JOURNAL
COMPUTER SCIENCES INSTITUTE

JCSI 23 (2022) 77–83
Received: 24 November 2021
Accepted: 2 December 2021

Web application performance analysis using Angular, React and Vue frameworks

Analiza obciążeniowa aplikacji internetowych z użyciem szkieletów Angular, React i Vue

Konrad Bielak*, Bartłomiej Borek*, Małgorzata Plechawska-Wójcik

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract
The aim of the work is to conduct a performance analysis and, on its basis, to indicate the most user-friendly and fastest operating framework. Three Internet applications have been written to test the speed of operation of selected technologies. Some of the most popular frameworks were analyzed: Angular, React and Vue. Static comparative criteria used in the work are: favorable and generally available documentation, application development, speed of application development, development and support by creators. The practical criteria are the times measured during the execution of the tests. The tests were performed with simple operations using the CRUD (Create Read Update Delete) function. The performance analysis carried out in this way shows the differences between the frameworks. The following developer-Persian tools were used for the comparative analysis: Google Analytics, Google Chrome, Mozilla Firefox, Chrome DevTools.

Keywords: Angular; React; Vue; Performance; JavaScript

Streszczenie
Celem pracy jest przeprowadzenie analizy wydajnościowej oraz wskazanie na jej podstawie najbardziej przyjaznego dla użytkownika i najszybciej działającego szkieletu. Do sprawdzenia szybkości działania wybranych technologii napisane zostały trzy aplikacje internetowe. Analizie poddano jedne z najpopularniejszych szkieletów, mianowicie: Angular, React i Vue. Statyczne kryteria porównawcze wykorzystane w pracy to: rozbudowana i ogólnodostępna dokumentacja, proces wytwarzania aplikacji, szybkość budowy aplikacji, rozwój oraz wsparcie przez twórców. Kryteriami praktycznymi są czasy mierzone podczas wykonywania testów. Testy wykonane zostały za pomocą prostych operacji korzystających z funkcji CRUD (ang. Create, Read, Update, Delete). Dokonano zestawienia wyników szybkości, w jakiej zrealizowane zostały poszczególne operacje oraz opis analizy technicznej danych szkieletów. Do analizy porównawczej wykorzystano narzędzia deweloperskie takie jak: Google Analytics, Google Chrome, Mozilla Firefox, Chrome DevTools.

Słowa kluczowe: Angular; React; Vue; analiza obciążeniowa; JavaScript

*Corresponding author
Email address: konrad.bielak@pollub.edu.pl (K. Bielak), bartlomiej.borek@pollub.edu.pl (B. Borek)

1. Wstęp

JavaScript to język skryptowy opracowany w latach 90-tych. Pierwotnie opracowany przez NetScape dla przeglądarki NetScape Navigator 2.0, aby uczynić strony internetowe bardziej przykuwającymi uwagę użytkowników. Przez lata JavaScript bardzo ewoluował, od języka skryptów witryny, do kodu po stronie serwera, aż do środowiska uruchomieniowego.

Zgodnie z rankingiem hackr.io, Angular, React i Vue to najpopularniejsze szkielety JavaScript [1]. React oraz Vue tworzą interfejsy graficzne za pomocą SPA (Single Page Application). Komponenty wielokrotnego użytku opisywane za pomocą JSX są wykorzystywane w interfejsie Reacta. Jest to zestawienie kodu JavaScript oraz HTML. Natomiast w Vue interfejs zbudowany jest z komponentów, które wykorzystują w swojej implementacji szablony HTML. Angular to następca szkieletu Google Angular JS. Jest on rozwijany przez ten sam zespół. Szkielet Angular narzuca podział poszczególnych komponentów na trzy elementy (logika, interfejs użytkownika i style), podczas gdy Vue zachowuje wszystkie komponenty w jednym pliku.

Większość nowoczesnych witryn internetowych używa JavaScript w pewnym kształcie lub formie, aby zwiększyć szybkość działania i obsługę stron interaktywnych. Tradycyjne strony internetowe to aplikacje wielostronicowe, gdzie nowy dokument HTML jest ładowany za każdym razem, gdy użytkownik zmienia stronę lub jej zawartość. Jest to stosunkowo powolna opcja w porównaniu z najnowszą wersją modelu rozwoju aplikacji jednostronicowej (ang. Single Page Application, SPA). W przypadku tego podejścia jedynie używane elementy, które uległy zmianie są pobierane z serwera i zaktualizowane.

