

TI	Dokumentacja Projektu HTML i CSS
Autor	Oliwer Strzałka
Kierunek, rok	Informatyka i ekonometria, II rok, st. Stacjonarne (3.5-l)
Temat	Implementacja strony opisującej mechanizm działania i wykorzystywania imprezentowanej biblioteki JS

Tematyka projektu.....	1
Realizacja projektu.....	2
Testowanie aplikacji	3
Testowanie responsywności menu	4
Testowanie responsywności elementów interfejsu	5
Testy optymalizacji.....	8
Raport SEO.....	9
Implementacja.....	11
Menu.....	11

Tematyka projektu

Główym celem niniejszego projektu jest stworzenie profesjonalnej strony dokumentacyjnej dla autorskiej biblioteki JavaScript implementującej algorytm łamania i justowania tekstu. Witryna ma służyć jako kompleksowe źródło informacji dla programistów, prezentując zarówno teoretyczne podstawy działania algorytmu, jak i praktyczne aspekty jego wykorzystania w projektach webowych. Strona powinna szczegółowo opisywać implementowany mechanizm, jego architekturę oraz podejście do przetwarzania ciągów tekstowych.

Jednym z istotnych komponentów serwisu będzie sekcja demonstracyjna z interaktywnymi przykładami, która umożliwi użytkownikom testowanie funkcjonalności biblioteki bezpośrednio w przeglądarce, co pozwoli lepiej zrozumieć jej działanie i zachęci do wdrożenia w własnych aplikacjach. Aby zilustrować efektywność i precyzję algorytmu, planuje się dodanie sekcji z przykładowymi fragmentami kodu, prezentującymi różne scenariusze zastosowań – od podstawowego justowania po zaawansowane łamanie wyrazów z łącznikami. Ważnym elementem serwisu będzie także część dokumentacyjna API, gdzie użytkownicy odnajdą wszystkie potrzebne informacje techniczne, w tym dostępne metody, parametry konfiguracyjne, typy zwracanych wartości oraz przykłady użycia. W tym dziale przewiduje się również

implementację interaktywnej konsoli testowej, która umożliwi programistom szybkie eksperymentowanie z kodem bezpośrednio w środowisku przeglądarki.

Witryna powinna charakteryzować się atrakcyjną i spójną gamą kolorystyczną zdominowaną przez odcienie niebieskiego i szarości, co wpłynie na jej profesjonalny wygląd i odbiór wizualny, podkreślając techniczny charakter projektu. Kolejnym kluczowym wymaganiem jest zapewnienie pełnej responsywności serwisu, tak aby działał on prawidłowo zarówno na komputerach stacjonarnych, jak i na urządzeniach mobilnych, umożliwiając wygodne studiowanie dokumentacji na ekranach o różnych rozmiarach. Strona powinna być również zoptymalizowana pod względem różnych przeglądarek internetowych (Chrome, Firefox, Safari, Edge), aby zapewnić programistom płynną i bezproblemową pracę z dokumentacją niezależnie od wybranego środowiska developerskiego.

Reasumując, projekt zakłada opracowanie funkcjonalnej i estetycznej strony internetowej, która dostarczy użytkownikom dogodny dostęp do szczegółowej dokumentacji biblioteki, wysoką wydajność w prezentacji przykładów kodu, pełną responsywność i kompatybilność z różnymi urządzeniami oraz przeglądarkami, a także interaktywne narzędzia wspomagające naukę i integrację implementowanego algorytmu.

Realizacja projektu

W celu realizacji założeń projektu wykorzystano języki HTML, CSS oraz JavaScript do implementacji interaktywnych funkcjonalności. Podstawą serwisu jest przejrzystość, funkcjonalność oraz responsywność, co gwarantuje użytkownikom komfortowe korzystanie z dokumentacji na różnych platformach.

Etap 1: Opracowanie identyfikacji wizualnej i struktury informacyjnej

Pierwszą fazą realizacji było przygotowanie spójnego schematu kolorystycznego i typograficznego, które odzwierciedlają techniczny charakter przedsięwzięcia. Dominująca paleta niebiesko-szara została zastosowana we wszystkich modułach, budując profesjonalny wizerunek. Układ serwisu zaprojektowano w oparciu o hierarchię informacji – od ogólnego wprowadzenia po szczegółową dokumentację API.

