

Serwis internetowy prezentujący implementację wybranych zagadnień dostępności cyfrowej na podstawie WCAG 2.1

Adrian Damaziak, Piotr Kopciał*

Warszawska Wyższa Szkoła Informatyki

Abstrakt

W niniejszym artykule opisano proces przygotowania formularza umożliwiającego użytkownikowi zgłoszenie strony internetowej do badania poziomu dostępności cyfrowej witryny. Badanie jest zlecane firmie audytorskiej, sprawdzającej stronę pod kątem spełnienia zaleceń WCAG 2.1, a w efekcie daje odpowiedzi na szereg pytań kluczowych. W pracy przedstawiono zagadnienie oceny witryn internetowych pod kątem dostępności przez osoby niepełnosprawne. Opracowano wymagania funkcjonalne i niefunkcjonalne systemu walidacji witryn, omówiono architekturę i działanie zaimplementowanego systemu. Przedstawiono wyniki testów dostępności w oparciu o listę zapytań kontrolnych oraz walidator WAVE.

Słowa kluczowe – WCAG 2.1, dostępność cyfrowa witryny, testy dostępności, WAVE, WCAG 3.0

* E-mail: kopcial@ms.wysi.edu.pl

1. Idea zagadnień WCAG

Możliwość użytkowania stron internetowych to istotny aspekt codzienności dla większości populacji, także dla ludzi z niepełnosprawnościami. Pomimo istnienia od ponad dekady technicznych wytycznych dla zapewniania dostępności, wciąż wiele stron nie jest dostosowanych dla wszystkich użytkowników. Wymagania WCAG (Web Content Accessibility Guidelines) pokrywają szeroki zakres rekomendacji do tworzenia zawartości stron internetowych. Stosowanie tych pryncypiów umożliwia wszystkim, w tym osobom niepełnosprawnym, postrzeganie, rozumienie, nawigowanie i interakcję z treściami zawartymi w Internecie [1].

Zagadnienia dostępności cyfrowej zostały stworzone, aby poprawić jakość korzystania z sieci użytkownikom opisanym poniżej. Mimo, że wytyczne obejmują szeroki zakres zagadnień, nie są w stanie zaspokoić potrzeb użytkowników ze wszystkimi rodzajami, stopniami i kombinacjami niepełnosprawności. Wytyczne sprawiają również, że treści internetowe są bardziej użyteczne dla osób starszych, których możliwości zmieniają się wraz z wiekiem. Typowe rodzaje niepełnosprawności to:

- niepełnosprawność wzrokowa,
- niepełnosprawność słuchowa,
- niepełnosprawność fizyczna,
- niepełnosprawność mowy,
- niepełnosprawność poznawcza,
- niepełnosprawność językowa,
- niepełnosprawność uczenia się.

1.1. *Standardy dostępności na świecie*

ADA (American with Disabilities Act) jest ustawą o prawach obywatelskich, która „zakazuje dyskryminacji osób niepełnosprawnych we wszystkich dziedzinach życia publicznego, w tym w pracy, szkołach, transporcie oraz we wszystkich miejscach publicznych i prywatnych, które są otwarte dla ogółu społeczeństwa” [2]. Prawo to gwarantuje równe szanse osobom niepełnosprawnym w zakresie zakwaterowania

publicznego, zatrudnienia, transportu, usług państwowych i samorządowych oraz telekomunikacji. Również wymagania co do witryn internetowych i aplikacji są regulowane wskazaną ustawą [3].

Wymagania zawarte w WCAG są częścią serii wytycznych dotyczących dostępności stron internetowych, opublikowanych przez Inicjatywę Dostępności Sieci (WAI) Konsorcjum World Wide Web (W3C), głównej międzynarodowej organizacji zajmującej się standardami dla Internetu. Są one zbiorem zaleceń dotyczących zwiększenia dostępności treści internetowych przede wszystkim dla osób niepełnosprawnych, ale także dla wszystkich użytkowników. Dotyczą również urządzeń o ograniczonych możliwościach, takich jak telefony komórkowe. WCAG 2.0 zostały opublikowane w grudniu 2008 roku i stały się normą ISO, ISO/IEC 40500:2012 w październiku 2012 roku. WCAG 2.1 zostały opublikowane przez W3C w czerwcu 2018 roku.

Wytyczne WCAG 2.1 zostały podzielone na trzy poziomy zgodności, aby zaspokoić potrzeby różnych grup i różnych sytuacji: A (najniższy), AA (średni zakres) i AAA (najwyższy). Zgodność na wyższych poziomach oznacza zgodność na poziomach niższych.

- Poziom A – minimalna dostępność – strona nie spełniająca tego zakresu kryterium tworzy bariery, których użytkownik nie pokona za pomocą urządzeń asystujących. Ten poziom dotyczy najszerzej grupy i przynosi użytkownikowi najwięcej korzyści. Jego implementacja jest niezbędna.
- Poziom AA – większa dostępność – kryteria na tym poziomie określają poziom dostępności, który powinien działać z większością technologii wspomagających na urządzeniach stacjonarnych i mobilnych. Zaleca się, aby strony internetowe spełniały ten zakres wymagań. Nadal będą występowały trudności w odbiorze strony dla pewnych grup użytkowników.
- Poziom AAA – jeszcze większa dostępność – zgodność na tym poziomie sprawia, że witryna jest dostępna dla maksymalnej liczby użytkowników i maksymalnie ułatwia korzystanie z jej zasobów.