Celem pracy jest przeprowadzenie analizy wydajności trzech wybranych szkieletów oraz wskazanie na jej podstawie szkieletu najbardziej przyjaznego i najszybszego dla użytkownika. Na potrzeby analizy zostały napisane trzy aplikacje, zaimplementowane w różnych szkieletach: Angular, React i Vue. Wszystkie programy

posiadają te same funkcjonalności i napisane w taki sposób, aby zminimalizować ilość kodu do niezbędnego działania. W artykule postawiono tezę badawczą: „Vue jest wydajniejszym szkieletem w mniejszych aplikacjach niż React i Angular". Niniejsze badanie ma na celu ukazanie ich zalet oraz wad, umożliwiając czytelnikowi podjęcie świadomej decyzji przy wyborze między nimi.

2. Przegląd literatury

W pracy przeanalizowano publikacje pod kątem wydajności tytułowych szkieletów. Okazało się to pomocne w doborze odpowiednich metod badawczych.

Pierwszy analizowany artykuł pod tytułem "Porównanie wydajności narzędzi do tworzenia interfejsu aplikacji typu SPA na przykładzie React i Vue.js" [2] przedstawia porównanie dwóch aplikacji posiadających ten sam interfejs użytkownika i oferujących taką samą funkcjonalność. Wykorzystano do tego wymienione w tytule narzędzia JavaScript. Zaimplementowano tabelę z możliwością zmiany liczby wyświetlanych wierszy. Dane zostały wygenerowane za pomocą biblioteki faker.js w postaci pliku JSON. Następnie aplikacje zostały uruchomione w dwóch przeglądarkach internetowych. Każda z nich posiadała własny silnik renderowania, co miało wpływ na różnice w wydajności. Zaimplementowane testy umożliwiały obliczenie czasu pomiędzy ich wywołaniami. Do argumentu funkcji wprowadza się identyfikator, służący do wyznaczenia bloku kodu jaki będzie poddany pomiarom. Rezultatem badania są różnice w wydajności wybranych narzędzi. Ostateczny wynik analizy wskazał, że React przewyższa nieznacznie wydajnością Vue.

W publikacji "Porównanie narzędzie do tworzenia aplikacji typu SPA na przykładzie Angular2 i React" [3] zostały porównane dwa najczęściej stosowane narzędzia do tworzenia aplikacji internetowych typu SPA. W badaniu wykorzystano aplikacje testowe, które posiadały takie same funkcjonalności. Skorzystano z narzędzi Google Chrome DevTools aby testować i porównać możliwości wybranych szkieletów. Pomaga to przeglądać i debugować pliki HTML, CSS oraz witryny JavaScript i sprawdza ruch sieciowy pobierany przez witrynę, sprawdza szybkość witryny i analizę wydajności środowiska wykonawczego. Szkielety porównywano pod kątek struktury aplikacji, dokumentacji, wsparcia społecznościowego oraz testów wydajnościowych. W rezultacie autor podał Angular2 jako lepsze narzędzie dla doświadczonego programisty, a React bardziej przystosowany do potrzeb początkującego użytkownika.

Kolejną analizowaną publikacją jest "DOM benchmark comparison of the front-end JavaScript frameworks React, Angular, Vue.js, and Svelte" [4]. Przy wykorzystaniu różnych wzorców DOM wykonano podstawową ocenę każdego szkieletu. Aplikacje testowe zostały skonfigurowane z minimalną liczbą wymaganych elementów zgodnie z odpowiednią dokumentacją każdego szkieletu. Wszystkie cztery aplikacje zawierają tę samą funkcjonalność i podobne pliki: główny plik HTML, zatytułowany index.html i folder src, w którym zawarto odpowiedni kod testu. Test DOM jest wykonywany przy użyciu bezpośrednich aktualizacji DOM i wstawiania różnych elementów HTML, takich jak elementy i <p>. W aplikacji porównawczej każdy rodzaj trwałego lub złożonego przechowywania danych poza stanem lokalnym wykluczono, ponieważ rzutowałoby to na wyniki badania. We wnioskach wskazano React jako najbardziej wydajny szkielet spośród badanych.

W publikacji „Supporting Web Development Decisions by Comparing Three Major JavaScript Framework: Angular, React and Vue.js" [5] szkielety zostały poddane analizie według określonych kryteriów dobranych na potrzeby firm oraz osób, które chcą poznać daną technologię. Wzięto pod uwagę m.in. wydajność każdego szkieletu, szybkość pisania kodu oraz jakość dokumentacji. W rezultacie autor określił, kiedy stosować badane szkielety które najlepiej spełnią postawione w aplikacji wymagania.