Etap 2: Budowa nawigacji i interfejsu użytkownika

Menu główne zostało zaprojektowane w sposób przejrzysty i intuicyjny, z wyraźnym podziałem na sekcje tematyczne. Na urządzeniach przenośnych zastosowano tzw. hamburger menu, które umożliwia łatwe rozwinięcie listy opcji bez zajmowania przestrzeni ekranu. Dodatkowo wdrożono przycisk przełączania między trybem jasnym i ciemnym, zwiększając dostępność i komfort użytkowania.

Etap 3: Konstrukcja strony głównej

Centralną część strony głównej zajmuje powitalny nagłówek z krótkim opisem biblioteki i wizualizacją przykładowego kodu. Poniżej znajduje się sekcja wprowadzająca prezentującą kluczowe informacje o projekcie w formie kart z ikonami – wyróżniono m.in. cel projektu, zasadę działania i technologie implementacyjne.

Etap 4: Prezentacja możliwości biblioteki

Dział funkcji przedstawia szczegółowy zakres możliwości implementowanego algorytmu, obejmując takie operacje jak łamanie wyrazów, justowanie tekstu, responsywność i konfigurowalność. Każda funkcjonalność jest opisana w sposób zrozumiały i uzupełniona odpowiednią ikoną oraz numeracją, co podnosi czytelność oferty.

Etap 5: Dokumentacja zastosowań i integracji

W celu ułatwienia wdrożenia biblioteki w projektach zewnętrznych, dodano sekcję z instrukcją szybkiego startu. Instrukcja przedstawiona została w postaci etapów zawierających fragmenty kodu, opisy konfiguracji i przykłady wywołań metod.

Etap 6: Opis algorytmu i interfejsu programistycznego

Część techniczna prezentuje szczegółowy opis architektury algorytmu oraz dokumentację interfejsu programistycznego. Informacje przedstawiono w postaci ustrukturyzowanych bloków zawierających nazwy metod, ich parametry, zwarcane wartości oraz krótkie opisy działania.

Etap 7: Interaktywna prezentacja działania

Dla zwiększenia użyteczności serwisu zaimplementowano dedykowaną stronę demonstracyjną (demo.html), umożliwiającą testowanie biblioteki w środowisku przeglądarki. Ta sekcja zawiera edytowalne pola tekstowe, suwaki konfiguracyjne oraz podgląd wyników w czasie rzeczywistym.

Etap 8: Elementy dodatkowe i aspekty dostępności

Wdrożono przycisk „Przejdź do głównej treści” dla użytkowników nawigujących klawiaturą, zapewniono odpowiedni kontrast kolorystyczny oraz strukturę nagłówków zgodną z wymaganiami WCAG 2.1. Wszystkie interaktywne komponenty są w pełni obsługiwane za pomocą klawiatury.

Testowanie aplikacji

Proces weryfikacji strony internetowej został zaplanowany w celu dokładnej analizy jej funkcjonalności, dostępności oraz optymalizacji, co zapewni użytkownikom komfortową i bezproblemową pracę z dokumentacją. Testy obejmują kilka kluczowych obszarów.

Pierwszym etapem jest sprawdzenie dostępności wszystkich sekcji serwisu na różnych urządzeniach. Witryna będzie testowana pod kątem poprawnego renderowania na komputerach, tabletach i smartfonach, aby upewnić się, że prezentuje się i funkcjonuje prawidłowo w każdej rozdzielcości ekranu. W tym celu zostaną wykorzystane narzędzia developerskie przeglądarki oraz emulatory urządzeń, które pozwolą ocenić responsywność strony.

Kolejną fazą jest optymalizacja serwisu. W ramach tych testów zostanie przeanalizowany czas ładowania witryny oraz jej wydajność. Narzędzia takie jak Google PageSpeed Insights i Lighthouse umożliwiają ocenę szybkości ładowania oraz wskażą obszary wymagające usprawnień. Zostaną również zweryfikowane rozmiary plików graficznych, arkuszy CSS oraz skryptów JavaScript, aby upewnić się, że są odpowiednio skompresowane i zoptymalizowane, co przełoży się na lepsze działanie serwisu.