1.2. Sposoby walidowania stron internetowych

Walidacja i finalna ocena strony pod kątem dostępności cyfrowej jest złożonym procesem, który można realizować następującymi sposobami [4]:

- Wykorzystanie walidatorów kodu – programów działających offline/online, desktopowych, jak również w formie rozszerzeń do przeglądarki internetowej (głównie Chrome/Firefox), umożliwiających podanie wskazanie adresu witryny lub wklejenie kodu HTML strony i jej weryfikację pod kątem poprawności zaimplementowanych rozwiązań.
- Parser HTML5 W3 (link) – oficjalny parser organizacji W3, sprawdzający poprawność użycia tagów HTML5, dobrania odpowiedniej semantyki znaczników, użytych atrybutów (ich poprawności oraz czy nie są przestarzałe), składni oraz swoistej gramatyki języka.
- Manualna weryfikacja kolejnych kryteriów sukcesu WCAG 2.1 i testowanie jej elementów na głównych funkcjonalnych podstronach portalu, przez doświadczonego audytora, na przykładzie listy kontrolnej.
- Manualne testy użytkowników z niepełnosprawnościami, ocena poprawności / sukcesu przejścia przez funkcjonalne scenariusze, do których wykonywania strona została stworzona (np. użytkownik niewidomy zakłada konto w banku, przy użyciu czytnika ekranu i klawiatury, bez użycia myszki). Oceniana jest finalnie łatwość / zrozumienie komunikacji ze stroną internetową i to czy polecenia były jasne oraz ogólny sukces operacji.

1.3. Badanie zgodności ze standardem WCAG 2.1

Celem niniejszego artykułu jest przedstawienie projektu oraz implementacji, a także audyt w pełni dostępnej strony formularza, służącej do zgłaszania witryny do badania WCAG 2.1. W ramach realizacji projektu powstała inżynierska praca dyplomowa [5]. Większość z istotnych jej wyników przedstawiono w niniejszym artykule. Zrealizowany w ramach realizacji pracy dyplomowej projekt jest w pełni dostępny cyfrowo, wykonany z użyciem najnowszych technologii frontendowych. W trakcie realizacji pracy wykorzystywano ogólnie dostępną dokumentację i zalecenia dotyczące

dostępności stron internetowych. Zakres pracy obejmował zaprojektowanie i stworzenie systemu, w ramach którego wykonano następujące czynności:

- przeanalizowano liczne witryny internetowe pod kątem dostępności,
- określono funkcjonalności aplikacji, wymagania funkcjonalne oraz wymagania pozafunkcjonalne,
- dobrano odpowiednie technologie i narzędzia,
- utworzono API dla potrzeb funkcjonowania aplikacji,
- wykonano aplikację,
- zrealizowano testy wykonanego rozwiązania z użyciem czytnika ekranu oraz rekomendowanej listy kontrolnej.

W dalszej części niniejszego artykułu zostaną opisane wszystkie istotne elementy zaprojektowanego systemu wraz z opisem walidacji przeprowadzonej dla przykładowej witryny utworzonej na potrzeby niniejszej pracy.

2. Założenia i wymagania dotyczące aplikacji

Podczas projektowania aplikacji ważnym elementem było wyznaczenie jej wymagań. Celem tego działania było opisanie funkcji i możliwości jakie musi zawierać projekt, aby spełnił on zamierzoną rolę. W poniższej pracy przedstawiono dwie główne kategorie wymagań systemu:

- Wymagania funkcjonalne – wymagania przedstawiające sposób działania systemu oraz czynności i operacje, które wykonuje.
- Wymagania pozafunkcjonalne – wymagania opisujące techniczne parametry i ograniczenia, przy których możliwa jest realizacji funkcji systemu.

2.1. Wymagania funkcjonalne

- Aplikacja umożliwi użytkownikowi niewidomemu odczyt opisu pól formularza oraz wypełnienie go m.in. przy użyciu samego czytnika ekranu lub/i klawiatury [6].
- Formularz potrafi reagować na błędnie wprowadzone dane (np. niezgodność z formatem, który przyjmie aplikacja) oraz wyświetlać podpowiedzi, sygnalizując błąd w sposób wizualny, jak również widoczny dla czytnika ekranu.

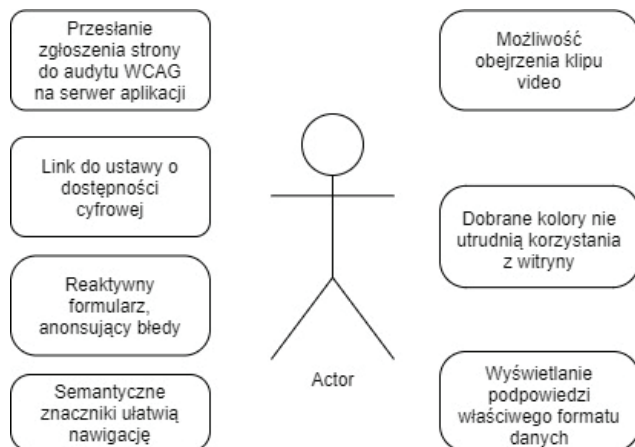
Przy błędnym wypełnieniu pola i wysłaniu danych, formularz wskaże miejsce wymagające poprawy oraz wyświetli tekstową podpowiedź z propozycją zmiany.

- Aplikacja posiada wbudowane walidatory wprowadzanych danych, zaimplementowane na polach numer telefonu, email oraz data. W przypadku ostatniego wymienionego pola, niezbędne jest obsadzenie w pełni dostępnego modułu wyboru daty.
- Po prawidłowym uzupełnieniu formularza, aplikacja wyśle dane przez REST API na testowy serwer oraz przejdzie do kolejnej strony. Na stronie podsumowania aplikacja wyświetli w tabeli dane pobrane z serwera, umożliwiając użytkownikowi sprawdzenie ich poprawności oraz wyświetli przycisk powrotu do strony głównej.
- Na stronie pojawi się możliwość obejrzenia klipu video, mogącego nawiązywać do działalności firmy, także dla użytkowników z niepełnosprawnościami, między innymi z możliwością włączenia napisów opisujących sytuację na ekranie, a także nawigacji po elementach odtwarzacza bez użycia myszki, wraz z anonsovaniem przez czytnik kolejnych pól (rysunek 1).

