|  |  |
| --- | --- |
|  |  |

**Kacper Tumulec**

nr albumu: 44535

kierunek studiów: Informatyka

specjalność: Inżynieria systemów informacyjnych

forma studiów*: stacjonarne*

**PROJEKT I IMPLEMENTACJA APLIKACJI DO NAUKI JAPOŃSKICH ZNAKÓW KANJI**

**PROJECT AND IMPLEMENTATION APPLICATION SUPPORTING LEARNING JAPANESE KANJI SIGNS**

praca dyplomowa inżynierska

napisana pod kierunkiem:

**dr inż. Luizy Fabisiak**

Katedra Inżynierii Oprogramowania i Cyberbezpieczeństwa

Data wydania tematu pracy: 24.06.2021

Data dopuszczenia pracy do egzaminu: …………………………………………………………………  
(uzupełnia pisemnie Dziekanat)

Szczecin, 2021

**OŚWIADCZENIE**

**AUTORA PRACY DYPLOMOWEJ**

Oświadczam, że praca dyplomowa inżynierska pn.

„Projekt i implementacja aplikacji do nauki japońskich znaków kanji” napisana pod kierunkiem: **dr inż. Luizy Fabisiak** jest w całości moim samodzielnym autorskim opracowaniem sporządzonym przy wykorzystaniu wykazanej w pracy literatury przedmiotu i materiałów źródłowych. Złożona w dziekanacie Wydziału Informatyki treść mojej pracy dyplomowej w formie elektronicznej jest zgodna z treścią w formie pisemnej.

Oświadczam ponadto, że złożona w dziekanacie praca dyplomowa ani jej fragmenty nie były wcześniej przedmiotem procedur procesu dyplomowania związanych z uzyskaniem tytułu zawodowego w uczelniach wyższych.

…….………………………..

podpis dyplomanta

Szczecin, dn. ……………

**Streszczenie pracy**

Celem pracy było zaprojektowanie oraz implementacja aplikacji do nauki japońskich znaków kanji. Głównym zadaniem aplikacji była nauka znaków poprzez całe zdania gdzie dany znak kanji byłby użyty. Aplikacja ma być przeznaczona dla osób przygotowujących się do Nihongo nōryoku shiken – test znajomości języka japońskiego dla obcokrajowców. Aplikacja podzielona jest na 5 sekcji znaków od N5 do N1.

**Słowa kluczowe**

kanji, python, nauka, znaki, javascript, react, fastapi, mysql, nauka języka

**Abstract**

The aim of this work was to design and implement an application for learning Japanese kanji characters. The main goal was to learn the characters through whole sentences where a kanji character would be used. The application is intended for people preparing for the Nihongo nōryoku shiken - Japanese language proficiency test for foreigners. The application is divided into 5 sections of characters from N5 to N1 as in the test.

**Keywords**

kanji, python. learning, signs, javasript, react, fastapi, mysql, language learning

Spis treści

[WSTĘP 6](#_Toc100303120)

[Rozdział 1 7](#_Toc100303121)

[Omówienie i przedstawienie problemu 7](#_Toc100303122)

[1.1 Sposoby nauki znaków 8](#_Toc100303123)

[1.2 Konkurencyjne rozwiązania 10](#_Toc100303124)

[**1.2.1 Duolingo** 11](#_Toc100303125)

[**1.2.2 Anki** 11](#_Toc100303126)

[**1.2.3 Kanji Study** 12](#_Toc100303127)

[1.3 Przedstawienie problemu 13](#_Toc100303128)

[Rozdział 2 14](#_Toc100303129)

[Wybór narzędzi środowiska programowania 14](#_Toc100303130)

[2.1 Narzędzia projektowe 14](#_Toc100303131)

[**2.1.1 Visual Studio Code** 14](#_Toc100303132)

[**2.1.2 Git** 14](#_Toc100303133)

[**2.1.3 XAMPP** 15](#_Toc100303134)

[**2.1.4 Marvel** 15](#_Toc100303135)

[2.2 Front-end 15](#_Toc100303136)

[**2.2.1 Javascript w wersji ES7** 15](#_Toc100303137)

[**2.2.2 HTML w wersji 5** 16](#_Toc100303138)

[**2.2.3 Kaskadowe arkusze stylów – CSS** 17](#_Toc100303139)

[**2.2.4 Material UI** 17](#_Toc100303140)

[**2.2.5 Recharts** 17](#_Toc100303141)

[2.3 Back-end 18](#_Toc100303142)

[**2.3.1 Python w wersji 3.7.4** 18](#_Toc100303143)

[**2.3.2 MySQL** 19](#_Toc100303144)

[Rozdział 3 20](#_Toc100303145)

[Projekt i implementacja aplikacji 20](#_Toc100303146)

[3.1 Wymagania funkcjonalne i niefunkcjonalne 20](#_Toc100303147)

[3.1.1 Wymagania funkcjonalne: 20](#_Toc100303148)

[3.1.2 Wymagania niefunkcjonalne 20](#_Toc100303149)

[3.2 Projekt wyglądu ekranów aplikacji 21](#_Toc100303150)

[3.3 Projekt bazy danych 23](#_Toc100303151)

[3.4 Skrypt pobierania danych 24](#_Toc100303152)

[3.5 Projekt i implementacja funkcjonalności aplikacji 25](#_Toc100303153)

[3.5.1 Rejestracja i logowanie do aplikacji 25](#_Toc100303154)

[**3.5.1.1 Rejestracja** 25](#_Toc100303155)

[**3.5.1.2 Logowanie** 27](#_Toc100303156)

[3.5.2 Strona główna aplikacji 29](#_Toc100303157)

[3.5.3 Moduł uczący – quiz z pytaniami 30](#_Toc100303158)

[3.5.4 Wyniki quizu 35](#_Toc100303159)

[3.5.5 Wykres słupkowy postępów 37](#_Toc100303160)

[4. Przeprowadzenie testów 38](#_Toc100303161)

[5. Podsumowanie oraz wnioski 38](#_Toc100303162)

[6. Bibliografia 38](#_Toc100303163)

# WSTĘP

# Rozdział 1

# Omówienie i przedstawienie problemu

Jeszcze niedawno Japonia, jej kultura i język była w oczach wielu czymś egzotycznym, nieosiągalnym i pełnym tajemnic. Dziś choć kraj kwitnącej wiśni dalej zadziwia, to jest dostępny dla ludzi z całego świata jak nigdy dotąd. Powszechna globalizacja i przeplatanie się kultur spowodowały znaczny wzrost zainteresowania Azją. Wśród krajów wiodących prym   
w zainteresowaniu na zachodzie są Chiny, Korea i Japonia.

Japonia zaczęła zyskiwać popularność początkowo przez coraz częściej pojawiające się w naszej kulturze filmy Akiro Kurosawy, budzące zachwyt wśród krytyków z zachodu. Jednak prawdziwa ekspansja japońskiej kultury zaczęła się na dobre w momencie rozpoczęcia importu do telewizji i kin japońskich animacji. Nagle okazało się, że gdzieś po drugiej stronie globu powstają kreskówki idące w parze czy nawet przewyższające te wydane przez Walta Disneya. Dokonania w dziedzinie animacji zostały przypieczętowane w 2003 roku kiedy to Hayao Miyazaki otrzymał Oscara za najlepszy pełnometrażowy film animowany.

Wraz ze wzrostem popularności japońskiej animacji, ludzie, a w szczególności młodzież odkrywali coraz to nowsze dzieła. Wzrost dostępności internetu pozwolił ludziom   
na oglądanie anime (japońskiej animacji, najczęściej jako serie wieloodcinkowe), zbieranie   
się w społeczności na forach i grupach w mediach społecznościowych. Powstała cała subkultura ‘otaku’ której podstawowym filarem była fascynacja Krajem Wschodzącego Słońca. Oczywistym był fakt, że obiektem zainteresowania stał się również język japoński.

W roku 2018 na świecie według danych zebranych przez Japan Foundation[1] było 3 851 774 osób uczących się języka japońskiego. Język japoński jest przez wielu uznawany jako jeden z trudniejszych do nauczenia. Chociaż sama wymowa i zasady gramatyczne nie odbiegają poziomem zaawansowania od innych języków obcych, prawdziwą barierą dla osoby z zachodu jest zapis zdań w języku japońskim.

W kraju kwitnącej wiśni używane są cztery alfabety, zazwyczaj używane razem, nawet w jednym zdaniu możemy napotkać się na znaki każdego z nich. Alfabet łaciński – używany   
w nazwach własnych i produktach zachodnich, katakana i hiragana – są to sylabariusze   
za pomocą których możemy zapisać każde zdanie używając opowiadających dźwięków,   
jest ich po 46 dla każdego z alfabetów. Wydawać by się mogło więc, że nie ma problemu,   
bo musimy nauczyć się 92 znaków łącznie dla hiragany i katakany, a z nimi jesteśmy w stanie zapisać już wszystko. Niestety jest to tylko po części prawda, ponieważ w Japonii na porządku dziennym jest używanie czwartego alfabetu – Kanji. Alfabet kanji to zbiór znaków zapożyczonych z języka chińskiego. W Dai Kan-Wa Jiten – czyli japońskim słowniku znaków kanji, zapisane jest ponad 50 000 znaków, jednak wiele z nich już dawno wyszła z użytku   
lub jest używana naprawdę sporadycznie. Osoby uczące się japońskiego, ale i sami Japończycy poznają jōyō kanji – czyli znaków codziennego użytku składających się z 2 136 znaków.   
Dla obcokrajowców stworzony został Nihongo nōryoku shiken – testu znajomości języka japońskiego (w późniejszej części pracy nazywany również JLPT od Japanese Language Proficiency Test). Test składa się z pięciu sekcji: kanji, słownictwo, gramatyka, czytanie  
 i słuchanie. Jednym z głównych zadań aplikacji jest więc przygotowanie osoby aspirującej   
do zdania wyżej wymienionego egzaminu do części dotyczącej znaków kanji.

## Sposoby nauki znaków

Na rynku możemy znaleźć wiele rozwiązań dotyczących nauki języków jako ogółu jak i konkretnych kursów czy aplikacji dotyczących danego zagadnienia, w przypadku opisanej pracy znaków Kanji. Do nauki stosowane są różne techniki które doskonale w swojej pracy „*Tackling the Kanji hurdle: Investigation of Kanji learning in Non-Kanji background learners*”[2] opisali *Simon Paxton, Chavalin Svetanant.*

**Nauka pamięciowa –** Uczenie na pamięć wydaje się główną metodą używaną przez zagranicznych studentów, do nauki pisma kanji. W badaniu przeprowadzonym przez Shimizu i Green (2002) [3], odkryto, że strategia uczenia się na pamięć była najpopularniejsza wśród zagranicznych studentów. Shimizu zauważa, że przyczyną popularności tej strategii jest fakt, że wielu rodowitych japońskich nauczycieli czerpie z własnych doświadczeń w nauce kanji,   
a większość z nich uczyła się metodami pamięciowymi. Ponadto, taka sama technika używana jest przez rodowitych użytkowników języka angielskiego do zapamiętywania pisowni angielskich słów, a zatem to może być również czynnikiem, dlaczego tak wielu uczniów wybiera tę strategię. Uczenie się na pamięć może być użyteczną metodą pomagającą   
w zapamiętywaniu kanji. Dodatkowo może pomóc studentom w rozwijaniu umiejętności pisania znaku z zachowaniem właściwych proporcji. Nesbitt [4] sugeruje, aby w pierwszym roku nauki kanji wykorzystywać uczenie się na pamięć jako specyficzne, ustrukturyzowane narzędzie do nauki, żeby rozwinąć silne ścieżki neuronalne dla automatyzacji. Uczenie się na pamięć będzie nadal miało swoje miejsce jako strategia nauki znaków mimo faktu, iż badania wydają się sugerować, że nie jest ono szczególnie efektywne w poprawie zdolności zapamiętywania.

**Mnemonika (nauka przez skojarzenia)** – Użycie mnemotechnik w nauce kanji polega na użyciu słów kluczowych do reprezentowania poszczególnych elementów kanji. Łącząc te słowa kluczowe w zdania, uczniowie otrzymują użyteczne narzędzie pomagające zapamiętać kanji.

Pomimo tego, że istnieje niewiele dowodów na to, iż mnemotechniki pomagają w długotrwałej zdolności zapamiętywania, strategia ta zyskuje na popularności.