W publikacji „Transitioning Angular 2 User Interface (UI) into React" [6] został opracowany interfejs użytkownika w Angularze 2. Następnie został on przeniesiony do React, co w rezultacie pokazało różnice między nimi. Badanie wykazało, że React dominuje w aspektach szablonowości, szybkości w przyswajaniu wiedzy i zarządzaniu pakietami.

W artykule „Analiza porównawcza wydajności frameworków Angular oraz oraz Vue.js" [7] zestawiono dwa tytułowe szkielety i porównano ich wydajność w kontekście tworzenia gier. Wzięto pod uwagę kryteria takie jak: czas wymiany danych z serwerem, renderowanie różnych komponentów aplikacji i ilość zajmowanej pamięci. W badaniu najwydajniejszy okazał się szkielet Vue.

W książce „Mastering Web Application Development with AngularJS" [8] pokazano jak AngularJS wpływa na tworzenie aplikacji a także jak wspomaga tworzenie czystego i łatwego kodu. Pozycja opisuje także jak wykorzystać szablony oparte na szkielecie Angular i czym różni się on od innych szkieletów.

Do napisania aplikacji wykorzystanych w niniejszym badaniu niezbędne były: dokumentacja Vue [9], dokumentacja React [10] oraz dokumentacja Angular [11]. Informacje zawarte w nich wyjaśniły ważne aspekty każdej badanej technologii. Pomogły odpowiednio wykorzystać badane szkielety w stworzeniu aplikacji testowej.

Artykuły „Biblioteki AngularJS i ReactJS – analiza wydajnościowa" [12] oraz „Analiza porównawcza narzędzi do budowania aplikacji SPA AngularJS, ReactJS, Ember.js" [13] zapewniły autorom wstępną wiedzę niezbędną do przeprowadzeniu badań. W pierwszym artykule analizowano szkielety pod względem trudności nauki, dodatkowych narzędzi charakterystycznych dla danego szkieletu, wydajności na urządzeniach stacjonarnych oraz mobilnych. W wyniku analizy została wybrana ta biblioteka, która okazała się najlepsza w kontekście wybranych kryteriów. W drugim artykule badano wydajność AngularJS i ReactJS na różnych

przeglądarkach. Wpłynęło to na wybór Google Chrome do przeprowadzenia badania. Autor pierwszego artykułu podał w jakich sytuacjach lepiej sprawdzą się badane szkielety. Natomiast wynikiem drugiego artykułu było wybranie Reacta jako najbardziej wydajnego oraz posiadającego najwięcej dodatkowych narzędzi. Autorzy zaznaczyli jednak, że warto wziąć pod uwagę fakt, iż jest on trudniejszy dla programistów rozpoczynających naukę.

3. Opis i uzasadnienie badanej technologii

Wybrane technologie są obecnie najbardziej funkcjonalnymi narzędziami do tworzenia aplikacji internetowych. Wszystkie oparte są na JavaScript. Budowa aplikacji na poszczególnych szkieletach, mimo podobieństw, wygląda jednak inaczej.

3.1 Angular

Angular jest szkieletem, który istnieje w dwóch wersjach, powszechnie określanych jako AngularJS i Angular 2+. AngularJS jest starszą wersją opartą na JavaScript która nie jest już rozwijana [10]. Natomiast Angular 2+ jest nowszy i oparty o TypeScript oraz HTML. TypeScript to język programowania typu open source, który został opracowany przez firmę Microsoft. Wykorzystanie go pozwala na statycznie pisanie i kompilowanie do JavaScript. Szkielet Angular 2+ służy do budowania aplikacji klienckich w HTML, CSS i JavaScript lub Typescript. Zapewnia aplikacjom czystą i luźno powiązaną strukturę, która jest łatwa do zrozumienia i prosta w utrzymaniu. Dzięki tej strukturze można między innymi wykorzystywać kod ponownie w różnych aplikacjach.

Aplikacje zbudowane przy użyciu szkieletu Angular składają się z modułów NgModule. Moduły te są kontenerami dla bloku kodu dedykowanego domenie aplikacji. Prosty przepływ pracy i ściśle powiązane zestawy możliwości generują łatwy i zrozumiały kod programistyczny. NgModules zapewniają kontekst kompilacji dla swojej zawartości. Zawierają składniki dostawców usług i inne pliki, co zwiększa jednak redundancję kodu. Funkcjonalności aplikacji bez problemu są eksportowane i importowane do innych modułów.