2.2. Wymagania pozafunkcjonalne

Wymagania pozafunkcjonalne projektu realizowanego w ramach niniejszej pracy są następujące:

- użytkownik potrzebuje dostępu do Internetu,
- użytkownik potrzebuje komputer z systemem Windows 7, 8 lub 10,
- użytkownik potrzebuje przeglądarki internetowej (Chrome, Mozilla, Edge) w najnowszej wersji,
- w przypadku osoby z niepełnosprawnością wzrokową, użytkownik potrzebuje także jednego z czytników ekranu, wspomnianego wcześniej.

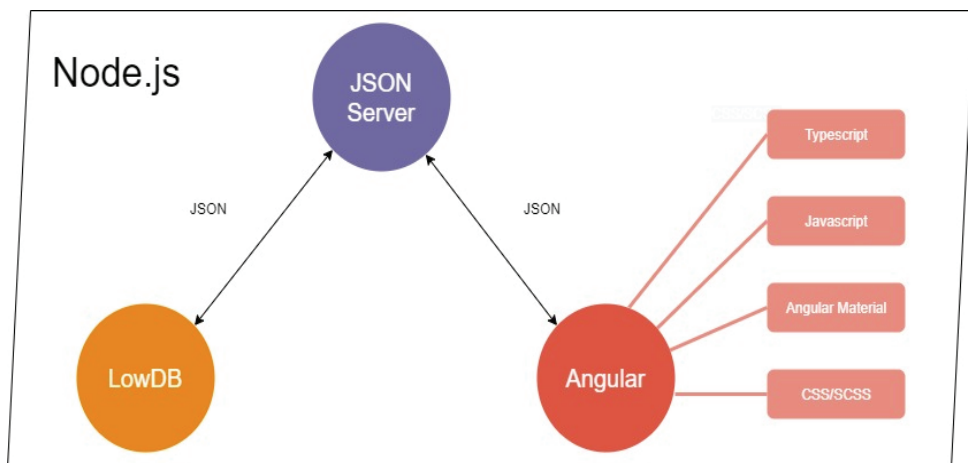


Rysunek 1. Wymagania funkcjonalne

2.3. Architektura systemu

Na działanie aplikacji składają się trzy główne elementy (rysunek 2):

- Angular,
- JSON Server,
- LowDB.



Rysunek 2. Wykres przedstawiający architekturę aplikacji

Warstwa widoku generowana jest przez framework Angular 10, komunikujący się poprzez REST API z częścią backendową, realizowaną przez aplikację JSON Server, która natomiast korzysta z wbudowanej prostej bazy danych LowDB, obsługującej pliki JSON oraz YAML.

2.4. Działanie systemu

Użytkownik wchodząc na stronę główną (rysunek 3), w celu zgłoszenia witriny do badania dostępności, ma do wypełnienia formularz, przekazujący informacje firmie audytorskiej.



Rysunek 3. Implementacja strony głównej aplikacji

Dla ułatwienia wprowadzania danych zaimplementowany został moduł wyboru daty, zbudowany w oparciu o tabele HTML i jej semantyczne znaczniki tak, aby

możliwe było nawigowanie przy pomocy samej klawiatury. Przyciski funkcyjne, służące do zmiany miesięcy oraz lat, zostały opatrzone atrybutami aria-label, sugerującymi ich przeznaczenie.

Dodatkowo, gdy wyświetlane jest okno wyboru daty, pozostałe interaktywne elementy strony są wyłączone z użycia do czasu jego zamknięcia, co jest korzystne z perspektywy użytkownika czytnika ekranu (rysunki 4 i 5).