Wang i Thomas [5] stwierdzili, że nie było większej przewagi w zapamiętywaniu mnemotechnik w porównaniu z uczeniem się na pamięć, a w rzeczywistości w ich badaniu odnotowano większe zapominanie wśród uczących się, którzy używali mnemotechnik. Chociaż mnemotechnika może pomóc czytelnikowi w zapamiętaniu struktury poszczególnych kanji,   
nie może ona dostarczyć mu informacji semantycznych i fonologicznych dotyczących różnych kanji.

Jednym z głównych zarzutów wobec mnemotechniki jako strategii nauki jest fakt, że może ona pomagać w zapamiętywaniu struktury znaku, ale nie pomaga uczniom w zapamiętywaniu czytania kanji.

**Metoda kontekstowa –** Studiowanie kanji w kontekście jest popularną strategią uczenia się kanji, a dowodem na to może być liczba popularnych podręczników kanji na rynku, które przyjmują to podejście; szczególnie książki "Basic Kanji Book"[6], "Intermediate Kanji Book"[7] i "Kanji in Context"[8] wydane przez Japan Times. Podstawą tego podejścia jest to, że kanji powinno być traktowane jako "słownictwo", a nie jako "alfabet".

Jest to często podejście stosowane w nauczaniu kanji na poziomie średniozaawansowanym. Kano[7] twierdzi, że celem edukacji kanji na poziomie średniozaawansowanym powinno być efektywne budowanie słownictwa za pomocą słów użytkowych, a nie nauczanie kanji jako znaków. Według Shimizu i Greena [3], strategie kontekstowe zostały podkreślane, ponieważ czytanie i znaczenie związków kanji jest często wysoce zależne od kontekstu.

Z uwagi na dużą liczbą homonimów w języku japońskim, studiowanie kanji w kontekście,   
w przeciwieństwie do studiowania pojedynczych znaków wyizolowanych z jakiegokolwiek kontekstu jest użyteczną strategią dla studenta kanji.

Na podstawie analizy prac i metodyk dostępnych na rynku oraz własnych doświadczeń w nauce języka japońskiego, w aplikacji jako główna metoda przyjęta została strategia nauki kontekstowej poprzez zmuszanie użytkownika do wyboru odpowiedniego znaku   
w wyznaczonym zdaniu.

## Konkurencyjne rozwiązania

Przed przystąpieniem do projektowania i implementacji, należy przenalizować istniejące już na rynku aplikacje. Jest to ważna część developmentu gdyż nie powinno się powtarzać istniejącego rozwiązania, a dodatkowo możemy sprawdzić jakie funkcjonalności można poprawić i zaimplementować w naszym projkecie.

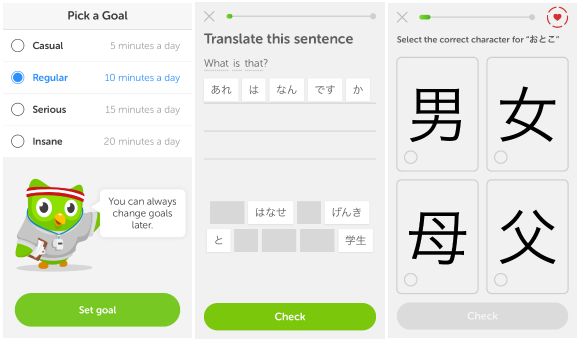
Aktualnie rynek nauki znaków kanji jest rozbudowany, na sklepie play   
jak i w internecie dominują aplikacje które głównie korzystają z metody pamięciowej. Duża część konkurencji to fiszki i ich klony gdzie nauka znaku nie różni się niestety dużo   
od przeglądania słownika kanji z wyszczególnieniem danych znaków. Często też brak wyraźnego podziału poziomów trudności, a jeśli już jest obecny to nie korzysta on   
z ustandaryzowanego podziału dla egzaminu JLPT. Kolejnym zastrzeżeniem jest powtarzalność przykładów jak i brak automatyzacji rozpoznawania poziomu znajomości danego znaku.

W pracy przeanalizowano 3 darmowe aplikacje które posiadają pewnego rodzaju mikropłatności. Wybrano te aplikacje jako reprezentantów pewnych kategorii sposobów nauki.

### 1.2.1 Duolingo

Duolingo jest aplikacją mobilną do nauki języków. Jej głównym celem jest nauka języka jako całości, a nie nauka poszczególnych elementów, jednak jej popularność   
i posiadanie w pewnym stopniu podobnej funkcjonalności sprawiło, że warto ją wspomnieć . Użytkownik jest zapoznawany ze znakami w sposób chaotyczny. Nie można wybrać opcji nauki samych znaków, a dodatkowo z jakiego poziomu znajomości języka powinny być one wyświetlane.

Dużym plusem aplikacji jest mieszanie metod nauczania i stosowanie metody kontekstowej. Dodatkowo wyróżnia się przyjazną szatą graficzną i systemem rankingu użytkowników. Mankamentem jest jednak mała ilość znaków jakie poznajemy podczas całego kursu i bardziej zaawansowany student japońskiego raczej nie znajdzie tutaj nic nowego.   
Na Rysunku 1 przedstawiono przykładowy ekran aplikacji.



Rysunek 1. Przykładowe ekrany aplikacji Duolingo

### 1.2.2 Anki

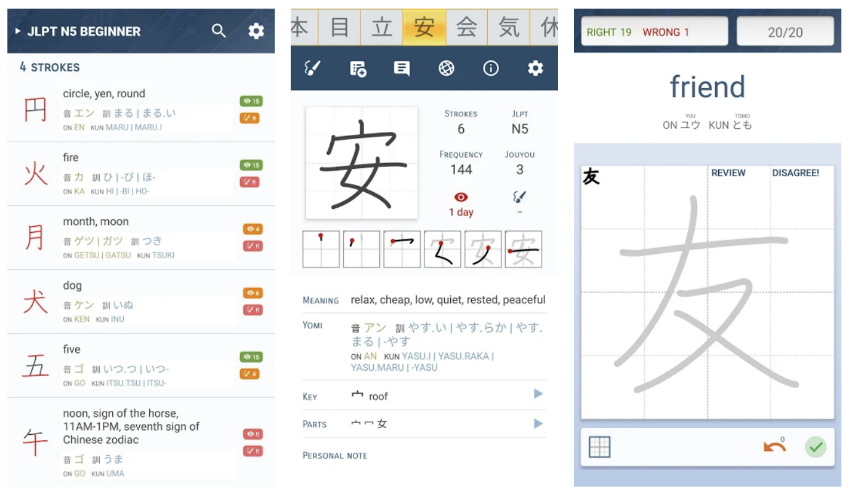
Anki - jest to aplikacja do tworzenia fiszek i zestawów do nauki, jest ona w pełni konfigurowalna i dużo zależy od zestawów używanych przez użytkownika. Minusem jest jednak powtarzalność fiszek i ryzyko, że użytkownik będzie dany znak kojarzył z konkretnym zdaniem na fiszce lub słowem, a kiedy znak zostanie użyty w innej formie nie będzie go rozpoznawał. Dodatkowo aplikacja bazuje czysto na naszym feedbacku.



Rysunek 2 Przykładowy ekran aplikacji Anki

### 1.2.3 Kanji Study

Kanji Study to aplikacja mobilna nastawiona na naukę znaków kanji. Posiada podział na poziomy N5 do N1 i przykłady zdań ze znakami kanji. Aplikacja jest bezpłatna dla najłatwiejszych znaków za wyższe pozimy należy wykupić wersję premium. Aplikacja głównie stawia na naukę poprzez fiszki opatrzone jednak w dużą ilość informacji gdzie sami oceniamy naszą znajomość znaku kanji. Wielką zaletą jest usystematyzowany podział i różne opcje fiszek, dodatkowo w przeciwieństwie do Anki, tutaj cały zestaw do nauki dostarcza aplikacja i nie musimy sami wgrywać go w aplikację. Jest to moim zdaniem najlepsza konkurencyjna aplikacja, jednak powtarzalność fiszek i konieczność wykupienia pełnej wersji, aby uzyskać dostęp do wyższych poziomów stanowi główną wadę aplikacji.



Rysunek 3 Przykładowe ekrany aplikacji Kanji Study

## Przedstawienie problemu

Mnogość sposobów nauczania znaków powoduje obecność na rynku wielu rozwiązań i aplikacji. Wiele z nich jednak nie naucza studentów w sposób usystematyzowany przez certyfikat z języka japońskiego. Często głównymi quizami nauczającymi są zwykłe fiszki, a dopiero po wejściu w konkretny znak możemy poznać jego użycie w kontekście.

Kolejnym problemem jest powtarzalność przykładów, zazwyczaj znaki mają przypisane stałe rodzaje użycia co powoduje u osoby uczącej się powstawanie skojarzeń ze znakami obok zamiast świadomego wybieranie odpowiedniego czytania lub znaczenia.

Duża ilość informacji jest rozproszona na wielu stronach internetowych czy książkach, a wiele materiałów jest zapisanych w przestarzałej formie np. w formatach arkuszy do wydrukowania. Nie jest ona przystępna dla młodego człowieka żyjącego cały czas w ruchu chcącego uczyć się na telefonie lub komputerze.

Podsumowując, problemem jest brak na rynku rozwiązania skupiającego się na nauce kontekstowej, ale przy tym posiadającego również możliwość nauki pojedynczych słówek(znaków) z wybranego zakresu. Brakuje również rozwiązania celującego głównie w naukę kanji pod kątem przystąpienia do egzaminu JLPT gdzie przykłady byłyby wzorowane na prawdziwym układzie testu certyfikującego.

# Rozdział 2

# Wybór narzędzi środowiska programowania

W tym rozdziale zostały opisane narzędzia użyte do zaprojektowania i implementacji aplikacji webowej. Narzędzia można podzielić na trzy główne kategorie:

-projektowe – użyte w kontekście całego projektu

-front-end – narzędzia użyte przy pisaniu i tworzeniu front-endu aplikacji

-back-end – narzędzia użyte przy pisaniu i tworzeniu back-endu aplikacji

## 2.1 Narzędzia projektowe

### 2.1.1 Visual Studio Code

Visual Studio Code jest darmowym edytorem kodu źródłowego stworzonym przez Microsoft. Kod źródłowy programu jest otwarty, na licencji MIT. Jest to bardzo popularne środowisko, jak podaje serwis stackoverflow, w 2020 roku według ankiety przeprowadzonej na 82 277 badanych, aż 71 % developerów używa VS Code. [9]

Oprogramowanie to jest proste w użyciu, a jego największą zaletą jest uniwersalność ponieważ użytkownik może dostosować wtyczki do technologii, które zainstaluje. Dzięki temu można pisać w wielu językach programowania używając jednego, zintegrowanego środowiska. Visual Studio Code pozwala na debugowanie, automatyczne uzupełnianie, wykrywanie błędów i refaktoryzację kodu.

### 2.1.2 Git

Git to system kontroli wersji, stworzony przez Linusa Torvaldsa. Stanowi on wolne oprogramowani na licencji GNU GPL. System kontoli wersji pozwala na zapisywanie zmian dokonanych w plikach, łączenie plików w których dokonano równolegle edycji czy powrót do danej wersji pliku. Git jest tak zwanym rozproszonym systemem kontroli wersji; oznacza to, że każda kopia repozytorium posiada wszystkie jego gałęzie i każdy klient posiada dokładną jego kopię. Badania przeprowadzone przez serwis stackoverflow[9] wykazały, że aż 93% programistów korzysta z Gita, co sprawia, że ten system kontroli wersji jest najczęściej używanym narzędziem przez programistów.

### 2.1.3 XAMPP

XAMPP to pakiet w którego skład wchodzą server Apache oraz bazy danych MySQL   
i interpretery dla PHP i Perl. W projekcie wykorzystano server Apache i bazę danych MySQL. Pakiet ten został użyty ze względu na mały rozmiar aplikacji i brak konieczności wystawiania aplikaji na serwerze produkcyjnym. XAMPP pozwala na uruchomienie całego środowiska lokalnie

### 2.1.4 Marvel

Marvel jest aplikacją internetową pozwalającą na projektowanie interfejsu użytkownika. Pozwala na stworzenie interaktywnej makiety będącej prototypem interfejsu   
i ogólnego wyglądu strony.

