

Kierunek: **Informatyka Techniczna (ITE)**
Specjalność: **Inżynieria Systemów Informatycznych (INS)**

PROJEKT

System do zarządzania biblioteką wykorzystujący głosowy interfejs użytkownika

inż. Bartosz Błyszcz
276951@student.pwr.edu.pl

Prowadzący zajęcia
dr inż. Dariusz Banasiak

1. Wstęp

1.1. Temat i cel projektu

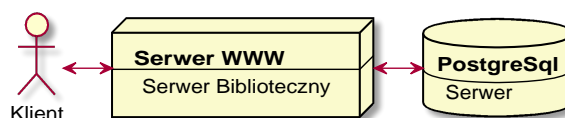
Tematem projektu jest, "System do zarządzania biblioteką wykorzystujący głosowy interfejs użytkownika", w ramach, którego został opracowany wielomodalny interfejs użytkownika służący do obsługi biblioteki. Celem projektu było zbadanie i wykorzystanie możliwości WebSpeechApi [1] w celu umożliwienia poruszania się na stronie internetowej w sposób tradycyjny (klikany) oraz głosowy.

1.2. Założenia techniczne

Tabela 1.1. Technologie wykorzystywane w projekcie
Źródło: Opracowanie własne

| Typ aplikacji | Aplikacja internetowa |
|-------------------------|-----------------------|
| Język oprogramowania | PHP 7.4 [2] |
| Baza danych | PostgreSQL [3] |
| Wykorzystane biblioteki | Symfony 5.4 [4] |
| | WebSpeechApi [1] |
| | Bootstrap [5] |

W tabeli 1.1 opisano założenia techniczne aplikacji internetowej, jakie przyjęto w trakcie wykonywania aplikacji. Założenia opisują wybrany język oraz technologie, jakie posłużyły w utworzeniu całej aplikacji. Przepływ informacji pokazujących Klientowi został przedstawiony na diagramie 1.1.



Rys. 1.1. Schemat systemu wraz z komunikacją
Źródło: Opracowanie własne

Cały system składa się serwera aplikacji internetowej napisanej w języku PHP 7.4 [2], wykorzystującej framework Symfony 5.4 [4] oraz z serwera bazodanowego wykorzystującego PostgreSQL 10 [3]. Do komunikacji głosowej aplikacja wykorzystuje WebSpeechApi [1], która umożliwia rozpoznawanie języka polskiego i przetwarzanie go na tekst. Dzięki czemu jest możliwość sterowania aplikacją przy pomocy głosu.

2. Dokumentacja techniczna

Poniższy rozdział przedstawia dokumentację techniczną projektu. W projekcie skorzystano z kilku gotowych rozwiązań takich jak:

- **Bootstrap** - jest to framework wykorzystywany do prostszego i szybszego tworzenia responsywnego interfejsu aplikacji internetowej. Dzięki dostarczonym narzędziom pozwala na utrzymywanie raz określonych specyfikacji interfejsu oraz na łatwe ich zmienianie [5],
- **Symfony 5.4** - to bardzo rozbudowany framework dla języka PHP, wykorzystywany do tworzenia skalowalnych i dużych aplikacji internetowych. Framework jest zbudowany z wielu komponentów wielokrotnego użytku, dzięki czemu jest możliwość tworzenia aplikacji na miarę potrzeb bez instalacji niepotrzebnych modułów [4],
- **WebSpeechApi** - jest to biblioteka języka JavaScript, która pozwala na rozpoznawanie oraz syntezę mowy [1].

2.1. Aplikacja internetowa

Serwer aplikacji został napisany przy użyciu języka PHP 7.4 oraz frameworku Symfony 5.4, co pozwoliło na utworzenie aplikacji w architekturze *MVC* (ang. *Model View Controller*). Umożliwia on zarządzanie aplikacją poprzez wykonywanie operacji na bazie danych, z którą jest połączony. Aplikacja służy do zarządzania małą biblioteką, ewentualnie prywatnym zbiorem. Pozwala na:

- dodawanie / usuwanie użytkowników
- dodawanie / usuwanie książek
- zarządzanie wypożyczeniami