Imię
Grzegorz

Nazwisko
Cwikliński

Adres strony internetowej
http://www.wcag.pl

Wpisz poprawny adres strony internetowej

Telefon
+48 501345643

Wpisz numer telefonu o maksymalnej długości 9 cyfr

Email
gcwiklinski@gmail.com

W adresie email wymagany jest

Wybierz datę audytu
7/14/2021

Powtórny audyt

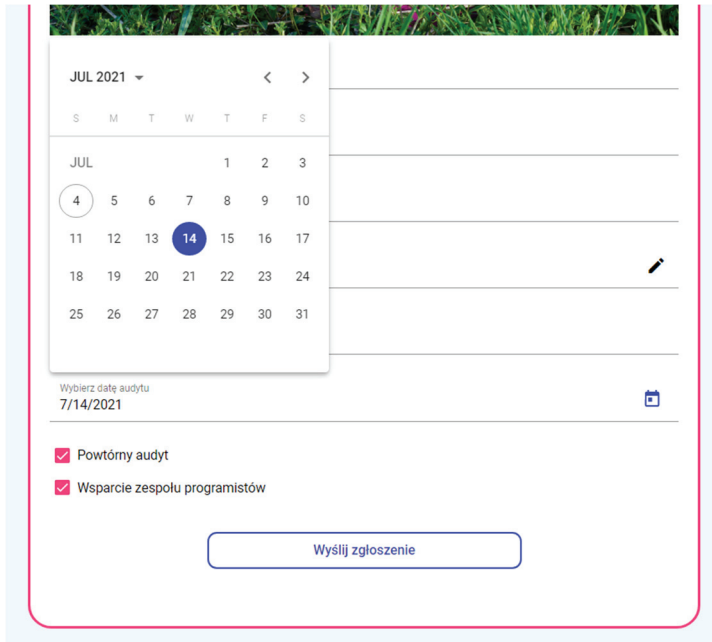
Wsparcie zespołu programistów

Wyślij zgłoszenie

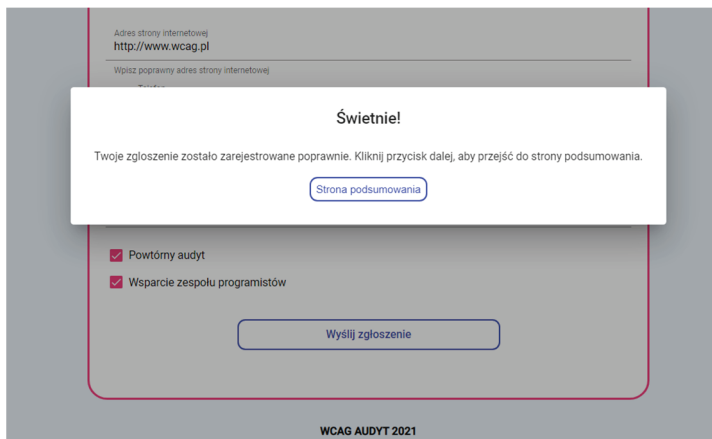
WCAG AUDYT 2021

Rysunek 4. Strona główna aplikacji – formularz

W przypadku poprawnego wypełnienia i akceptacji danych przez wbudowane walidatory, wyświetlane jest okno modalne informujące o sukcesie operacji (rysunek 6). Z punktu widzenia WCAG istotne jest, że fokus przeglądarki zostaje uwięziony wewnątrz okna, przez co nie jest możliwe nawigowanie po pozostałych elementach witryny, zarówno myszką jak i przy pomocy klawiatury, do momentu potwierdzenia przejścia do strony podsumowania.

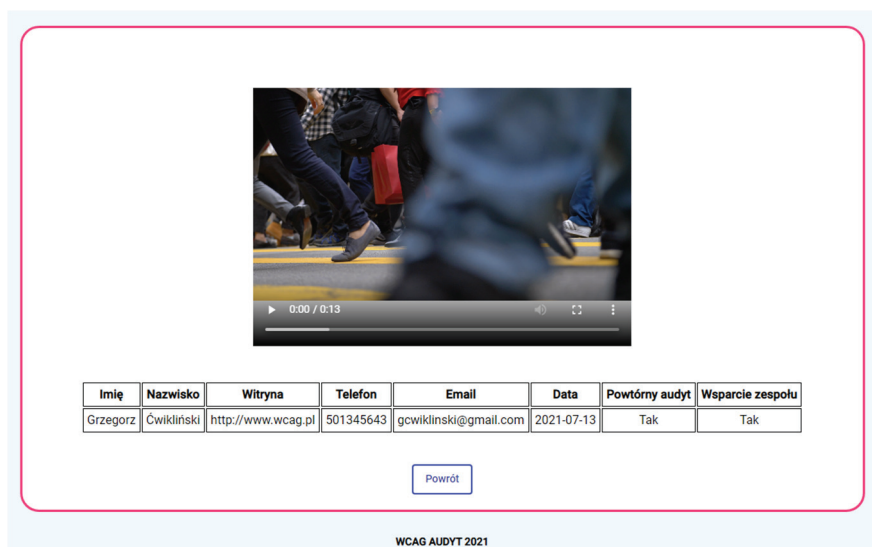


Rysunek 5. Moduł wyboru daty



Rysunek 6. Okno modalne aplikacji

Po przejściu do strony podsumowania, użytkownik może obejrzeć film instruktażowy informujący o tym jak przebiega audyt oraz może przejrzeć wprowadzone dane (rysunek 7). Wyświetlane są one przy pomocy modułu `mat-table` z biblioteki Angular Material. Jego zaletą jest prezentacja danych w sposób czytelni dla urządzeń wspomagających, z użyciem właściwej semantyki HTML (prawidłowe użycie tagów `<table>`, `<tr>`, `<th>`, `<td>` (odpowiednio – tabela, wiersz, nagłówek, zawartość).



Rysunek 7. Strona podsumowania

3. Testy dostępności z użyciem listy kontrolnej

Lista kontrolna zaproponowana przez Ministerstwo Cyfryzacji pozwala w sposób łatwy, przekrojowy oraz dosyć ogólny przeanalizować stronę WWW pod kątem zagadnień dostępności [7]. Wyszczególnione podpunkty wymagań WCAG 2.1 wraz z pytaniami pomocniczymi oraz merytorycznym opisem są bardzo dobrym wstępem do określenia poziomu dostępności witryny. Jednak nawet pozytywna odpowiedź na wszystkie z postawionych pytań nie może gwarantować dostępności na wskazanych poziomach oraz być traktowana jako pełny audyt strony.

