



Fundusze  
Europejskie  
Wiedza Edukacja Rozwój



Politechnika Wrocławskiego

Unia Europejska  
Europejski Fundusz Społeczny



„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

**Politechnika Wrocławskiego**

**Wydział Informatyki i Zarządzania**



Politechnika  
Wrocławskiego

# Zaawansowane Technologie Webowe

Laboratorium

Temat: Projekt grupowy – wybór tematu i narzędzi

Opracował: mgr inż. Piotr Jóźwiak

Data: lipiec 2020

Liczba godzin: 2 godziny

## Table of Contents

Wstęp.....	3
Projekt w mini grupach.....	3
Wybór tematu projektu .....	3
Repozytorium kodu.....	5
Model Bazy Danych.....	5
Dokumentacja prac cząstkowych.....	6
Uwagi końcowe.....	6

### Wstęp

Poczynając od dzisiejszego laboratorium charakter prac ulega zasadniczej zmianie. Do tego czasu praca była nastawiona na indywidualne wykonywanie zadań laboratoryjnych. Jednakże w normalnych warunkach pracy taka sytuacja jest raczej rzadko spotykana. Najczęściej nad opracowaniem serwisu internetowego pracuje zespół ludzi. Wynika to z faktu, że praca pojedynczego programisty zajmuje zbyt wiele czasu. Wykorzystanie większego zespołu pozwala znaczco skrócić czas wykonania zadania, jednakże wiążą się z tym zadaniem także dodatkowe zagadnienia. Zatem, aby przećwiczyć realne problemy związane ze współpracą przy projekcie, na kolejnych laboratoriach praca przyjmie charakter wykonania większego zadania.

### Projekt w mini grupach

Na początku Studenci zostaną podzieleni na zespoły dwu (w wyjątkowych sytuacjach trzy) osobowe. Dobór składu grup projektowych pozostawia się Studentom do samodzielnego ustalenia. Skład należy przestawić prowadzącemu zajęcia. Jeśli w grupie laboratoryjnej jest nieparzysta liczba Studentów to jedna grupa projektowa może składać się z trzech osób. W takiej sytuacji przy większej grupie ilość pracy do rozliczenia będzie odpowiednio większa, aby każdy z uczestników miał możliwość wykazania się.

### Wybór tematu projektu

Każda z grup projektowych w trakcie kolejnych spotkań, aż do końca semestru będzie pracowała nad indywidualnym zadaniem. Temat tego zadania ustala lub wybiera grupa projektowa samodzielnie, a następnie przedstawia temat oraz założenia prowadzącemu zajęcia. Opis zadania projektowego należy oddać w postaci sprawozdania zawierającego listę uczestników, wybrany temat serwisu do zaimplementowania oraz opis podstawowych założeń i problemów do rozwiązania. Na podstawie tego opisu każda z grup na koniec semestru będzie rozliczana, czy osiągnęła postawione sobie cele projektowe. Cele te należy tak dobrać, aby była możliwość wykonania ich w czasie jaki pozostał do końca semestru. W tym celu warto przedyskutować z prowadzącym zajęcia zakres prac. Prowadzący pomoże określić, czy ilość pracy jest wystarczająca – nie jest zbyt uboga lub za bogata.

Temat pracy musi zostać tak dobrany, aby była możliwość przedstawienia także bardziej zaawansowanych rozwiązań czy też technologii webowych. Poziom skomplikowania projektu np. prostego serwisu blogowego jest raczej za niski. Taki projekt opiera się jedynie na prostym przetwarzaniu formularzy. Dlatego temat należy tak dobrać, aby można było wykorzystać takie elementy jak np.:

- Praca z mapami
- Budowanie interaktywnych diagramów i przepływów
- Proste interaktywne gry oparte o HTML oraz JavaScript
- Generowanie wykresów
- Praca z tabelami zagnieżdzonymi
- Etc.

Aby osiągnąć powyższe cele należy wykonać reaserch dostępnych technologii i bibliotek, z których należy wybrać najlepiej dostosowaną do zrealizowania zadania. Jeśli na rynku nie istnieją biblioteki wspomagające osiągnięcie danej funkcjonalności wprost, to należy wykonać implementację samodzielnie.

Jak już zostało wspomniane wcześniej każda z grup projektowych ma przedstawić temat zadania do opracowania. W ostateczności, jeśli żaden z zaproponowanych tematów przez daną grupę nie zostanie zaakceptowany przez prowadzącego, można posiłkować się jednym z tematów przedstawionych poniżej:

1. **System rejestracji czasu pracy** – w tym zadaniu należy opracować rozwiązanie umożliwiające definiowanie listy pracowników oraz zadań do wykonania. Pracownicy mogą być członkami grup projektowych. Najistotniejszą częścią tego serwisu jest opracowanie strony z definiowaniem wykonania zadania w danym dniu. Najlepiej, aby do tego wykorzystać widok kalendarza, w którym użytkownik może w łatwy oraz interaktywny sposób dodawać zadania, urlopy, zwolnienia lekarskie. Dobrze by było, aby system umożliwiał dodanie raportu czasu pracy poprzez wysłanie e-maila do systemy z adresu danego pracownika.
2. **Generator siatki zajęć lekcyjnych** – zadanie zakłada opracowanie systemu, w którym definiujemy listę przedmiotów, klas oraz nauczycieli w szkole podstawowej. Najistotniejszym elementem tego systemu jest interaktywna siatka tygodnia zajęć, w którym z przybornika możemy układać przedmioty w godzinach lekcyjnych. System, musi także zadbać, aby liczba przedmiotów zapewniała realizację minimum programowego danej klasy oraz, aby dany nauczyciel posiadał zajęcia z jedną klasą w danym czasie.
3. **System rezerwacji sali konferencyjnych w firmie** – w tym zadaniu należy opracować serwis wspomagający zarządzanie salami konferencyjnymi w firmie. Istotą problemu jest zapewnienie rezerwacji danej sali przez organizatora spotkania. W danej sali może być tylko jedno spotkanie w danym czasie. Najważniejszą częścią tego systemu jest opracowanie rozwiązania przedstawiającego widok piętra w danym budynku wraz z przedstawieniem dostępności sali.
4. **Scheduler zadań** – zadanie polegające na opracowaniu systemu umożliwiającego graficzne modelowanie wykonania szeregu zadań za pomocą diagramów przepływów. Poprzez zadanie rozumie się np. wykonanie polecenia shell lub wysłanie zapytania do bazy danych. W wersji minimalnej system nie musi posiadać modułu egzekucyjnego. Natomiast na wyższą ocenę taki moduł musi zostać zaimplementowany.
5. **System rezerwacji miejsc kinowych** – w tym zadaniu należy opracować system rezerwacji miejsc w kinie wraz ze sprzedażą biletu na seans. Proponowane rozwiązanie powinno odzwierciedlać podstawowe problemy napotykane przy projektowaniu tego typu systemów, jak: definiowanie seansów, zapewnienie transakcyjności rezerwacji, etc.
6. **System wspomagający logistykę transportu** – projektowany system ma za zadanie wspomóc planowanie tras dostawy towaru na terenie miasta/kraju. Przy projektowaniu systemu należy zapewnić także wizualizacje tras na mapie oraz ich automatyczne generowanie dla zdefiniowanych punktów dostawy.
7. **Moduł obsługi przychodni** – tematem tego zadania jest opracowanie systemu wspomagającego umawianie wizyt do lekarza przez pacjentów. Należy zaproponować rozwiązanie uwzględniające dostępność lekarzy w godzinach pracy oraz umożliwić zapisy na wizytę przez portal WWW. W danym rozwiązaniu nacisk powinien zostać położony na opracowanie jak najwygodniejszego rozwiązania rezerwacji wizyty. Wykorzystanie czystego formularza nie jest akceptowanym rozwiązaniem. Lepszym jest wygenerowanie siatki terminów z możliwością rezerwacji jednego z nich.

## Repozytorium kodu

Z uwagi na to, że zadanie do wykonania wymaga pracy w mini zespole, niezbędnym staje się przygotowanie odpowiedniego środowiska repetytoryjnego – umożliwiającego pracę w zespole. Dlatego każda z grup musi przygotować repozytorium kodu w taki sposób, aby wszyscy członkowie grupy mieli nieograniczony dostęp do źródeł w każdym miejscu i czasie.

Aby zapewnić takie rozwiązanie można skorzystać z jednego z publicznych repozytoriów dostępnych w Internecie. Wiele z tych rozwiązań jest oferowana za darmo. Proponuję skorzystać z jednego z poniższych rozwiązań:

- **GitHub** (<https://github.com/>): jest publicznym serwisem hostingowym udostępnionym dla projektów programistycznych. GitHub udostępnia hosting systemu kontroli wersji Git. Rozwiązanie jest darmowe dla projektów open source.
- **GitLab** (<https://about.gitlab.com/>): jest kolejnym serwisem hostingowym systemu kontroli wersji Git z darmowym planem dla projektów open source. Oprócz samego repozytorium kodu udostępnia także pełen wachlarz rozwiązań DevOps – umożliwiających modelowanie pełnego procesu wytwarzania i dostarczania oprogramowania. GitLab jest bardzo ciekawym rozwiązaniem, które warto poznać, jeśli chce się wytwarzać profesjonalne oprogramowanie, które będzie w pełni automatycznie zarządzane, testowane i publikowane.
- **Azure DevOps** (<https://azure.microsoft.com/pl-pl/services/devops/>): jest rozbudowanym rozwiązaniem DevOps przygotowanym przez firmę Microsoft. Tak jak GitLab rozwiązanie to oprócz samego repozytorium kodu umożliwia w pełni zautomatyzowane przetwarzanie, analizowanie, testowanie oraz dostarczanie aplikacji. W podstawowej wersji do 5 użytkowników można skorzystać z darmowej oferty dla małych projektów.