Aplikacja jest zabezpieczona panelem logowania, dzięki czemu *użytkownik* o roli administratora może zarządzać zbiorem książek oraz użytkowników. Przykładowy kod został przedstawiony we **fragmentcie 2.1**, pozwala on na uzyskanie informacji o książce oraz wyświetleniu ich użytkownikowi.

```

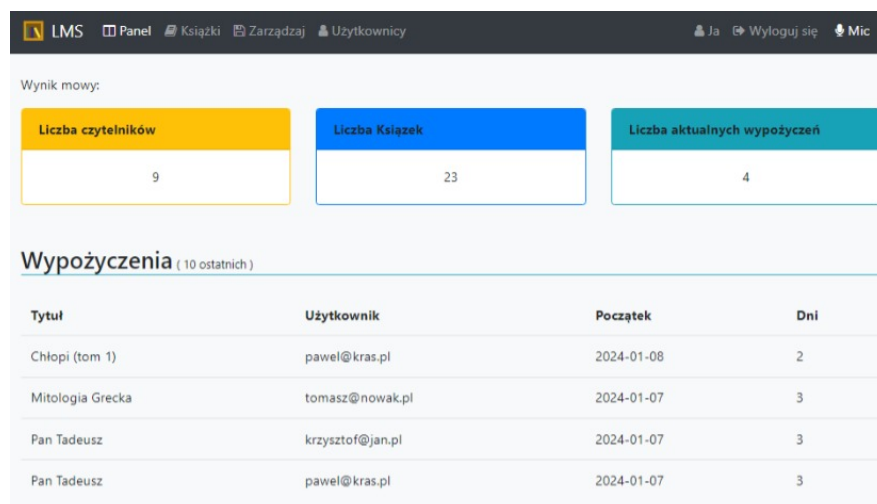
1  /**
2   * @Route("/book/{book}", name="admin_book_site", requirements={"book"="\d{1,9}"})
3   */
4  public function bookSiteAction(Book $book): Response
5  {
6      if ($book->getDeletedAt()) {
7          $this->addFlash('danger', 'Książka została usunięta');
8
9          return $this->redirectToRoute('home');
10     }
11
12     return $this->render('admin/book.html.twig', [
13         'book' => $book,
14     ]);
15 }

```

Fragment kodu 2.1. Metoda bookSiteAction

2.2. Interfejs graficzny

Interfejs graficzny został napisany w językach Html, Css, JavaScript oraz przy użyciu frameworka Bootstrap. Pozwoliło to na stworzenie interfejsu graficznego z działającymi przyciskami, zaciąganych z serwera danymi oraz z prostą kolorystyką co zostało przedstawione na **zdjęciu 2.1**.



Rys. 2.1. Interfejs graficzny w aplikacji
Źródło: Opracowanie własne

2.3. Interfejs głosowy

Do obsługi interfejsu wykorzystano WebSpeechApi, który został wykorzystany w dwojaki sposób. Po pierwsze wykorzystano możliwości biblioteki do rozpoznawania i przetwarzania mowy. Dzięki temu po utworzeniu słownika słów kluczowych dostępnego dla aplikacji jest możliwe sterowanie większością funkcjonalności w sposób głosowy. Słownik ten został podany w **tabeli 2.1**. Dodatkowo wykorzystane zostały możliwości syntezy mowy, które najlepiej zostały zobrazowane na formularzu dodawania nowych książek, gdzie po

wysłaniu niepoprawnych danych aplikacja zwraca się do użytkownika, opisując mu jakie błędy popełnił.

Tabela 2.1. Słownik słów kluczowych
Źródło: Opracowanie własne

| Słownik słów kluczowych | | | |
|-------------------------|-------------|----------------|--------------------|
| login | hasło | zaloguj się | ja |
| wyloguj się | użytkownicy | zarządzaj | książki |
| panel | dodaj | dodaj nową | dodaj nową książkę |
| dodaj nowego | tytuł | autor | wiek |
| granica wieku | zapisz | powrót | książka |
| usuń | dodaj kopię | edytuj książkę | wypożycz |
| użytkownik | oddaj | imię | nazwisko |
| email | data | wyszukaj | indeks |
| index | | | |

Kod do obsługi tego interfejsu został w pełni napisany w języku Java Script. Kod opisany we fragmencie 2.2 Przedstawia wstępną konfigurację biblioteki WebSpeechApi, która jest wykorzystywana w projekcie. Powyższe fragmenty kodu przedstawiają kod napisany w języku JavaScript, którego oddziaływania na interfejs graficzny zostały opisane w części dokumentacji użytkowej, gdzie wskazywane są poszczególne funkcjonalności.

```

1  speechRecognition = window.SpeechRecognition || window.webkitSpeechRecognition
2  speechGrammarList = window.SpeechGrammarList || window.webkitSpeechGrammarList
3  speechRecognitionEvent = window.SpeechRecognitionEvent || window.webkitSpeechRecognitionEvent
4  grammar = '#JSGF V1.0;';
5  recognition = new speechRecognition();
6  speechRecognitionList = new speechGrammarList();
7  speechRecognitionList.addFromString(grammar, 1);
8  recognition.grammars = speechRecognitionList;
9  recognition.continuous = true;
10 recognition.lang = 'pl-PL';
11 recognition.interimResults = false;
12 recognition.maxAlternatives = 1;
13 var recognizing = false;
14 var focus = false;
15 var number = false;
16 var element = null;
17 var book = false;
18 var user = false;
19 var ddate = false;
20 var index = false;