Podczas ewaluacji witryny utworzonej na potrzeby niniejszego artykułu opisano wybrane podpunkty listy. Pominęto te z podpunktów, które nie mają zastosowania do utworzonego projektu strony. Przykładami pytań niemających zastosowania do pracy, są między innymi:

- Czy na stronie znajdują się elementy, które powodują gwałtowne zmiany jasności lub szybko błyskają na czerwono?
- Czy wszystkie znaczące elementy animowane lub dźwiękowe posiadają odpowiedni opis (tytuł) wyjaśniający co przedstawiają lub czego dotyczą?
- Czy na stronie jest mechanizm otwierający nowe okno przeglądarki bez udziału użytkownika?
- Czy jest na stronie aktualna mapa strony bądź wyszukiwarka, jeżeli jest to uzasadnione?
- Czy istnieje rozwiązanie powodujące dostępność CAPTCHA?
- Czy na stronie jest mechanizm zmieniający kontekst bez udziału użytkownika?
- Czy są na stronie elementy uruchamiające dźwięk, którego nie da się zatrzymać?
- Czy są na stronie migające lub poruszające się elementy, których nie da się zatrzymać?
- Czy na stronie jest pułapka klawiaturowa?
- Czy linki prowadzące do dokumentów do pobrania zawierają informację o ich języku, formacie oraz rozmiarze?
- Czy nie ma informacji przekazywanych jedynie za pomocą koloru?

Natomiast pytania mające zastosowanie do niniejszej pracy przedstawiono poniżej:

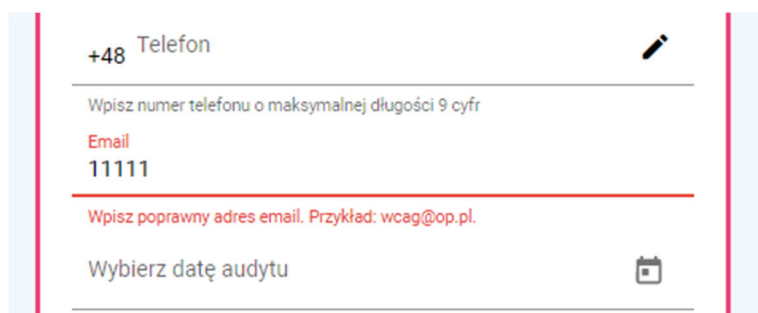
- **Czy wszystkie elementy aktywne w serwisie są dostępne za pomocą klawiatury?**

Tak, semantyczne znaczniki HTML5 zapewniają możliwość nawigowania po wszystkich interaktywnych elementach strony, takich jak przyciski, linki, pola formularza, a także elementy nawigacyjne odtwarzacza video. Brak użycia atrybutu „tab-index=0”, powodującego wyłączenie elementu z zakresu nawigacyjnego.

- **Czy informacja o błędzie w formularzu jednoznacznie ten błąd identyfikuje, jest dostępna i zrozumiała dla wszystkich użytkowników? Oraz Czy w przypadku pojawienia się błędów w danych wpisanych w formularzu**

pojawia się informacja sugerująca sposób w jaki można ponownie, poprawnie wpisać te dane?

Formularz na stronie jest napisany w sposób w pełni dostępny. W przypadku błędnego wypełnienia pola, czyli braku spełnienia warunku walidatora (na przykład dla formatu adresu email lub numeru telefonu, także w przypadku niepoprawnie wpisanej daty), aplikacja wizualnie podświetli błędne pole w kolorze czerwonym, przekaże informacje do czytnika ekranu poprzez atrybut „aria-invalid”, oraz wyświetli użytkownikowi podpowiedź poniżej pola (rysunek 8).



Rysunek 8. Zachowanie formularza w przypadku nie spełnienia warunku walidatora

- **Czy jest ostrzeżenie przed otwarciem nowego okna/zakładki w przeglądarce?**

W przypadku linku otwierającego nową kartę przeglądarki, informacja ta zawarta jest w tagu span z klasą css „sr-only”, która sprawia, że element nie jest widoczny w warstwie wizualnej witryny ani nie wpływa na jej wygląd. Istnieje tylko z perspektywy czytnika ekranu, który odczyta tekst znacznika gdy znajdzie się na nim fokus przeglądarki (rysunek 9).

```
<a _ngcontent-ubq-c133 href target="_blank" class="directive" == $0
  "Ustawa o dostępności cyfrowej"
  <span _ngcontent-ubq-c133 class="sr-only">link
  otwiera się w nowej karcie</span>
</a>
```

Rysunek 9. Sposób przekazania informacji o otwarciu nowej karty przeglądarki

- **Czy wszystkie multimedialne elementy wizualne zawierające ścieżkę dźwiękową posiadają poprawnie sformułowane napisy dla niesłyszących?** Aplikacja umożliwia wgranie pliku video wraz z napisami (format „.vvt”) oraz prezentację ich w odtwarzaczu HTML5. Ze względów bezpieczeństwa uruchomienie napisów z tagu <track> możliwe jest tylko w witrynie „otwartej na świat”, czyli umieszczonej na serwerze. Korzystając z zasobów lokalnych (pliki video oraz napisy) przeglądarka nie przedstawi ich jednocześnie (rysunek 10).

```
<div class="video-container">
  <video id="video" controls preload="metadata" width="500" height="400">
    <source src="..\assets\movie.mp4" type="video/mp4" >
    <track label="English" kind="subtitles" srclang="en" src="..\assets\captions.vtt" default>
  </video>
</div>
```