## 2.2 Front-end

### 2.2.1 Javascript w wersji ES7

Javascript jest skryptowym językiem programowania służącym głównie przy tworzeniu stron internetowych. Jedną z głównych zalet javascriptu jest odciążenie serwera   
z wykonywania programu – kod wykonuje się po stronie klienta. Według definicji zawartej na stronie developer.mozilla.org „Każda karta przeglądarki jest swoim własnym kontenerem dla kodu, który w niej się wykonuje (te kontenery są nazywane technicznie "środowiskami wykonywania" (ang. "execution environments"). Oznacza to, że w większości przypadków kod w każdej karcie jest wykonywany oddzielnie i kod z jednej z kart nie jest w stanie bezpośrednio wpłynąć na ten wykonujący się w innej karcie.”[10]

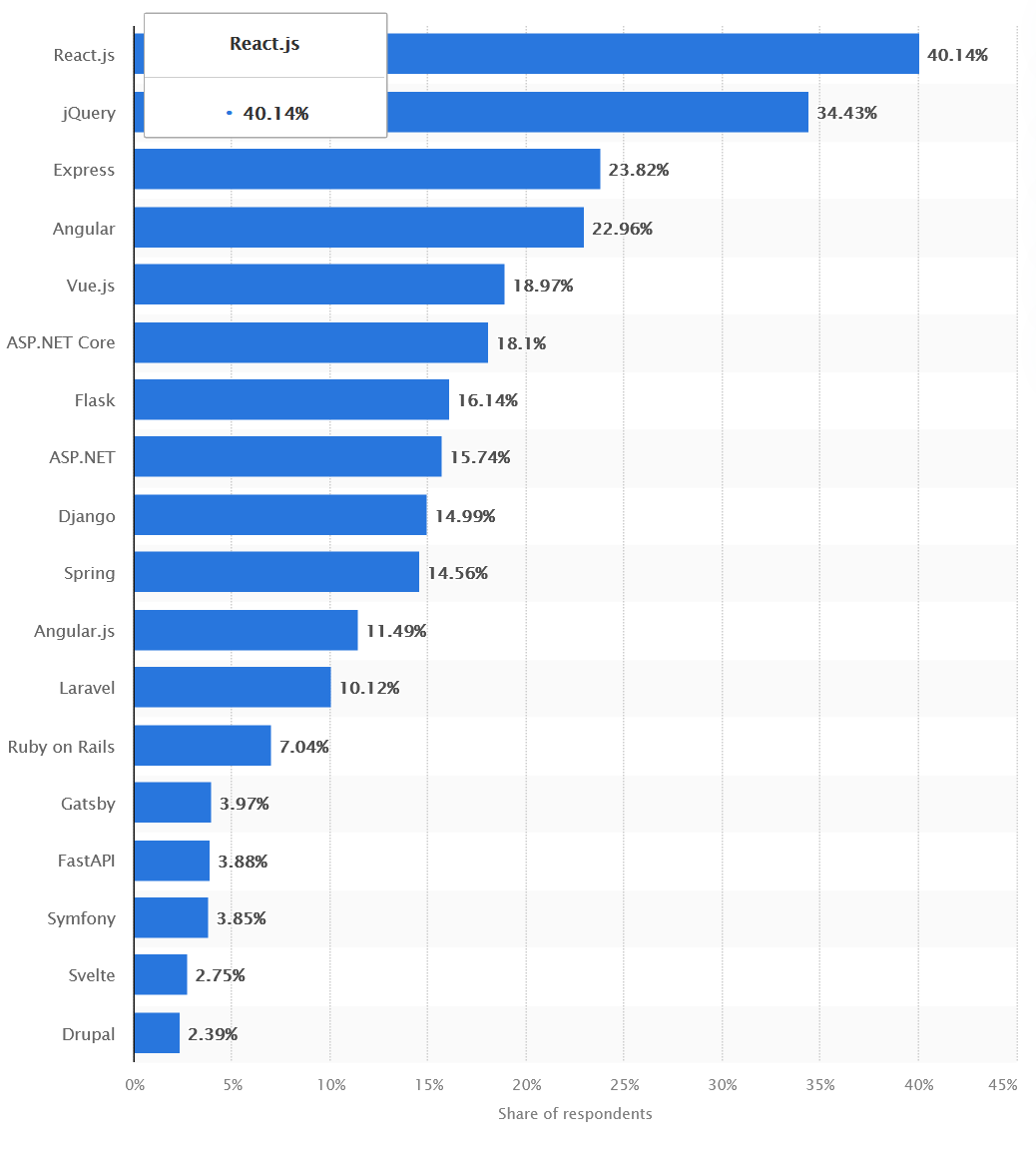
Javascript posiada również wiele bibliotek usprawniających pisanie stron. Najważniejszą biblioteką po stronie front-endu jest React.

#### 2.2.1.1 React.js

Biblioteka React pozwala na łatwiejsze projektowanie i implementację stron internetowych. Główną ideą jest tworzenie wyizolowanych komponentów, zamiast szablonów. Komponenty możemy dowolnie łączyć w jeden większy moduł. Kolejną zaletą Reacta jest posiadanie Vitual DOM które pozwala na aktualizację drzewa DOM w wybranym fragmencie zamiast całości.

React pozwala również na budowę aplikacji jednostronnicowych, gdzie użytkownik nie musi czekać na ładowanie kolejnych podstron aplikacji. Sprawia ona wrażenie bardziej interaktywnej i szybszej.

Ważnym argumentem dla wyboru biblioteki React jest również duża społeczność programistów używających jej. Pozwala to na znalezienie rozwiązań wielu problemów programistycznych oraz ułatwia wzajemną pomoc innych użytkowników.



Rysunek 4 Najbardziej popularne frameworki wśród web developerów w 2021 [11]

#### 2.2.1.2 Bulma

Bulma jest darmowym frameworkiem, który dostarcza gotowe do użycia komponenty frontendowe, które można łatwo połączyć w celu zbudowania responsywnych interfejsów internetowych. Pozwala on budować responsywne i skomplikowane style bez zaawansowanej znajomości CSS. Bulma w projekcie została użyta do ostylowania strony.

### 2.2.2 HTML w wersji 5

HTML jest językiem znaczników używanym do tworzenia struktur stron internetowych. Odpowiada też za to, co wyświetlane jest na ekranie. HTML 5 obsługuje tradycyjną składnię HTML 5 oraz XHTML.

### 2.2.3 Kaskadowe arkusze stylów – CSS

CSS jest jednym z podstawowych języków otwartej sieci i jest standaryzowany   
w przeglądarkach internetowych zgodnie ze specyfikacją W3C. CSS służy do ustalenia sposobu wyświetlania strony internetowej. Kaskadowe arkusze stylów określają reguły wyświetlania danych elementów HTML strony przez przeglądarkę. CSS pozwala również na ustalenie wyglądu poszczególnych elementów, takich jak kolor i wielkość czcionki, wielkość marginesu czy układ list.

W projekcie wraz z pomocą frameworku Bulma CSS odpowiada za wygląd i układ stron aplikacji.

### 2.2.4 Material UI

Material UI jest biblioteką przeznaczoną dla projektów stworzonych w React JS. Paczka ta zapewnia dużą ilość gotowych komponentów do wykorzystania w projekcie. Dużą zaletą Material UI jest rozległa dokumentacja i baza użytkowników. Wygląd elementów jest nowoczesny i idealnie nadaje się zarówno w widoku przeglądarkowym jak i mobilnym.

### 2.2.5 Recharts

Recharts to biblioteka opensource która pozwala na tworzenie wykresów różnego typu. Wszystkie komponenty Recharts są wyraźnie oddzielone. Dzięki temu można w prosty sposób dostosowywać komponent do potrzeb użytkownika.

## 2.3 Back-end

### 2.3.1 Python w wersji 3.7.4

Python jest interpretowanym, obiektowym, wysokopoziomowym językiem programowania. Python posiada wiele zastosowań, na co pozwala duża ilość dostępnych bibliotek. Aktualnie python jest używany między innymi do:

-Obliczeń naukowych i numerycznych

-Aplikacji biznesowych

-Tworzenia serwisów webowych

-Tworzenia skryptów automatyzujących pracę

W projekcie Python odpowiada za back-end aplikacji, zbieranie danych i zarządzanie bazą danych przez użycie bibliotek i frameworków.

#### 2.3.1.1 FastAPI

FastAPI jest frameworkiem webowym pozwalającym na budowanie interfejsów API   
za pomocą języka Python. Przy jego użyciu można szybko zbudować backend aplikacji. FastAPI wyróżnia się na tle innych webowych frameworków pythona możliwością obsługi asynchronicznych zapytań.

W projekcie FastAPI jest odpowiedzialny za przyjmowanie requestów i zwracanie odpowiednich odpowiedzi niezbędnych do poprawnego działania aplikacji. FastAPI automatycznie generuje dokumentację zgodną z OpenAPI przy użyciu interaktywnego Swagger UI. Serwer jest obsługiwany przez lekki framework Uvicorn wspierający asynchroniczne działanie FastAPI.

#### 2.3.1.2 Beautiful Soup

Beautiful Soup jest pakietem pozwalającym na zebranie z danej strony jej kodu HTML oraz wyciągnięcie z niego potrzebnych informacji tak zwanego web scrapingu. W projekcie posłużył do zbierania danych do nauki znaków za pomocą stworzonych na te potrzebę skryptów.

#### 2.3.1.3 SQLAlchemy

SQLAlchemy to biblioteka Pythona służąca obsłudze i zarządzaniu relacyjnymi bazami danych SQL. Biblioteka ta pozwala definiować klasy, które w sposób automatyczny stworzą tabelę w bazie danych. SQLAlchemy posiada ORM, dzięki któremu można zarządzać bazą danych za pomocą obiektów bez pisania czystych zapytań SQL, co z kolei chroni przed atakami typu SQL injection.

#### 2.3.1.4 Pandas

Pandas jest biblioteką służąca do manipulacji danych. W projekcie używana głównie jest do obróbki danych zebranych za pomocą web scrapingu. Pandas pozwala na spreparowanie danych gotowych do grupowego insertu do bazy danych.

#### 2.3.1.5 Bcrypt

Bcrypt to prosta biblioteka pozwalająca na haszowanie haseł, oparta na szyfrze Blowfish.

#### 2.3.1.6 Googletrans

Googletrans to biblioteka która korzystając z Google Translate Ajax APi pozwala na tłumaczenie słów lub zdań.

#### 2.3.1.7 Selenium

Biblioteka umożliwiająca zebranie kodu źródłowego strony HTML wraz   
z dynamicznie ładowaną zawartością. Używa silnika chrome, aby móc załadować stronę.

### 2.3.2 MySQL

MySQL jest systemem zarządzania relacyjną bazą danych. Zaletami MySQL jest:

-skalowalność - możliwość obsługi bardzo dużej ilości zapytań naraz przy bardzo dużej wydajności

-uniwersalność – działa na wielu platformach i systemach operacyjnych

-ochrona danych – MySQL posiada wiele mechanizmów pozwalających na skuteczną ochronę danych zapisanych w bazie jak uwierzytelnianie użytkownika, wparcie dla protokołów SSH i SSL, a także enkrypcja oraz dekrypcja danych.[13]

MySQL został wybrany ze względu na wydajność i wygodną platformę do zarządzania bazą w postaci phpmyadmin zapewnioną przez XAMPP. W projekcie MySQL posłużył   
do przechowywania, dodawania, usuwania czy edytowania danych użytkowników i zestawów do nauki.

# Rozdział 3

# Projekt i implementacja aplikacji

W tym rozdziale opisany został proces projektowania i implementacji aplikacji.

# 3.1 Wymagania funkcjonalne i niefunkcjonalne

Wymaganiami funkcjonalnymi są określane odpowiednie funkcje, czynności   
i operacje, które będą wykonywane przez aplikacje, zaś niefunkcjonalnymi określane są cechy systemu dotyczące wydajności, dostępności i skalowalności.