```

Fragment kodu 2.2. Konfiguracja WebSpeechApi

Za rozpoznawanie i przetwarzanie słów na cyfry odpowiada metoda zawarta we fragmencie 2.3.

```

1  textToNumber = (text) => {
2    switch (text) {
3      case 'jeden':
4      case 'jeden':
5        return 1;
6      case 'dwa':

```

```

7      case 'dwa':
8          return 2;
9      case 'trzy':
10     case ' trzy':
11         return 3;
12     case 'cztery':
13     case ' cztery':
14         return 4;
15     case 'piec':
16     case ' piec':
17         return 5;
18     case 'szesc':
19     case ' szesc':
20         return 6;
21     case 'siedem':
22     case ' siedem':
23         return 7;
24     case 'osiem':
25     case ' osiem':
26         return 8;
27     case 'dziewiec':
28     case ' dziewiec':
29         return 9;
30     case 'dziesiec':
31     case ' dziesiec':
32         return 10;
33     case 'jedenascie':
34     case ' jedenascie':
35         return 11;
36     case 'dwanascie':
37     case ' dwanascie':
38         return 12;
39     case 'trzynascie':
40     case ' trzynascie':
41         return 13;
42     case 'czternascie':
43     case ' czternascie':
44         return 14;
45     case 'ępitnascie':
46     case ' ępitnascie':
47         return 15;
48     case 'szesnascie':
49     case ' szesnascie':
50         return 16;
51     case 'siedemnascie':
52     case ' siedemnascie':
53         return 17;
54     case 'osiemnascie':
55     case ' osiemnascie':
56         return 18;
57     default:
58         return parseInt(text)
59     }
60 }

```

Fragment kodu 2.3. Metoda zamieniająca słowa na cyfry

Kod ukazany we fragmencie 2.4 odpowiada za konfigurację wydarzeń na stronie, które uruchamiają i wyłączają nasłuchiwanie. Dodatkowo odpowiada za obsłużenie wydarzeń biblioteki. Wydarzenia obsługiwane to:

- nomatch - nie rozpoznano mowy
- speechend - wykryto koniec mowy
- error - inny błąd

```
1  recognitionEvent = () => {
2      document.getElementById('mic').onclick = () => {
3          if(!recognizing) {
4              document.getElementById('mic').style.backgroundColor = 'red'
5              recognizing = true
6              recognition.abort();
7              recognition.start();
8          }
9      };
10     recognition.addEventListener('nomatch', () => {
11         document.getElementById('result').innerText = 'Nie rozpoznano mowy, rozpocznij na nowo';
12         document.getElementById('mic').style.backgroundColor = '#343a40'
13     });
14     recognition.addEventListener('speechend', () => {
15         document.getElementById('result').innerText = 'Wykryto koniec mowy, rozpocznij na nowo';
16         document.getElementById('mic').style.backgroundColor = '#343a40'
17         recognizing = false;
18     });
19     recognition.addEventListener('error', (event) => {
20         document.getElementById('result').innerText = 'Wykryto błąd: ${event.error}, rozpocznij na nowo';
21         document.getElementById('mic').style.backgroundColor = '#343a40'
22         recognizing = false;
23     });
24     recognition.onstart = function () {
25         recognizing = true;
26     };
27 }
```

Fragment kodu 2.4. Obsługa wydarzeń związanych z mową

Fragment kodu 2.5 odpowiada za wykorzystanie rezultatu przetwarzania mowy na tekst i obsłużenie różnych akcji m.in. uzupełniania formularzy.

```
1  useRecognition((text) => {
2      var tt = text.toLowerCase()
3      document.getElementById('result').innerText = text
4      if (index) {
5          const sss = String(text).trim().slice(0, -1);
6          const _class = 'index-' + textToNumber(sss);
7          document.getElementById(_class).click()
8      }
9      if (focus) {
10         var sss = String(text).trim().slice(0, -1);
11         if (number) {
12             element.value = textToNumber(sss);
13         } else if (ddate) {
14             element.value = Date.parse(String(text).trim());
15         } else {
16             element.value = sss;
17         }
18     }
19 }
```



```

18     element.dispatchEvent(new Event('keyup'))
19     element = null
20     number = false;
21     focus = false
22     return
23 }
24 if (book || user) {
25     var sss = String(text).toLowerCase().trim().slice(0, -1);
26     document.getElementById(sss).click()
27 }
28 tt = tt.slice(0, -1)
29 let utterance = new SpeechSynthesisUtterance('Rozpoznano: ' + tt);
30 speechSynthesis.speak(utterance);
31 })