Rysunek 10. Implementacja odtwarzacza video

- **Czy kontrast tekstu w stosunku do tła wynosi co najmniej 4,5:1?** Tak, taki rezultat przestawił walidator WAVE na obu stronach, co kwalifikuje zgodność kontrastu na poziomie AAA.
- **Czy wszystkie tabele prezentujące dane mają poprawnie zdefiniowane nagłówki połączone z danymi?** Tabela na stronie podsumowania jest zbudowana z właściwych znaczników. Komórki nagłówków są zgrupowane w wierszu, a ten należy do znacznika <thead> (table head), natomiast odpowiadające nagłówkom komórki zgrupowane są w znaczniku <tbody> (table body) (rysunek 11).

```
▼<table _ngcontent-uaq-c56 cdk-table class="cdk-table" role="grid" ng-reflect-data-source="[object Object]">
  ▼<thead role="rowgroup">
    ▼<tr _ngcontent-uaq-c56 role="row" cdk-header-row class="cdk-header-row">
      <th _ngcontent-uaq-c56 role="columnheader" cdk-header-cell class="cdk-header-cell cdk-column-name"> Imię </th>
      <th _ngcontent-uaq-c56 role="columnheader" cdk-header-cell class="cdk-header-cell cdk-column-surname"> Nazwisko </th>
      <th _ngcontent-uaq-c56 role="columnheader" cdk-header-cell class="cdk-header-cell cdk-column-website"> Witryna </th>
      <th _ngcontent-uaq-c56 role="columnheader" cdk-header-cell class="cdk-header-cell cdk-column-phone"> Telefon </th>
      <th _ngcontent-uaq-c56 role="columnheader" cdk-header-cell class="cdk-header-cell cdk-column-email"> Email </th>
      <th _ngcontent-uaq-c56 role="columnheader" cdk-header-cell class="cdk-header-cell cdk-column-date"> Data </th>
      <th _ngcontent-uaq-c56 role="columnheader" cdk-header-cell class="cdk-header-cell cdk-column-reaudyt"> Powtórny audyt </th>
      <th _ngcontent-uaq-c56 role="columnheader" cdk-header-cell class="cdk-header-cell cdk-column-support"> Wsparcie zespołu </th>
      <!--ng-container-->
    </tr>
    <!--ng-container-->
  </thead>
  ▼<tbody role="rowgroup">
    ▼<tr _ngcontent-uaq-c56 role="row" cdk-row class="cdk-row">
      <td _ngcontent-uaq-c56 role="gridcell" cdk-cell class="cdk-cell cdk-column-name"> Grzegorz </td> == $0
      <td _ngcontent-uaq-c56 role="gridcell" cdk-cell class="cdk-cell cdk-column-surname"> Ćwikliński </td>
```

Rysunek 11. Implementacja tabeli na stronie podsumowania

- **Czy nie ma efektu jąkania?**

Testy czytnikiem ekranu NVDA nie wykazały efektu jąkania, czyli powtórnego czytania tego samego tekstu. Taka sytuacja może mieć miejsce, na przykład, gdy element <buton> ma wpisany tekst „wyślij formularz” w środku znaczniku (widoczny w witrynie) oraz dodatkowo błędnie w atrybucie „title” (z założenia mającym go dodatkowo identyfikować, jednak realnie zaciemnia on i utrudnia odbiór strony przez osobę niewidomą).

- **Czy wszystkie graficzne elementy czysto dekoracyjne mają pusty atrybut alt?**
Atrybut „alt” oznacza tekst alternatywny wyświetlany w przypadku, gdy obrazek nie może zostać załadowany na stronie. Jeśli obraz zawiera informację, która powinna zostać przekazana użytkownikowi, wartość wspomnianego atrybutu będzie zawierała jego opis (rysunek 12).

```
 == $0
```

Rysunek 12. Opis głównej grafiki w kodzie HTML

Elementem graficznym czysto dekoracyjnym jest ikona pióra przy polu numer telefonu, więc powinna ona mieć atrybut „alt” (alternative text), jednak jest schowana przed czytnikiem poprzez użycie atrybutu „aria-hidden”, przez co ta zasada nie ma zastosowania.

- **Czy umiejscowienie etykiet pól formularzy (zarówno w kodzie jak i wizualnie) nie pozostawia żadnych wątpliwości?**

```
<input _ngcontent-oig-c133 matinput formcontrolname="name" type="text" class="mat-input-element mat-form-field-autofill-control ng-tns-c106-0 ng-untouched ng-pristine ng-invalid cdk-text-field-autofill-monitored" ng-reflect-type="text" ng-reflect-name="name" id="mat-input-0" aria-invalid="false" aria-required="false" control-id="ControlID-1">  
▼<span class="mat-form-field-label-wrapper ng-tns-c106-0">  
▼<label class="mat-form-field-label ng-tns-c106-0 mat-empty mat-form-field-empty ng-star-inserted" ng-reflect-disabled="true" id="mat-form-field-label-1" ng-reflect-ng-switch="true" for="mat-input-0" aria-owns="mat-input-0">
```

Rysunek 13. Powiązanie etykiety z odpowiadającym jej polem

W bibliotece Angular Material, pola oraz ich etykiety są zagnieżdżone, zarówno w kodzie HTML jak i wizualnie, przez co użytkownik jednoznacznie uzna ich korelację. Programistycznie pole z etykietą jest związane atrybutami `for` (dla tagu `<label>`) oraz `id` (dla tagu `<input>`) (rysunek 13).

- **Czy każda strona ma poprawną deklarację języka?**

Tak, definicja języka jest ustalona globalnie dla całego projektu. W przypadku wielojęzycznej wersji strony atrybut „`lang`” nadany tagowi głównemu `<html>` powinien automatycznie się uaktualniać.

- **Czy na każdej stronie jest przynajmniej jeden nagłówek h1?**

Tak, nagłówki informują o organizacji treści na stronie. Przeglądarki internetowe, wtyczki i technologie wspomagające mogą ich używać do nawigacji po stronie, przy zachowaniu ich hierarchiczności. Najważniejszy nagłówek ma rangę 1 (`<h1>`), najmniej ważny rangę 6 (`<h6>`). Nagłówki o takiej samej lub wyższej randze rozpoczynają nową sekcję, nagłówki o niższej randze rozpoczynają nowe podrozdziały.