Niezwykle istotnym elementem testów jest walidacja kodu. Zostanie ona przeprowadzona przy użyciu narzędzi takich jak W3C Validator, aby upewnić się, że kod HTML i CSS jest zgodny z obowiązującymi standardami. Ponadto, przy pomocy serwisu Can I Use, zostanie przeanalizowana kompatybilność zastosowanych znaczników i stylów z różnymi przeglądarkami, co zagwarantuje spójne działanie strony na platformach takich jak Chrome, Firefox, Safari i innych.

Istotną częścią testów są również testy funkcjonalne, które mają na celu sprawdzenie poprawności działania interaktywnych komponentów serwisu. Weryfikacji poddane zostaną menu nawigacyjne, przycisk zmiany motywu, interaktywne przykłady kodu oraz strona demonstracyjna. Dodatkowo, nawigacja klawiszowa zostanie przetestowana pod kątem obsługi wszystkich interaktywnych elementów, aby upewnić się, że użytkownicy niekorzystający z myszy mogą w pełni wykorzystywać witrynę.

Ostatnim, ale równie ważnym aspektem testów, jest analiza dostępności serwisu dla osób z niepełnosprawnościami. Zostanie oceniona czytelność strony, w tym kontrast kolorystyczny oraz struktura nagłówków, a także jej obsługa za pomocą czytników

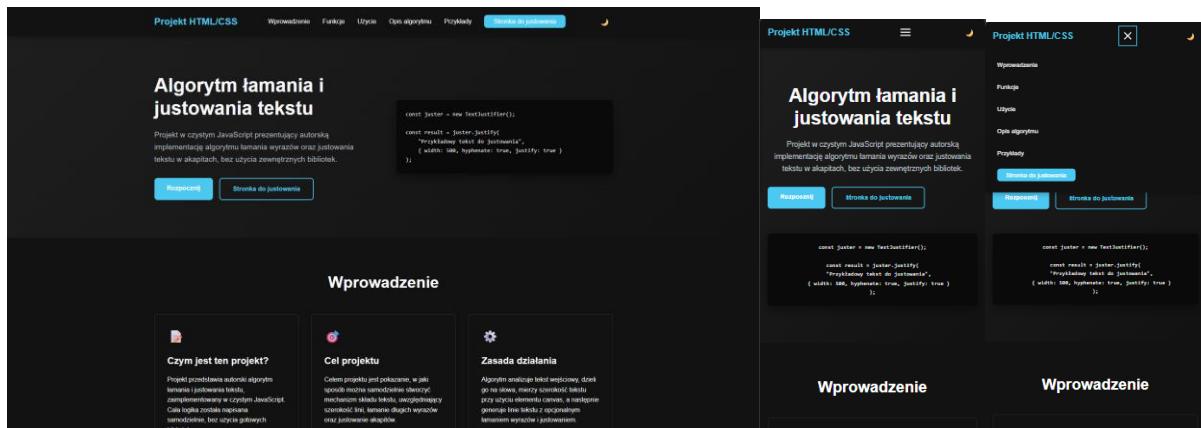
ekranowych, co pozwoli na dostosowanie witryny do potrzeb szerokiego grona odbiorców.

Przeprowadzenie tych szczegółowych testów pozwoli zoptymalizować stronę internetową pod kątem wygody użytkowania, wysokiej wydajności oraz zgodności ze standardami, co w efekcie zapewni pozytywne doświadczenia dla wszystkich użytkowników.

Testowanie responsywności menu

Założenia testu: Poprawne renderowanie na różnych rozdzielcościach ekranu.

- **Na urządzeniach desktopowych:** Menu powinno być w pełni widoczne w formie poziomego paska z opcjami nawigacyjnymi, z logo po lewej stronie i przyciskiem zmiany motywu po prawej.
- **Na urządzeniach mobilnych i tabletach:** Menu powinno być zwinięte do formy hamburger menu (ikona trzech poziomych linii), która rozwija listę opcji po aktywacji, z zachowaniem wszystkich elementów nawigacyjnych.



Rysunek 1. Prawa strona: Pokazanie widoku wersji desktopowej (1920x1080) Środek: Pokazanie widoku wersji mobilnej (578x1080). Lewa strona: Ten sam widok jednak z rozwiniętym menu

Rezultaty testów:

- Na urządzeniach stacjonarnych menu renderuje się w formie poziomego paska nawigacyjnego z widocznymi wszystkimi opcjami. Elementy menu są właściwie rozmiieszczone i w pełni aktywne. Nawigacja działa płynnie, a opcje kierują do odpowiednich sekcji serwisu.

