

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

KIERUNEK: Informatyka (INF)
SPECJALNOŚĆ: Inżynieria systemów informatycznych (INS)

**PRACA DYPLOMOWA
MAGISTERSKA**

Analiza porównawcza popularnych frameworków
webowych.

Comparative analysis of most popular web
frameworks.

AUTOR:
Marcin Mantke

PROWADZĄCY PRACĘ:
dr inż. Roman Ptak

OCENA PRACY:

Spis treści

1	Wstęp	3
1.1	Ogólny opis pracy	3
1.2	Cel pracy	3
1.3	Wymagania	3
1.4	Zarys koncepcji	3
2	Przedstawienie omawianych technologii	4
2.1	Historia i rozwój technologii webowych	4
2.2	Framework webowy	5
3	Przegląd wybranych rozwiązań	6
3.1	Ruby on Rails	6
3.2	Phoenix	6
3.3	Express	6
4	Analiza porównawcza	7
4.1	Opisowe porównanie charakterystyki wybranych frameworków	7
4.1.1	Ruby on Rails vs Phoenix	7
4.1.2	Ruby on Rails vs Express	7
4.1.3	Phoenix vs Express	7
4.2	Subiektywna ocena	7
5	Implementacja	8
5.1	Ruby on Rails	8
5.2	Phoenix	8
5.3	Express	8
6	Projekt komputerowego środowiska eksperymentalnego	9
6.1	Plan prowadzenia eksperymentów	9
6.2	Wykorzystane narzędzia	9
6.2.1	Nginx	9

SPIS TREŚCI	2
6.2.2 Docker	9
6.3 Benchmarki	9
6.3.1 Ruby	9
6.3.2 Elixir	9
6.3.3 JavaScript	9
7 Analiza wydajnościowa	10
7.1 Metody mierzenia wydajności aplikacji internetowych	10
7.2 Wyniki badań	10
8 Podsumowanie	11
Literatura	12

Rozdział 1

Wstęp

1.1 Ogólny opis pracy

W momencie pisania pracy istnieje niezliczona ilość frameworków webowych. Prawie codziennie, dla samego języka JavaScript, powstaje jeden nowy (micro) framework. Fakt ten pokazuje jak ważnym elementem stały się dla programistów frameworki. Niestety, tak dynamiczny rozwój frameworków powoduje spory problem jeśli chodzi o wybór technologii.

1.2 Cel pracy

1.3 Wymagania

1.4 Zarys koncepcji

Rozdział 2

Przedstawienie omawianych technologii

2.1 Historia i rozwój technologii webowych

Od lat 90 XX wieku świat obserwuje bardzo dynamiczny rozwój technologii związanych z Internetem. Począwszy od roku 1991, kiedy to naukowcy z instytutu badawczego **CERN** (ang. *European Organization for Nuclear Research*) opracowali standard WWW, przed programistami zaczęła się otwierać nowa gałąź tworzenia aplikacji, którą są aplikacje internetowe. Początkowo aplikacje te były jedynie statycznymi stronami WWW, na których znajdował się jedynie tekst. Wprowadzenie kaskadowych arkuszy styli (*CSS*) w roku 1996 sprawiło, że strony internetowe przybrały graficzną formę. Rok 1997 przyniósł obsługę języka *JavaScript* w przeglądarkach internetowych. Oznaczało to, że strony internetowe, poza statycznymi elementami, zyskały elementy dynamiczne, np. reagujące na akcje użytkownika.

Wraz ze wzrostem dostępu ludzi do Internetu rozwijały się technologie odpowiedzialne za strony internetowe. Za punkt początkowy istnienia nie stron, a aplikacji internetowych, można przyjąć rok 1997 i powstanie języka *PHP*. Był to pierwszy interpretowany skryptowy język programowania, który służył do budowania aplikacji internetowych działających w czasie rzeczywistym. Wraz z rozwojem języka PHP oraz innych, podobnych mu języków, np. *Python* i *Ruby*, zmienił się sposób budowania aplikacji. Programiści zaczęli rezygnować ze standardowych klientów w postaci aplikacji desktopowych i przechodzili na tzw. cienkich klientów (ang. *thin client*). Trend ten przyspiesza rozwój oraz różnorodność aplikacji serwerowych posiadających interfejs graficzny w formie strony internetowej, które nazywane są aplikacjami internetowymi [1].

2.2 Framework webowy

Aby ułatwić korzystanie z coraz liczniejszych technologii wykorzystywanych w tworzeniu aplikacji internetowych, powstały narzędzia nazywane frameworkami webowymi. Framework jest uniwersalnym środowiskiem programistycznym, które dostarcza niezbędne narzędzia wymagane do stworzenia aplikacji internetowej w wybranym języku programowania [2]. Każdy z frameworków dostarcza pewną abstrakcję, która znajduje się wokół kodu napisanego przez programistę. Przykładem takiej abstrakcji jest system mapowania ścieżki podstrony (np. */users/3*) na konkretną akcję w aplikacji (zwykle akcja *SHOW* dla kontrolera *Users*), czyli *routing*. Twórcy frameworków zauważyli, że w każdej aplikacji webowej są stosowane te same typy rozwiązań, więc w wielu przypadkach wprowadzili dane rozwiązania jako integralne części frameworków. W efekcie programiści mogą korzystać z gotowych, dogłębnie przetestowanych rozwiązań, które znajdują się w 90% aplikacji webowych.

Rozdział 3

Przegląd wybranych rozwiązań

3.1 Ruby on Rails

3.2 Phoenix

3.3 Express

Rozdział 4

Analiza porównawcza

4.1 Opisowe porównanie charakterystyki wybranych frameworków

4.1.1 Ruby on Rails vs Phoenix

4.1.2 Ruby on Rails vs Express

4.1.3 Phoenix vs Express

4.2 Subiektywna ocena

Rozdział 5

Implementacja

5.1 Ruby on Rails

5.2 Phoenix

5.3 Express

Rozdział 6

Projekt komputerowego środowiska eksperymentalnego

6.1 Plan prowadzenia eksperymentów

6.2 Wykorzystane narzędzia

6.2.1 Nginx

6.2.2 Docker

6.3 Benchmarki

6.3.1 Ruby

6.3.2 Elixir

6.3.3 JavaScript

Rozdział 7

Analiza wydajnościowa

7.1 Metody mierzenia wydajności aplikacji internetowych

7.2 Wyniki badań

Rozdział 8

Podsumowanie

Literatura

- [1] *Historia Internetu*, dostępne pod adresem: https://pl.wikipedia.org/wiki/Historia_Internetu, aktualne na dzień 1.04.2017r.
- [2] *Software framework*, dostępne pod adresem: https://en.wikipedia.org/wiki/Software_framework, aktualne na dzień 1.04.2017r.
- [3] *Ruby on Rails Guides*, dostępne pod adresem: <http://guides.rubyonrails.org/>, aktualne na dzień 14.06.2017r.
- [4] Martin, R. *Czysty kod. Podręcznik dobrego programisty*, Gliwice, Wydawnictwo Helion 2010
- [5] Diwan, A. *Tools for Testing Website Performance*, dostęp pod adresem: <https://www.sitepoint.com/tools-testing-website-performance/>, aktualne na dzień 18.03.2017r.