Jak podaje dokumentacja [7], każda aplikacja Angular ma moduł główny, według konwencji nazwany AppModule. Zapewnia on mechanizm ładowania początkowego, który uruchamia aplikację. Moduł główny może zawierać nieograniczoną hierarchę modułów podrzędnych. Szkielet ten częściowo oparty jest na wzorcu MVC (Model View Controller), mając komponent jako kontroler i szablon jako widok. Komponent kontroluje widok, który reprezentuje obszar ekranu jak również hierarchię widoków złożoną z poszczególnych modułów.

Aby zapewnić jak największą szybkość oraz wydajność aplikacji internetowej, Angular daje kontrolę nad skalowalnością. Pozwala to na spełnienie wymagań dotyczących danych, budując modele danych w RxJS, Imutable.js albo innym modelu push.

3.2 React

React jest biblioteką początkowo rozwijaną przez firmę Facebook jako port JavaScript XHP biblioteki PHP. XHP jest modyfikacją PHP, która pozwalała na tworzenie własnych komponentów. Dokumentacja React [9] podaje, że pierwsza wersja została wydana w maju 2013 roku. Zakres jest mniejszy niż w innych badanych technologiach, ponieważ renderowany jest tylko interfejs użytkownika aplikacji. Korzyścią używania tej technologii jest lekka konstrukcja, mniej kosztowna w nauce oraz użytkowaniu. Rozwój ten może być postrzegany jako bardzo ważny krok w ogólnej zmianie tworzenia aplikacji internetowych.

Największym sukcesem biblioteki React jest to, że zoptymalizowane są funkcjonalności modelu DOM. Model ten jest sposobem reprezentacji dokumentów XML i HTML w postaci modelu obiektowego. Manipulacja DOM jest stosunkowo kosztowna pod względem wykorzystywanych zasobów obliczeniowych. React został zaprojektowany, aby wykonywać jak najmniej manipulacji przy użyciu tego modelu. Zmniejszenie wymagań zużycia w tym szkielecie spowodowane jest zarządzaniem wirtualnym DOM. Podstawy JavaScript zawarte w tym szkielecie przyspieszają cały mechanizm działania aplikacji.

3.3 Vue

Vue można uznać za najnowszy szkielet JavaScript spośród wszystkich analizowanych w tym badaniu. Został stworzony przez firmę Google. W porównaniu do szkieletu Angular, jest bardziej uproszczony i skupiony tylko na widoku. Vue obsługuje DOM podobnie jak React, budując i używając oddzielnego wirtualnego DOM do zarządzania rzeczywistym DOM. Jedną z różnic jest to, że omawiany szkielet nie używa w tym celu oddzielnej modułowej biblioteki. Vue jest często opisywane jako postępowy szkielet, który jest używany do tworzenia interfejsu użytkownika na stronie internetowej. Nie jest ściśle związany z wzorcem MVVM (Model View ViewModel).

Vue może być wykorzystywany do małych projektów, gdzie podstawowa biblioteka używana jest w połączeniu z innymi technologiami jak i dla pełnowymiarowych SPA, zatem główną zaletą Vue jest skalowalność. Nie jest rozwijany przez duże przedsiębiorstwo, lecz przez aktywnych programistów. Kompilacja środowiska wykonawczego Vue jest odpowiedzialna za tworzenie instancji wraz z renderowaniem i zarządzaniem wirtualnym DOM. Istnieje również alternatywna pełna wersja, która zawiera środowisko wykonawcze i kompilator.

Informacje zawarte w dokumentacji [8] podają, że zależności pojedynczych komponentów we Vue są automatycznie śledzone podczas renderowania. System jest w stanie zweryfikować, które komponenty muszą się ponownie renderować, podczas zmiany stanu. To optymalizuje czas i pracę programisty, dzięki czemu może on skupić się na istotnych funkcjonalnościach w tworzeniu aplikacji. Podobnie jak React, jest to szkielet optymalny i szybki w swoim działaniu.

4. Metoda badań

Analiza porównawcza została przeprowadzona z wykorzystaniem kryteriów statycznych i praktycznych. Analiza porównawcza polegała na zestawieniu wyników uzyskanych przez poszczególne szkielety dla określonych kryteriów badawczych. Aby zapewnić wiarygodność wyników wszystkie pomiary zostały powtórzone 100 razy. Parametry sprzętu, na którym utworzono środowisko testowe przedstawiono w Tabeli 1.