- Na urządzeniach przenośnych i tabletach menu zostało prawidłowo dostosowane do mniejszych ekranów. Hamburger menu funkcjonuje zgodnie z założeniami – ikona trzech poziomych linii jest czytelna i łatwo dostrzegalna. Po aktywacji ikony menu rozwija się płynnie, prezentując listę opcji nawigacyjnych. W przypadku zamknięcia menu proces ten również przebiega bez opóźnień czy nieprawidłowości wizualnych.
- Kompatybilność menu została potwierdzona na różnych przeglądarkach, takich jak Opera, Edge, oraz na systemach operacyjnych Android. Testy przeprowadzone potwierdziły pełną funkcjonalność i poprawne działanie menu na szerokiej gamie urządzeń.

Testowanie responsywności elementów interfejsu

Założenia testu: Weryfikacja prawidłowego dostosowania trzech kluczowych komponentów serwisu do różnych rozdzielczości ekranu.

Test 1: Sekcja powitalna z przykładem kodu

- **Na urządzeniach desktopowych:** Układ dwukolumnowy (opis tekstowy po lewej, blok kodu po prawej). Tekst jest czytelny, przyciski właściwie rozmieszczone.
- **Na urządzeniach mobilnych:** Układ jednokolumnowy, tekst wycentrowany, blok kodu umieszczony pod opisem, zachowuje czytelność i nie powoduje przewijania poziomego

Algorytm łamania i justowania tekstu

Projekt w czystym JavaScript prezentujący autorską implementację algorytmu łamania wyrazów oraz justowania tekstu w akapitach, bez użycia zewnętrznych bibliotek.

[Rozpoznać](#) [Stronka do justowania](#)

Algorytm łamania i justowania tekstu

Projekt w czystym JavaScript prezentujący autorską implementację algorytmu łamania wyrazów oraz justowania tekstu w akapitach, bez użycia zewnętrznych bibliotek.

[Rozpoznać](#) [Stronka do justowania](#)

Wprowadzenie

Czym jest ten projekt?

Projekt przedstawia autorską algorytm łamania i justowania tekstu, zaimplementowany w czystym JavaScript. Cała logika została napisana samodzielnie, bez użycia gotowych bibliotek.

Cel projektu

Celem projektu jest pokazanie, w jaki sposób można samodzielnie stworzyć mechanizm składu tekstu, uwzględniający szerokość linii, łamanie długich wyrazów oraz justowanie akapitów.

Zasada działania

Algorytm analizuje tekst wojcieczy, dzieli go na słowa, mierzy szerokość tekstu przy użyciu elementu canvas, a następnie generuje linię tekstu z ojściowym łamaniem wyrazów i justowaniem.

Wprowadzenie

Rysunek 2. Screen dokumentujący ułożenie sekcji powitalnej (na urządzeniach mobilnych 1 kolumnowy, a na urządzeniach desktopowych 2 kolumnowy)

Test 2: Karty informacyjne w dziale wprowadzenia

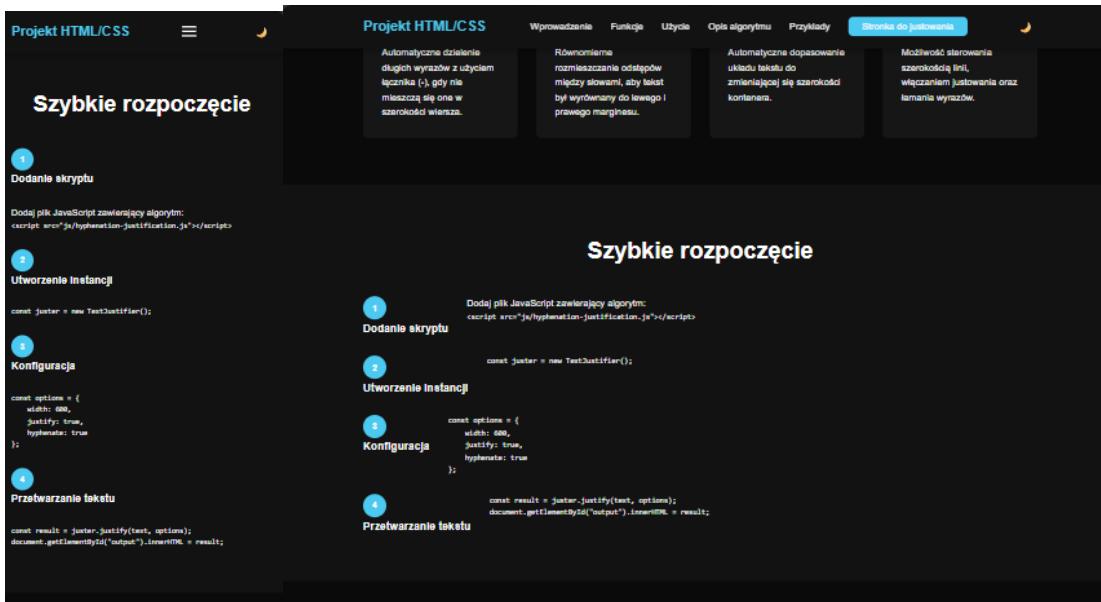
- **Na urządzeniach desktopowych:** Trzy karty prezentowane obok siebie w jednym rzędzie, z równymi odstępami
- **Na tabletach:** Dwie karty w rzędzie, trzecia przechodzi do kolejnego wiersza
- **Na smartfonach:** Jedna karta na pełną szerokość ekranu, ułożone w kolumnie