## 3.1.1 Wymagania funkcjonalne:

1. Możliwość rejestracji i logowania

2. Możliwość wylogowania

3. Moduł uczący w formie quizu posiadający wiele kategorii nauki na poziomach od N5 do N1

4. Inteligentne dobieranie przykładów na podstawie tego, jak często użytkownik poprawnie odpowiedział na dane pytanie

5. Możliwość uzupełniania bazy pytań o nowe przykłady

6. Wizualne przedstawienie aktualnego poziomu poznanych znaków

7. Zapisywanie wyników każdej sesji nauki

## 3.1.2 Wymagania niefunkcjonalne

1. Możliwość korzystania z systemu tylko po zalogowaniu

2. Responsywność – aplikacja wygodna do użytkowania na komputerze oraz urządzeniach mobilnych

3. Szybki czas reakcji aplikacji

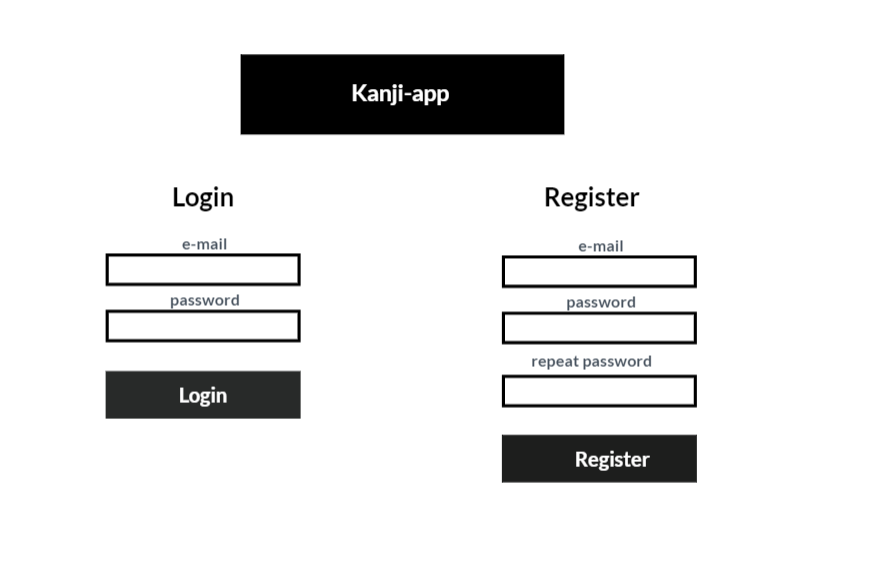
4. Pozbycie się ładowania podstron – wykorzystanie stanów biblioteki React

5. Krótki czas otrzymywania zestawu pytań od backendu aplikacji

# 3.2 Projekt wyglądu ekranów aplikacji

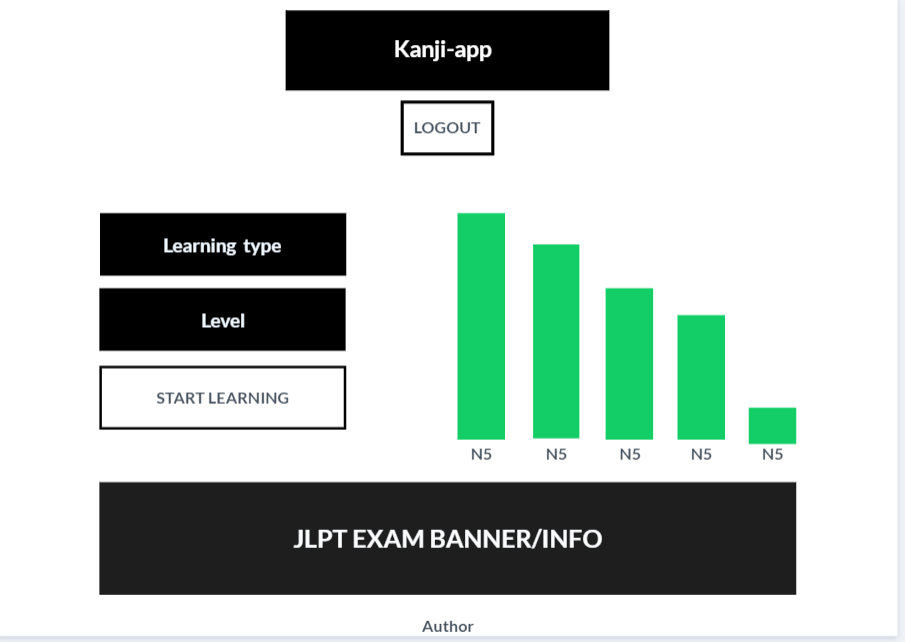
Przy projektowaniu wyglądu aplikacji jednym z podstawowych założeń była prostota interfejsu użytkownika. W wielu aplikacjach użytkownika przytłacza ilość funkcji   
i informacji, co nie pozwala skupić się na głównym celu jakim jest nauka.

Ekran logowania i rejestracji został połączony w jeden dla zmniejszenia konieczności ładowania kolejnych stron. Umieszczenie obydwu funkcjonalności na tej samej stronie pozwala zaoszczędzić czas użytkownikowi nie tracąc przy tym na przejrzystości aplikacji.



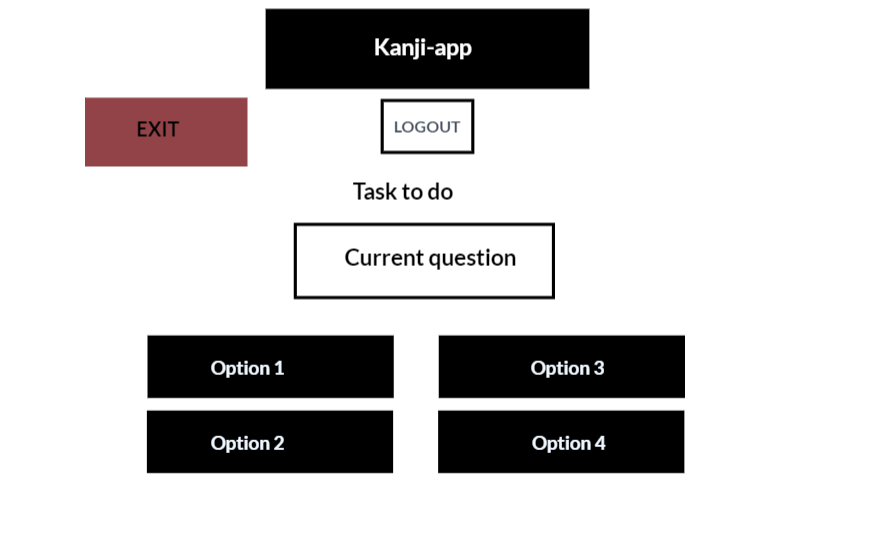
Rysunek 5 Projekt ekranu logowania i rejestracji

Główny ekran aplikacji pozwala łatwo zlokalizować opcje nauki i rozpocząć studiowanie już po kilkunastu sekundach po zalogowaniu. Posiada również podstawową opcję wylogowania użytkownika.



Rysunek 6 Projekt ekranu głównego aplikacji

Ekran odpowiadający za naukę został zaprojektowany w formie quizu gdzie użytkownik wybiera odpowiednie opcje dla danego znaku lub zadanego zdania. Każda odpowiedź jest przyciskiem który pozwala na wybór danej opcji.



Rysunek 7 Projekt ekranu nauki

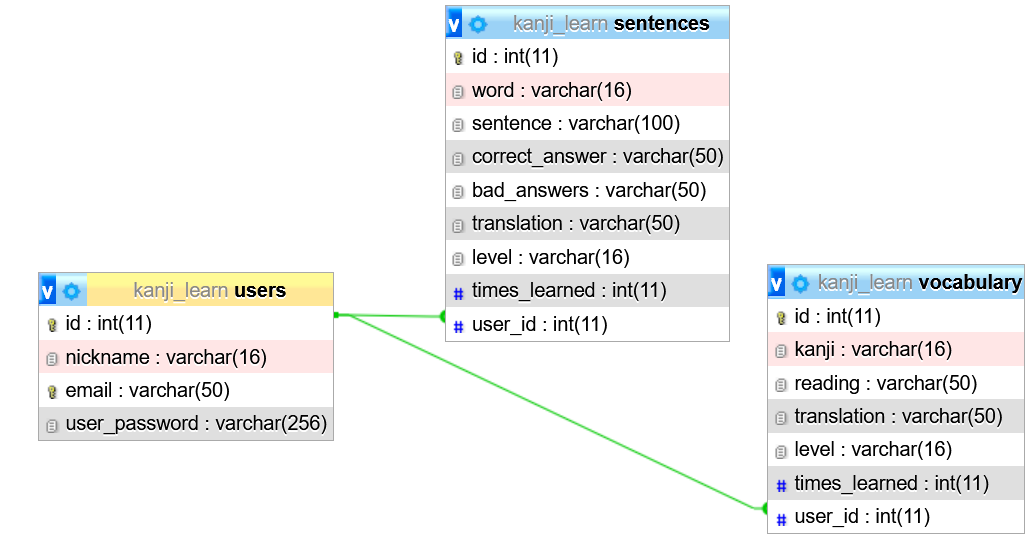
# 3.3 Projekt bazy danych

W projekcie została wykorzystana baza danych MySQL, backend komunikuje   
się z bazą za pomocą ORM biblioteki sqlalchemy. Dla każdej z tabeli zostały zdefiniowane odpowiednie modele, które określają pola, indeksy i relacje z innymi tabelami.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek Zdefiniowane modele sqlalchemy



Rysunek Struktura bazy danych

# 3.4 Skrypt pobierania danych

Aplikacja do nauki znaków opiera się na danych które zostaną wykorzystane podczas generowania zestawów uczących. Dane do nauki i zakres nie jest niestety oficjalnie udostępniany przez organizatorów JLPT. Mimo braku oficjalnych danych w internecie można znaleźć wiele darmowych słowników pogrupowanych przez użytkowników. W aplikacji wykorzystany został słownik ze strony japanesetest4you[16]. Do zebrania danych słownikowych użyto biblioteki pythona **Beautiful Soup,** która pozawala na tzw. web scrapping czyli zbieranie informacji z kodu html strony. Zebrane dane zostały następnie poddane odpowiedniej obróbce jak podział na kolumny i wpisane do pliku csv.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 10 Główne wywołanie skryptu zbierającego dane słownictwa

Po zapisaniu pliku, dane zostały dodane do tabeli **vocabulary** przy użyciu biblioteki Pandas zapewniającej łatwą integracje sqlalchemy z plikami csv.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 11 Dodawanie słownictwa do bazy danych

Następnie aby zdobyć zdania do nauki kontekstowej pobierane są dane z oficjalnej strony informacyjnej NHK news[17]. Problem stwarzają dane ładowane dynamicznie które nie pokazują się w dokumencie HTML. Aby pobrać takie dane potrzebne jest załadowanie najpierw całego kodu źródłowego używając biblioteki **selenium**. Wykorzystuje ona silnik chrome, po czym odczytuje zawartość strony, którą można analizować już przy użyciu Beautiful Soup.

Ze strony pobrane zostały zdania, następnie porównano występujące znaki kanji   
z obecnymi w bazie danych, jeśli dany znak się zgadzał wpisywany zostawał do listy.

Na koniec, aby zapewnić tłumaczenie znaku użyta została biblioteka **googletrans** która używa API udostępnionego przez google na tłumaczenie przez wybrany przez użytkownika język. Zebrane dane zostają następnie wpisane do tabeli **sentences**.

Skrypt do uzupełniania przykładów zdań może być uruchamiany co jakiś czas kiedy pojawią się nowe artykuły, aby dodać nowe dane.

Dużą wadą rozwiązania bazującego na takim zbieraniu danych jest to, że jeśli twórcy strony NHK news zmienią strukturę strony, web scrapper przestanie działać w pożądany sposób co spowoduje błędy. Najlepszym rozwiązaniem jest monitorowanie działania skryptu   
i wprowadzanie ewentualnych poprawek w przypadku awarii.

# 3.5 Projekt i implementacja funkcjonalności aplikacji

## 3.5.1 Rejestracja i logowanie do aplikacji

Żeby móc korzystać z funkcjonalności aplikacji należy dokonać pierw rejestracji,   
a następnie się zalogować.