- **Czy po powiększeniu strony do 200% rozmiarów początkowych utrzymana jest widoczność całości informacji na stronie?**

Tak, responsywność serwisu zostaje zachowana przy takim powiększeniu, dzięki prawidłowo zaimplementowanym stylom CSS. Cała zawartość strony jest możliwa do przejrzenia, nie wymagając poziomego przesunięcia. Żaden z jej elementów nie zostaje utracony bądź ukryty.

- **Czy wszystkie multimedialne elementy wizualne zawierające ścieżkę dźwiękową niezbędną dla zrozumienia zawartości posiadają poprawnie sformułowane napisy dla niesłyszących?**

Aplikacja umożliwia dodane do folderu `/assets` pliku z napisami w formacie „`vvt`”, który definiuje znaki mające się pojawić w odpowiednich momentach nagrania.

- **Czy wszystkie elementy multimedialne niosące treść są dostępne bez użycia myszki lub posiadają dostępną alternatywę?**

Elementem multimedialnym witryny jest odtwarzacz HTML5, którego przyciski nawigacyjne są dostępne z poziomu samej klawiatury oraz mają wyróżniony fokus, dzięki użyciu białej czarnej ramki aktywnego elementu, widocznej na różnych klatkach nagrania.

- **Czy wszystkie graficzne elementy klikalne lub podzielone na strefy klikalne są dostępne z klawiatury?**

Brak jest na stronie elementów, do których w sposób sztuczny (czyli przy użyciu JavaScript) dodawana jest możliwość interakcji. Taka sytuacja miałaby miejsce, gdyby na przykład znacznik „div” został przypisany do funkcji kierującej użytkownika do następnej strony. Czytnik ekranu nie uzyskałby tej informacji, przez co element byłby dla niego statycznym kontenerem.

- **Czy jakiś element na badanej stronie wskazuje elementy za pomocą koloru?**

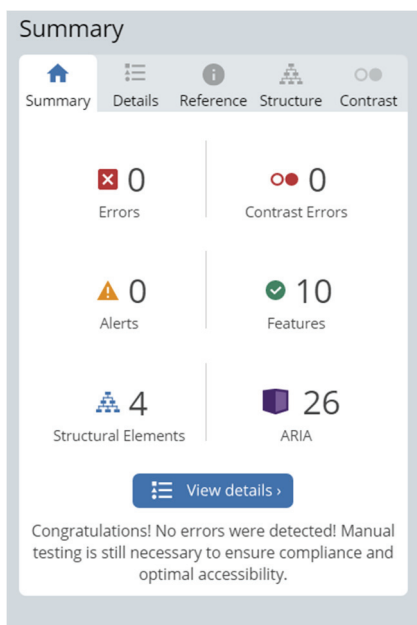
Żaden z elementów strony nie wskazuje na inny element, opisując go pod kątem wizualnym, zarówno w formie tekstu widocznego na stronie (na przykład instrukcja mówiąca o czerwonym podkreśleniu błędnie wypełnionych pól), jak też zawartej w atrybutach aria-label widocznych dla czytników ekranu.

4. Testy przeprowadzone walidatorem WAVE

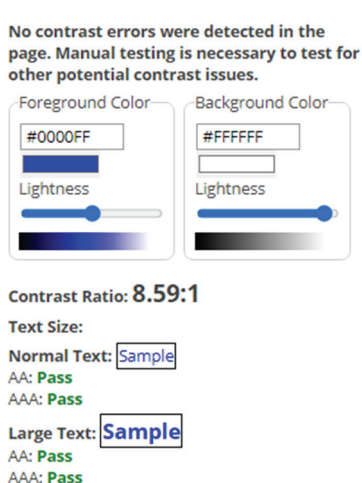
WAVE to zestaw narzędzi ułatwiających ocenę dostępności stron internetowych, poprzez zapewnienie wizualnej reprezentacji problemów z dostępnością w obrębie strony [8]. Program dostępny jest w formie serwisu online lub jako rozszerzenie do przeglądarek Firefox i Chrome. Automatyczny audyt przeprowadzany przez to narzędzie sprawdza dostępność strony w wąskim zakresie i może być wykorzystywany tylko częściowego potwierdzenia zgodności z WCAG 2.1.

Raport programu WAVE widoczny na rysunkach 14 i 15 informuje o braku błędów oraz ostrzeżeń związanych z dostępnością witryny, a także o poprawnym stosunku kolorów, w celu zachowania odpowiedniego kontrastu (zarówno na poziomie AA jak

i AAA). Wskazuje na liczbę użytych atrybutów „aria”, tworzących dla czytelnika pewnego rodzaju mapę, uzupełniając semantyczne elementy HTML5.



Rysunek 14. Wynik audytu walidatora WAVE



Rysunek 15. Wynik audytu kontrastu walidatora WAVE

Stosunek kolorów na stronie o wartości ponad 8:1 wskazuje na zgodność ze standardem WCAG 2.1 w wersji AAA. Badany był zarówno tekst wraz ze swoim tłem, jak i kontrast elementów nietekstowych (np. przycisk i otaczająca go ramka). Automatyczny walidator ma jednak swoje ograniczenia, warto więc manualnie sprawdzać stosunek kolorów w niewrażliwych punktach aplikacji.

5. WCAG 3.0

Aktualnie przedmiotem konsultacji prowadzonych przez konsorcjum W3C jest WCAG 3.0. Wersja 3.0 zachowuje skrót WCAG, ale w treści wiele się zmienia [9]. Przykładowo, WCAG 3.0 ma nie być wstecznie kompatybilny. Zmieniła się także geneza skrótu. Zamiast nawiązania do „Web content”, obecnie pochodzi ono od „W3C accessibility guidelines”. Odniesienie do treści zostało usunięte, co może być subtelnym wskaźnikiem zwiększonego zakresu wytycznych. Publikacja nowego standardu jest przewidziana na rok 2023.

