

„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

Politechnika Wrocławska
Wydział Informatyki i Zarządzania



Politechnika
Wrocławska

Zaawansowane Technologie Webowe

Laboratorium

Temat: Projekt grupowy – wybór frontend
Opracował: mgr inż. Piotr Jóźwiak
Data: lipiec 2020
Liczba godzin: 2 godziny

Table of Contents

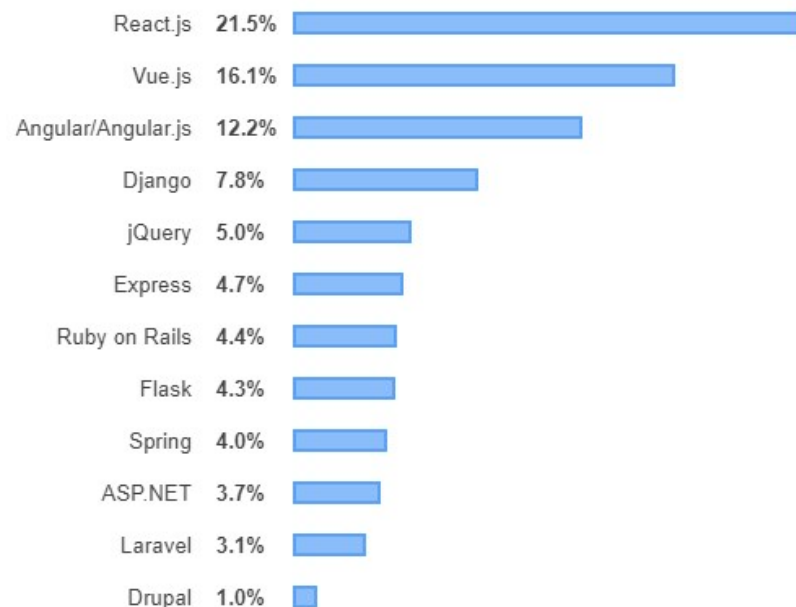
Wstęp	3
Cel laboratorium	3
Przegląd wybranych technologii frontendowych	4
React	4
Vue	5
Angular	6
Flutter	7
Podsumowanie	8

Wstęp

Framework frontendowy to przede wszystkim grupa narzędzi i bibliotek, które pozwalają na ułatwienie i przyspieszenie procesu tworzenia projektu. Korzystając z czystego kodu bez wsparcia jakiegokolwiek frameworku na pewno napotkamy wiele problemów związanych z poprawną modularyzacją, złożonością kodu czy też standaryzacją sposobu implementacji wewnątrz dużych zespołów programistycznych. Wszystko to pochłonie wiele czasu programistów jak i testerów.

Opisane problemy można wyeliminować za pomocą odpowiedniego frameworku frontendowego. Framework ma za zadanie wprowadzić zespół formalizmów dla programistów, które pozwolą uporządkować i ułatwić utrzymanie kodu. Ulepszą organizację kodu, jego używalność oraz pomogą wdrożyć dobre wzorce projektowe.

Jak wynika z ankiety StackOverflow z roku 2019 największą popularnością cieszą się takie frameworki jak React, Vue oraz Angular. Poniższy wykres przedstawia ranking technologii frontendowych z roku 2019:



% of developers who are not developing with the language or technology but have expressed interest in developing with it

Jak widać z powyższego rankingu najpopularniejsze są frameworki w technologiach JavaScript.

Cel laboratorium

Celem tego laboratorium jest przyjrzenie się dostępnym technologiom front-end'owym oraz, na podstawie zdobytej wiedzy, podjęcie decyzji o wybraniu najlepiej dopasowanej technologii do implementacji projektu grupowego. Aby wspomóc decyzję pokrótce scharakteryzujemy najpopularniejsze (choć jedynie wybrane spośród wielu) technologie wykorzystywane do implementacji warstwy prezentacji strony internetowej.

Przegląd wybranych technologii frontendowych

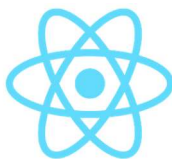
Dokonując wyboru, należy pamiętać o kilku wymaganiach stawianych przed UI frameworkami. Wybierając jedną z najlepszych platform frontendowych, musimy wziąć pod uwagę perspektywę zarówno użytkowników, jak i zespołów programistycznych. Funkcje te obejmują etapy projektowania, rozwoju, testowania i konserwacji. Najlepsze frameworki front-end jednocześnie upraszczają i przyspieszają proces tworzenia aplikacji. Jednocześnie framework musi zapewnić elastyczność dla dalszych prac nad projektem, a klient musi otrzymać doskonały UX oraz UI. Jeśli chodzi o zespół programistów, musimy szukać frameworka, z którym wygodnie się pracuje. Należy także wziąć pod uwagę łatwość integracji z backendem. Najlepsze frameworki front-end są zawsze łatwe w utrzymaniu i testowaniu. Regularne aktualizacje są niezbędne ze względu na szybko zmieniające się wymagania rynku, nowe zagrożenia bezpieczeństwa i wersje technologii.