Przy wyborze i przygotowaniu repozytorium należy także zadbać o dostęp przynajmniej do odczytu dla prowadzącego zajęcia. Prowadzący przy wystawianiu ocen Studentom będzie także mógł przeanalizować wkład poszczególnych członków zespołu na docelowe rozwiązanie.

Skorzystanie przez zespół z dodatkowych funkcjonalności DevOps przy pracy nad projektem będzie miało pozytywny wpływ na ocenę końcową.

## Model Bazy Danych

Na dzisiejszych zajęciach należy także podjąć decyzję co do wyboru bazy danych, jeśli takowa będzie wymagana. Najpopularniejszym rozwiązaniem oczywiście są wciąż bazy relacyjne, jednak należy rozważyć, czy w konkretnym zadaniu taki rodzaj bazy danych jest najlepiej dopasowany do zadania.

Po wybraniu silnika bazodanowego, należy także opracować model danych, który zostanie w późniejszym czasie wykorzystany w serwisie internetowym. Model należy opisać oraz przedstawić w formie diagramów.

Przy wyborze silnika należy także zastanowić się nad planowanym wykorzystaniem reszty stosu technologicznego. Jest to o tyle ważne zadane, że błędne wybranie stosu może skończyć się niemożnością integracji poszczególnych technologii (np. ze względu na brak driverów bazodanowych).

## Dokumentacja prac cząstkowych

Całość prac nad projektem należy opisać w sprawozdaniu z laboratorium. Sprawozdanie w formie dokumentu PDF należy przedstawić na zajęciach w trakcie oddawania rezultatów cząstkowych. Należy zadać, aby sprawozdanie było poprawnie sformatowane (nie ilość, a jakość), zawierało wszelkie niezbędne informacje o autorach, temacie, terminie zajęć oraz numerze laboratorium.

Wystarczające jest opracowanie jednego sprawozdania dla całej grupy projektowej. Nie jest wymagane oddawanie sprawozdań indywidualnych.

W pierwszym sprawozdaniu należy opisać temat wybranej pracy wraz z opisem problematyki do rozwiązania.

## Uwagi końcowe

Na koniec pozostaje przedstawić kilka uwag, które będą pomocne przy oddaniu ostatecznego rezultatu. Na ostatnich zajęciach każda z grup zobowiązana jest zaprezentować swoje rozwiązanie w postaci krótkiej prezentacji rezultatów na forum grupy laboratoryjnej.

Prezentacja ta oprócz przedstawienia działania systemu powinna także zawierać krótki opis założeń do projektu oraz problemów do rozwiązania. Dodatkowo należy także przedstawić wykorzystany stos technologiczny wraz z umotywowaniem jego wyboru.



Fundusze  
Europejskie  
Wiedza Edukacja Rozwój



Politechnika Wrocławskiego

Unia Europejska  
Europejski Fundusz Społeczny



„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

**Politechnika Wrocławskiego**

**Wydział Informatyki i Zarządzania**



Politechnika  
Wrocławskiego

# Zaawansowane Technologie Webowe

Laboratorium

Temat: Projekt grupowy – wybór backend

Opracował: mgr inż. Piotr Jóźwiak

Data: lipiec 2020

Liczba godzin: 2 godziny

## Table of Contents

Wstęp.....	3
Cel laboratorium .....	3
Przegląd wybranych technologii backend'owych .....	3
Ruby on Rails.....	4
Laravel.....	5
Express.js.....	6
Django .....	7
Pyramid .....	8
Flask .....	9
Podsumowanie .....	10

### Wstęp

O roli backendu w technologiach webowych zdążyliśmy już coś powiedzieć na poprzednich laboratoriach. Dla przypomnienia należy przytoczyć jedynie istniejący podział developerów (czy też programistów) pracujących nad aplikacjami internetowymi na kategorie:

- Front-end
- Back-end
- Full stack

Na dzisiejszym laboratorium skupimy się na wybraniu i rozpoczęciu pracy nad Back-end'em projektu grupowego. Jak już mówiliśmy back-end to część aplikacji znajdująca się na zewnętrznym serwerze, do którego użytkownik nie ma bezpośredniego dostępu (ang. Server-side). Ponieważ ta część systemu dobrze jest odizolowana od części publicznej, dlatego to tutaj odbywają się procesy zarządzania całym systemem. Wynika to oczywiście z zapewnienia bezpieczeństwa. System ten pełni rolę, można by rzec, szarej eminencji. Nie widać go na zewnątrz, ale jest on niezbędny do działania kluczowych funkcjonalności.

Oczywiście istnieją klasy serwisów internetowych, które nie posiadają back-endu. Takim przykładem mogą być proste statyczne serwisy, które pełnią jedynie rolę wizytówek czy portfolio. Nie jest wymagana w takich sytuacjach żadna skomplikowana dynamika, przez co nie ma potrzeby wprowadzania backendu.

### Cel laboratorium

Celem tego laboratorium jest przyjrzenie się dostępnych technologiom back-end'owym oraz na podstawie zdobytej wiedzy podjęcie decyzji nad wybraniem najlepiej dopasowanej technologii do implementacji projektu grupowego. Aby wspomóc decyzję pokrótkę scharakteryzujemy najpopularniejsze (choć jedynie wybrane spośród wielu) technologie wykorzystywane do implementacji logiki biznesowej.

Oczywiście żadna z grup nie jest zmuszona do wybrania jednej z omówionych technologii. Nic nie stoi na przeszkodzie, aby wybrać dowolną inną technologię niż przedstawione w niniejszym opracowaniu. Każdy wybór należy oczywiście umotywować.

### Przegląd wybranych technologii backend'owych

Wybór odpowiedniej technologii pracującej na tyłach aplikacji webowej jest kluczową decyzją umożliwiającą osiągnięcie sukcesu danego rozwiązania. Niezależnie od tego czy jest to prosty startup, czy implementacja wielkiego systemu, bardzo ważne jest ustalenie jakiego języka użyć w projekcie. Dobrze dobrana technologia zapewni szybkość działania, skalowalność oraz łatwość modernizacji i zarządzania wdrożonym produktem.

Pytanie o wybór najlepszej technologii back-end może zostać sprowadzona do odpowiedzi na poniższe pytania:

1. Jaka jest dziedzina problemu – każda technologia ma swoje własne zastosowania biznesowe, a tym samym zestaw zalet i wad.
2. Charakterystyka języka programowania – zależy od tego co chce się osiągnąć.
3. Jaka jest dostępność do ekspertów w danej technologii w zespole – jeśli zespół głównie składa się z programistów np. Java, to trudno oczekwać, że w szybki i bezbolesny sposób będą oni pisali np. Pythonie.

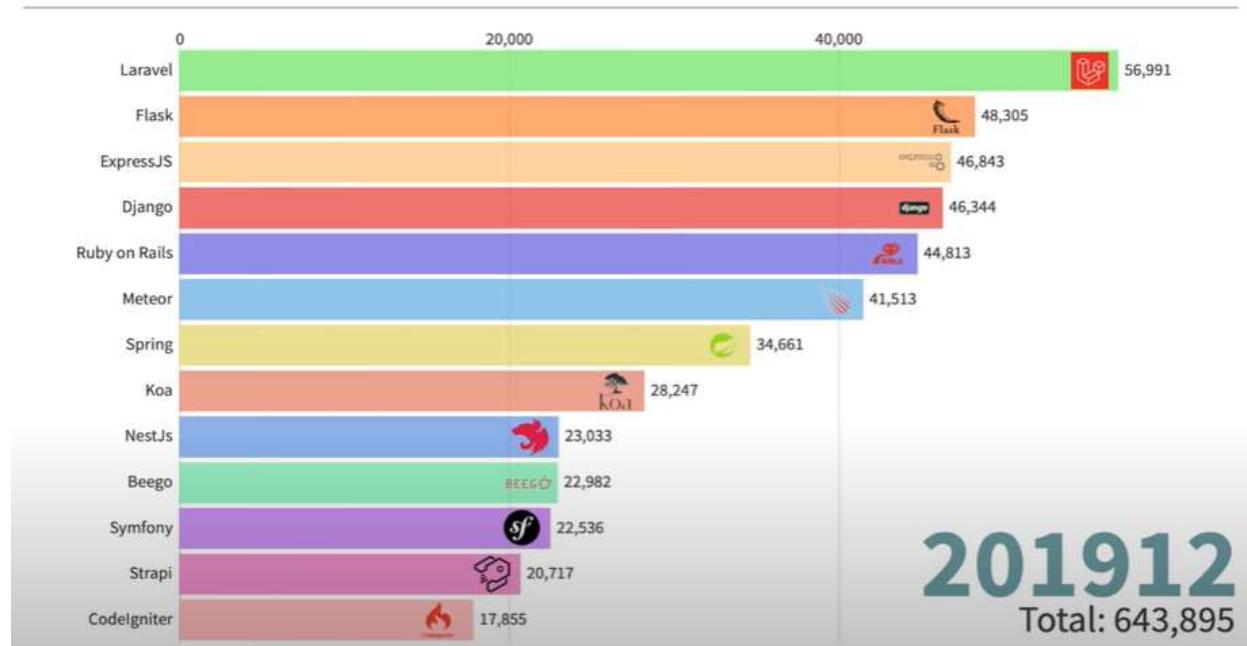
Niezwyczajne jest określić, jaka jest różnica między dwoma językami programowania. Dzieje się tak, ponieważ wszystkie języki programowania mają własną rzeszę zwolenników, którzy odmawiają wszelkiego rodzaju krytyki. Dlatego niezwyczajne jest, aby nie tylko zrozumieć różnicę między nimi, ale także zdawać sobie sprawę z wpływu, jaki może wywarzyć zły język.