### 3.5.1.1 Rejestracja

Rejestracja wymaga od użytkownika podania:

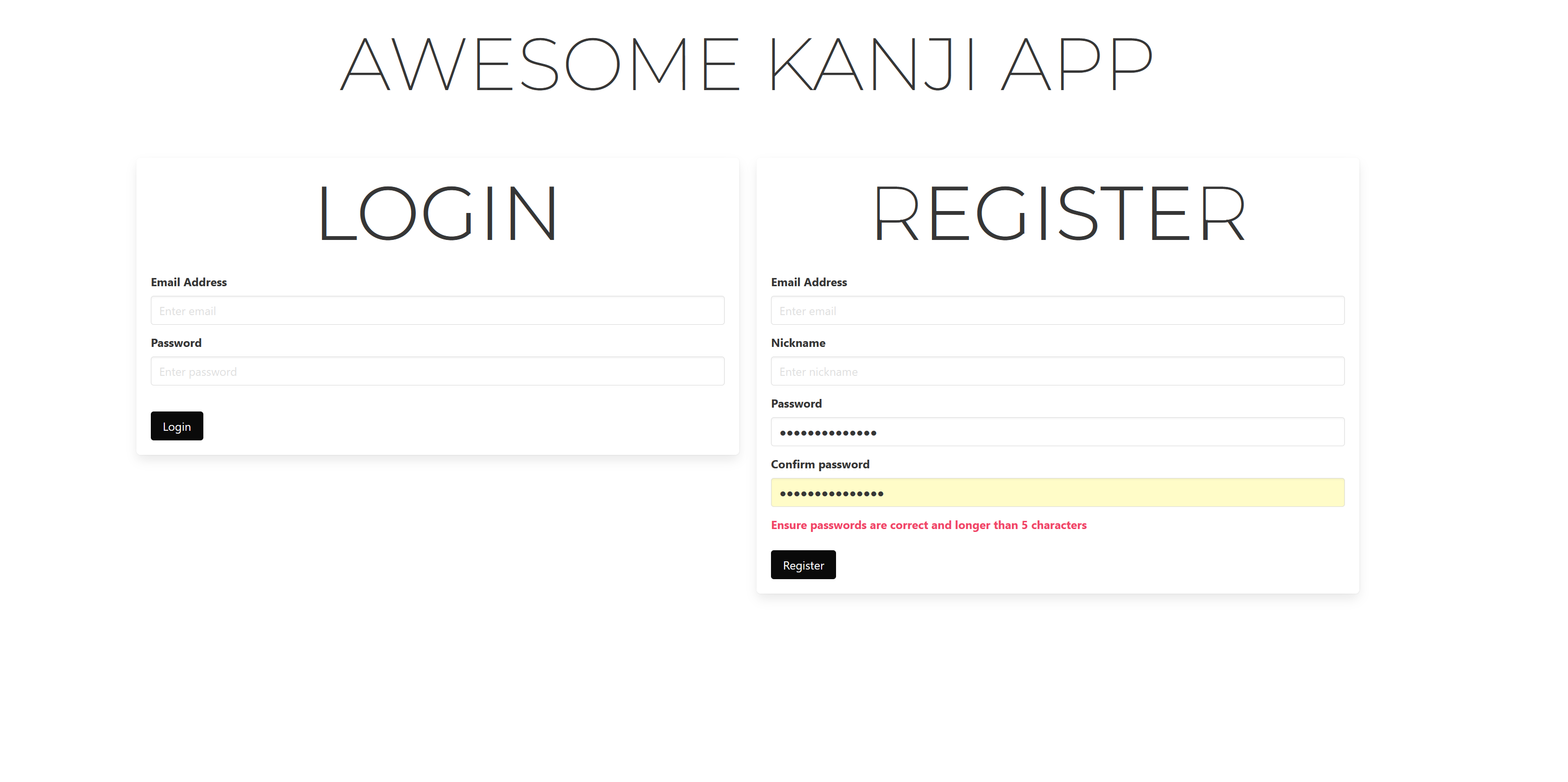
-adresu e-mail – weryfikowanie jest czy adres posiada odpowiedni format tzn. identyfikator użytkownika, znak @ oraz pełną nazwę domenową serwera poczty elektronicznej

-nickname’u – pseudonim w przeciwieństwie do adresu e-mail nie musi być wyjątkowy i może się powtarzać z istniejącym już w bazie danych Nickiem innego użytkownika

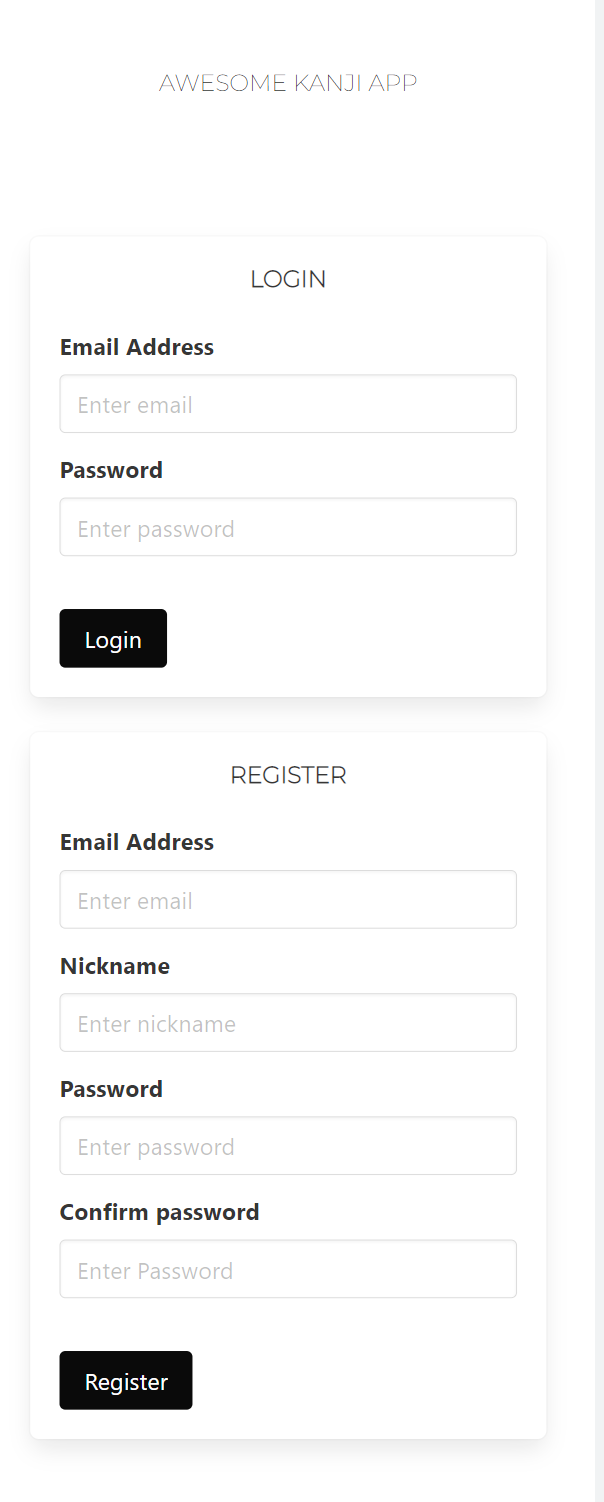
-hasła oraz potwierdzenia hasła poprzez ponowne wpisanie – hasło posiada ustawione bardzo luźne restrykcje bo tylko 5 znaków. Podczas projektowania aplikacji hasło miało początkowo posiadać wymóg 8 znaków, wielkich i małych liter oraz znaku specjalnego.

Finalnie w aplikacji zrezygnowano z tak skomplikowanych zabezpieczeń ponieważ zwiększało to ryzyko wykorzystania przez użytkowników haseł z social mediów czy co gorsza haseł bankowych. Aplikacja nie posiada żadnych wrażliwych danych osobowych, a więc potencjalny wyciek danych lub włamanie na konto nie wyrządzi tak dużych szkód jak gdyby haker poznał „silne” hasło użytkownika.

W przypadku podania zbyt krótkiego hasła lub popełnienia błędu pojawia się wiadomość błędu.



Rysunek 12 Ekran logowania i rejestracji z aplikacji wraz z błędem hasła



Rysunek 13 Ekran logowania i rejestracji w widoku mobilnym

Po poprawnym wypełnieniu formularza i zatwierdzeniu, zostaje on przesłany na backend aplikacji, tam dane zostają wpisane do modelu.

Hasło zostaje dodatkowo „haszowane” funkcją bcrypt która została zaprojektowana przez Nielsa Provosa i Davida Mazièresa, wzorując się na szyfrze Blowfish. Algorytm ten wprowadza do haszowania „sól” czyli losową wartość doklejaną do każdego hasła przed wykonaniem haszowania co broni przed złamaniem hasła przy użyciu tablic tęczowych.

Wynikiem rejestracji jest wpisanie użytkownika do tabeli *users* i zalogowanie użytkownika.

Obraz zawierający stół

Opis wygenerowany automatycznie

Rysunek 14 struktura tabeli users

### 3.5.1.2 Logowanie

Logowanie do aplikacji odbywa się poprzez podania do formularza poprawnego adresu e-mail i hasła.

Po wysłaniu danych z formularza za pomocą metody POST na backend aplikacji, dane są weryfikowane. Najpierw weryfikowane jest hasło z odpowiednim hasłem w bazie danych przypisanym do danego użytkownika. Odbywa się to przy pomocy metody zapewnionej przez bibliotekę bcrypt . Jeśli wszystkie dane są poprawne wygenerowany zostaje JWT token, który jest przechowywany w lokalnym magazynie danych (local storage) przeglądarki.

JWT to standard definiujący sposób wymiany danych w bezpieczny sposób poprzez obiekt JSON, przesyłane weryfikowane są przy pomocy cyfrowego podpisu będącego częścią tokenu. JWT pozwala na uzyskanie dostępu do zasobów strony dostępnych tylko dla zalogowanych użytkowników. Token posiada 3 części składowe:

1. Nagłówek – zawiera on informację o rodzaju tokena oraz używanego algorytmu szyfrującego
2. Zawartość – są to dane które przesyłamy za pomocą tokenu np. dane identyfikacyjne
3. Sygnatura – jest to podpis cyfrowy który potwierdza autentyczność danych zawartych w JWT.



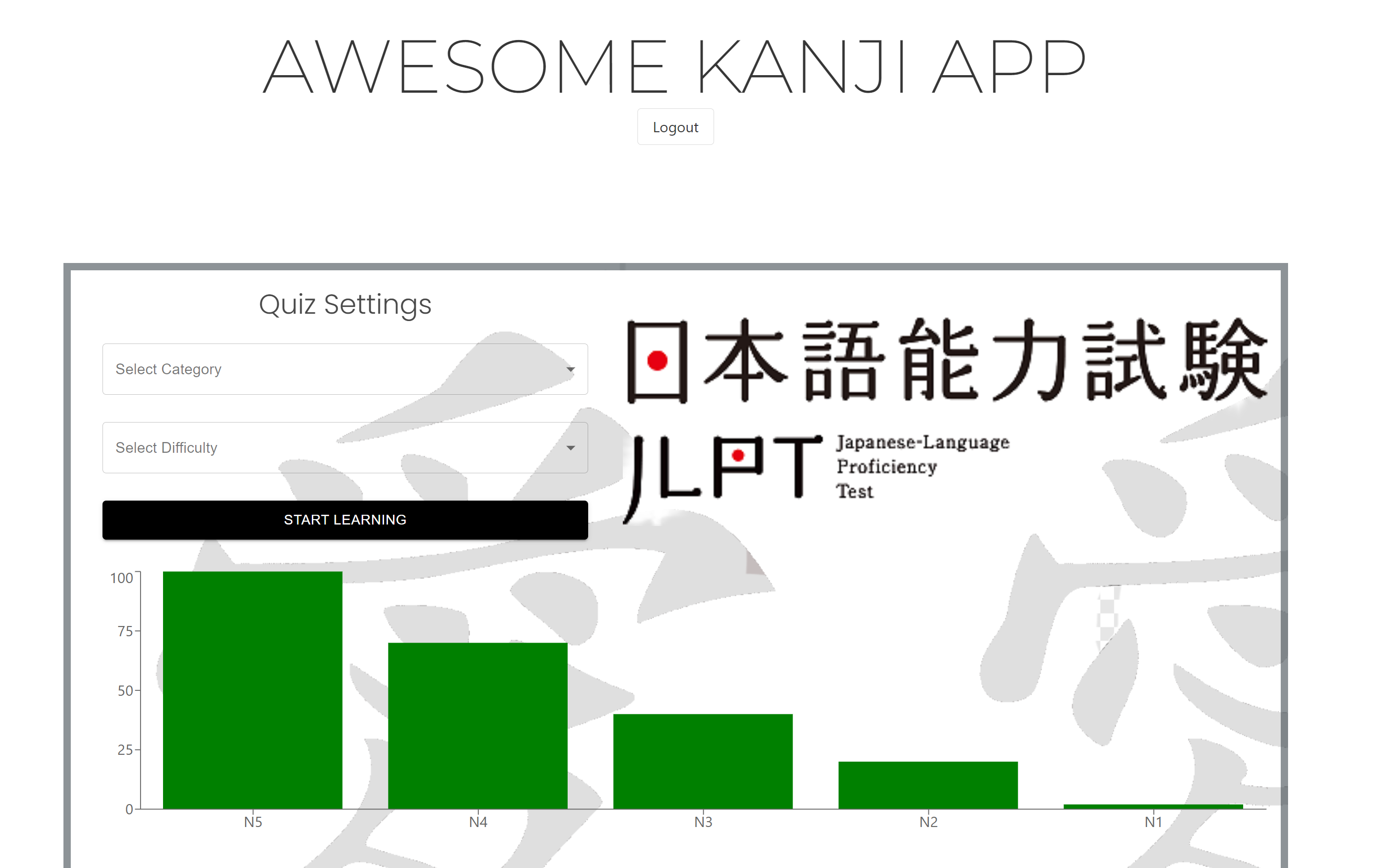
Rysunek 15 Schemat uzyskiwania JWT.