Większość frameworków do tworzenia stron internetowych oferuje narzędzia programistyczne i obsługuje testy jednostkowe oraz zapewnia wsteczną kompatybilność. Stabilność i wsparcie społeczności to również czynniki, o których nie można zapomnieć. Aby osiągnąć pożądane rezultaty, należy znaleźć najlepszą platformę front-endową, która będzie spełniała ogólne i indywidualne wymagania. Omówimy kilka najpopularniejszych frameworków.

Programiści UX najczęściej zwracają uwagę na takie czynniki jak wsparcie w zarządzaniu stanem (sesja), sposób przetwarzania formularzy, walidacji formularzy, szablonowanie strony (templates), komunikacja HTTP oraz wsparcie routingu. Właśnie ze względu na rozbudowane wsparcie w powyższych kategoriach to właśnie React, Angular i Vue są tak popularne wśród deweloperów. Czasami programiści muszą zastosować dodatkowe narzędzia z zewnętrznych bibliotek. Uniknięcie tego dodatkowego zgiełku podczas integracji i dysponowanie wszystkimi potrzebnymi narzędziami w jednym miejscu jest dużą korzyścią dla programistów. Analizując dane ankietowe, możemy uzyskać następujące wyniki dotyczące najczęściej używanych frameworków: React jest stosowany głównie przez programistów Node.js, Python i Rails, Angular jest często używany przez programistów Java, PHP i C #, a Vue.js jest popularny wśród PHP, SCC i Node.js.

Przedstawmy krótką charakterystykę tych frameworków.

React



React

React to framework stworzony i rozwijany przez Facebooka. Facebook zaczął implementację z myślą o uzyskaniu frameworka z wysoką wydajnością. Jordan Walke zaproponował utworzenie biblioteki w oparciu o kombinację XHP i JavaScript w 2011 roku. W rezultacie otrzymał bibliotekę do budowy interfejsu internetowego z JavaScript, w którym aktualizacja wiadomości odbywa się w tym samym czasie, gdy klienci korzystają z czatu. Facebook wypuścił otwarte narzędzie JavaScript w 2013 roku. Kontynuował rozwój frameworka i wydał React Native na Androida i iOS w 2015 roku.

W ciągu dwóch lat Facebook ogłosił nowy podstawowy algorytm - React Fibre. React stał się najlepszym frameworkiem front-end ze względu na swoje charakterystyczne cechy.

Po pierwsze, działa z wirtualnym DOM i stosuje abstrakcyjne kopie prawdziwego DOM. Dlatego aktualizuje wszystkie zmiany użytkownika, ale nie ma wpływu na inne części interfejsu. Po drugie, ten front-end

sieciowy sprawia, że aktualizacje są dość szybkie, ponieważ React łączy DOM z funkcjonalnością interfejsu użytkownika. Ma specjalną możliwość ponownego wykorzystania komponentów kodu na każdym poziomie bez wprowadzania w nich zmian. Po trzecie, stabilny kod jest zapewniany przez jednokierunkowy przepływ danych. Po zastosowaniu jakichkolwiek aktualizacji zmienione zostaną tylko wybrane komponenty. Oprócz Facebooka, takie firmy jak Instagram i Netflix przyjęły React, który zoptymalizował ich działanie w sieci i wysiłki programistyczne znacznie zwiększając wydajność.

React charakteryzuje się następującą liczbą zalet i zaledwie kilkoma wadami:

Zalety:

- Virtual DOM poprawia zarówno wrażenia użytkownika, jak i pracę programisty - Virtual DOM pomaga aktualizować dowolne zmiany użytkownika bez ingerencji w inne części, stosując izolowane komponenty. Takie rozwiązanie pozwala budować bardzo szybkie interfejsy cieszące się dobrym *user experience*.
- Oszczędność czasu podczas ponownego używania komponentów React - React radzi sobie z izolowanymi komponentami, dlatego można je użyć ponownie w dowolnym momencie. Aktualizacje systemu nie wpływają na system ani go nie zmieniają.
- Stabilny kod jest zapewniany przez jednokierunkowy przepływ danych - bezpośrednia praca z każdym komponentem wymaga jednokierunkowego przepływu danych. Inną rzeczą jest to, że w tej strukturze JavaScript możliwe jest tylko wiązanie danych w dół.
- Biblioteka typu open source z różnymi narzędziami - wszystkie aktualizacje są udostępniane społeczności. React dostarcza bibliotekę open source, co pozwala programistom dodawać swoje rozwiązania.

