytkowników bez utraty wydajności. Dodatkowo, MongoDB Atlas oferuje wsparcie dla replikacji danych, co zwiększa niezawodność i dostępność systemu. Funkcja replikacji zapewnia, że dane są automatycznie kopiowane na wiele serwerów, co chroni przed utratą danych i zapewnia ciągłość działania aplikacji. Dzięki tym funkcjom MongoDB Atlas jest idealnym wyborem dla naszej aplikacji, zapewniając jej wydajność, skalowalność i elastyczność w zarządzaniu danymi.

1.2 Przegląd aktualnych rozwiązań

1.2.1 Anki

Anki to popularna aplikacja edukacyjna oparta na systemie powtórek rozłożonych w czasie (Spaced Repetition System – SRS). Dzięki temu mechanizmowi nauka jest bardziej efektywna, ponieważ aplikacja prezentuje użytkownikowi informacje w odpowiednich odstępach czasowych, co pomaga w utrwaleniu materiału. Anki wyróżnia się uniwersalnością i możliwością dostosowania do różnych potrzeb, takich jak nauka języków obcych, przygotowanie do egzaminów czy zapamiętywanie faktów w innych dziedzinach.

Funkcjonalności:

- Możliwość ręcznego dodawania kart z różnymi typami treści, w tym tekstów, obrazów, dźwięków i nagrań wideo.
- Obsługa dodatków (pluginów), które rozszerzają możliwości aplikacji, np. importowanie napisów filmowych.
- Analiza postępów użytkownika z wykorzystaniem statystyk i wykresów.

Ograniczenia:

- Podczas oglądania filmu użytkownik musi ręcznie przerywać oglądanie, aby dodawać niezbędne informacje do kart. Utrudnia to płynność procesu i może obniżać komfort nauki.
- Mimo tych niedogodności Anki pozostaje dobrym rozwiązaniem do nauki języków, szczególnie dzięki możliwości personalizacji kart i śledzenia postępów.

1.2.2 Language Reactor

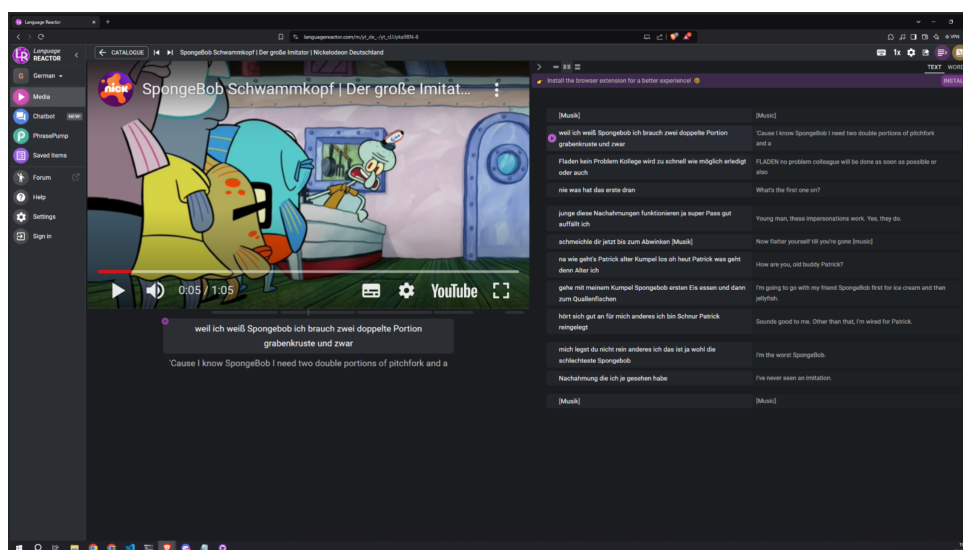
Language Reactor to narzędzie edukacyjne, które umożliwia naukę języków obcych w sposób przyjemny i efektywny poprzez oglądanie filmów i seriali. Aplikacja oferuje interaktywne napisy, które umożliwiają tłumaczenie słów i zwrotów bezpośrednio na ekranie. Dzięki tej funkcji użytkownicy mogą szybko sprawdzić znaczenie nowego słowa, klikając na nie podczas oglądania, lub oznaczyć jako słowo do nauki, ale jest to możliwe dopiero po zapłaceniu za usługę "pro" na stronie.