Rysunek 11 [14] przedstawia schemat autoryzacji za pomocą JWT:

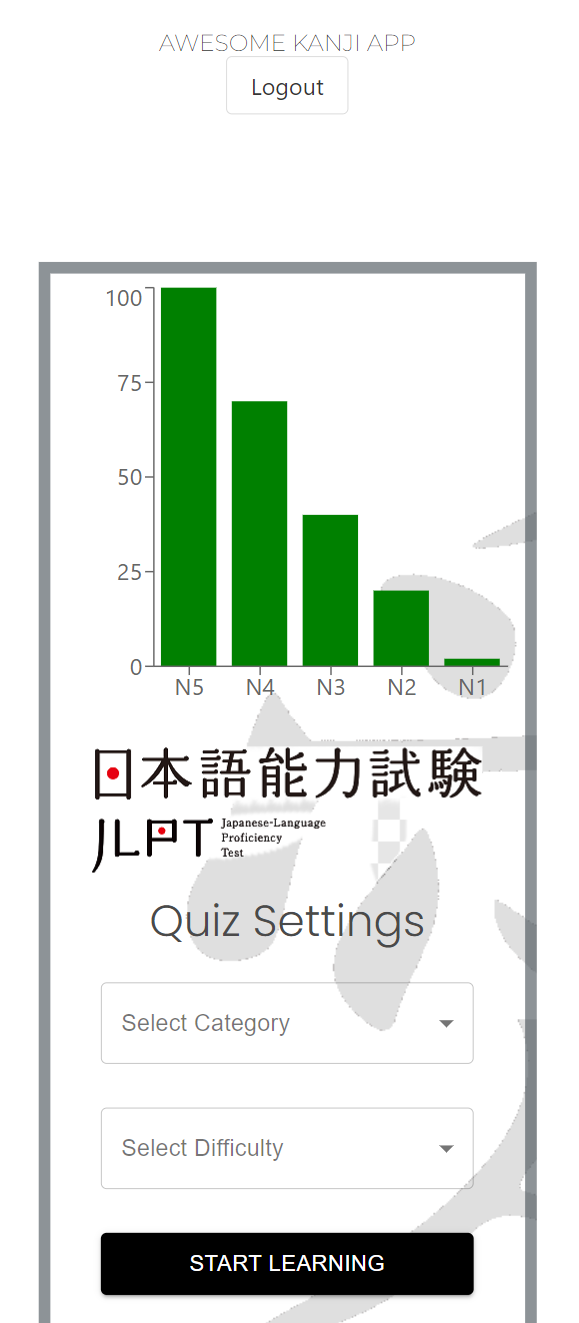
1. Wysłanie danych do funkcji generującej token
2. Uzyskanie tokenu
3. Autoryzacja użytkownika przy pomocy tokenu.

## 3.5.2 Strona główna aplikacji

Po zalogowaniu się do aplikacji widoczna jest strona główna aplikacji.



Rysunek 16 Strona główna aplikacji



Rysunek 17 Strona główna aplikacji w widoku mobilnym

Na ekranie strony głównej możemy wydzielić cztery podstawowe sekcje:

1. Baner z nazwą aplikacji wraz z przyciskiem wylogowania
2. Moduł uczący – quiz z pytaniami
3. Baner testu znajomości języka japońskiego dla obcokrajowców będący również odnośnikiem do strony głównej testu gdzie możemy się zapisać na JLPT
4. Wykres słupkowy postępu nauki znaków z danego poziomu

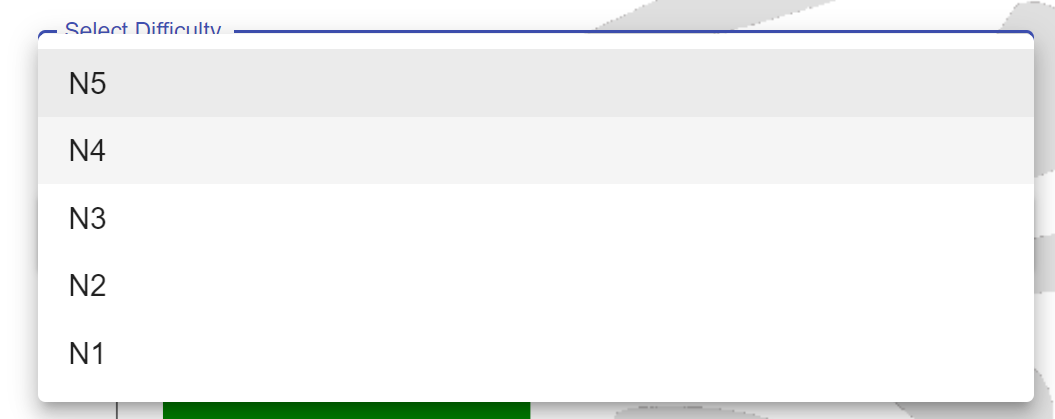
## 3.5.3 Moduł uczący – quiz z pytaniami

Aby wejść do modułu uczącego należy wybrać odpowiednie opcje ze strony głównej

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 18 Dostępne typy nauki



Rysunek 19 Dostępne poziomy trudności

W aplikacji zostały zaimplementowane następujące moduły widoczne na ryc. 14:

- Sentence context learning – nauka znaków poprzez wybieranie odpowiedniego kanji zależnie od kontekstu w zdaniu.

- Kanji reading – wybór odpowiedniego czytania wybranego słowa zapisanego znakiem kanji

- Kanji meaning – wybór prawidłowego znaczenia znaku

- Kanji from reading – wybór odpowiedniego znaku z czytania

- Kanji from meaning – wybór odpowiedniego znaku na podstawie znaczenia

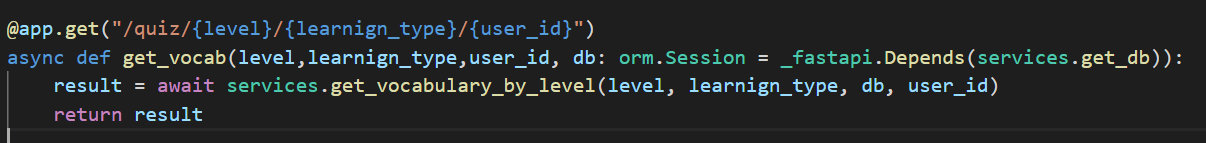
Po wyborze odpowiedniej opcji i naciśnięciu przycisku ”START LEARNING”, ustawiana jest wartość kategorii, poziomu trudności i id użytkownika. Wysyłana jest metoda **GET** do backendu.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 20 Wywołaie metody GET przez frontend

Wywołanie powoduje odpowiedź endpointu zaimplementowanego w bibliotece pythona – FastAPI.



Rysunek 21 Endpoint do uzyskania pytań do modułu uczącego

Po przekazaniu odpowiednich argumentów backend aplikacji zwraca odpowiednio opakowane dane w formie obiektu JSON jak przedstawiono na rys. 18.

Dokumentacja z której został wykonany przykładowy GET to OpenAPI który jest dostępny wraz z FastAPI. Jest to bardzo wygodne narzędzie, które generuje dokumentacje wszystkich endpointów zdefiniowanych w aplikacji wraz z parametrami oraz odpowiedziami danego endpointu.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 22 Przykładowe wywołanie z ednpointu za pomocą interaktywnej dokunetacji OpenAPI

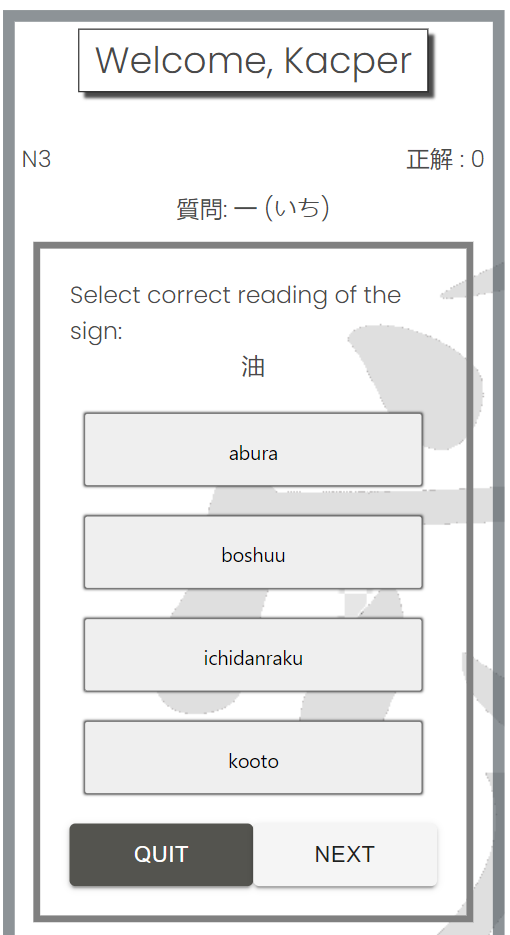
Aby uzyskać dane do quizu uczącego odpytywana jest baza danych przy pomocy ORM sqlalchemy. Jest to bezpieczniejsze rozwiązanie od używania czystego zapytania SQL ponieważ zapobiega atakom typu SQL injection. Podatność ta polega na „wstrzyknięciu” złośliwego kodu SQL który może spowodować wyciek danych lub ich usunięcie.

Wybierane są pytania z określonego poziomu i co ważne priorytetowo będą pokazywane pytania których użytkownik jeszcze nie poznał lub odpowiadał na nie źle w przeszłości.