Tabela 1: Parametry sprzętu

Parametr	**Wartość**
Procesor	Intel Core i3-1005G1
Ram	16GB
Dysk	256 GB SSD
Karta graficzna	Intel UHD Graphics
System operacyjny	Windows 10 Home

Statyczne kryteria porównawcze posłużyły do technicznej analizy wybranych technologii. Do kryteriów tych należą:

- wytwarzanie aplikacji testowej,
- szybkość budowy aplikacji lokalnie,
- szybkość uruchomienia aplikacji lokalnie,
- rozwój danego szkieletu.

Testy praktyczne wykonano z zastosowaniem aplikacji testowych uruchomionych lokalnie. Testy wykonano dla siedmiu przypadków:

- wczytywanie 1 000 wierszy,
- odświeżanie 1 000 wierszy,
- wybór pojedynczego wiersza,
- usunięcie 50 wierszy,
- edycja 3 losowych wierszy,
- stworzenie 10 000 wierszy,
- usunięcie 1 000 wierszy.

Dodatkowym kryterium była ilość kodu programistycznego niezbędnego do rozwinięcia aplikacji.

Wszystkie testy rozpoczynają się z 1 000 załadowanych na stronie wierszy, które ładowane są przed każdym pomiarem testowym. Kryteria porównawcze, za każdym razem są odświeżane i wykonywane z tego samego punktu. Dane w aplikacjach uzupełniane są automatycznie, a więc zniwelowany do zera jest błąd ludzki.

Celem aplikacji jest sprawdzenie szybkości działania wybranych szkieletów. Każda z aplikacji została napisana w możliwie najprostszy sposób, jaki został opisany w dokumentacji. Można zatem przybliżyć szybkość działania wszystkich testowych aplikacji. Aplikacje włączane były lokalnie na uprzednio zrestartowanych maszynach, które nie miały połączenia z Internetem. Wszystkie dane wynikowe zostały zapisane do osobnych plików a następnie zaimportowane na potrzeby analizy do formatu .xlsx.

Do przeprowadzenia badań użyto zestawu narzędzi Chrome DevTools. Jest to zestaw narzędzi wbudowanych bezpośrednio w przeglądarkę Google Chrome. Narzędzia te umożliwiają wykonanie pomiarów podczas wczytywania strony internetowej. Google Analytics użyto w celu monitorowania wydajności aplikacji.

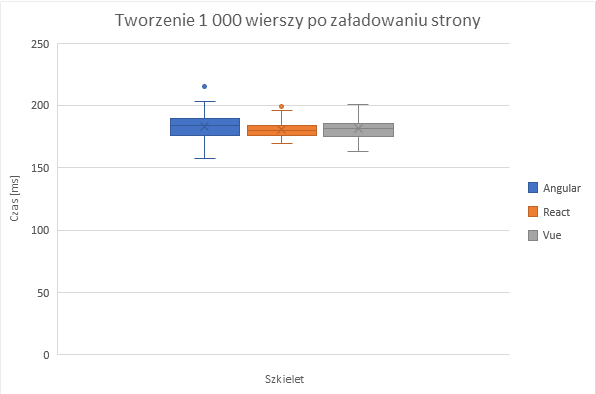
5. Statyczna analiza wyników

Oceną wydajności technologii poza wykonywaniem testów jest opis wsparcia technicznego, które dostarcza zespół twórców danego szkieletu. Najszybciej budującym i uruchamiającym się szkieletem jest Angular. Poprzez najszybsze przetwarzanie procesów związanych ze środowiskiem uruchomieniowym najlepiej radzi również sobie z przeglądarkami internetowymi. Następnym szkieletem jest Vue. Wykorzystywany jest on do tworzenia małych i średnich aplikacji, dlatego korzystanie z niego jest wygodne. Budowa i uruchomienie aplikacji w tym szkielecie jest bardzo zbliżone do Reacta, dlatego nie można jednoznacznie stwierdzić który z tych szkieletów jest szybciej działającym.

Angular jest szkieletem, w którym często tworzone są aktualizacje. Wspierany jest on przez Google, a cykl wydawania nowych wersji trwa 6 miesięcy. Twórcy przykładają bardzo dużą uwagę do aktualizacji tego szkieletu. React rozwija się poprzez komunikację ze społecznością programistów. Twórcy jak i zewnętrzni współtwórcy przesyłają swoje zmiany poprzez platformę GitHub, następnie wszystkie informacje przechodzą przez proces recenzji. Każdy programista może wprowadzić zmianę lub informację o błędzie, które zostaną zaakceptowane przez twórców. Zespół bardzo szybko poprawia krytyczne błędy znalezione w starszych aktualizacjach. Najmłodszym szkieletem spośród wybranych jest Vue. Autorski projekt byłego twórcy Angulara nie posiada dużej społeczności profesjonalistów, którzy wspierają tą technologie. Licencja open source w tym szkielecie ciągle powiększa swoją społeczność.