Funkcjonalności:

- **Interaktywne napisy:** Jedną z głównych zalet Language Reactor jest możliwość wyświetlania tłumaczeń słów i zwrotów w trakcie oglądania, co sprawia, że nauka odbywa się w naturalnym kontekście. Dzięki temu użytkownik może natychmiast zobaczyć, jak dane słowo funkcjonuje w zdaniu.
- **Integracja z platformami streamingowymi:** Aplikacja działa z popularnymi serwisami, takimi jak YouTube, Netflix czy Disney+, co oznacza, że użytkownicy mogą korzystać z niej podczas oglądania ulubionych filmów i seriali w obcym języku.
- **Słownik i lematyzacja:** Language Reactor automatycznie przetwarza słowa na ich formy podstawowe (lematy), co pomaga w nauce gramatyki oraz zapamiętywaniu nowych słówek bez względu na ich odmianę.
- **Baza słówek:** Użytkownicy mogą zapisywać słówka, które napotkali podczas oglądania, tworząc spersonalizowaną listę do późniejszego przyswajania. To rozwiązanie pozwala na systematyczną naukę i powtórki.
- **System powtórek SRS:** Aplikacja wprowadza system powtórek rozłożonych w czasie, co wspomaga długotrwałe zapamiętywanie materiału, podobnie jak w przypadku Anki.
- **Dostosowanie poziomu trudności:** Użytkownicy mogą dostosować poziom trudności materiałów, co pozwala na naukę dostosowaną do ich umiejętności i tempa.

Ograniczenia:

- **Ograniczona dostępność:** Language Reactor jest dostępny tylko na wybranych platformach streamingowych, co ogranicza możliwość korzystania z aplikacji do nauki języków z innych źródeł.
- **Płatne funkcje:** Pełna funkcjonalność aplikacji, w tym możliwość zapisywania słówek i korzystania z systemu powtórek, jest dostępna tylko w wersji płatnej, co może być barierą dla niektórych użytkowników.



Rysunek 1. Interaktywne napisy na stronie Language Reactor

Możliwość nauki tylko z wybranych kanałów youtube nie można wybrać filmu który nie jest na liście strony internetowej.

1.2.3 Trancy

Trancy to aplikacja stworzona z myślą o nauce języków obcych, która integruje naukę słówek z oglądaniem filmów, seriali i innych materiałów wideo. Aplikacja oferuje funkcje, które pozwalają użytkownikom uczyć się języka poprzez interaktywne napisy, tłumaczenia oraz dodatkowe ćwiczenia, co sprawia, że proces nauki staje się bardziej angażujący i efektywny.

Funkcjonalności:

- **Interaktywne napisy:** Trancy umożliwia wyświetlanie napisów w różnych językach, z tłumaczeniami słów i zwrotów. Dzięki temu użytkownicy mogą szybko zrozumieć, co oznacza dane słowo, a także zobaczyć je w kontekście. Napisy są dostosowane do poziomu zaawansowania użytkownika, co pozwala na naukę w sposób dostosowany do indywidualnych potrzeb.
- **Zbieranie słówek:** Użytkownicy mogą zapisywać napotkane słówka, które następnie mogą zostać przekształcone w fiszki do nauki. Taki system pozwala na systematyczne utrwalanie materiału i umożliwia szybkie powtórki w dogodnym czasie.
- **Tłumaczenia i definicje:** Aplikacja oferuje tłumaczenie słów na różne języki oraz wyświetlanie ich definicji, co wspomaga naukę gramatyki i budowanie słownictwa. Użytkownicy mogą korzystać z wbudowanego słownika, aby na bieżąco zgłębiać znaczenie nowych wyrazów.
- **Fiszki SRS:** Trancy wykorzystuje system powtórek rozłożonych w czasie (SRS), który pomaga w długotrwałym zapamiętywaniu słówek. Użytkownik dostaje powiadomienia o konieczności powtórki, co pozwala na regularne utrwalanie materiału i skuteczniejszą naukę.