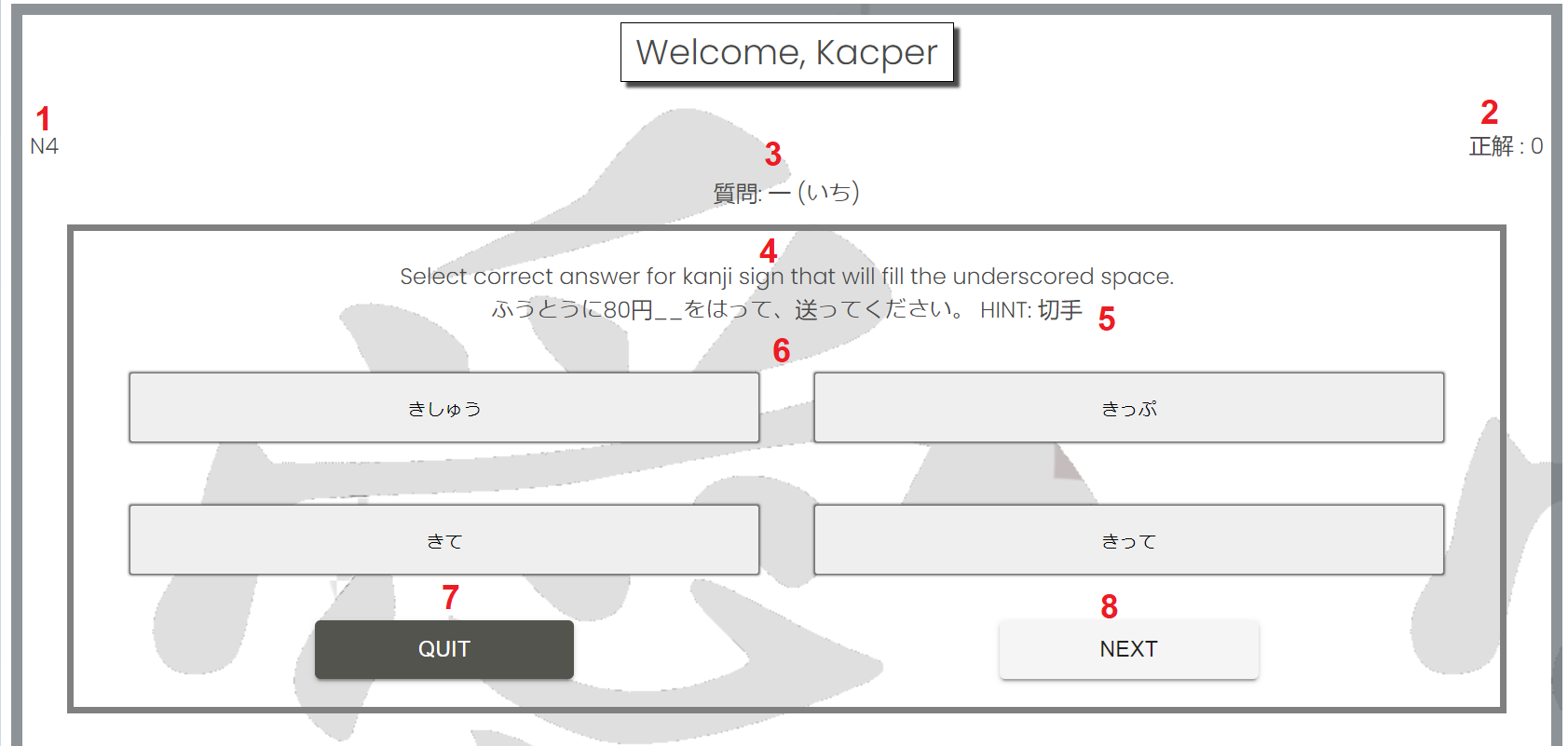
result = db.query(models.Vocabulary).filter(models.Vocabulary.level == level).filter(models.Vocabulary.user\_id == user\_id).order\_by(models.Vocabulary.times\_learned.asc()).limit(10).all()

Kod odpowiedzialny za odpytanie bazy danych po dane uczące za pomocą ORM

Po wyborze odpowiednich opcji zostaje załadowany ekran quizu uczącego, w którym użytkownik może wybrać odpowiednią odpowiedź. Każdy moduł składa się z dziesięciu pytań. Na ten moment nie ma możliwości zmiany ilości pytań na jeden quiz, ponieważ aplikacja miała pozwalać na szybkie lekcje i powtórki. Możliwe będzie w przyszłości rozszerzenie strony o takie możliwości.



Rysunek 23 Moduł quizu dla widoku mobilnego



Rysunek 24 Moduł quizu wraz z podpisanymi numerami poszczególnymi elementami

W quizie uczącym można wydzielić następujące elementy:

1. Poziom trudności aktualnego quizu
2. Liczba zdobytych punktów
3. Numer pytania (numer jest zapisany w języku japońskim)
4. Polecenie dla wybranego quizu
5. Aktualne zdanie lub znak dla danego pytania
6. Dostępne odpowiedzi
7. Przycisk wyjścia – przeniesie nas do strony głównej bez zapisywania postępów
8. Przycisk przejścia do następnego pytania

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 25 Moduł quizu po wyborze odpowiedzi

Wybranie odpowiedzi powoduje zaznaczenie prawidłowej na zielono, zaś błędnej na czerwono. Aplikacja nie przechodzi automatycznie do następnego pytania, dzięki temu użytkownik może od razu zweryfikować swoją odpowiedź. Po przejściu dalej zapisywany do stanu jest fakt, czy udzielona odpowiedź była prawidłowa oraz sumowane są punkty.

W przypadku próby przejścia do następnego pytania kiedy nie zaznaczymy żadnej odpowiedzi na ekranie generowana jest wiadomość błędu ryc. 22.



Rysunek 26 Wiadomość błędu w przypadku braku zaznaczenia odpowiedzi

## 3.5.4 Wyniki quizu

Po skończeniu się pytań przycisk **NEXT** przenosi na ekran wyników, gdzie można zobaczyć podsumowanie całego quizu.

Obraz zawierający stół

Opis wygenerowany automatycznie

Rysunek 27 Ekran wyniku wraz z podsumowaniem odpowiedzi

Na ekranie wyników w górnej sekcji przedstawiona jest liczba prawidłowych odpowiedzi oraz przycisk zapisujący wyniki i przenoszący użytkownika na stronę główną.

Poniżej znajduje się tabela podsumowująca przebieg całego quizu. Tabela posiada trzy stałe kolumny:

**Kanji** - znak japoński

**Sentence** – zdanie w języku japońskim lub Reading – czytanie danego znaku (zależnie od wybranego trybu uczenia)

**Translation** – tłumaczenie danego znaku

**Selected answer** – wybrana przez użytkownika odpowiedź w czasie quizu

**Correct answer** – prawidłowa odpowiedź z danego modułu uczącego

Przycisk SAVE RESULTS AND RETURN powoduje wysłanie do backendu aplikacji metody POST która zawiera w sobie zapis o postępach użytkownika. Każda poprawna odpowiedź zostaje wpisane do danego przykładu w tabeli vocabulary lub sentences na polu **times\_learned.** Dodatkowo aktualizowana jest informacja jaki procent znaków w danym przedziale zaawansowania został poznany. W odpowiedzi do frontendu przychodzi aktualny procent poznanych znaków, a następnie metoda ustawia wynik ponownie na wartość zero   
i przenosi użytkownika na stronę główną, gdzie może zobaczyć zaktualizowane postępy   
w formie wykresu słupkowego.

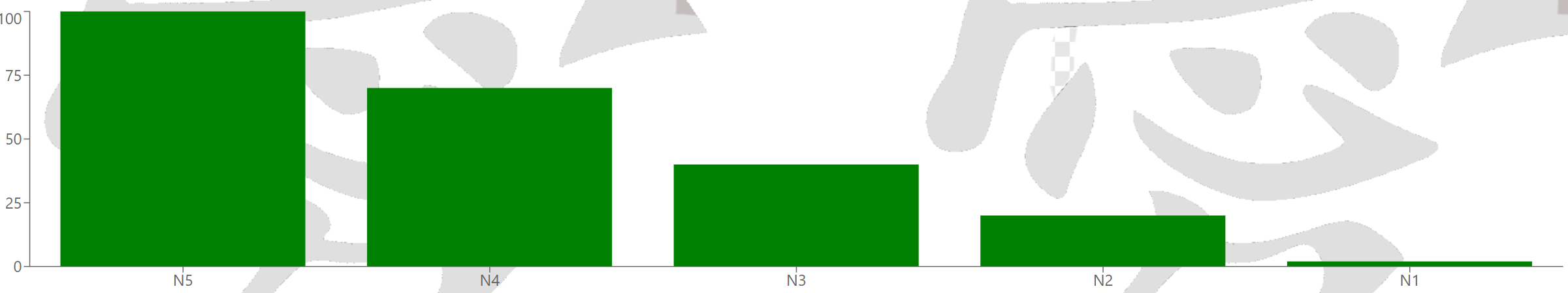
Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek Metoda POST wywoływana przy zapisywaniu wyników

## 3.5.5 Wykres słupkowy postępów

Wykres na stronie głównej aplikacji jest generowany za pomocą biblioteki recharts, biblioteka ta umożliwia stworzenie wykresów. Atutem biblioteki jest komponent ResponsiveContainer który zapewnia responsywność wykresu.



Rysunek 29 Wykres znajomości znaków z każdego poziomu zaawansowania

Obraz zawierający tekst, ekran, zrzut ekranu

Opis wygenerowany automatycznie

Rysunek 30 Implementacja komponentu wykresu

Po każdym zakończonym module uczącym, wartość postępu użytkownika zostaje zaktualizowana i przedstawiona w formie wykresu.

# Przeprowadzenie testów

## Testy funkcjonalne

Testy funkcjonalne zwane też testami czarnej skrzynki to testy które sprawdzają aplikację z perspektywy osoby która będzie potencjalnym użytkownikiem. Testy funkcyjne dają wgląd na to jak aplikacja reaguje na określone zachowania użytkownika i jak realizuje poszczególne zadania [18]

### Rejestracja

Analiza scenariuszy testowych dla rejestracji:

1. Poprawna rejestracja poprzez wpisanie prawidłowego adresu e-mail, nicku oraz hasła dłuższego niż 4 znaki

**Oczekiwany rezultat:** poprawne zarejestrowanie użytkownika i zalogowanie do aplikacji

1. Nieudana rejestracja przy próbie rejestracji bez wypełnienia formularza

**Oczekiwany rezultat:** komunikat mówiący o niewypełnionych polach

1. Nieudana rejestracja przy wpisaniu adresu e-mail w złym formacie

**Oczekiwany rezultat:** komunikat o błędzie dotyczącym nieprawidłowego adresu e mail.

1. Nieudana rejestracja po wpisaniu hasła o mniejszej liczbie znaków niż 5

**Oczekiwany rezultat:** komunikat błędu o zbyt krótkim haśle

1. Nieudana rejestracja po wpisaniu nie pokrywających się haseł w formularzu

**Oczekiwany rezultat:** komunikat błędu o nieprawidłowych hasłach

**Rezultat:** aplikacja pomyślnie przeszła wszystkie testy związane z rejestracją użytkownika. Komunikaty o błędach zostały poprawnie wyświetlone, zaś po poprawnym wypełnieniu formularza następuje założenie konta użytkownika i zostaje on zalogowany.

### Logowanie

Analiza scenariuszy testowych do logowania:

1. Poprawne logowanie w przypadku wpisania prawidłowego adresu e-mail oraz hasła

**Oczekiwany rezultat:** poprawne zalogowanie i przeniesienie na stronę główną

1. Nieudane logowanie przy niewypełnieniu formularza żadnymi danymi

**Oczekiwany rezultat:** komunikat o braku danych i prośbie wypełnienia pól

1. Nieudane logowanie przy wpisaniu poprawnego adresu e-mail, ale błędnego hasła

**Oczekiwany rezultat:** komunikat o nieprawidłowych danych

1. Nieudane logowanie przy wpisaniu poprawnego hasła, ale błędnego adresu e-mail

**Oczekiwany rezultat:** komunikat o nieprawidłowych danych

**Rezultat:** aplikacja pozytywnie przeszła wszystkie przypadki testowe. W przypadku podania prawidłowych danych użytkownik zostaje prawidłowo zalogowany i przeniesiony na stronę główną.

Wszystkie komunikaty zostały poprawnie wyświetlone. Przy podawaniu nieprawidłowych danych wiadomość jest ogólna (nie precyzuje który element jest błędny), aby nie ułatwiać łamania dostępu przez osoby trzecie.

### Zabezpieczenia ścieżek linku

Analiza scenariuszy testowych dostępu do postron:

1. Próba wejścia na stronę quizu (/quiz) bez zalogowania się do aplikacji

**Oczekiwany rezultat:** przekierowanie na stronę logowania i rejestracji

1. Próba wejścia na nieistniejącą podstronę po zalogowaniu do aplikacji

**Oczekiwany rezultat:** pokazanie strony z kodem 404 i przyciskiem przekierowującym do strony głównej.

1. Próba wejścia do aplikacji na stronę główną po wcześniejszym zalogowaniu i zamknięciu karty z aplikacją.

**Oczekiwany rezultat:** poprawne pokazanie strony głównej aplikacji dla danego zalogowanego użytkownika

**Rezultat:** aplikacja zachowała się tak jak oczekiwano. Kod błędu został poprawnie wyświetlony i weście dla wcześniej zalogowanego użytkownika zapewnione.