Rzućmy okiem na najpopularniejsze technologie według badania *Stack Overflow* z udziałem 100 000 programistów. Według nich w roku 2019 najpopularniejsze technologie stosowane w back-end to:

1. Ruby on Rails
2. Laravel
3. Express.js
4. Django
5. Pyramid
6. Flask

Podobny skład wyłania się z badania na podstawie Github:

**Most popular backend frameworks 2012-2019 (based on Github stars)**



Dla wielu może wydawać się ta lista dość zaskakująca. Nie ma tu np. .Net, a Java Spring wcale nie jest na samym szczycie. Dlatego przyjrzyjmy się krótkiej charakterystyce tych technologii. Przypominam, że wybór grupy projektowej nie jest ograniczony do w/w listy. Nic nie stoi na przeszkodzie by wybrać inną, nieopisaną technologię w swoim projekcie.

### Ruby on Rails

Ruby on Rails jest bardzo dobrze znany od dawna z tego, że jest łatwy do czytania i pisania. W Railsach popularna jest także platforma dodatkowa dla Rubiego, która bardzo ułatwia tworzenie aplikacji internetowych.

Ruby on Rails obecnie znajduje się na szczycie listy najlepszych technologii backendowych do wykorzystania w 2019 roku.

Ponad 10% wszystkich zapytanych programistów wskazało Ruby, jako ich język pierwszego wyboru. Jego popularność szybko rośnie - ponad 46% respondentów postrzega Ruby jako swoje ulubione środowisko do pracy.

Rozwój Ruby on Rails jest bardzo szybki, co czyni go wyjątkowo dobrym dla startupów. Został stworzony w oparciu o podejście MVC (Model-View-Controller), co sprawia, że nawet najbardziej skomplikowane aplikacje można łatwo rozszerzyć o nowe funkcje lub logikę biznesową. Podobnie jak Django, istnieje kilka naprawdę zadziwiających przykładów działania Ruby on Rails, czyniąc go godnym zaufania frameworkm.

Głównym celem Ruby on Rails jest szybkość i łatwość pisania kodu, co oszczędza czas i podnosi jakość implementacji. Bardzo poważną wadą tego rozwiązania jest powolne działanie w porównaniu z konkurencją. Także wymagania wobec serwera i problemy ze skalowaniem są często wskazywaną bolączką przez programistów.

Społeczność Ruby on Rails jest ogromna - a mówiąc o niej, mamy na myśli to, że w zasadzie w każdym dużym mieście społeczność spotyka się regularnie. Na wiele pytań, które może zadać zespół programistów, gdzieś już udzielono odpowiedzi. Istnieją różne biblioteki open source (znane również jako gems), więc ktoś wykonał już wiele pracy. Znalezienie dobrej dokumentacji dla wspomnianych gem'ów może być trudne, szczególnie dla mniej popularnych bibliotek.

Przykłady wdrożeń: Airbnb, Couchsurfing, Kickstarter, Shopify, GitHub, Bloomberg.

Przykład kodu:

```
# fibonacci sequence

# input n
n = 7
def fibonacci(n)
  # if n = 1, we want to output 1
  if n == 1
    1
  # if n = 2, we want to output 1
  elsif n == 2
    1
  # if n > 2, we want to output the sum of the previous two values
  else
    fibonacci(n-1) + fibonacci(n-2)
  end
end

# print the output of n
puts "#{n}'s fibonacci value is #{fibonacci(n)}"
```

### Laravel

Język PHP znajduje się w pierwszej dziesiątce w sieci, głównie ze względu na niezwykłą popularność WordPressa i często używane frameworki. Ogromna popularność zawsze wpływa na dużą społeczność skupioną wokół PHP, dzięki czemu w sieci odnajdziemy wiele opublikowanych już rozwiązań, które pomogą nam rozwijać się szybciej przy niższych kosztach.

PHP jest dobrze znane z możliwości integracji wielu baz danych, jest bardzo dynamicznym językiem. Zapewnia wszystkie niezbędne funkcje, których programiści potrzebują do tworzenia aplikacji internetowych, pracy z bazą danych i problemów z uwierzytelnianiem.

Wśród frameworków PHP warto zająć się Laravel. Laravel jest nieprzypadkowo jedną z najlepszych technologii backendowych do wykorzystania w 2019 roku.

Laravel to framework PHP MVC, który przyspiesza proces programowania dzięki eleganckiemu i prostemu wzorcowi składni, migracji bazy danych i narzędzi do tworzenia schematów. Korzystne jest również proste i bardzo bezpieczne uwierzytelnianie.

Dokumentacja Laravel jest bardzo przejrzysta, uporządkowana i zrozumiała. Posiada wbudowane wsparcie do MVC, jednak architektura systemu nie ogranicza nas do tego jednego modelu. Laravel umożliwia programistom tworzenie zarówno małych, jak i dużych aplikacji. Zapewnia także bezpieczną strukturę budowania API, która może wspierać budowanie aplikacji hybrydowych.

PHP ma swoich zwolenników i krytyków. Nie każdy zespół programistów jest chętny do pracy z PHP, ale mimo wielu krytyków, deweloperzy często z niego korzystają.

Przykłady realizacji: [laracasts.com](https://laracasts.com), [events.arsenal.com](https://events.arsenal.com)

```
1 <p class="message-controls no-margin">
2   <!-- <i class="sprite-project sprite-project-pin-gray pull-right float-none-xs"></i> -->
3
4   <button class="pull-right">
5     <i class="fa fa-ban pull-right"></i>
6   </button>
7
8   @if ($data->owner_id == app('user')->id)
9     <button type="button" class="js-button-message-edit pull-right" data-id="{{ $data->id }}>
10       <i class="fa fa-pencil"></i>
11     </button>
12   @endif
13
14   <button class="pull-right">
15     <i class="fa fa-thumb-tack pull-right"></i>
16   </button>
17 </p>
```

### Express.js

Patrząc na listę najpopularniejszych technologii, JavaScript nadal pozostaje jednym z najczęściej używanych języków programowania. Pozostaje niezmiennym liderem od dłuższego czasu. Prawie 70% programistów wskazało JavaScript jako framework pierwszego wyboru.

Express.js zasłużenie znajduje się na liście najlepszych technologii backendowych w 2019 roku. Prawdopodobnie dlatego także większość programistów przyjęła framework Node.js jako preferowaną technologię. Express.js zapewnia niezwykłą skalowalność i niesamowitą obsługę równoległości. Obecnie technologia ta jest wykorzystywana w usługach lub aplikacjach, które wymagają szybkiego wykonania wielu zadań z ogromną liczbą użytkowników w tym samym czasie.

Express.js jest platformą server-side dla Node.js. Jest to również jeden z najlepiej wspieranych frameworków Node.js. Posiada bardzo rozbudowaną społeczność miłośników tej technologii. Express.js

wraz z Node.js mogą być używane do tworzenia interfejsów API dla jednostronicowych, wielostronicowych, hybrydowych aplikacji mobilnych i internetowych.

Framework Express.js ma jeszcze jedną ważną zaletę. Pozwala zespołowi programistów używać tego samego języka, którym jest JavaScript, zarówno w back-end, jak i w front-end. W rezultacie proces tworzenia jest znacznie szybszy i łatwiejszy, ponieważ jedna osoba może zarządzać zarówno warstwami prezentacji, jak i dostępu do danych.

Przykłady wdrożeń: Uber, PayPal, Netflix, LinkedIn

```
1 // Load required packages
2 var express = require('express');
3
4 // Create our Express application
5 var app = express();
6
7 // Create our Express router
8 var router = express.Router();
9
10 // Initial dummy route for testing
11 // http://localhost:3000/api
12 router.get('/', function(req, res) {
13   res.json({ message: 'You are running dangerously low on beer!' });
14 });
15
16 // Register all our routes with /api
17 app.use('/api', router);
18
19 // Start the server
20 app.listen(3000);
```

### Django

Z wyników ankiety Stack Overflow wynika, że Python staje się najszybciej rozwijającym się głównym językiem programowania. Okazuje się, że Python jest także jednym z najbardziej poszukiwanych języków wśród programistów. Dlatego też programiści coraz częściej wskazują na Python'a jako na język, który chcieliby się nauczyć – 25,1% deweloperów.

Django jest obecnie najpopularniejszym frameworkm sieciowym Pythona, który zapewnia szybki rozwój i bardzo czysty, pragmatyczny projekt. Obecnie Python działa na dowolnej platformie oraz jest open source – dostarczając łatwo skalowalne rozwiązania. Ponieważ wykorzystuje zestaw komponentów, zapewnia standardowy sposób tworzenia witryn internetowych w bardzo łatwy i szybki sposób. Głównym celem Django jest ułatwienie tworzenia złożonych witryn internetowych opartych na bazie danych.

Django jest obecnie jednym z najlepszych rozwiązań do tworzenia minimalnie opłacalnego produktu, na którym można dalej budować, ponieważ staje się w pełni funkcjonalny. Oznacza to, że zawiera już wszystkie niezbędne narzędzia do tworzenia praktycznie każdej dodatkowej funkcji dla tego produktu. Django realizuje wzorzec MTV (model-template-view).

Ponadto Django oferuje jeden z najwyższych poziomów bezpieczeństwa. Jednocześnie może łączyć się z wieloma istniejącymi aplikacjami innych firm, takimi jak Twitter, Facebook czy aplikacje płatnicze.