Ograniczenia:

- **Wymaga płatnej subskrypcji:** Aby w pełni skorzystać z funkcji aplikacji, użytkownik musi zapłacić, w wersji darmowej są tylko niezbędne narzędzia jak tłumaczenie i limit do 100 słów do zapisania.
- **Ograniczona integracja z platformami:** Trancy oferuje integrację z wybranymi platformami streamingowymi, i nie daje możliwości nauki z własnych zapisanych filmów.
- **Wymagana wtyczka w przeglądarce:** Żeby dodawać słowa lub oglądać filmy musimy robić to poza stroną Trancy z użyciem wtyczki Trancy.

Rozdział 2

Cel i zakres pracy

2.1 Cel pracy

Głównym celem niniejszej pracy jest zaprojektowanie i implementacja aplikacji webowej wspomagającej naukę języków obcych poprzez oglądanie filmów i seriali. Aplikacja ma umożliwiać użytkownikom naukę wybranego języka obcego podczas oglądania treści multimedialnych własnego wyboru. Główne założenia to:

1. **Łatwość w nauce:** Umożliwienie użytkownikom nauki języków poprzez napisy w filmach i serialach bez potrzeby tworzenia kart do nauki w innych aplikacjach, takich jak Anki oszczędzając przy tym dużo czasu użytkownika.
2. **Połączenie nauki z rozrywką:** Użytkownicy będą mogli uczyć się języka podczas oglądania swoich ulubionych filmów i seriali, co czyni naukę bardziej przyjemną i efekowną.
3. **Baza danych:** Umożliwienie użytkownikom zapisywania trudnych słów do bazy danych, unikając duplikacji słów za pomocą blokady w bazie danych do wyciągnięcia z tego 100% skuteczności pomagać będzie lematyzacja słów, czyli proces upraszczania słowa do jego formy słownikowej.
4. **Spersonalizowane środowisko nauki:** Dostęp do panelu użytkownika z podsumowaniem postępów w nauce, różnorodne metody nauki słownictwa i gramatyki (fiszki, quizy, gry edukacyjne), możliwość śledzenia statystyk i wykresów przedstawiających postępy w nauce, a także gamifikacja procesu nauki poprzez nagradzanie użytkowników za osiągnięcia.
5. **Różne sposoby nauki:** Wgląd do danych za pomocą różnych metod jak flashcards, moduł do edycji napisów, quiz, wpisywania tłumaczenia słowa itp.
6. **Motywacja i śledzenie postępów:** Aplikacja będzie motywować użytkowników do nauki poprzez wizualizację ich postępów za pomocą statystyk, wykresów i informacji widocznych na profilu użytkownika.

7. **Zapewnienie kompatybilności:** Aplikacja będzie dostępna z różnych przeglądarek internetowych i urządzeń.
8. **Innowacyjne rozwiązania technologiczne:** Wykorzystanie sztucznej inteligencji do tłumaczenia napisów i ich przetwarzanie przy użyciu NLP, oraz zastosowanie nowoczesnych technologii webowych do zapewnienia płynnego działania aplikacji.