```

Fragment kodu 2.5. Obsługa rezultatu rozpoznawania mowy

Ostatni fragment 2.6 obsługuje słownik słów kluczowych na stronie, wykonując odpowiednie akcje przypisane do poszczególnych słów.

```

1  switch(String(tt).trim()) {
2      case 'login':
3          document.getElementById('inputEmail').focus()
4          focus = true;
5          break;
6      case 'haslo':
7          document.getElementById('inputPassword').focus()
8          focus = true;
9          break;
10     case 'zaloguj sie':
11         document.getElementById('submit').click()
12         break;
13     case 'ja':
14         document.getElementById('Ja').click()
15         break;
16     case 'wyloguj sie':
17         document.getElementById('Wyloguj sie').click()
18         break;
19     case 'uzytkownicy':
20         document.getElementById('Uzytkownicy').click()
21         break;
22     case 'zarzadzaj':
23         document.getElementById('Zarzadzaj').click()
24         break;
25     case 'ksiazki':
26         document.getElementById('Ksiazki').click()
27         break;
28     case 'panel':
29         document.getElementById('Panel').click()
30         break;
31
32     case 'dodaj':
33     case 'dodaj ąnow':
34     case 'dodaj ąnow ęksiazk':
35     case 'dodaj nowego':
36         document.getElementById('dodaj').click()
37         break;
38
39     case 'tytul':

```

```

40     document.getElementById('book_form_title').focus()
41     element = document.getElementById('book_form_title')
42     focus = true
43     break;
44
45     case 'autor':
46         document.getElementById('book_form_author').focus()
47         element = document.getElementById('book_form_author')
48         focus = true
49         break;
50     case 'wiek':
51         document.getElementById('book_form_ageThreshold').focus()
52         element = document.getElementById('book_form_ageThreshold')
53         focus = true
54         number = true
55         break;
56     case 'granica wieku':
57         document.getElementById('book_form_ageThreshold').focus()
58         element = document.getElementById('book_form_ageThreshold')
59         focus = true
60         number = true
61         break;
62
63     case 'zapisz':
64         document.getElementById('zapisz').click()
65         break;
66     case 'powrot':
67         document.getElementById('powrot').click()
68         break;
69
70     case 'ksiazka':
71         document.getElementById('ażksika').style.backgroundColor = 'red'
72         book = true
73         break;
74     case 'usun':
75         document.getElementById('book-del').click()
76         break;
77     case 'dodaj kopie':
78         document.getElementById('book-new-copy').click();
79         break;
80     case 'edytuj ksiazke':
81         document.getElementById('book-edit').click();
82         break;
83     case 'wypożycz':
84         document.getElementById('new-loan').click();
85         document.getElementById('new-loan').style.backgroundColor = 'red';
86         break;
87     case 'uzytkownik':
88         document.getElementById('user').style.backgroundColor = 'red'
89         user = true
90         break;
91     case 'oddaj':
92         document.getElementById('loan-return').click();
93         break;
94     case 'imie':
95         document.getElementById('user_form_name').focus();
96         element = document.getElementById('user_form_name');
97         focus = true;

```

```

98         break;
99     case 'nazwisko':
100         element = document.getElementById('user_form_surname');
101         document.getElementById('user_form_surname').focus();
102         focus = true;
103         break;
104     case 'email':
105         element = document.getElementById('user_form_email');
106         document.getElementById('user_form_email').focus();
107         focus = true;
108         break;
109     case 'data':
110         element = document.getElementById('user_form_birthDate');
111         document.getElementById('user_form_birthDate').focus();
112         focus = true;
113         ddate = true;
114         break;
115     case 'wyszukaj':
116         element = document.querySelector('#booksCopies_filter label input');
117         element.focus();
118         focus = true;
119         break;
120     case 'indeks':
121     case 'index':
122         element = document.querySelectorAll('.index');
123         element.forEach(n => {
124             n.style.backgroundColor = 'red'
125         })
126         index = true
127         break;
128 }

```

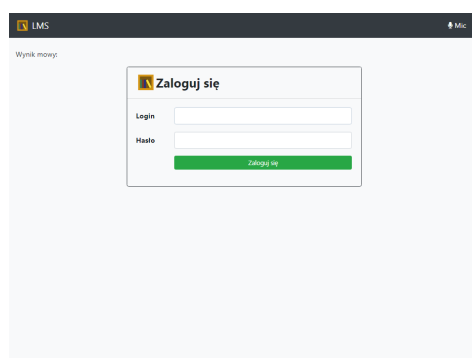
Fragment kodu 2.6. Obsługa słów kluczowych

3. Dokumentacja użytkowa

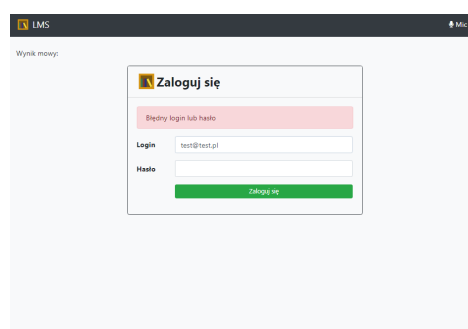
W poniższym rozdziale zostanie zaprezentowana dokumentacja użytkowa aplikacji wytworzonej na potrzeby projektu. W dokumentacji zostaną zaprezentowane poszczególne ekrany oraz interakcje.