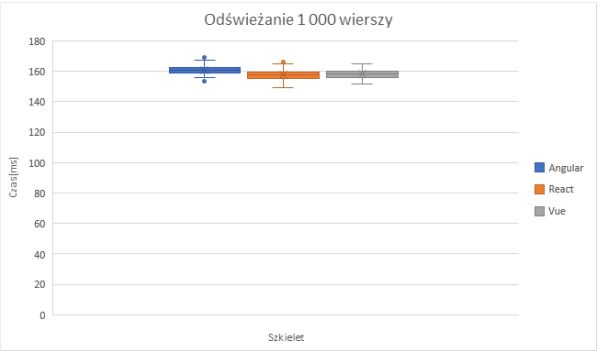
6. Praktyczna analiza wyników

Wyniki ze wszystkich testów zostały przedstawione na Rysunkach 1-6. Wyniki otrzymane zostały podzielone na 7 kategorii bazując na operacjach CRUD. Do każdego pomiaru stworzone zostały wykresy „skrzynka i wąsy”. Ten typ wykresu zawiera informacje odnośnie do położenia, rozproszenia i kształtu rozkładu danych. Dolna linia łącząca się z prostokątem oznacza minimalną wartość w zbiorze danych, a górna wyznacza maksymalną. Dolny bok oznacza pierwszy kwartyl, czyli środkową wartość między minimum, a medianą, a górny bok oznacza trzeci kwartyl, czyli wartość środkową między medianą, a maksymalną. Linia w środku prostokąta to mediana, a punkt „X” wyznacza średnią zbioru wartości. Pojedyncze punkty wyznaczają wartości odstające.



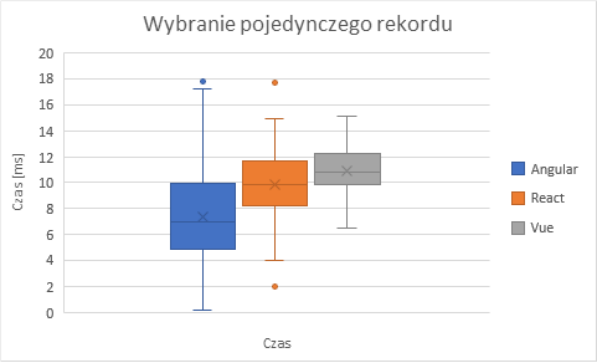
Rysunek 1: Porównanie czasów dla renderowania wierszy.

Czas tworzenia wierszy podczas ładowania strony jest przedstawiony na Rysunku 1. Najszybciej z tym zadaniem radzi sobie React. Stosowane w tej technologii mikroserwisy, najkrócej komunikują się z interfejsem użytkownika.



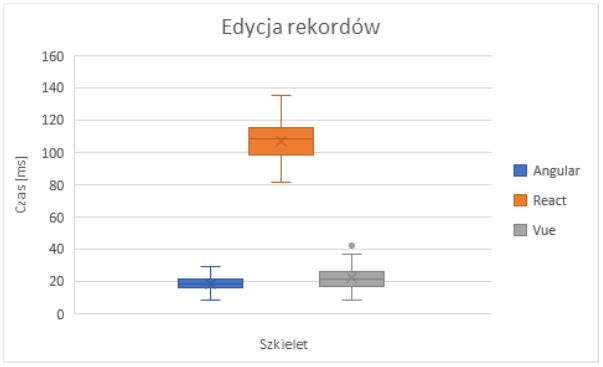
Rysunek 2: Porównanie czasów odświeżania wierszy.

Test odświeżania strony wypadł podobnie dla wszystkich trzech szkieletów. Technologie posiadają niemal identyczne przetwarzanie procesów, dotyczących podglądu załadowanych informacji.