2.2 Zakres pracy

Zakres pracy obejmuje szczegółowe opisanie i realizację następujących etapów:

- **Opracowanie architektury systemu.**
- **Projektowanie i implementacja bazy danych.**
- **Tworzenie logiki aplikacji oraz jej frontendu i backendu.**
- **Zabezpieczenie aplikacji przed redundancją danych oraz niepoprawnymi wpisami.**
- **Umożliwienie użytkownikom logowania z różnych urządzeń obsługujących przeglądarki.**
- **Implementacja różnych metod nauki (słownik, flashcards, edycja napisów) oraz elementów gamifikacji.**

Na rynku dostępne są różne aplikacje wspomagające naukę języków obcych, takie jak Duolingo, Anki, Quizlet, Memrise, Babbel, Rosetta Stone, LingQ, FluentU, Clozemaster, itp.

Rozdział 3

Projekt

3.1 Wprowadzenie

Celem tego rozdziału jest przedstawienie procesu implementacji aplikacji webowej wspomagającej naukę języków obcych za pomocą filmów i seriali. W kolejnych sekcjach opisano architekturę systemu, projekt bazy danych, implementację logiki aplikacji oraz jej frontendu i backendu, zabezpieczenia przed redundancją danych oraz niepoprawnymi wpisami, możliwość logowania z różnych urządzeń obsługujących przeglądarki, implementację różnych metod nauki oraz elementów gamifikacji.

3.2 interfejs użytkownika

Interfejs użytkownika aplikacji składa się z kilku głównych widoków, które umożliwiają użytkownikom korzystanie z różnych funkcji. Wszystkie widoki zostały zaprojektowane z myślą o prostocie i intuicyjności, aby zapewnić użytkownikom łatwą nawigację i szybki dostęp do potrzebnych informacji. Poniżej przedstawiono najważniejsze elementy interfejsu użytkownika.

3.2.1 Widok startowy

Strona główna aplikacji zawiera przyciski kierujące do innych sekcji aplikacji, takich jak nauka, słownik, statystyki, profil użytkownika, ustawienia, itp. Użytkownicy mogą również przeglądać najnowsze filmy i serie dostępne w aplikacji oraz korzystać z wyszukiwarki, aby znaleźć interesujące ich treści.

3.2.2 Strona Nauki

Strona Nauki to główne miejsce, w którym użytkownicy mogą korzystać z różnych metod nauki, takich jak fiszki, quizy, gry edukacyjne, wpisywanie tłumaczenia słowa itp.

3.2.3 Strona Profilu

3.2.4 Strona Słownika

3.2.5 Strona Statystyk

3.2.6 Strona logowania oraz rejestracji

3.2.7 Strona oglądania filmów

Rozdział 4

Implementacja

4.1 Wprowadzenie

W tej sekcji przedstawiono szczegółowy opis implementacji projektu, w tym wybór technologii, narzędzi programistycznych oraz środowiska, w którym został zrealizowany. Omówione zostaną również kluczowe aspekty techniczne, takie jak struktura bazy danych, architektura backendu i frontendu, a także proces testowania i napotkane problemy implementacyjne. Celem tej sekcji jest dostarczenie pełnego obrazu technicznego projektu oraz uzasadnienie wyboru poszczególnych rozwiązań technologicznych.

4.2 Środowisko i narzędzia programistyczne

4.2.1 Środowisko programistyczne

Do implementacji projektu wykorzystano następujące narzędzia i środowiska programistyczne:

- **Visual Studio Code:** Środowisko programistyczne do tworzenia aplikacji internetowych w JavaScript i Python.
- **Git:** System kontroli wersji do zarządzania kodem źródłowym projektu.
- **MongoDB Atlas:** Usługa do hostowania bazy danych MongoDB w chmurze.
- **Flask:** Środowisko do tworzenia aplikacji webowych w Pythonie.
- **TypeScript:** Język programowania, który kompiluje się do JavaScriptu i dodaje typy danych w kodzie.

4.2.2 Wybór technologii

W trakcie realizacji projektu wykorzystano następujące technologie:

- **Next.js:** Framework do tworzenia aplikacji internetowych w React, który oferuje wiele wbudowanych funkcji, takich jak routing, server-side rendering czy generowanie statyczne [2].