Bardzo istotną zaletą tego framework'a jest funkcjonalność automatycznego generowania panelu administratora, z możliwością dalszego dostosowywania. Pozwala to zaoszczędzić wiele czasu związanego z implementacją tej części systemu.

Właściwa dokumentacja jest bardzo ważna dla zaoszczędzenia czasu programisty, co przekłada się na oszczędność pieniędzy klientów. Skalowalność Django jest również jedną z najważniejszych części rozwoju i wzrostu projektu. Posiada także jedną z największych społeczności wraz z biblioteką open source, która zapewnia programistom wspaniałe narzędzie pomagające im w dostarczaniu funkcjonalności w krótszym czasie.

Przykłady realizacji: Instagram, Disqus, NASA.

```
8  {% if message_list %} 
9   <table class="message_list">
10    <thead>
11      <tr>
12        <th>Date</th>
13        <th>Time</th>
14        <th>Message</th>
15      </tr>
16    </thead>
17    <tbody>
18      {% for message in message_list %}
19        <tr>
20          <td>{{ message.log_date | date:'d M Y' }}</td>
21          <td>{{ message.log_date | date:'H:i:s' }}</td>
22          <td>
```

### Pyramid

Pyramid bardzo ułatwia pisanie aplikacji internetowych. Wraz z rozwojem aplikacji Pyramid będzie oferować wiele funkcji, dzięki którym pisanie złożonego oprogramowania będzie wymagało mniej wysiłku. Ponadto Pyramid działa we wszystkich obsługiwanych wersjach Pythona.

Pyramid jest najczęstszą alternatywą dla Django. Niektórzy programiści twierdzą, że jest to framework do pisania frameworków, natomiast inni nazywają go Flask'iem na sterydach.

Pyramid zawiera wbudowane podstawowe mechanizmy routingu adresów URL wraz z widokami, przydatne funkcjonalności autoryzacyjne oraz całkiem spory zestaw wtyczek i aplikacji firm trzecich, które mogą zapewnić np. szkielet będący odpowiednikiem panelu administracyjnego, jaki otrzymujemy w Django. Jego zestaw narzędzi jest nieco większy niż ten dostarczony w Flask. Jednak w tym środowisku nie ma wbudowanego wsparcia dla ORM, szablonów czy też usług formularzy.

Pyramid nadaje się zarówno do małych aplikacji, jak i dużych serwisów. Jedną z najczęstszych skarg na Pyramid jest to, że oferuje tak wiele opcji, że rozpoczęcie nowego projektu może być trudne ze względu na problem z podjęciem decyzji z czego będzie się korzystać.

Przykład wdrożenia: Dropbox, SurveyMonkey, Reddit.

```
from wsgiref.simple_server import make_server
from pyramid.config import Configurator
from pyramid.response import Response

def hello_world(request):
    return Response('Hello World!')

if __name__ == '__main__':
    with Configurator() as config:
        config.add_route('hello', '/')
        config.add_view(hello_world, route_name='hello')
    app = config.make_wsgi_app()
    server = make_server('0.0.0.0', 6543, app)
    server.serve_forever()
```

### Flask

Flask jest prawie bezpośredniem przeciwnieństwem Django. Jest to microframework, który zawiera tylko minimalne narzędzia i wymaga ręcznej konfiguracji wszystkiego innego. Oznacza to, że programiści mają pełną elastyczność w doborze narzędzi.

Bez problemu można użyć innego ORM niż popularny SQLAlchemy, (przy okazji, polecam przetestowanie PonyORM, chociaż nie jest jeszcze gotowy do produkcji, ponieważ nie obsługuje migracji). „Chcesz innego języka szablonów niż Jinja2, bądź moim gościem!” Dzięki takiemu podejściu Flask świetnie sprawdza się wszędzie tam, gdzie programiści chcą mieć pełną kontrolę nad każdym elementem swojej aplikacji.

Flask najlepiej sprawdza się w projektach, w których architektura jest oparta na mikro serwisach. Z drugiej strony, uruchomienie projektu za pomocą Flaska może być trudniejsze ze względu na czas i wysiłek wymagany do napisania i skonfigurowania wszystkiego ręcznie - podczas gdy wszystkie te elementy są po prostu dostarczane w Django. Wybierając Flask, w zasadzie zamieniamy wygodę na większą elastyczność.

Przykład wdrożenia: Pinterest, LinkedIn.

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello():
    return "Hello World!"

@app.route('/<name>')
def hello_name(name):
    return "Hello {}!".format(name)