3.1. Logowanie

Ekran logowania to pierwszy ekran widoczny dla użytkownika niezalogowanego do systemu. Tak jak wskazano na **rysunku 3.1a** na ekranie występuje prosty formularz pozwalający na wpisanie loginu i hasła użytkownika w celu weryfikacji tożsamości logowanego. Po wciśnięciu przycisku *”Zaloguj się”* użytkownik zostaje przekierowany do swojego profilu. W przypadku błędnych danych uwierzytelniających użytkownik otrzyma komunikat zwrotny tak jak na **rysunku 3.1b**.



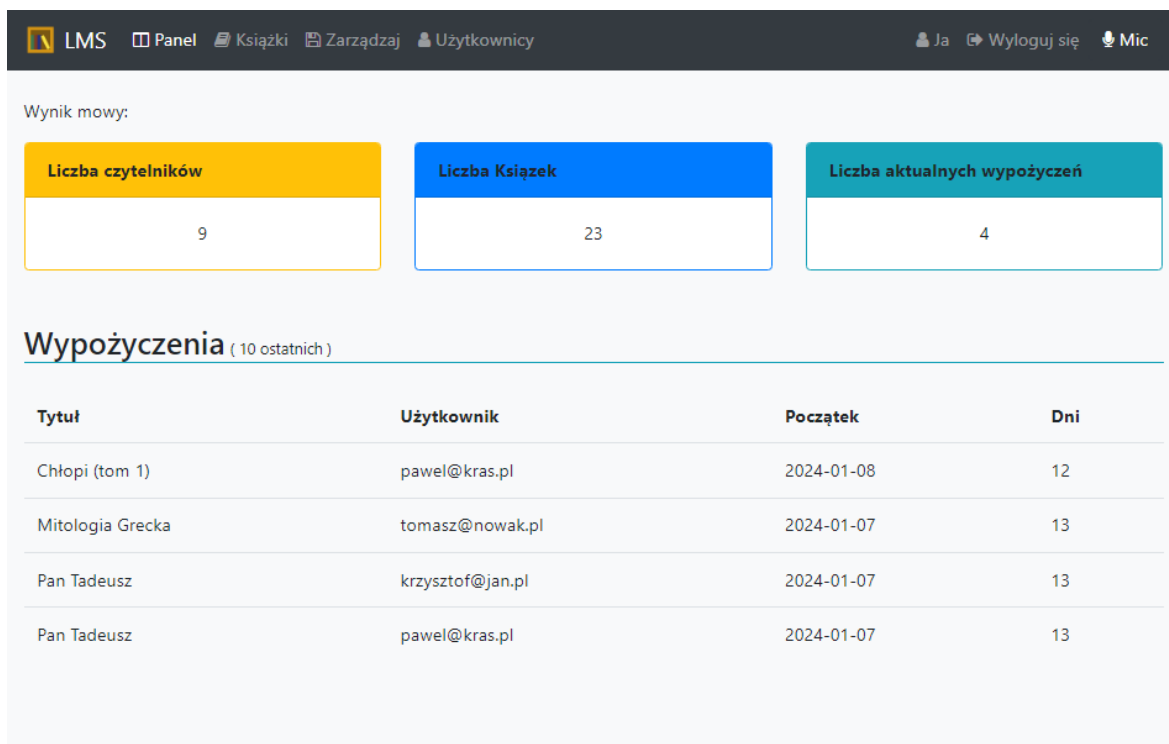
(a) Ekran logowania
Źródło: Opracowanie własne



(b) Ekran logowania - błędne dane
Źródło: Opracowanie własne

Rys. 3.1. Logowanie
Źródło: Opracowanie własne

Po zalogowaniu, użytkownik przechodzi do ekranu głównego, który jednocześnie jest ekranem ukazującym ogólne statystyki. Został on ukazany na **rysunku 3.2**



Rys. 3.2. Ekran początkowy
Źródło: Opracowanie własne

Na ekranie zostały zaprezentowane takie informacje jak:

- liczba czytelników
- liczba książek
- liczba aktualnych wypożyczeń
- ostatnie 10 wypożyczeń

Pozwala to użytkownikowi zorientować się w aktualnej sytuacji biblioteki.