The figure consists of two screenshots of a mobile application interface. The top screenshot, titled 'Wprowadzenie' (Introduction), displays three cards side-by-side. The first card is titled 'Czym jest ten projekt?' (What is this project?) and contains text about an auto-breaking algorithm. The second card is titled 'Cel projektu' (Project goal) and describes the algorithm's purpose. The third card is titled 'Zasada działania' (Operation principle) and explains the algorithm's internal logic. The bottom screenshot, also titled 'Wprowadzenie', shows a different layout. The first card is split into two columns. The left column contains the same information as the first card in the top screenshot. The right column contains the same information as the second and third cards in the top screenshot. A fourth card, 'Użycie' (Usage), is shown below the first card, indicating a two-column layout for tablets and a single-column layout for mobile devices.

Rysunek 3. Screeny dokumentujące ułożenie sekcji wprowadzenia. Na górze wersja desktopowa z 3 kolumnami, poniżej po prawej wersja mobilna jednokolumnowa oraz wersja (na tablety) 2 kolumnowa

Test 3: Sekcja etapów użycia (usage-steps)

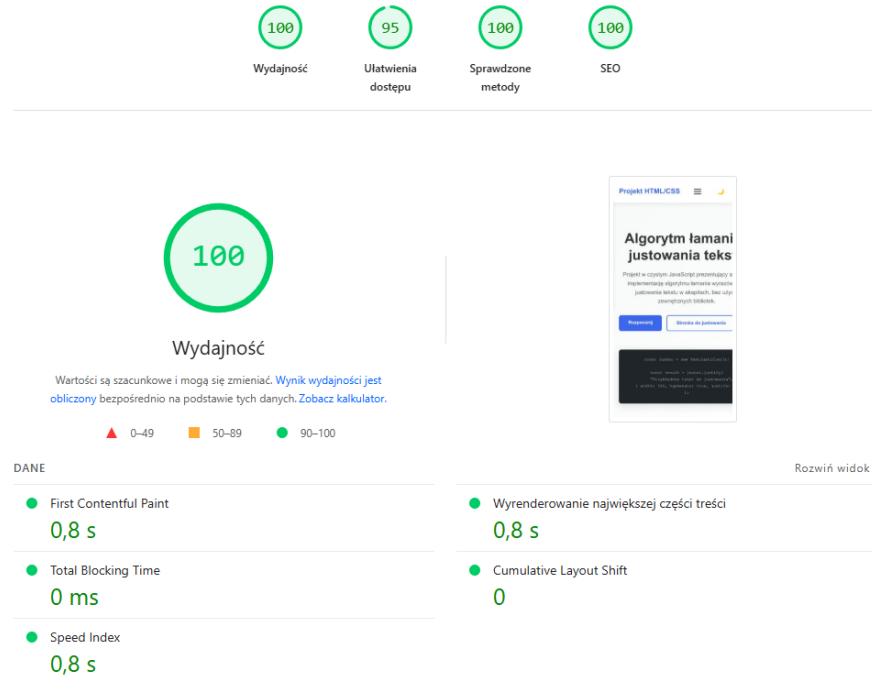
- **Na urządzeniach desktopowych:** Kroki wyświetlane w postaci poziomej sekwencji z numeracją i treścią obok siebie.
- **Na urządzeniach mobilnych:** Kroki układają się w kolumnę, numeracja pozostaje widoczna, zawartość przechodzi pod numer.