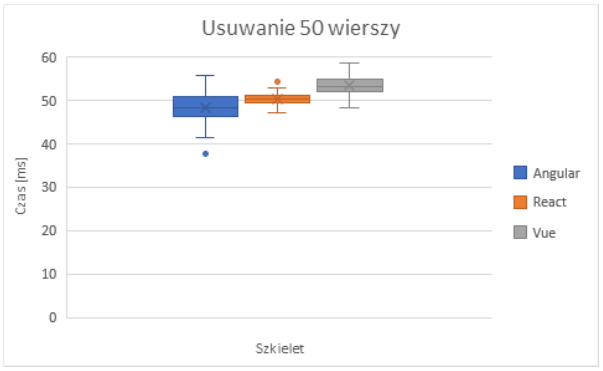
Rysunek 3: Porównanie czasów wyboru pojedynczego wiersza.

Angular najszybciej wykonał test mający na celu wybór wiersza poprzez jego zaznaczenie. Średni wynik tego szkieletu jest aż 30% krótszy od dwóch pozostałych. Komponenty w tej technologii bardzo dobrze się ze sobą komunikują, dzięki temu zaznaczenie danych w bardzo szybkim tempie dociera do interfejsu użytkownika.



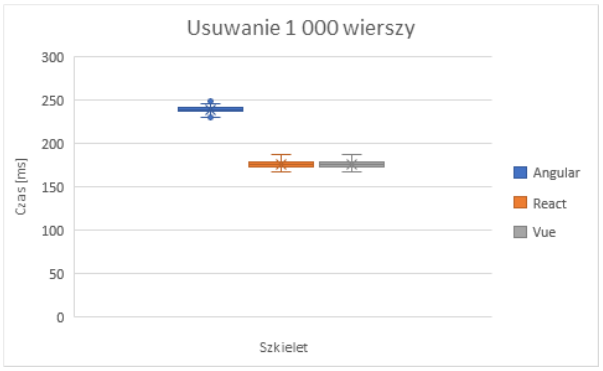
Rysunek 4: Porównanie czasów edycji losowego wiersza.

Edycja poszczególnych losowych wierszy jest testem, w którym React okazał się najsłabszy. Dwa pozostałe szkielety mają wynik na podobnym poziomie. Czas wykonania tego testu przez Reacta jest około pięciokrotnie większy niż Angulara i Vue. Zewnętrzne funkcje służące do edycji danych wykonują się zbyt długo, dlatego React nie jest wydajny podczas zmiany danych.



Rysunek 5: Porównanie czasów usunięcia 50 wierszy.

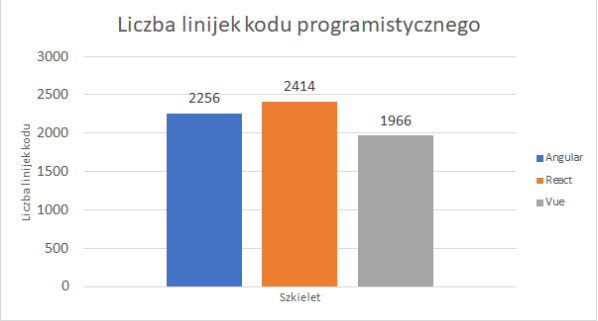
Usuwanie wierszy było przeprowadzone dla 50 i 1000 rekordów. Angular bardzo szybko usuwa małe ilości danych. Częste oczyszczanie danych z małych pakietów jest najlepszym rozwiązaniem w tej technologii. React i Vue są szkieletami, które szybciej usuwają większe pakiety danych. W tych technologiach czas usuwania danych ma rozkład logarytmiczny: im więcej danych jest do usunięcia, tym mniejszy jest czas usuwania jednej jednostki danych.



Rysunek 6: Porównanie czasów usunięcia 1 000 wierszy.

Rysunek 7: Porównanie czasów tworzenia 10 000 wierszy.

Ostatnim wykonywanym testem było utworzenie 10 000 nowych rekordów. W tym zadaniu Vue okazał się najbardziej wydajny, prawdopodobnie dzięki sposobie działania procesów. React jest szkieletem najwolniej uzupełniającym dane, ze względu na korzystanie z zewnętrznej funkcji. Angular zbliżony jest czasem do Vue, dobra komunikacja komponentów sprawia, iż ta technologia radzi sobie z dodawaniem danych.



Rysunek 8: Liczba linijek kodu programistycznego.

Ostatni wykres (Rysunek 8) przedstawia ilość kodu programistycznego niezbędnego do utworzenia aplikacji testowych. Używanie szkieletów JavaScript wpływa na lepszą organizację kodu, co ułatwia programiście zarządzanie aplikacją. Aplikacja wykorzystująca szkielet Vue zawiera najmniejszą liczbę linii kodu programu. Mimo to, różnice między konkurencyjnymi technologiami są niewielkie.