W górnym prawym rogu ekranu dostępna jest ikona mikrofonu, po której przyciśnięciu aplikacja nasłuchuje wydawanych komend. Po wypowiedzeniu nazwy zakładki z górnego paska użytkownik zostanie do niej przekierowany.

3.2. Książki

Ekran "Książki" pozwala na zarządzanie zbiorem książek. Pierwszy ekran pozwala podejrzeć cały zbiór oraz wyszukiwać konkretne tytuły. Dodatkowo z tego okna można przejść na zakładkę informacyjną książki, oraz dodać nową pozycję. Na tym ekranie po włączeniu opcji *mikrofonu*, można dodać nową książkę bądź wybrać konkretny tytuł. Aby wybrać tytuł należy powiedzieć słowo "Książka", a następnie powiedzieć tytuł książki. Działanie ekranu zostało przedstawione na **rysunkach 3.3a 3.3b**.

| # | Autor | Tytuł | Gatunek | Wiek | Kopie | Dostępność |
|----|-----------------------------|------------------|---------|------|-------|------------|
| 1 | Adam Mickiewicz | Pan Tadeusz | EPIKA | 12 | 3 | tak |
| 2 | Władysław Stanisław Reymont | Chłopi (tom 1) | EPIKA | 12 | 1 | tak |
| 3 | Witold Gombrowicz | Ferdynand | EPIKA | 14 | 1 | tak |
| 4 | - | Biblia | EPIKA | 10 | 1 | tak |
| 5 | Jan Parandowski | Mitologia Grecka | EPIKA | 8 | 1 | tak |
| 6 | Homer | Iliada | EPIKA | 10 | 1 | tak |
| 7 | Homer | Odyseja | EPIKA | 12 | 1 | tak |
| 8 | Sofokles | Antygona | EPIKA | 14 | 1 | tak |
| 9 | Gali Anonim | Kronika polska | EPIKA | 10 | 1 | tak |
| 10 | William Szekspir | Makbet | DRAMAT | 14 | 1 | tak |

(a) Ekran Książki
Źródło: Opracowanie własne

| # | Autor | Tytuł | Gatunek | Wiek | Kopie | Dostępność |
|---|-----------------------------|------------------|---------|------|-------|------------|
| 1 | Adam Mickiewicz | Pan Tadeusz | EPIKA | 12 | 3 | tak |
| 2 | Władysław Stanisław Reymont | Chłopi (tom 1) | EPIKA | 12 | 1 | tak |
| 3 | Witold Gombrowicz | Ferdynand | EPIKA | 14 | 1 | tak |
| 4 | - | Biblia | EPIKA | 10 | 1 | tak |
| 5 | Jan Parandowski | Mitologia Grecka | EPIKA | 8 | 1 | tak |
| 6 | Homer | Iliada | EPIKA | 10 | 1 | tak |
| 7 | Homer | Odyseja | EPIKA | 12 | 1 | tak |

(b) Ekran Książki po wypowiedzeniu słowa książka
Źródło: Opracowanie własne

Rys. 3.3. Książki
Źródło: Opracowanie własne

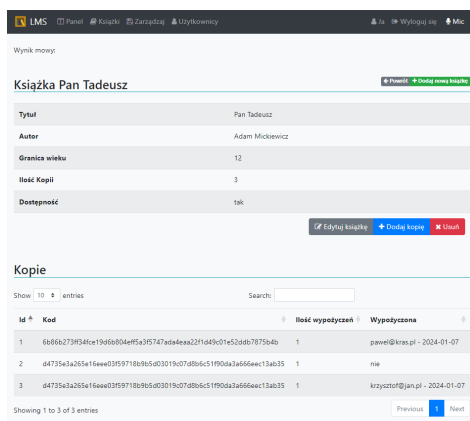
Po przejściu do ekranu pozwalającego dodać nową książkę użytkownik zostaje poproszony o wypełnienie krótkiego formularza w którym zawarte są podstawowe informacje o nowej pozycji tak jak to zostało przedstawione na **rysunku 3.4a**. Operacje dodania książki można wykonać również w sposób głosowy, podając etykiety pól formularza a następnie wypowiadając to co ma się w danym polu znajdować.