Wady:

- Stosunkowo długa krzywa uczenia się - jest mało prawdopodobne, aby nauczyć się tego w jeden dzień. Po prostu potrzeba trochę więcej czasu, aby być świadomym wszystkich szczegółów, niż na przykład w Angular.
- Brak dokumentacji ze względu na szybkie tempo rozwoju - popularność tego frameworka front-endowego jest znacznie wyższa niż innych. Ma tak wiele aktualizacji i innowacji, że czasami trudno jest znaleźć wszystkie szczegółowe informacje.

Vue



Vue.js to platforma do tworzenia interfejsów użytkownika. Jest to niezależne narzędzie, które tworzy interfejsy internetowe oraz nie wymaga dodatkowych rozszerzeń. Vue.js został stworzony przez Evan You i wydany początkowo w lutym 2014.

Vue zachęca do korzystania z niego dostarczając wiele funkcji. Jest łatwy do uczenia się przez programistów oraz prosty w integracji. Vue.js można zastosować zarówno do implementacji pojedynczych komponentów, jak i do pełnych aplikacji jednostronicowych. Obejmuje układ plików komponentów i strukturę logiczną. Struktura JavaScript obsługuje dwukierunkowe reaktywne wiązanie danych i nie wymaga żadnych dodatkowych bibliotek.

Jednocześnie framework jest niezwykle elastyczny. Vue.js można łatwo zintegrować z różnymi bibliotekami i zastosować do większych projektów. Zespół Vue.js wydał kilka świetnych rozszerzeń do przeglądarek (Devtools) wspomagających debugowanie aplikacji. Pozwala sprawdzić aktualną sytuację komponentów i ręcznie zmienić niektóre z nich. Deweloperów przyciąga możliwość połączenia konstrukcji w wybrany przez siebie sposób. Nie powinniśmy zapominać o rozmiarze Vue.js, który jest naprawdę mały w porównaniu z innymi frameworkami. W rezultacie przyspiesza to ogólny czas ładowania strony. Wreszcie Vue.js jest znany z kompleksowej dokumentacji, dlatego programiści zaoszczędzą czas na uczenie się tego front-endowego frameworka. Vue.js stosuje wzorce szablonów oparte na HTML.

Jeśli chodzi o listę zalet i wad Vue.js, przyjrzyjmy się krótko:

Zalety:

- Przejrzystość i prostota – framework posiada bardzo przejrzystą strukturę oraz posiada niski próg wejścia dla nowych programistów. Dość łatwo jest nie tylko zacząć się uczyć, ale w krótkim czasie zacząć z niego korzystać.
- Szczegółowa i obszerna dokumentacja.
- Prosta integracja i możliwość ponownego wykorzystania kodu - dzięki swojej logicznej strukturze programiści interfejsu użytkownika mają możliwość tworzenia elastycznych komponentów i ich późniejszego ponownego wykorzystania, nawet w innych projektach.
- Rozszerzenia przeglądarki devtools - jak wspomniano powyżej, Vue.js nie wymaga żadnych dodatkowych bibliotek, ale posiada rozszerzenia przeglądarki devtools. Dzięki temu można ręcznie zmieniać poszczególne komponenty oraz łatwo debugować aplikację.

Wady:

- Zbyt duża elastyczność prowadzi do nieprawidłowości w kodzie - niektórzy eksperci zwrócili uwagę, że duża elastyczność nie jest zbyt dobra. Wszyscy inżynierowie front-endu mogą przyczynić się do jego rozwoju, więc może to powodować pewne nieprawidłowości.
- Społeczność mniejszych programistów - jest to stosunkowo nowa i stale ulepszana platforma. Vue.js jest obsługiwany głównie przez indywidualnych programistów, ale ich społeczność szybko się powiększa.

Angular



Angular lub Angular v2 lub Angular 2+ to kompletna przeróbka AngularJS przez ten sam zespół w Google, który opracował oryginalny front-end frameworka w 2010 roku. Został wydany we wrześniu 2016 roku jako zupełnie nowy framework z nową logiką. Użycie tak podobnej nazwy do jego poprzednika wywołało sporo kontrowersji wśród programistów ze względu na brak wstecznej kompatybilności do AngularJS.

Angular jest rzeczywiście kompleksowym rozwiązaniem frameworkowym, a nie tylko zestawem bibliotek. Programiści mogą bardziej skupić się na wykonywaniu swoich zadań niż na szukaniu bibliotek i rozwiązaniach dla swoich problemów. To zdecydowanie sprawia, że Angular jest obecnie jednym z najlepszych frameworków interfejsu użytkownika.