if __name__ == '__main__':
    app.run()
```

## Podsumowanie

Przedstawiona powyżej charakterystyka opisuje jedynie wybrane dostępne technologie. Zanim podejmiemy decyzję o dobraniu odpowiedniego frameworku należy przyjrzeć się jeszcze innym ofertom, chociażby takim jak ASP.Net Core czy Java Spring.

Na koniec tego laboratorium należy przygotować krótkie sprawozdanie przedstawiające wybraną technologię do pisania Back-end'u. Jednocześnie należy rozpocząć prace implementacyjne nad projektem. Należy także zapewnić odpowiedni podział zadań pomiędzy członków zespołu. Mile widziane będzie wykorzystanie wsparcia z np. tablic kanbanowych dostępnych w środowiskach DevOps opisanych na poprzednim laboratorium. Postęp prac implementacyjnych będzie prezentowany prowadzącemu na kolejnych laboratoriach.



Fundusze  
Europejskie  
Wiedza Edukacja Rozwój



Politechnika Wrocławskiego

Unia Europejska  
Europejski Fundusz Społeczny



„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

**Politechnika Wrocławskiego**

**Wydział Informatyki i Zarządzania**



Politechnika  
Wrocławskiego

# Zaawansowane Technologie Webowe

Laboratorium

Temat: Projekt grupowy – wybór frontend

Opracował: mgr inż. Piotr Jóźwiak

Data: lipiec 2020

Liczba godzin: 2 godziny

## Table of Contents

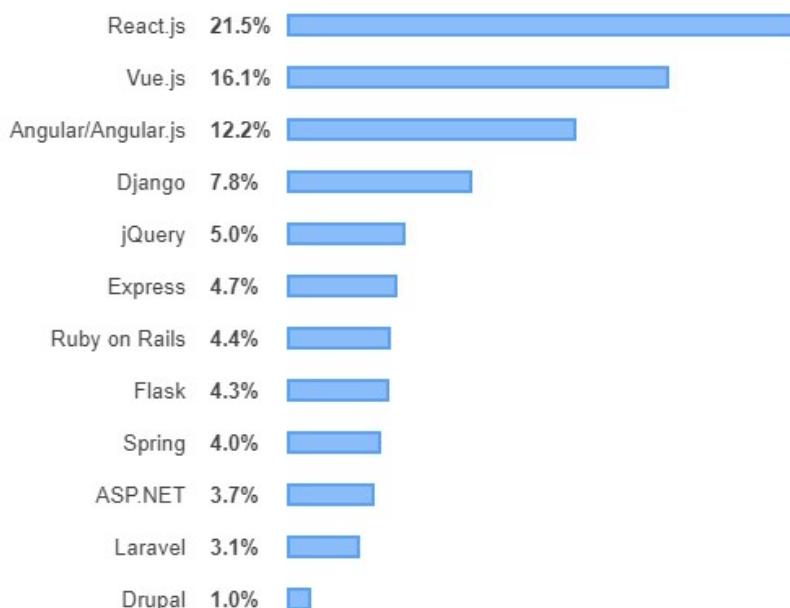
Wstęp .....	3
Cel laboratorium .....	3
Przegląd wybranych technologii frontendowych .....	4
React .....	4
Vue .....	5
Angular .....	6
Flutter.....	7
Podsumowanie .....	8

### 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 pokrótkę scharakteryzujemy najpopularniejsze (choć jedynie wybrane spośród wielu) technologie wykorzystywane do implementacji warstwy prezentacji strony internetowej.

## Przegląd wybranych technologii frontendowych

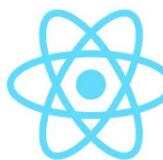
Dokonując wyboru, należy pamiętać o kilku wymaganiach stawianych przed UI frameworkami. Wybierając jedną z najlepszych platform frontendowych, musimy wziąć pod uwagę perspektywę zarówno użytkowników, jak i zespołów programistycznych. Funkcje te obejmują etapy projektowania, rozwoju, testowania i konserwacji. 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 wstępную 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 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 Androidea i iOS w 2015 roku.

W ciągu dwóch lat Facebook ogłosił nowy podstawowy algorytm - React Fibre. React stał się najlepszym frameworkm 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 wprawdzania 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 zwiększały 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 expirience*.
- 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 dłuża 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 framework'a 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 jednostronnicowych. 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 framewokami. 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 framework'a. 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 zwróciли 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 framework 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 wstępnej 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 rozwiązań 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 czegokolwiek 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 utrzymanie 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 przeciwagę 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 komplikowanego 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 odrębna funkcjonalnością. Język programowania Dart oferuje opcje łańcucha narzędzi dla Intel, ARM i JavaScript, które umożliwiają komplikację 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.
- Zwiększyły 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 dostęonej ofercie. Dlatego zachęcam, aby pogłębić poszukiwania we własnym zakresie i dobrać taką technologię, która z jednej strony umożliwia 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.



Fundusze  
Europejskie  
Wiedza Edukacja Rozwój



Politechnika Wrocławskiego

Unia Europejska  
Europejski Fundusz Społeczny



„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

**Politechnika Wrocławskiego**

**Wydział Informatyki i Zarządzania**



Politechnika  
Wrocławskiego

# Zaawansowane Technologie Webowe

Laboratorium

Temat: Projekt grupowy – implementacja Single Sign On

Opracował: mgr inż. Piotr Jóźwiak

Data: lipiec 2020

Liczba godzin: 2 godziny

## Table of Contents

Wstęp.....	3
Uwierzytelnianie vs Autoryzacja .....	3
SAML .....	4
OAuth2.....	5
OpenID Connect.....	7
Przykład wykorzystania SSO.....	8
Implementacja SSO w projekcie grupowym .....	9

## Wstęp

Duże organizacje poszukiwały sposobu na skonsolidowanie swoich systemów uwierzytelniania, celem łatwiejszego zarządzania użytkownikami i bezpieczeństwem. Rozwiązaniem tego problemu jest Single Sign On (SSO). SSO jest skonsolidowanym serwisem zarządzającym sesją oraz autentykacją, w taki sposób, że pojedynczy zestaw login/hasło może być wykorzystany do uzyskania dostępu do wielu aplikacji. Piękno tego rozwiązania jest ukryte w jego prostocie. Serwis autentykuje użytkownika na dowolnej platformie, a tym samym w sposób przeźroczysty dla użytkownika dostarcza mu dostęp do wszystkich innych serwisów bez ponownego procesu logowania.



Naturalnym rozwinięciem technologii SSO była chęć skorzystania z tego API przez zewnętrznych programistów. Chcieli oni zapewnić usługi autoryzacji do własnych serwisów w oparciu o istniejące rozwiązania SSO wdrożone w dużych firmach. Oczywiście jest, że takie podejście staje się sporym wyzwaniem. Do tego wszystkiego pojawiły się różnego rodzaju sieci społecznościowe, które jeszcze bardziej komplikowały sprawę.

Obecnie mamy tysiące aplikacji, które wykorzystują uwierzytelnianie za pośrednictwem sieci społecznościowych jak Facebook, Google, Twitter czy LinkedIn. Największy problem w danej architekturze polega na utrzymaniu prostoty systemu bez narażania poziomu zabezpieczeń. Rozwiązaniem w tej sytuacji okazuje się koncepcja Federated Identities (FI).

Aktualnie posiadamy trzy główne protokoły FI: SAML, OAuth2 oraz OpenID Connect. Zanim pokróćce się im przyjrzymy, omówmy kilka powszechnych pojęć, które są często mylone.

## Uwierzytelnianie vs Autoryzacja

Problematyka bezpieczeństwa dostępu do zasobów definiuje dwa terminy: uwierzytelnianie i autoryzacja, które często rozumiane są zamiennie. Uwierzytelnianie to weryfikacja tożsamości, podczas gdy autoryzacja to weryfikacja tego, do czego uzyskuje się dostęp.

Uwierzytelnianie polega na sprawdzaniu danych logowania przed udzieleniem użytkownikowi dostępu do systemu. Zazwyczaj w najprostszym ujęciu jest to formularz logowania monitujący o podanie nazwy użytkownika oraz hasła.

Autoryzacja następuje po procesie uwierzytelniania poprzez weryfikację praw przed udzieleniem dostępu do wymaganych zasobów, takich jak bazy danych, pliki, repozytoria itp. Praktycznym przykładem może być sprawdzenie czy aktualnie zalogowany użytkownik może edytować posty na blogu.



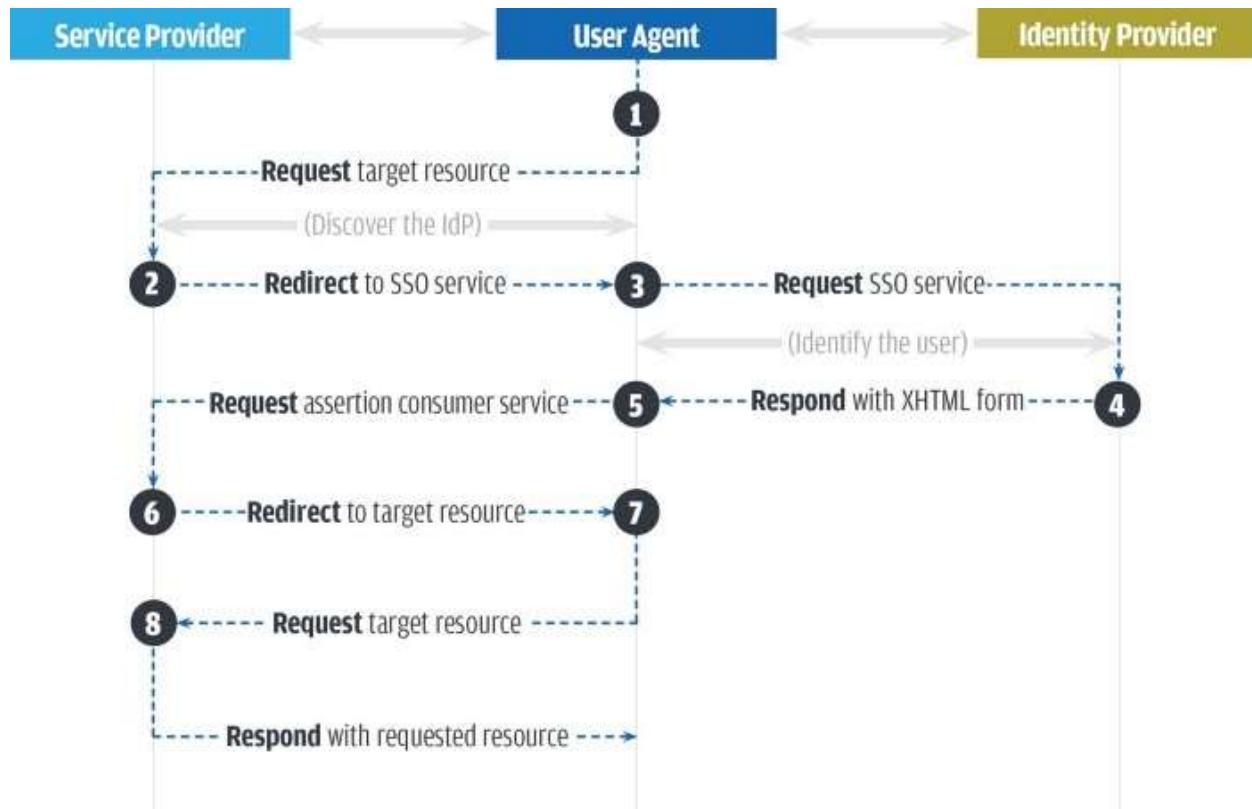
Zarówno autoryzacja, jak i uwierzytelnianie mają fundamentalne znaczenie dla bezpieczeństwa i zarządzania dostępem. Te dwie koncepcje, mają kluczowe znaczenie dla bezpieczeństwa infrastruktury systemu, a ich zrozumienie jest niezbędne do zarządzania tożsamością i dostępem.

### SAML

**Security Assertion Markup Language** (SAML) to oparty na języku XML otwarty standard używany do implementacji logowania Single Sign On (SSO). SAML 2.0 został opracowany w 2005 roku i jest aktualną wersją tego standardu.

SAML jest używany zarówno do uwierzytelniania, jak i autoryzacji między dwiema stronami: dostawcą usług (Office365, Salesforce, G Suite itp.) i dostawcą tożsamości (Okta, OneLogin, Ping Identity itp.). Z założenia Service Provider (SP) ufa dostawcy tożsamości (Identity Provider- IdP) w procesie uwierzytelniania. Odbywa się to za pomocą dokumentu SAML XML wysłanego przez IdP, zawierającego autoryzację i uwierzytelnienie użytkownika, a następnie przekierowanie do usługodawcy SP.

Przyjrzymy się procesowi uwierzytelniania przedstawionemu na poniższym diagramie. Jako Identity Providera (IdP) możemy przyjąć serwis Okta. Natomiast jako Service Provider (SP) występuje aplikacja webowa – np. nasza z projektu.



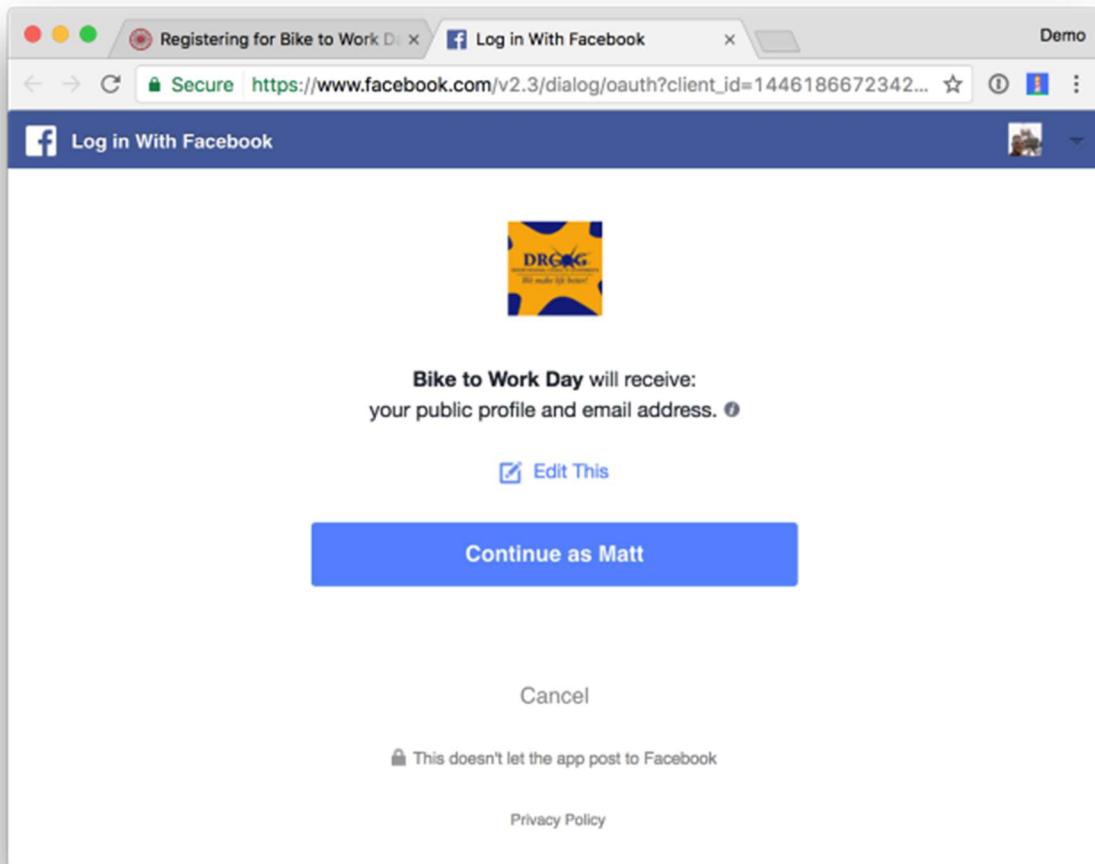
1. Użytkownik próbuje zalogować się do aplikacji webowej z przeglądarki.
2. Aplikacja webowa odpowiada, generując żądanie SAML.