(a) Formularz dodawania książek
Źródło: Opracowanie własne

(b) Formularz wypełniony błędnie
Źródło: Opracowanie własne

Rys. 3.4. Dodawanie książek
Źródło: Opracowanie własne

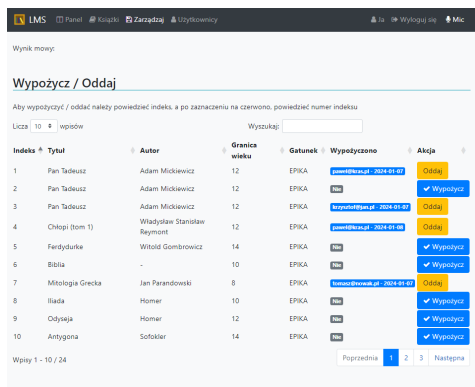
Jeśli po zapisaniu formularza nastąpi błąd. Błędne pola zostaną podkreślone, a błędy zostaną wypisane i powiedziane (z wykorzystaniem syntezy mowy) przez formularz. Wygląd błędnego formularza został ukazany na **ekranie 3.4b**. Po poprawnym wypełnieniu i zapisaniu formularza zostanie dodana nowa książka. Panel informacyjny o książce został wskazany na **obrazku 3.5**. Na tym ekranie dostępne są podstawowe informacje na temat książki oraz informacje o jej wypożyczeniach. Z zaprezentowanego **ekranu 3.5** jest możliwość zarządzać samą książką.



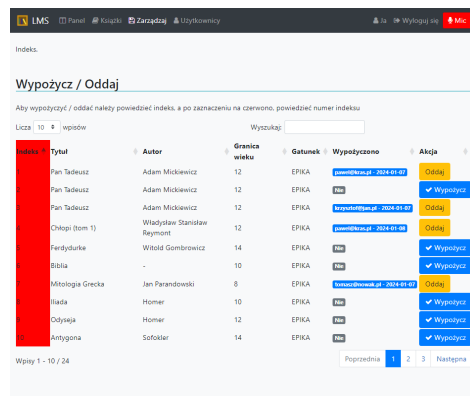
Rys. 3.5. Ekran informacyjny o książce
Źródło: Opracowanie własne

3.3. Zarządzaj

Ekran "Zarządzaj" pozwala na zarządzanie listą wypożyczeń oraz dostępnych książek. Dodatkowo jest to też wyszukiwarka, która pozwala na wyszukiwanie konkretnych tytułów. Został on przedstawiony na **zdjęciu 3.6a**. Ekran pozwala na wypożyczanie oraz oddawanie książek. Można tą operację wykonać również głosowo po wypowiedzeniu słowa "indeks", co zaznaczy kolumnę i poczeka na wypowiedzenie numeru indeksu tak jak na **rysunku 3.6b**. Aby oddać książkę należy wybrać przycisk "Oddaj", bądź wypowiedzieć nazwę indeksu z tym przyciskiem.



(a) Ekran Zarządzaj
Źródło: Opracowanie własne



(b) Ekran z zaznaczoną kolumną Indeks
Źródło: Opracowanie własne

Rys. 3.6. Ekran Wypożyczenia
Źródło: Opracowanie własne

Aby wypożyczyć należy zrobić czynność analogiczną ale dla słowa "Wypożycz". Dodatkowo po wybraniu przycisku "Wypożycz" użytkownik zostanie przekierowany na odpowiedni ekran, który został ukazany na **zdjęciu 3.7**. Tam należy wskazać odpowiedniego użytkownika i kliknąć "Zapisz".

Rys. 3.7. Ekran wypożyczenia
Źródło: Opracowanie własne

3.4. Użytkownicy

Kolejnym panelem jest panel ”*Użytkownicy*”, który pozwala na zarządzanie użytkownikami. Główne okno prezentuje listę użytkowników z możliwością dodania nowego użytkownika oraz zobaczenia informacji o użytkowniku. **Rysunek 3.8a** pokazuje listę użytkowników z możliwością wyszukiwania po niej. **Obraz 3.8b** pokazuje okno informacyjne o użytkowniku, a **ekran 3.8c** pokazuje formularz dodania nowego użytkownika.

| # | Imię | Nazwisko | Login | Liczba wypożyczonych książek |
|---|-----------|----------|------------------|------------------------------|
| 1 | Tomasz | Nowak | tomasz@nowak.pl | 1 |
| 2 | Krzysztof | Jan | krzysztof@jan.pl | 1 |
| 3 | Tomasz | Chabla | tomasz@chabla.pl | 0 |
| 4 | Barbara | Kowal | barbara@kowal.pl | 0 |
| 5 | Paweł | Kucan | kucan@kucan.pl | 0 |
| 6 | Tomasz | Wójcik | tomasz@wojcik.pl | 0 |
| 7 | Jan | Wojcik | jan@wojcik.pl | 0 |
| 8 | Paweł | Kucan | pawel@kucan.pl | 2 |
| 9 | Michał | Nowak | michal@nowak.pl | 1 |

(a) Ekran Użytkownicy
Źródło: Opracowanie własne

| # | Tytuł | Author | Data wypożyczenia | Okaz | Oddane |
|---|--------|-----------------------------|-------------------|------|--------|
| 1 | Władcy | Adam Mickiewicz | 2024-01-02 | 12 | nie |
| 2 | Chłopa | Władysław Stanisław Reymont | 2024-01-08 | 11 | nie |

(b) Ekran informacyjny
Źródło: Opracowanie własne

(c) Formularz użytkownika
Źródło: Opracowanie własne

Rys. 3.8. Użytkownicy
Źródło: Opracowanie własne

3.5. Ja

Ostatnim panelem jest ekran z informacjami zalogowanego użytkownika, pokazany na **rysunku 3.9a**. Pozwala on na sprawdzenie informacji o koncie oraz ich zmodyfikowaniu za pomocą formularza pokazanego na **zdjęciu 3.9b**.