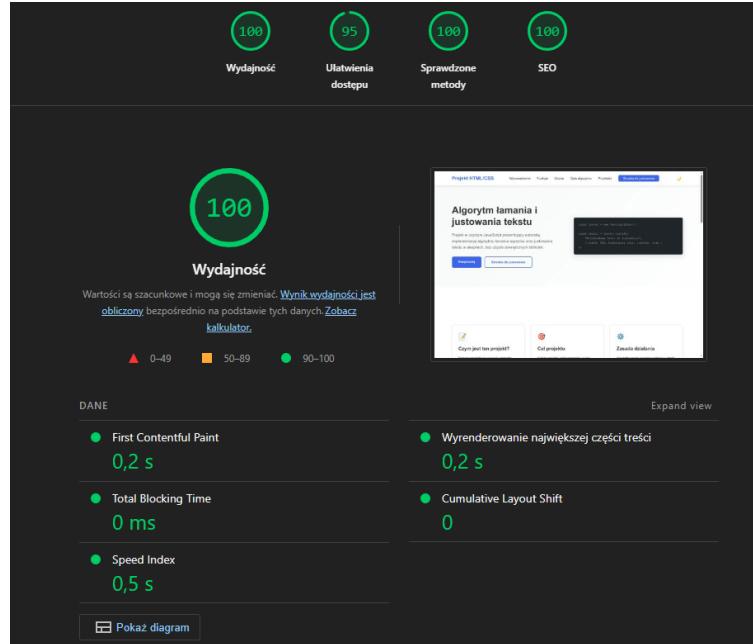
Rysunek 4. Screeny dokumentujące ułożenie etapów użycia (po lewej stronie widok jednokolumnowy z urządzenia mobilnego, a po prawej z desktopowego)

Wyniki weryfikacji: Wszystkie testowane elementy zachowują pełną responsywność. Przejścia między układami są płynne, nie zaobserwowano zakłóceń w layoutcie ani utraty funkcjonalności. Elementy interaktywne (przyciski, linki) utrzymują odpowiednie wymiary dotykowe na urządzeniach przenośnych (min. 44x44px).