3. Przeglądarka przekierowuje użytkownika na adres URL do logowania jednokrotnego, Okta analizuje żądanie SAML, uwierzytelnia użytkownika (może to być za pomocą nazwy użytkownika i hasła, uwierzytelniania dwuskładnikowego lub MFA, jeśli użytkownik nie znajduje się wewnętrznej sieci firmy; jeśli użytkownik jest już uwierzytelniony w Okta, ten krok zostanie pominięty) i generuje odpowiedź SAML.
4. Okta ponownie wysyła zakodowaną odpowiedź SAML do przeglądarki klienta.
5. Przeglądarka przekierowuje odpowiedź SAML do aplikacji webowej.
6. Jeśli weryfikacja się powiedzie, użytkownik zostanie zalogowany do aplikacji webowej i uzyska dostęp do zasobów.

## OAuth2

**OAuth2** to otwarty standard używany do autoryzacji, który umożliwia aplikacjom dostarczanie funkcjonalności z „delegowaną autoryzacją”. W przeciwieństwie do innych struktur, które zapewniają uwierzytelnianie, OAuth2 autoryzuje tylko urządzenia, API, serwery z tokenami dostępu, a nie poświadczaniami. Działa przez HTTPS.

Technologia OAuth2 powszechnie wykorzystywana jest w takich serwisach jak Facebook czy Google. Serwisy te umożliwiają skorzystanie z ich wewnętrznych mechanizmów autoryzacji do logowania w zewnętrznych serwisach, nie związanych z dostawcą tożsamości.

Zwróćmy uwagę na poniższy przykład. Przykładowa strona z pierwszej zakładki korzysta z Facebookowego mechanizmu logowania, widocznego na drugiej zakładce.

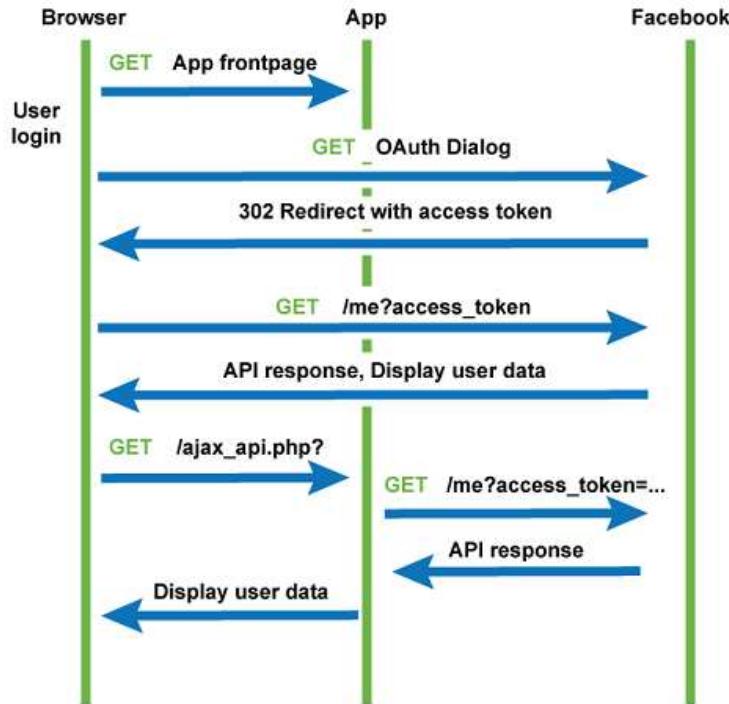


Przypomina to system kart hotelowych, ale dla aplikacji. Jeśli dysponujemy hotelową kartą-kluczem, możemy uzyskać dostęp do pokoju. Jak zdobyć kartę hotelową? Aby ją otrzymać, musimy przeprowadzić proces uwierzytelniania w recepcji. Po uwierzytelnieniu i uzyskaniu karty-klucza możemy uzyskać dostęp do zasobów w hotelu.

Oauth2 wyróżnia cztery role:

- Właściciel zasobu: ogólnie sam użytkownik.
- Klient: aplikacja żądająca dostępu do serwera zasobów.
- Serwer zasobów: serwer obsługujący chronione dane (na przykład Facebook hostujący profil i dane osobowe).
- Serwer autoryzacyjny: serwer wystawiający klientowi token dostępu. Ten token zostanie użyty, aby uzyskać dostęp do żądanych zasobów.

Poniższy diagram przedstawia proces uwierzytelniania protokołem Oauth2:



1. Spotify chce uzyskać dostęp do listy znajomych z Twojego konta na Facebooku.
2. Spotify przekierowuje Cię na serwer autoryzacyjny (w tym przypadku Facebook).
3. Jeśli autoryzujemy dostęp, serwer autoryzacyjny wysyła kod autoryzacyjny do klienta (Spotify) w odpowiedzi zwrotnej.
4. Następnie ten kod jest wymieniany na token dostępu między Facebookiem a Spotify.
5. Teraz Spotify może używać tego tokena dostępu do wysyłania zapytań do serwera zasobów (Facebook) i pobierania listy znajomych.

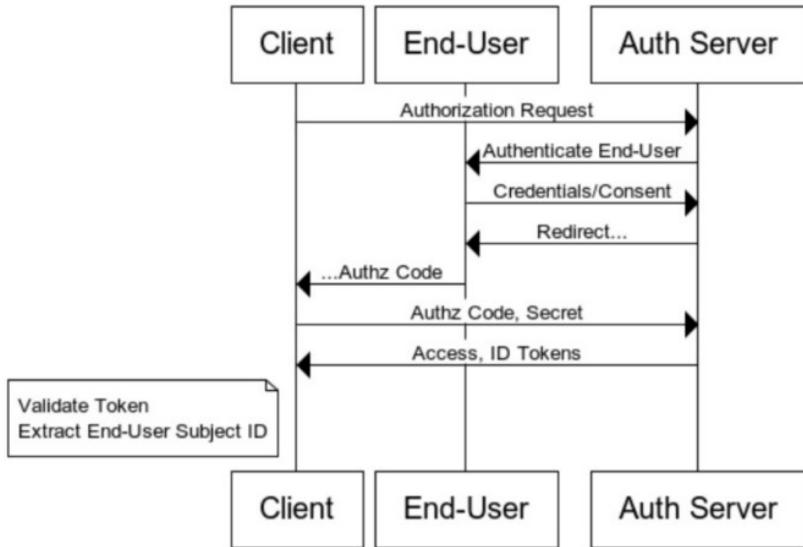
Należy zwrócić uwagę na to, że użytkownik nigdy nie zobaczy tokenu dostępu, będzie on przechowywany w sesji. Serwer autoryzacji wysyła również inne informacje, takie jak okres ważności tokenu i token odświeżania.

## OpenID Connect

**OpenID Connect** to warstwa tożsamości uzupełniająca protokół OAuth 2.0, która rozszerza OAuth2 i umożliwia „uwierzytelnianie federacyjne” (Federated Authentication).

Przebieg procesu OpenID Connect jest podobny do przepływu autoryzacji OAuth2, a główną różnicą jest „token id”, który umożliwia uwierzytelnienie użytkownika.

Należy pamiętać, że uwierzytelnianie federacyjne to coś zupełnie innego niż autoryzacja delegowana. Przyjrzyjmy się ponownie przykładowi Facebooka i Spotify.



- Uwierzytelnianie federacyjne polega na logowaniu do Spotify przy użyciu danych logowania z Facebooka.
- Autoryzacja delegowana to możliwość uzyskania dostępu do zasobów przez aplikację zewnętrzną. W tym przypadku Spotify próbuje uzyskać dostęp do listy znajomych na Facebooku, aby zaimportować ją do Spotify.

Wszystkie powyższe informacje można podsumować w poniższej tabeli:

	SAML 2.0	OAuth2	OpenID Connect
Czym jest?	Otwarty standard do autoryzacji oraz uwierzytelniania	Otwarty standard do autoryzacji	Otwarty standard do uwierzytelniania
Twórca	Opracowany przez OASIS w 2001 roku	Opracowany przez Twitter oraz Google w 2006 roku	Opracowany przez OpenID Foundation w 2014 roku
Docelowy odbiorca	SSO dla aplikacji korporacyjnych	API autoryzacyjne	SSO dla aplikacji konsumenckich
Format komunikacji	XML	JSON	JSON

## Przykład wykorzystania SSO

Aby rozpocząć prace deweloperskie polegające na zbudowaniu SSO w aplikacji webowej można skorzystać z wielu gotowych rozwiązań. Do zbudowania uwierzytelniania w oparciu o SAML warto użyć aplikacji napisanej w PHP o nazwie SimpleSAMLphp (<https://simplesamlphp.org/>). Jest to prosty skrypt dostarczający zarówno funkcjonalności Service Providera (SP) oraz Identity Providera (IdP). Najlepiej pierwsze kroki rozpoczęć od tego narzędzia. Sam przykład implementacji jest mocno powiązany z wykorzystaną technologią po stronie klienta, tak więc nie da się tutaj przedstawić kompletnego przykładu wykonania tej funkcjonalności. Zatem należy przeprowadzić własne poszukiwania w celu przygotowania aplikacji na SSO z SAML. Dobrym zestawieniem możliwości zestawienia SAMLa w wielu różnych

technologiach jest artykuł dostępny pod tym adresem: <https://medium.com/the-new-control-plane/i-need-a-saml-stack-now-63d9691e2d43>

W przypadku standardu OAuth2 oraz Angulara warto spojrzeć na ten materiał: <https://www.baeldung.com/rest-api-spring-oauth2-angular>. Implementacja OAuth2 może nie być taka prosta. Więcej szczegółów na temat OAuth2 można odnaleźć w tym artykule: <https://sekurak.pl/oauth-2-0-jak-dziala-jak-testowac-problemy-bezpieczenstwa/>.

### Implementacja SSO w projekcie grupowym

Kolejnym zadaniem do wykonania przez grupy projektowe jest implementacja dowolnego sposobu funkcjonalności Single Sign On. Można skorzystać z dowolnej technologii SSO (SAML, OAuth2). Grupie projektowej pozostawia się do wyboru, czy logowanie odbędzie się za pośrednictwem zewnętrznego serwisu ogólnie dostępnego jak Facebook, Google, etc., czy zaimplementowany zostanie własny serwer uwierzytelniania i autoryzacji.



Fundusze  
Europejskie  
Wiedza Edukacja Rozwój



Politechnika Wrocławskiego

Unia Europejska  
Europejski Fundusz Społeczny



„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

**Politechnika Wrocławskiego**

**Wydział Informatyki i Zarządzania**



Politechnika  
Wrocławskiego

# Zaawansowane Technologie Webowe

Laboratorium

Temat: Projekt grupowy – Migracja bazy danych

Opracował: mgr inż. Piotr Jóźwiak

Data: lipiec 2020

Liczba godzin: 2 godziny

## Table of Contents

Wstęp.....	3
Jak działa migracja bazy danych.....	3
Poszczególne zmiany jako skrypty .....	3
Migracje z użyciem zewnętrznych narzędzi .....	4
Jak przeprowadzić migrację? .....	4
Wykorzystanie frameworkowych narzędzi i bibliotek.....	5
Wykorzystanie narzędzi zewnętrznych.....	5
Które podejście jest najlepsze?.....	5
Podsumowanie .....	5

## Wstęp

W czasie, gdy GIT zdobywał swoją popularność, powstał trend do implementowania aplikacji webowych z wykorzystaniem mapowania obiektowo-relacyjnego (*object relational mapping* – ORM). Główną ideą powstania tej funkcjonalności było zauważenie, że skoro programiści wprowadzają zmiany w kodzie, które z łatwością można wycofać za pomocą Git, to podobne rozwiązanie należy zastosować w schemacie bazy danych. W końcu każda większa nowa funkcjonalność obejmuje zmiany kodu i schematu jednocześnie. Tak popularne narzędzia jak **Rails** i **Django** dodaly ORM i migrację bazy danych (znaną również jako migracja schematu) jako część framework'a.

Jednak koncepcja migracji bazy danych nie ogranicza się do popularnych platform internetowych. Istnieją niezależne biblioteki do migracji baz danych, takie jak Flyway (<https://flywaydb.org/>) i Liquibase (<https://www.liquibase.org/>). Wyobraźmy sobie możliwość cofania szczegółowych zmian w schemacie podczas pisania kodu.

## Jak działa migracja bazy danych

Skoro wiemy skąd pojawiła się koncepcja migracji schematów to przyjrzyjmy się sposobie funkcjonowania.

### Poszczególne zmiany jako skrypty

Wspominaliśmy już, że migracja schematu bazy danych polega w zasadzie na szczegółowym śledzeniu zmian w jej strukturze (czasem także zmian w samych danych). Śledzenie odbywa się poprzez odzwierciedlenie każdej ze zmian jako oddzielne pliki skryptów. W ten sposób każda zmiana zostaje sformalizowana w postaci kodu, który w łatwy sposób oddaje się przechowywaniu w systemie kontroli wersji.

Jako przykład pliku migracji niech posłuży poniższy listing z framework'a Rails:

```
class CreateProducts < ActiveRecord::Migration[5.0]
  def change
    create_table :products do |t|
      t.string :name
      t.text :description

      t.timestamps
    end
  end
end
```

Jak widać z powyższego przykładu, plik migracji to formalny zapis zmian w schemacie bazy danych. Więcej o działaniu migracji w frameworku Ruby on Rails można znaleźć tutaj: <https://www.c-sharpcorner.com/article/create-migration-file-in-ruby-on-rails/>. Oczywiście można utworzyć stan na podstawie aktualnego stanu bazy danych, więcej o tym tutaj: [https://edgeguides.rubyonrails.org/active\\_record\\_migrations.html#schema-dumping-and-you](https://edgeguides.rubyonrails.org/active_record_migrations.html#schema-dumping-and-you).

Pliki migracji tworzą pełną historię zmian jaką należy wykonać z danego punktu w przeszłości, aby osiągnąć aktualną wersję schematu – pasującą do kodu aplikacji (w naszym przypadku webowej).

### Migracje z użyciem zewnętrznych narzędzi

W poprzednim przykładzie przedstawiliśmy sposób migracji bazy danych wbudowany we framework Ruby. Tylko co zrobić, jeśli używany przez nas framework nie posiada takiej funkcjonalności? Wtedy do tego celu należy wykorzystać zewnętrzne narzędzie.

Poniższy przykład przedstawia formalizm migracji z Liquibase:

```
<?xml version="1.0" encoding="UTF-8"?>

<databaseChangeLog
    xmlns="http://www.liquibase.org/xml/ns/dbchangelog"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog
        http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-3.1.xsd">

    <changeSet id="1" author="bob">
        <createTable tableName="department">
            <column name="id" type="int">
                <constraints primaryKey="true" nullable="false"/>
            </column>
            <column name="name" type="varchar(50)">
                <constraints nullable="false"/>
            </column>
            <column name="active" type="boolean" defaultValueBoolean="true"/>
        </createTable>
    </changeSet>