Wynik mojej:

Użytkownik

| | |
|---------------|--------------|
| Imię | test |
| Nazwisko | test |
| Email | test@test.pl |
| Rok Urodzenia | 2000-01-01 |

[Edytuj dane](#)

Wypożyczenia

Show 10 entries

Search:

| # | Tytuł | Użytkownik | Początek | Dni | Oddano |
|---|-------------|--------------|------------|-----|------------|
| 1 | Pan Tadeusz | test@test.pl | 2024-01-07 | 1 | 2024-01-08 |

Showing 1 to 1 of 1 entries

[Previous](#) [Next](#)

(a) Ekran Ja
Źródło: Opracowanie własne

Wynik mojej:

Edycja danych

[Edytuj](#)

| | |
|--------------|--------------|
| Imię | test |
| Nazwisko | test |
| Data Urodzin | Jan 1 2000 |
| Email | test@test.pl |

[Zapisz dane](#)

Zmień hasło

| | |
|-----------------|--|
| Nowe Hasło | |
| Potwierdź Hasło | |

[Zapisz hasło](#)

(b) Formularz modyfikacji danych
Źródło: Opracowanie własne

Rys. 3.9. Ekran o mnie
Źródło: Opracowanie własne

4. Wnioski

W trakcie tego projektu zostały wykorzystane biblioteka WebSpeechApi do języka Java Script. Jest to bardzo prosta w użyciu biblioteka dająca duże możliwości i prosta w obsłudze. Dużym plusem tej biblioteki jest możliwość skorzystania z możliwości przeglądarek do obsługi rozpoznawania i syntezy mowy. Pozwoliło to na unowocześnienie aplikacji prostej biblioteki przez dodanie do niej obsługi interfejsu głosowego. Dzięki czemu osoby niepełnosprawne z problemami ruchowymi mogą obsługiwać aplikację poprzez wykorzystanie głosu. Kolejnym atutem jest możliwość syntezy głosu, dzięki czemu jest możliwość zwracania informacji zwrotnej tak jak zostało to pokazane na formularzu dodawania książek. Biblioteka bardzo dobrze sobie radzi z tekstem ciągłym i znakami specjalnymi np. przy dyktowaniu adresu email. Jednakże zdania należało wypowiadać bardzo wyraźnie ponieważ samo rozpoznawanie mowy nie zawsze "rozumiało" celu wypowiedzi albo słów, wypowiedzianych jeśli były robione w sposób niezrozumiały. Narzędzia dostarczane przez tę bibliotekę, wskazują jak duży potencjał, mają aplikacje przeglądarkowe, które mogą być uruchamiane bezpośrednio w przeglądarce bądź jako PWA (*ang. Progressive Web App*) czyli aplikacja "instalowana" lokalnie która może w pewnym stopniu działać bez dostępu do internetu.

Wykaz rysunków

| | | |
|-----|--|----|
| 1.1 | Schemat systemu wraz z komunikacją | 3 |
| 2.1 | Interfejs graficzny w aplikacji | 5 |
| 3.1 | Logowanie | 12 |
| 3.2 | Ekran początkowy | 13 |
| 3.3 | Książki | 14 |
| 3.4 | Dodawanie książek | 14 |
| 3.5 | Ekran informacyjny o książce | 15 |
| 3.6 | Ekran Wypożyczenia | 15 |
| 3.7 | Ekran wypożyczenia | 16 |
| 3.8 | Użytkownicy | 16 |
| 3.9 | Ekran o mnie | 17 |

Bibliografia

- [1] WebSpeechApi. „*WebSpeechApi*”. URL: <https://dvcz.w3.org/hg/speech-api/raw-file/tip/webspeechapi> (term. wiz. 2024-01-10).
- [2] PHP. „*PHP*”. URL: <https://www.php.net/> (term. wiz. 2023-01-10).
- [3] PostgreSQL. „*Postgresql*”. URL: <https://www.postgresql.org/pl/> (term. wiz. 2023-01-10).
- [4] Symfony. „*Symfony*”. URL: <https://www.symfony.com/> (term. wiz. 2023-01-10).
- [5] Bootstrap. „*Bootstrap*”. URL: <https://www.getbootstrap.com/> (term. wiz. 2023-01-10).