- **Python:** Język programowania, który wykorzystałem do implementacji algorytmów przetwarzania języka naturalnego (NLP).
- **React Virtuoso:** Biblioteka do wirtualizacji list w aplikacjach internetowych.
- **Node.js:** Środowisko uruchomieniowe JavaScript, które pozwala na tworzenie aplikacji serwerowych.
- **MongoDB:** Baza danych NoSQL, która umożliwia przechowywanie danych w formacie JSON.
- **React.js:** Biblioteka do tworzenia interfejsów użytkownika w aplikacjach internetowych.
- **Shadcn:** Biblioteka gotowych komponentów do budowy interfejsu użytkownika w React.

4.2.3 Opis technologii

Next.js

Next.js to framework do tworzenia aplikacji internetowych w React, który oferuje wiele wbudowanych funkcji, takich jak routing, server-side rendering czy generowanie statyczne. Dzięki temu można tworzyć wydajne i skalowalne aplikacje internetowe, które są przyjazne dla SEO i łatwe w utrzymaniu. Next.js oferuje również wiele gotowych rozwiązań, takich jak automatyczne generowanie ścieżek, obsługa dynamicznych routów czy optymalizacja obrazów. Dzięki temu można skupić się na tworzeniu funkcjonalności, zamiast martwić się o konfigurację i optymalizację aplikacji [2].

Python

Python to język programowania, który wykorzystałem do implementacji algorytmów przetwarzania języka naturalnego (NLP). Python jest popularny w dziedzinie analizy danych i uczenia maszynowego, dzięki czemu można znaleźć wiele gotowych bibliotek i narzędzi do przetwarzania tekstu. W moim projekcie wykorzystałem biblioteki takie jak NLTK, spaCy czy TextBlob do lematyzacji, oznaczania części mowy i analizy sentymentu tekstu.

React Virtuoso

React Virtuoso to biblioteka do wirtualizacji list w aplikacjach internetowych. Umożliwia renderowanie długich list danych w sposób efektywny i wydajny, co przyczynia się do poprawy wydajności i płynności interfejsu użytkownika. Dzięki React Virtuoso można renderować tylko widoczne elementy i kilka dodatkowych poza obszarem.

Node.js

Node.js to środowisko uruchomieniowe JavaScript, które pozwala na tworzenie aplikacji serwerowych. Dzięki Node.js można pisać zarówno frontend, jak i backend w jednym języku programowania, co ułatwia rozwój i utrzymanie aplikacji. Node.js oferuje również wiele gotowych modułów i bibliotek, które ułatwiają tworzenie aplikacji internetowych, takich jak Express.js, Socket.io czy Mongoose.

MongoDB

MongoDB to baza danych NoSQL, umożliwiająca przechowywanie danych w formacie JSON. Baza jest skalowalna i elastyczna, co pozwala na przechowywanie różnych typów danych i szybki dostęp do nich. W projekcie baza danych została wykorzystana do przechowywania danych o użytkownikach, słowach do nauki, napisach oraz postępach w nauce.

React.js

React.js to biblioteka do tworzenia interfejsów użytkownika w aplikacjach internetowych. React.js oferuje wiele funkcji i narzędzi, które ułatwiają tworzenie interaktywnych i responsywnych interfejsów. Dzięki React.js można tworzyć komponenty UI, zarządzać stanem aplikacji i reagować na interakcje użytkownika w sposób efektywny i wydajny.