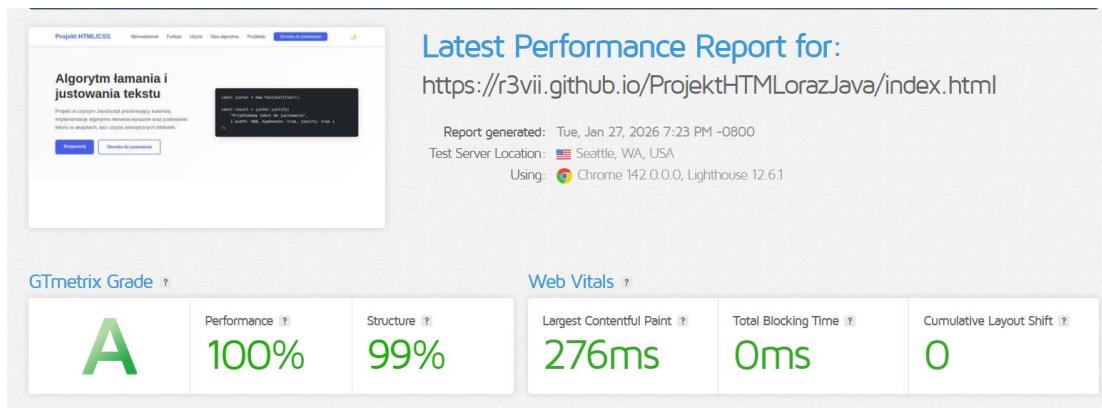
Testy optymalizacji



Rysunek 5. Wynik testu ze strony pagespeed.web



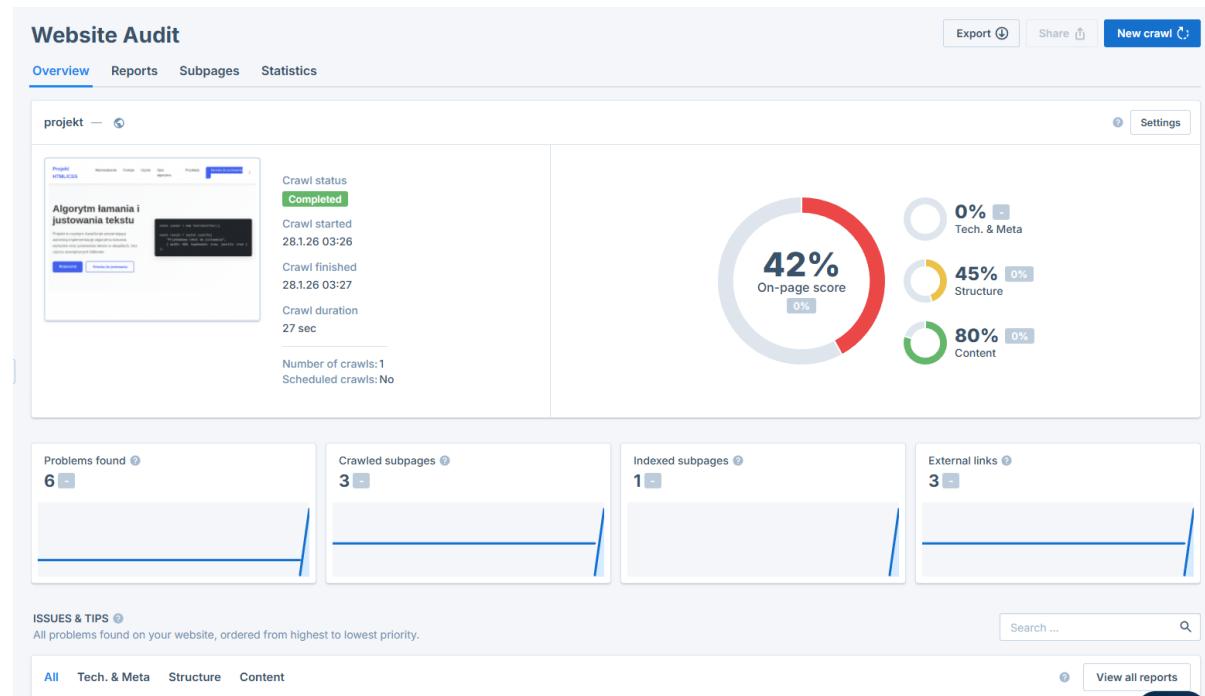
Rysunek 6. Wynik testu z narzędzia lighthouse



Rysunek 7. Wynik testu z strony gtmetrix.com

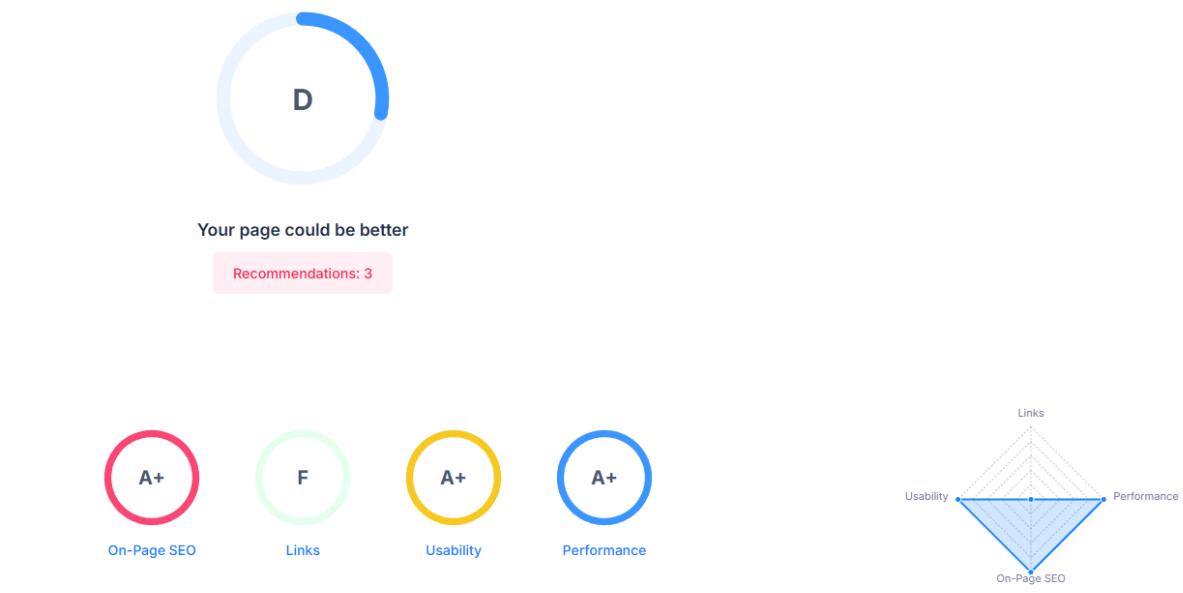
Wyniki testu optymalizacji wykazują, że strona internetowa jest w dużej mierze zoptymalizowana pod względem wydajności, z uwzględnieniem kluczowych aspektów takich jak czas ładowania, minimalizacja zasobów oraz optymalizacja mediów.