Niektóre z celi stawianych przed nowym standardem to:

- chęć zaspokojenia potrzeb szerszego grona użytkowników,
- obsługa nowych technologii, takich jak wirtualna rzeczywistość (VR) i rzeczywistość rozszerzona (AR),
- bardziej precyzyjnie zdefiniowane procedury testowe,
- różnorodne podejścia do testowania (nie tylko stwierdzenie: prawda / fałsz),
- wprowadzenie częstszych aktualizacji normy,
- traktowanie zapewniania zgodności jako procesu,
- wytyczne sformułowane w bardziej zrozumiały dla użytkowników.

Poprzednia wersja WCAG, gdy zaczęła obowiązywać w krajach Europy (szczególnie w projektach finansowanych z budżetów państw), wywołała duże zmiany w podejściu do projektowania stron i ich implementacji, jak również w wymaganiach odnośnie kandydatów i trendach na rynku pracy. Popularniejszy także stał się zawód „audytor stron internetowych”, jeszcze niedawno praktycznie nieistniejący. Zakładać można, że czas, w którym WCAG 3.0 będzie stawał się obowiązującym standardem, może również przynieść pewnego rodzaju ewolucję w podejściu do pracy oraz użytkowania stron internetowych.

6. Podsumowanie

W ramach pracy opisanej w niniejszym artykule zaimplementowano stronę WWW umożliwiającą zebranie danych od użytkownika z niepełnosprawnościami oraz przesłanie ich do części typu backend aplikacji i późniejsze wykorzystanie. Rezultatem przedstawionej w pracy analizy fragmentu portalu internetowego pod kątem zaleceń WCAG 2.1 jest spełnienie większości kryteriów na wszystkich trzech poziomach zgodności.

Jako że liczba implementacji portali jest nieskończona, ocena zgodności strony pod kątem osób niepełnosprawnych jest zagadnieniem, którego poziom złożoności rośnie wraz z poziomem skomplikowania portalu, możliwych podstron, oraz prezentowanych funkcji. Na finalną dostępność strony ma wpływ szereg czynników, takich jak właściwy projekt UX/UI (User Experience/User Interface) witryny, użycie odpowiednich tagów i atrybutów języka znaczników, użycie zgodnie z WCAG 2.1 kaskadowych arkuszy styli oraz poprawna implementacja zewnętrznych bibliotek napisana w omawianym standardzie.

Można stwierdzić, że obowiązek i ciężar tworzenia dostępnych treści leży nie tylko po stronie programisty. Cały zespół dostarczający rozwiązania programistyczne, w skład którego wchodzi także architekci, analitycy, projektanci oraz testerzy, powinien mieć świadomość zasad oraz celu tworzenia dostępnych rozwiązań IT. W przypadku portali internetowych współtworzonych przez redaktorów, mogących na przykład tworzyć i dodawać artykuły zawierające obrazy, filmy czy style tekstu, istotne może być ich przeszkolenie z obowiązujących kryteriów sukcesu WCAG 2.1.

Literatura

- [1] The World Wide Web Consortium (W3C), <https://www.w3.org/>. [8.11.2021].
- [2] *What is the Americans with Disabilities Act (ADA)?*, ADA National Network, <https://adata.org/learn-about-ada>. [8.11.2021].
- [3] J. Pan, *508, ADA, WCAG: What's the difference?*, Logic Solutions 2017, <https://www.logicsolutions.com/508-ada-wcag-accessibility-difference/>. [8.11.2021].

- [4] Ch. Ward, *7 Ways to Make Your Angular App More Accessible*, Progress, Telerik 2019, <https://www.telerik.com/blogs/7-ways-to-make-your-angular-app-more-accessible>. [8.11.2021].
 - [5] A. Damaziak, *Serwis internetowy prezentujący implementację wybranych zagadnień dostępności cyfrowej na podstawie wymagań WCAG 2.1*, Warszawa: Praca dyplomowa inżynierska. Warszawska Wyższa Szkoła Informatyki, 2021.
 - [6] *Screen readers*, AFB – American Foundation for the Blind, <https://www.afb.org/blindness-and-low-vision/using-technology/assistive-technology-products/screen-readers>. [8.11.2021].
 - [7] *Lista kontrolna do badania zgodności stron internetowych z wymaganiami ustawy z dnia 4 kwietnia 2019 r. o dostępności cyfrowej stron internetowych i aplikacji mobilnych podmiotów publicznych*, gov.pl <https://www.gov.pl/attachment/2d7c2610-512b-4796-a0ba-5b4e50b858f6>. [8.11.2021].
 - [8] *WAVE Web Accessibility Evaluation Tool*, WAVE, <https://wave.webaim.org/>. [8.11.2021].
 - [9] *W3C Accessibility Guidelines (WCAG) 3.0*, <https://www.w3.org/TR/wcag-3.0/>. [8.11.2021].
-

Website presenting the implementation of selected issues of digital accessibility based on WCAG 2.1

Abstract

This article describes the process of preparing a form that allows a user to submit a website for testing the level of digital site accessibility. The audit is commissioned to an audit company that checks the website in terms of meeting WCAG 2.1 recommendations. The study returns answers to a number of key questions. The issue of website evaluation in terms of accessibility by people

with disabilities is presented. The functional and non-functional requirements of the website validation system were developed, the architecture and operation of the implemented system were discussed. The results of availability tests based on the list of control inquiries and the WAVE validator are presented.

Keywords: *WCAG 2.1, website digital accessibility, accessibility tests, WAVE, WCAG 3.0*