### Moduł quizu

Analiza zachowania komponentu quizu włącznie z elementem na stronie głównej odpowiedzialnym za kategorie nauki i wybór trudności:

1. Próba rozpoczęcia modułu uczącego po prawidłowym wyborze dostępnych opcji dla kategorii i poziomu trudności

**Oczekiwany rezultat:** poprawne wejście na podstronę quizu dla wybranego trybu nauki oraz poziomu trudności

1. Próba rozpoczęcia quizu bez wyboru kategorii nauki i bez wyboru poziomu trudności.

**Oczekiwany rezultat:** wyświetlenie komunikatu z prośbą o wybranie kategorii, oraz poziomu trudności

1. Próba rozpoczęcia quizu z wybranym jedynie poziomem trudności

**Oczekiwany rezultat:** wyświetlenie komunikatu z prośbą o wybranie kategorii

1. Próba rozpoczęcia quizu z wybraną jedynie kategorią

**Oczekiwany rezultat:** wyświetlenie komunikatu z prośbą o wybranie poziomu trudności

1. Zaznaczenie poprawnej odpowiedzi i próba przejścia do następnego pytania

**Oczekiwany rezultat:** zaznaczenie odpowiedzi na kolor zielony i przejście do kolejnego pytania

1. Zaznaczenie błędnej odpowiedzi i próba przejścia do następnego pytania

**Oczekiwany rezultat:** zaznaczenie wybranej odpowiedzi na czerwono, poprawnej na zielono i przejście do kolejnego pytania

1. Próba przejścia do kolejnego pytania bez zaznaczenia odpowiedzi

**Oczekiwany rezultat:** komunikat błędu proszący o zaznaczenie odpowiedzi

1. Próba zaznaczenia kolejnej odpowiedzi po wcześniejszym wybraniu innej.

**Oczekiwany rezultat:** brak możliwości zmiany odpowiedzi, kolejne kliknięcia nie zmieniają zaznaczonej opcji.

1. Wyjście do strony głównej nie kończąc quizu

**Oczekiwany rezultat:** przejście do strony głównej bez zapisywania dotychczasowych wyników

1. Przejście do kolejnego pytania kiedy jesteśmy na ostatnim zadaniu

**Oczekiwany rezultat:** wyświetlenie tablicy wyników z poprawnie i błędnie zaznaczonymi odpowiedziami

1. Kliknięcie przycisku zapisującego wyniki na ekranie podsumowania quizu

**Oczekiwany rezultat:** zapis danych o poprawnych odpowiedziach do bazy danych i przejście na stronę główną

**Rezultat:** aplikacja zachowała się jak oczekiwano dla wszystkich przypadków testowych. Pytania zostały poprawnie wyświetlone i przebieg modułu uczącego odbył się według założonych scenariuszy. W przypadku błędów, na ekranie zostały przedstawione odpowiednie komunikaty. Rezultaty pokazały odpowiednio jak i błędnie zaznaczone odpowiedzi, a następnie poprawnie zapisane w bazie danych.

## Testy niefunkcjonalne

Testy niefunkcjonalne pozwalają dowiedzieć się jak działa system. W przypadku aplikacji takimi testami będą to głównie testy wydajnościowe. Wyniki testów pozwalają ocenić czy strona mieści się we współczesnych standardach i ewentualnie wskazać miejsca które wymagają poprawy.

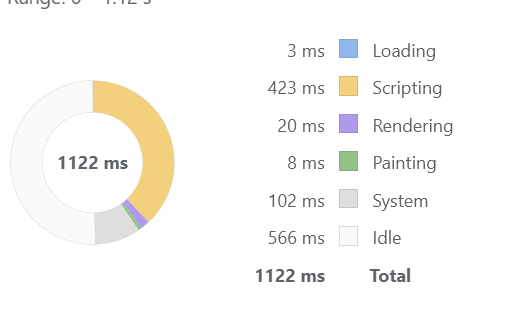
Do przeprowadzenia testów zostało użyte narzędzie developerskie wbudowane w przeglądarkę Google Chrome. Badana będzie wydajność strony w czasie wykonania, dodatkowo mierzona będzie ilość klatek na sekundę podczas działania strony. Badanie jest przeprowadzane dla przeglądarki komputerowej i dla urządzeń mobilnych. Opcja z której skorzystano podczas badania to symulowanie throttlingu CPU, aby przekonać się czy aplikacja będzie działała prawidłowo również na słabszych urządzeniach.

Poszczególne elementy strony mają różne czasy wykonania które zostały zinterpretowane odpowiednio na stronie web.dev[19] która jest polecana w artykule Google[20] na podstawie którego wykonano badania.

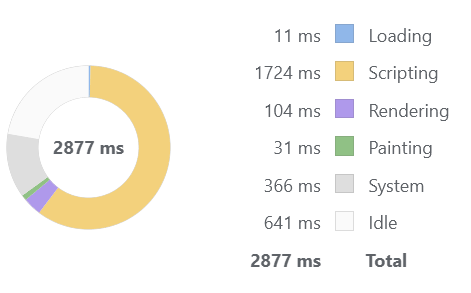
|  |  |
| --- | --- |
| 0 – 16 ms | Użytkownicy odbierają akcję lub animację jako płynną, ponieważ w każdej sekundzie renderowanych jest 60 klatek. Oznacza to 16 ms na klatkę. |
| 0 – 100 ms | W tym zakresie użytkownik ma wrażenie, że reakcje na jego działania są natychmiastowe. |
| 100 – 300 ms | Użytkownicy odczuwają niewielkie, odczuwalne opóźnienie. |
| 300 - 1000 ms | Działania przebiegają powoli, a użytkownik dostrzega, że coś trzeba zrobić z działaniem strony |
| +1000 ms | Użytkownicy tracą koncentrację na wykonywanym zadaniu. |
| +10 s | Użytkownicy są sfrustrowani i prawdopodobnie wyjdą. |

Tabela 1 Interpretacja czasów działania strony na podstawie witryny web.dev[19]

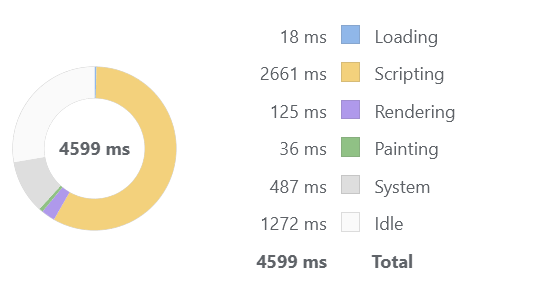
**Badanie dla urządzeń mobilnych**



Rysunek 31 Badanie dla urządzenia mobilnego bez spowolnienia z szybkim łączem internetowym



Rysunek 32 Badanie dla urządzenia mobilnego ze średniej półki cenowej z szybkim łączem 3G

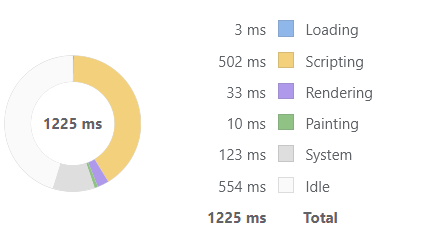


Rysunek 33 Badanie dla słabego urządzenia mobilnego z wolnym łączem 3G

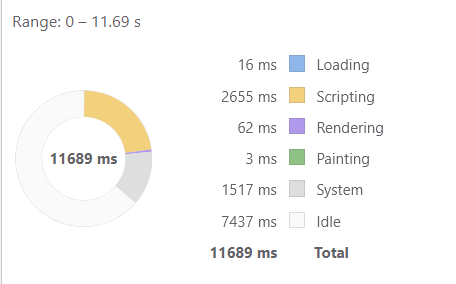
**Interpretacja wyników**

Wyniki dla ładowania przy wszystkich krokach mieściły się w granicy 100 ms nawet przy słabszych telefonach. Jedynym miejscem gdzie aplikacja ma wyraźne problemy jest proces działania skryptów. Nawet przy symulowaniu szybkiego telefonu wraz z dobrym połączeniem internetowym długość wykonywania skryptu trwała 423 ms co znacznie przekracza oczekiwany czas mieszczący się w 100 ms. Takie spowolnienie jest dostrzegalne przez użytkownika i powinno być zredukowane. Żadne z urządzeń nie przekracza krytycznego czasu 10 sekund, ale już przy telefonie ze średnimi specyfikacjami osiągany jest czas ponad 1 sekundy który spowoduje u użytkownika utratę koncentracji na wykonywanym zadaniu.

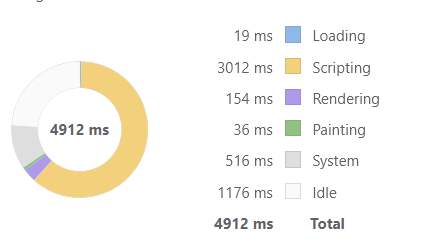
**Badanie dla przeglądarki komputerowej**



Rysunek Badanie dla przeglądarki internetowej bez spowolnienia i szybkiego łącza internetowego



Rysunek Badanie dla przeglądarki internetowej z 4 krotnie spowolnionym procesorem i szybkiego łącza 3G



Rysunek Badanie dla przeglądarki internetowej z 6 krotnie spowolnionym procesorem i wolnego łącza 3G

**Interpretacja wyników**

Wyniki dla przeglądarki komputerowej bez spowolnienia przy szybkim łączu internetowym były bardzo dobre dla ładowania, renderowania i rysowania strony (rys. 34). Problemem ponownie jak w przypadku symulacji mobilnego środowiska jest wykonywanie skryptów które nie mieści się w pożądanym czasie poniżej 100ms.

W przypadku spowolnienia procesora czterokrotnie (rys. 35) czas wykonywania skryptu przekraczał 2 sekundy co spowoduje utratę uwagi użytkownika.

Przy sześciokrotnym spowolnieniu (rys. 36) czas ten wyniósł 3 sekundy, a dodatkowo renderowanie strony trwało ponad 154 sekundy co również przekracza pożądany czas wynoszący mniej niż 100ms.

## 4.3 Podsumowanie testów

Wszystkie najważniejsze elementy aplikacji zostały pokryte testami i przeszła pozytywnie testy funkcjonalne. Strona zachowuje się tak jak założono w scenariuszach testowych. Testy wykonywano manualnie.

Testy niefunkcjonalne pozwoliły rzucić światło na problem aplikacji którym jest zbyt długie wykonywanie skryptów. Żeby poprawić działanie strony należy wykonać optymalizację poprzez podział kodu i wysyłanie użytkownikom tych jego fragmentów które są aktualnie potrzebne. Można również skompresować obecny kod i usunąć nieużywane jego fragmenty.

Mimo części związanej ze skryptami aplikacja działa szybko i poprawnie, ale wyniki testów pomogą w poprawie strony.

# Podsumowanie oraz wnioski

# Bibliografia

1. <https://www.jpf.go.jp/j/project/japanese/survey/result/dl/survey2018/Report_all_e.pdf>
2. Tackling the Kanji hurdle: Investigation of Kanji learning in Non-Kanji background learners - *Simon Paxton, Chavalin Svetanant*
3. Japanese Language Educators’ Strategies for and Attitudes toward Teaching Kanji - *Hideko Shimizu, Kathy E. Green*
4. Achieving unconscious recall of kanji: Can rote learning help? New Zealand Studies in Applied Linguistics - *Nesbitt, D.*
5. The effect of imagery-based mnemonics on the long-term retention of Chinese characters. Language Learning - *Wang, A. Y., & Thomas, M. H.*
6. "Basic Kanji Book", *Chieko Kano*
7. "Intermediate Kanji Book", *Chieko Kano*
8. "Kanji in Context", *Chieko Kano*
9. <https://insights.stackoverflow.com/survey/2021/#technology> (10.12.2021)
10. <https://developer.mozilla.org/pl/docs/Learn/JavaScript/First_steps/What_is_JavaScript> (15.12.2021)
11. Most popular web frameworks among developers *Shanhong Liu*
12. <https://fastapi.tiangolo.com> (28.12.2021)
13. <https://vavatech.pl/technologie/bazy-danych/mysql> (29.12.2021)
14. <https://blog.i-systems.pl/json-web-tokens-jwt/> (05.01.2022)
15. <https://wolski.pro/2015/02/rodzaje-wymagan/> (05.04.2022)
16. <https://japanesetest4you.com/> (07.04.2022)
17. <https://www3.nhk.or.jp/news/easy/> (07.04.2022)
18. <https://thestory.is/pl/proces/faza-wdrozenia/testy-funkcjonalne-aplikacji/> (10.04.2022)
19. <https://web.dev/rail/> (15.04.2022)
20. <https://developer.chrome.com/docs/devtools/evaluate-performance/> (15.04.2022)