Shadcn

Shadcn to kolekcja komponentów, które można kopiować i wklejać do swoich aplikacji. Nie jest to biblioteka komponentów, którą można zainstalować jako zależność. Shadcn nie jest dostępny ani dystrybuowany przez npm (node package manager). Użytkownik wybiera potrzebne komponenty, kopiuje i wkleja kod do swojego projektu, a następnie dostosowuje go do swoich potrzeb. Kod jest własnością użytkownika. Shadcn może służyć jako odniesienie do budowy własnych bibliotek komponentów. do rozbudowy interfejsu użytkownika w React. Shadcn oferuje wiele gotowych rozwiązań, takich jak przyciski, formularze, tabele czy karty, które można łatwo dostosować do własnych potrzeb. Dzięki Shadcn można tworzyć interfejsy użytkownika w sposób szybki i efektywny, co przyczynia się do skrócenia czasu potrzebnego na rozwój aplikacji.

4.3 Baza danych

4.4 Backend

4.5 Frontend

4.6 Testy

4.7 Problemy implementacyjne

4.7.1 Wirtualizacja List

Wirtualizacja listy w aplikacjach internetowych to technika, która optymalizuje renderowanie długich list danych. Bez niej aplikacja renderuje wszystkie elementy listy na raz, co może prowadzić do problemów z wydajnością, zwłaszcza gdy lista jest duża. Wirtualizacja polega na renderowaniu

jedynie tych elementów, które aktualnie są widoczne w przeglądarce użytkownika, dzięki czemu zużycie zasobów jest minimalne, a aplikacja działa płynnie. Bez wirtualizacji listy, użytkownik mógłby doświadczać opóźnień w interakcji z interfejsem, tym większych im dłuższa lista danych przy 2 tysiącach wierszy opóźnienie stawało się uciążliwe ponieważ czekało się parę sekund na reakcję interfejsu listy i pokazanie okna dodawania trudnych słów do systemu.

Jak działa wirtualizacja listy

- **Obserwacja widocznych elementów:** Komponent śledzi pozycję widoku użytkownika w liście. Renderowane są tylko te elementy, które mieszczą się w aktualnie widocznym obszarze (viewport) oraz kilka dodatkowych elementów „na zapas” wokół tego obszaru.
- **Renderowanie na żądanie:** Gdy użytkownik przewija listę, niewidoczne elementy są dynamicznie usuwane z DOM-u, a nowe – wczytywane na ich miejsce.
- **Stała wysokość elementów (lub szacowana):** Dla prawidłowego działania, komponent wirtualizujący często wymaga, aby elementy listy miały stałą lub przynajmniej przewidywalną wysokość. Dzięki temu może łatwo obliczać, które elementy powinny być aktualnie wyświetlane.
- **Oszczędność zasobów:** Dzięki renderowaniu tylko niewielkiej liczby elementów, zmniejsza się zużycie pamięci i obciążenie procesora, co prowadzi do szybszego działania aplikacji.

Dlaczego React Virtuoso

React Virtuoso jest biblioteką do wirtualizacji list, która znacznie upraszcza implementację tego mechanizmu w React. Automatycznie obsługuje:

- **Przewijanie:** Zajmuje się wykrywaniem widocznych elementów, reagując na przewijanie użytkownika.
- **Niestandardowe wysokości elementów:** Obsługuje zarówno stałe, jak i zmienne wysokości elementów, co czyni go bardziej elastycznym.
- **Lazy loading:** Umożliwia ładowanie danych w locie, co jest kluczowe dla dużych list z elementami, które mogą być dynamicznie ładowane z serwera.

Zastosowania

Virtualizacja listy jest szczególnie przydatna w przypadku:

- **Długich list:** Kiedy lista zawiera setki lub tysiące elementów.
- **Aplikacji mobilnych:** Gdzie zasoby są ograniczone i każda optymalizacja wydajności jest istotna.
- **Interfejsów użytkownika z dużą ilością dynamicznych danych:** Takich jak portale społecznościowe, aplikacje e-commerce czy dashboardy.

Wady

Mimo licznych zalet, wirtualizacja listy ma również pewne wady:

- **Złożoność implementacji:** Wprowadzenie wirtualizacji może wymagać dodatkowego kodu i konfiguracji, co może zwiększyć złożoność projektu.
- **Problemy z dostępnością:** Renderowanie dynamiczne może wpływać na narzędzia do czytania ekranu i inne technologie wspomagające, co może utrudniać dostępność aplikacji.