Raport SEO



Rysunek 8. Rysunek testu z witryny www.seobility.net

Audit Results for r3vii.github.io



Wyniki: Niestety nie byłem w stanie odpowiednio zrealizować wyników, gdyż witryna seobility.net wzięła dwa linki (odnośniki do moich podstron), które podsunął jej hosting na którym jest stronka. Dlatego wyniki mogą być zaniżone, z kolei seoptimer.com w darmowej wersji programów nie uznaje podstron, dlatego nie jest w stanie zanalizować linków które na niej są. Tak czy inaczej patrząc na ogólny uwagę, że strona odpowiednio spełnia swoje oczekiwania względem raportu SEO.

Implementacja

W ramach projektu opracowano arkusz stylów CSS, który umożliwia kompleksową kontrolę nad wyglądem i układem strony internetowej. Arkusz ten definiuje style dla różnych elementów HTML, takich jak nagłówki, akapity, przyciski, karty, sekcje i inne komponenty, zapewniając spójność wizualną i responsywność serwisu na różnych platformach.

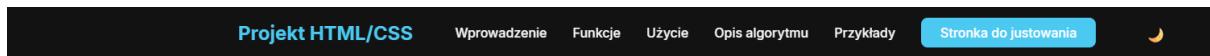
Struktura CSS:

- Zmienne CSS (Custom Properties):** Zdefiniowano zestaw zmiennych dla kolorów, rozmiarów, cieni i przejść, co ułatwia zarządzanie motywami (jasny/ciemny) i zapewnia spójność wizualną.

- **System siatki (Grid):** Wykorzystano CSS Grid do tworzenia zaawansowanych układów, szczególnie w sekcjach hero i galerii kart.
- **Flexbox:** Zastosowano do prostszych konfiguracji, takich jak menu nawigacyjne i przyciski.
- **Media Queries:** Wdrożono punkty przerwania dla urządzeń mobilnych (max-width: 768px) i tabletów (max-width: 1024px)

Menu

Menu nawigacyjne zostało zaprojektowane z wykorzystaniem Flexboxa, aby zapewnić responsywność na urządzeniach desktopowych i przenośnych. Na komputerach stacjonarnych prezentuje się jako poziomy pasek, a na urządzeniach mobilnych przekształca się w hamburger menu.



Rysunek 9. Widok menu gdy szerokość ekranu większa niż 770px



Rysunek 10. Widok menu gdy szerokość ekranu mniejsza niż 769px

```
<nav class="header-nav">
  <div class="company-logo">
    <a href="#" class="logo">Projekt HTML/CSS</a>
  </div>
  <nav class="navbar" aria-label="Główne menu">
    <a href="#" class="logo">Projekt HTML/CSS</a>
    <button class="menu-toggle" aria-label="Otwórz/zamknij menu">
      <span class="bar"></span>
      <span class="bar"></span>
      <span class="bar"></span>
    </button>
    <ul class="nav-menu">
      <li><a href="#introduction" class="nav-link">Wprowadzenie</a></li>
      <!-- pozostałe linki -->
    </ul>
    <button class="theme-toggle">🌙</button>
  </nav>
</nav>
```

- **.navbar** – kontener głównej nawigacji
- **.menu-toggle** – przycisk hamburger menu (widoczny tylko na urządzeniach mobilnych)
- **.nav-menu** – lista linków nawigacyjnych

- **.theme-toggle** – przycisk przełączania motywu