7. Wnioski

Teza badania została potwierdzona – Vue jest wydajniejszym szkieletem niż React i Angular. Analiza wyników wykazała, że najszybszym szkieletem jest Vue. Szybkość działania tej technologii opiera się na małych mikroserwisach, które bezproblemowo komunikują się między sobą. Najlepiej radzi on sobie z dużą ilością danych. Operacje usuwania i edycji danych również wykonują się tu szybko.

Test związany z edycją danych wykazał, że Angular jest drugim najszybciej działającym szkieletem. Edycja i usuwanie małej liczby pakietów danych bardzo dobrze przebiega w tej technologii. Badania pokazują, że dzięki szybkości usuwania małej ilości danych, ciągle można utrzymać aplikacje w czystości i pełnej sprawności.

Najmniej wydajnym szkieletem jest React. Uzyskany czas edycji danych wpływa na to, iż jest on najwolniejszą technologią. Renderowanie strony i usuwanie dużej ilości danych pokazuje jednak, że wewnętrzne funkcje tego szkieletu działają bardzo szybko. Jednak korzystanie z zewnętrznych funkcjonalności spowalnia działanie całej aplikacji.

Podczas przeprowadzonej analizy wykonanej na niewielkich aplikacjach, Vue okazał się być najbardziej wydajnym szkieletem. Jednak nie można uogólnić tego stwierdzenia bez dodatkowych badań. Każdy program stawia inne wymagania, przez co badanie innej aplikacji może przynieść inny rezultat. Aby wyniki badań były bardziej precyzyjne, warto powtórzyć je na większej liczbie próbek. Ponadto warto zaznaczyć, iż do badań została użyta aplikacja wykorzystująca podstawowe funkcjonalności, dlatego rozszerzenie jej o nowe możliwości dałoby więcej wniosków w badaniu.

Podsumowując, wszystkie badane szkielety mają wysoką wydajność. Mimo że Vue osiągnął najlepsze wyniki w badaniu, warto też wziąć pod uwagę inne biblioteki. Warto też być na bieżąco z nowymi publikacjami jakie powstają i śledzić dokumentacje.

Literatura

[1] S. K. Arora, 10 Best JavaScript Frameworks to Use in 2021, https://hackr.io/blog/best-javascript-frameworks, [23.01.2021].

[2] K. Boczkowski, B. Pańczyk, Comparison of the performance of tools for creating a SPA application interface - React and Vue.js, Journal of Computer Science Institute 14 (2020), 73-77, https://doi.org/10.35784/jcsi.1579.

[3] J. Kalinowska, B. Pańczyk, Comparison of tools for creating SPA applications using the examples of Angular 2 and React, Journal of Computer Sciences Institute 10 (2019) 1-4, https://doi.org/10.35784/jcsi.183.

[4] M. Levlin, DOM benchmark comparison of the front-end JavaScript frameworks React, Angular, Vue, and Svelte, Åbo Akademi, 2020.

[5] E. Wohlgethan, Supporting Web Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue. js. Diss. Hochschule für Angewandte Wissenschaften Hamburg, 2018.

[6] K. Simkhada, Transitioning Angular 2 User Interface (UI) into React, Helsinki Metropolia University of Applied Sciences, [12.04.2017].

[7] R. Baida, M. Andriienko M. Plechawska-Wójcik, Perfomance analysis of frameworks Angular and Vue.js, Journal of Computer Sciences Institute 14 (2020) 59-64, https://doi.org/10.35784/jcsi.1577.

[8] P. Kozlowki, Mastering Web Application Development with AngularJS, [23.08.2013].

[9] Dokumentacja Vue.js https://vuejs.org/v2/guide/, [30.11.2020].

[10] Dokumentacja React https://pl.reactjs.org/docs/getting-started.html, [22.03.2021].

[11] Dokumentacja Angular https://angular.io/docs, [30.11.2020].

[12] R. Nowacki, M. Plechawska-Wójcik, Comparative analysis of tools dedicated to building Single Page Applications – Angular Js, ReactJS, Ember.js, Journal of Computer Sciences Institute 2 (2016) 98-103, https://doi.org/10.35784/jcsi.122.

[13] K. Kowalczyk, M. Plechawska-Wójcik, Angular JS and ReactJS libraries – performance analysis, Journal of Computer Sciences Institute 2 (2016) 114-119, https://doi.org/10.35784/jcsi.126.