Wykorzystanie React Virtuoso przyczyniło się do poprawy wydajności i płynności interfejsu użytkownika aplikacji, co miało kluczowe znaczenie dla zadowolenia użytkowników i jakości doświadczenia użytkownika.

4.7.2 Przetwarzanie języka naturalnego (NLP)

W pracy inżynierskiej zastosowano techniki lematyzacji oraz oznaczania części mowy (POS tagging) w ramach przetwarzania języka naturalnego (NLP). Obie te techniki odegrały kluczową rolę w analizie tekstu i umożliwiły bardziej precyzyjne przetwarzanie danych z napisów, przy zapisywaniu wybranych przez użytkownika słów do nauki, lub przy wyświetlaniu częstości występowania słów w napisach [1].

Lematyzacja

Lematyzacja

Proces lematyzacji polega na sprowadzaniu różnych form gramatycznych wyrazów do ich podstawowej formy, zwanej lematem. Dzięki temu możliwe jest ujednolicenie wyrazów, które w zależności od kontekstu występują w różnych odmianach gramatycznych. Przykładowo, formy takie jak "chodzę", "chodził" czy "chodziliśmy" są sprowadzane do podstawowej formy "chodzić". Umożliwia to bardziej spójne analizowanie tekstów i wyciąganie wniosków na temat ich zawartości, np. poprzez obliczanie częstotliwości występowania poszczególnych słów [1].

POS Tagging (oznaczanie części mowy)

Drugą techniką było oznaczanie części mowy, czyli przypisywanie każdemu słowu w tekście odpowiedniej etykiety gramatycznej (rzeczownik, czasownik, przymiotnik itd.) [1]. Dzięki temu możliwe było lepsze zrozumienie struktury zdań oraz funkcji słów w kontekście. Oznaczanie części mowy okazało się kluczowe w procesie analizy tekstu, umożliwiając podział słów na kategorie zależne od ich funkcji gramatycznej. W przyszłości może to być przydatne do tworzenia funkcji umożliwiających użytkownikom wybór nauki określonych kategorii słów, takich jak czasowniki czy przymiotniki itd.

Rozdział 5

Podsumowanie

5.1 Wnioski

Projekt aplikacji webowej wspomagającej naukę języków obcych za pomocą filmów i seriali stanowi nowoczesne i efektywne narzędzie edukacyjne. Integracja funkcji nauki i oglądania w jednym miejscu ułatwia użytkownikom proces przyswajania wiedzy, eliminując konieczność korzystania z dodatkowych aplikacji. Aplikacja umożliwia przechowywanie trudnych słów, śledzenie postępów oraz korzystanie z różnych metod nauki, co czyni ją wszechstronnym narzędziem wspomagającym naukę języków.

5.2 Kierunki dalszego rozwoju

Dalszy rozwój aplikacji może obejmować dodanie nowych funkcji, takich jak integracja z innymi platformami streamingowymi, rozwój zaawansowanych metod nauki oraz ulepszenia interfejsu użytkownika. Dodatkowe modele NLP do innych języków również mogą okazać się przydatne.

Bibliografia

- [1] **Rada Mihalcea, Hugo Liu i Henry Lieberman**, „NLP (Natural Language Processing) for NLP (Natural Language Programming)”, w *Computational Linguistics and Intelligent Text Processing*, **Alexander Gelbukh**, red., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, s. 319–330, ISBN: 978-3-540-32206-1.
- [2] **Guillermo Rauch**. „Building Modern Web Applications with Next.js”. Dostęp z dnia 12 grudnia 2024 roku. (), adr.: <https://nextjs.org/docs>.

Spis rysunków

1	Interaktywne napisy na stronie Language Reactor	9
---	---	---

Spis tabel

Listings