Angular wykorzystuje TypeScript, przez co programiści mają dostęp do szerokiego spektrum funkcjonalności: arrow functions, async/await, class syntax itp. Wprowadzenie dobrze znanych praktyk OOP znacznie ułatwia przejście z języków takich jak C# czy Java.

Angular został stworzony do pracy w dużych zespołach. Wykorzystuje moduły, dzięki czemu każda część zespołu może pracować nad swoją częścią kodu bez obawy o złamanie czegoś w kodzie innej osoby. Zmniejsza to również obciążenie nakładania się prac i kontroli jakości.

Przyjrzyjmy się zaletom i wadom.

Zalety:

- Architektura Angular jest oparta na komponentach co pozwala na tworzenie interfejsu użytkownika, w którym łatwo można reużywać raz napisane części funkcjonalności. Komponentyzacja upraszcza również testowanie i konserwację kodu.
- Wykorzystanie TypeScript w Angular ułatwia proces kodowania.
- Wysoka wydajność.
- Angular Material - zespół Angular nieustannie odświeża swoją strukturę komponentami Material Design.
- Dwukierunkowe bindowanie.

Wady:

- Problemy kompatybilności wstecznej do AngularJS.
- Framework jest całkiem skomplikowany przez co posiada wyższy próg wejścia.
- Trudność z utrzymaniem komponentów poprzez wieloplikową definicję pojedynczego elementu.
- Słaba dokumentacja CLI – często trzeba przeszukiwać fora, aby dotrzeć do szczegółów.

Flutter



Wraz z obecnymi trendami warto zwrócić uwagę na szybko rozwijające się projekty. Jako przeciwwagę do najpopularniejszych frameworków warto także spojrzeć na rozwiązania, które szybko zyskują nowych zwolenników. Flutter okazuje się być jednym z najszybciej rozwijających się frameworków open source. Został pierwotnie wydany w 2015 roku jako framework dla programistów

Androida. Przez kolejne trzy lata zespół Flutter w Google pracował nad obsługą obu platform mobilnych. I wreszcie, w grudniu 2018 roku, wydał pierwszą oficjalną wersję Flutter 1.0 na iOS i Androida. Platforma jest wspierana przez społeczność w zakresie dobrej wydajności oraz dopracowanego interfejsu użytkownika.

Flutter to zestaw narzędzi do tworzenia interfejsu użytkownika kompilowanego z jednego kodu źródłowego zarówno na urządzenia mobilne, strony internetowe oraz komputery stacjonarne. Używa on innego podejścia niż większość tego typu rozwiązań, gdyż zamiast pracy w WebView lub mapowania elementów na natywne odpowiedniki Flutter sam bierze odpowiedzialność za tworzenie całego UI, wykorzystując natywny Canvas. Flutter posiada wiele zalet jak hot-reload, duża płynność interfejsu, łatwość tworzenia layoutów oraz pomocne community.

Flutter dostarcza swój język programowania zwany Dart, który jest wzorowany na popularnych składniach Java oraz C#, przez co próg wejścia jest niski.

Zalety:

- Open source - Otwartość na społeczność zaowocowała wieloma osiągnięciami. Zespół Flutter oferuje świetne wsparcie.
- Jeden kod bazowy - tworzenie jednej aplikacji na różnych platformach zawsze pozostanie odrębną funkcjonalnością. Język programowania Dart oferuje opcje łańcucha narzędzi dla Intel, ARM i JavaScript, które umożliwiają kompilację na prawie każdą platformę.
- Szczegółowa dokumentacja - Flutter pomimo, że jest całkiem młodym frameworkiem dostarcza bardzo dobrą dokumentację wraz z ciekawymi przykładami.
- Dostępność wtyczek - umożliwia programistom uzyskanie dostępu do innych natywnych bibliotek JS.

Wady:

- Wciąż rozwijany – przez co istnieje ryzyko wynikające z niepewności jak potoczy się przyszłość tego frameworka.
- Zwiększony wymóg testowania – w związku, że produkt jest młody, to testowanie po stronie deweloperów powinno być intensywniejsze, aby mieć pewność wyeliminowania problemów młodego frameworka.

Podsumowanie

Należy po raz kolejny tutaj wspomnieć, że opisane powyżej frameworki są znikomym wycinkiem tego co można dzisiaj znaleźć w dostępnej ofercie. Dlatego zachęcam, aby pogłębić poszukiwania we własnym zakresie i dobrać taką technologię, która z jednej strony umożliwi sprawne napisanie interfejsu, a z drugiej nie będzie trudna do opanowania. Czasu, jak to na studiach i w życiu, bywa zawsze za mało. Przypominam, że przy wyborze frameworka należy dokładnie sprawdzić czy trudniejsze części frontendu będzie można łatwo napisać z wykorzystaniem bibliotek wspomagających, takich jak np. diagramy czy planery pomieszczeń.