</databaseChangeLog>
```

Tym razem jest on w formacie XML. Liquibase wspiera zapis migracji lub inaczej changeset (zgodnie z tym jak nazywają to w Liquibase) w wielu formatach, np. JSON lub YAML (<https://docs.liquibase.com/concepts/basic/changelog.html>).

Jeśli więc nie chcemy ręcznie pisać migracji bazy danych w formacie SQL, nie ma żadnych standardów dotyczących sposobu tworzenia plików migracji. Oczywiście w plikach SQL można napisać własne niestandardowe skrypty migracji bazy danych. Ale po co pisać odręcznie i przypadkowo wprowadzać błędy, skoro można użyć narzędzi do automatycznego generowania migracji bazy danych?

### Jak przeprowadzić migrację?

Teraz, kiedy wiemy czym jest migracja bazy danych, przyjrzymy się (przynajmniej koncepcyjnie), jak ją przeprowadzić. Niestety na tę chwilę nie ma żadnych standardów dotyczących sposobu zapisu migracji, co oznacza, że nie możemy wdać się tutaj w zbytnie szczegóły tworzenia. Innymi słowy, sposób przeprowadzenia migracji bazy danych w dużej mierze zależy od konkretnego narzędzia używanego do tego zadania.

Omówmy dwa najpopularniejsze sposoby przeprowadzania migracji bazy danych.

### Wykorzystanie frameworkowych narzędzi i bibliotek

Jeśli używamy popularnego języka (Ruby, PHP, Python, itp.) lub frameworka (Rails, Django, itp.), to istnieją dobrze udokumentowane biblioteki do migracji dostępne wewnątrz frameworka. Funkcjonalność taka często jest dostarczna jako część frameworka. Czasem można spotkać więcej niż jedną bibliotekę dostępną w danym środowisku.

Zazwyczaj w tym scenariuszu pliki migracji są generowane za pomocą wiersza polecenia. Czasami może być konieczne ręczne napisanie kodu niestandardowego w przypadku niektórych zmian, takich jak migracja danych, a nawet sposób cofnięcia samej zmiany. Poza tym, większość zmuszającej pracy wykonuje narzędzie do migracji.

### Wykorzystanie narzędzi zewnętrznych

W niektórych przypadkach będziemy zmuszeniu (lub chcieli) użyć oprogramowania zewnętrznego, takiego jak Flyway lub Liquibase, które działa na zasadzie kontroli wersji schematu bazy danych. Skorzystanie z tego typu rozwiązania ma dużo zalet. Unika się w ten sposób silnego związania z określonym frameworkiem, a zatem łatwiej będzie można zmienić język programowania – bez zbędnego martwienia się o porting migracji.

Niestety same narzędzia także wprowadzają swego rodzaju ograniczenia. Dla przykładu Flyway wspiera pracę z bazami danych jak Oracle, MySQL czy MariaDB, ale katalog dostępnych baz jest zamknięty. Samo skorzystanie z Flywaya wymaga użycia Javy, co także może okazać się kolejnym ograniczeniem technologicznym.

Dobra wiadomość jest taka, że Flyway i Liquibase są stosunkowo łatwe w użyciu. Umożliwiają one również generowanie migracji z poziomu wiersza polecenia i umożliwiają niestandardowe kodowanie w celu przechwycenia migracji schematu bazy danych.

### Które podejście jest najlepsze?

Zatem, które rozwiązanie jest najlepsze? Większość programistów wybiera opcję pierwszą.

Zwykle dzieje się tak dlatego, że programiści mają już preferowany język / strukturę, więc mówiąc kognitywnie, nie wydaje się im, że uczą się (jeszcze) innego oprogramowania. A uzależnienie (w przypadku wariantu pierwszego dotyczyły ram i języka) jest całkowicie dobrowolne i pożądane. Opcja druga wydaje się dobrym rozwiązaniem, jeśli zespół nie jest w pełni przywiązany do określonego języka lub struktury.

Niezależnie od tego, na co się zdecydujemy, unikajmy niepotrzebnej zmiany swojego wyboru w połowie rozwoju aplikacji. Migracja bazy danych nie jest obszarem, w którym zmiana narzędzi daje ogromne korzyści. Większość korzyści wynika z prostego skonfigurowania migracji bazy danych w pierwszej kolejności.

## Podsumowanie

Migracja bazy danych to jedno z tych niezbędnych narzędzi dla programisty, które znaczco ułatwia pracę. W przyszłości przewiduję, że migracja baz danych będzie ewoluować z najlepszej praktyki programistycznej do standardowej praktyki programistycznej. Mimo to ważne jest, aby pamiętać, jak

migracje baz danych mogą obrócić się przeciwko programiście, szczególnie w przypadku trudnych do odwrócenia zmian schematu. Miejmy absolutną pewność co do zmian, zanim wprowadzimy je w życie.

Dlatego, każdy zespół ma za zadanie przeanalizować dostępne rozwiązania we własnym stosie technologicznym i dobranie najbardziej dopasowanego do zastanej sytuacji. Należy wprowadzić do własnego projektu funkcjonalność migracji bazy danych.



Fundusze  
Europejskie  
Wiedza Edukacja Rozwój



Politechnika Wrocławskiego

Unia Europejska  
Europejski Fundusz Społeczny



„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

**Politechnika Wrocławskiego**

**Wydział Informatyki i Zarządzania**



Politechnika  
Wrocławskiego

# Zaawansowane Technologie Webowe

Laboratorium

Temat: Projekt grupowy – Konteneryzacja web aplikacji

Opracował: mgr inż. Piotr Jóźwiak

Data: lipiec 2020

Liczba godzin: 2 godziny

## Table of Contents

Wstęp.....	3
Wymagania .....	3
Przykładowa aplikacja webowa .....	4
Przygotowanie do konteneryzacji.....	5
Definicja kontenera.....	5
Budowanie obrazu kontenera.....	7
Uruchomienie kontenera z aplikacją .....	8
Jak debugować.....	9

### Wstęp

Podczas implementacji aplikacji często pojawia się problem z czynnością przekazania projektu innemu programiście. Tym innym programistą może być zarówno kolejny deweloper, tester czy po prostu wdrożeniowiec na produkcję (DevOps). Odkąd powstały narzędzia ułatwiające zarządzanie zależnościami w projekcie takie jak Composer, Maven czy też Ant, sam program przestał być głównym źródłem problemów. W to miejsce jednak weszły jednak inne problemy, jak środowisko uruchomieniowe oraz jego konfiguracja – system operacyjny, odpowiedni interpreter (w odpowiedniej wersji), baza danych czy też inne usługi. Oznacza to, iż przekazując nasz kod innemu programiście, nie zawsze możemy być pewni, iż będzie on działał w sposób identyczny jak na naszym komputerze. Czasem, mimo nawet dokładnej analizy porównawczej środowisk, brak jednej biblioteki może zdecydowanie opóźnić lub też uniemożliwić szybkie rozpoczęcie pracy w nowym projekcie.

Opisany powyżej problem jest szczególnie dotkliwy w sytuacji, gdy okazuje się, że aplikacja poprawnie uruchamia się w środowiskach deweloperskich i testowych, a z jakiś przyczyn nie chce wystartować w produkcji.

Rozwiążanie tego problemu może być konteneryzacja aplikacji webowych. Do tego celu wykorzystać można Docekra. Czym zatem jest Docker? Jest to technologia umożliwiająca zapakowanie aplikacji wraz ze środowiskiem uruchomieniowym w postać kontenera. Zaletą takiego podejścia jest to, że dystrybucja oprogramowania nie polega na przesłaniu jedynie źródeł/binariów aplikacji, ale całości skonfigurowanej wraz z systemem operacyjnym. Od tej pory konfiguracja będzie działać zawsze i to niezależnie od tego, gdzie zostanie uruchomiona. Kontener Dockera zapewni nam dostarczenie odpowiedniej wirtualizacji systemu operacyjnego oraz zapisanej konfiguracji. To podejście jest codziennością i podstawą działania aplikacji w oparciu o mikro serwisy funkcjonujące pod dużym obciążeniem. Scenariusz jest bardzo prosty. Gdy pojawia się zwiększyony ruch do aplikacji webowej, to *load balancer* może podjąć decyzję o uruchomieniu dodatkowej instancji aplikacji webowej celem zwiększenia chwilowej przepustowości. Na takim założeniu pracuje dzisiaj środowisko Kubernetes.

Celem dzisiejszych zajęć będzie przyjrzenie się sposobowi przygotowania kontenera z aplikacją webową nad którą pracujecie w grupach.

Oczywiście każdy zespół posiada różny stos technologiczny, ale w przykładzie, który zostanie omówiony, chodzi o zrozumienie pewnej idei oraz sposobu działania tego rozwiązania. Jeśli opanuje się te podstawowe rozumowanie koncepcji konteneryzacji web aplikacji, to wykonanie poniższych kroków dostosowane do wybranych technologii nie będzie stanowiło większego problemu.

### Wymagania

Do wykonania poniższych przykładów musimy zainstalować dodatkowe oprogramowanie Docker'a, którego proces instalacji został opisany na tej stronie: <https://docs.docker.com/engine/install/>. Aby upewnić się, że środowisko jest gotowe należy wykonać poniższe polecenie:

```
λ docker -v  
Docker version 19.03.12, build 48a66213fe
```

Jeśli w wyniku powyższego polecenia otrzymaliśmy wersję zainstalowanego Dockera, to oznacza, że środowisko jest gotowe do pracy z kontenerami.

### Przykładowa aplikacja webowa

Zanim zaczniemy opisywać proces tworzenia kontenera z aplikacją webową, musimy najpierw przygotować samą aplikację. W naszym przypadku posłużymy się aplikacją wygenerowaną przez generator Express, z tej racji, że nie proces tworzenia aplikacji jest dla nas istotny, lecz sam proces konteneryzacji.

Dlatego całość przykładu będzie opierać się o aplikację webową wygenerowaną w poniższy sposób.

Na początek instalujemy menadżerem pakietów express generatora:

```
npm install express-generator -g
```

Następnie przechodzimy do folderu, w którym chcemy utworzyć szkielet aplikacji webowej i wydajemy polecenie:

```
express mywebappname
```

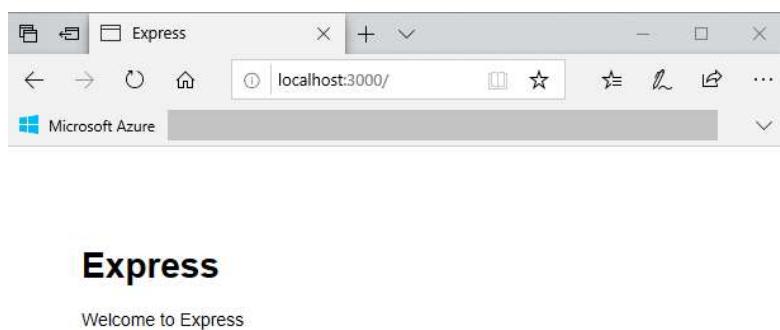
Powyższe polecenie utworzy aplikację webową o nazwie mywebappname w folderze o tej samej nazwie. Aby sprawdzić jej działanie musimy jeszcze zainstalować niezbędne zależności. Zatem wchodzimy do folderu z aplikacją i wydajemy poniższe polecenie:

```
cd mywebappname  
npm install
```

Czas uruchomić aplikację, zatem można wykorzystać poniższe polecenie:

```
SET DEBUG=mywebappname:* & npm start
```

Po chwili powinno pojawić się polecenie informujące o tym, że aplikacja jest dostępna pod adresem <http://localhost:3000>. Sprawdźmy to:



Zatem samo środowisko jest już gotowe.

## Przygotowanie do konteneryzacji

Zanim rozpoczęniemy proces konteneryzacji aplikacji, musimy zastanowić się nad składowymi naszego systemu. W najbardziej rozbudowanej wersji możemy wyróżnić trzy składowe systemu:

- Front-end
- Back-end
- Baza danych

Oczywiście to, jakie elementy w rzeczywistości wystąpią w danym systemie jest mocno powiązane z tym, co dany system implementuje oraz jakie technologie wykorzystuje. Bo przecież baza danych nie zawsze będzie potrzebna – np. w stronach statycznych jej nie ma. W innej sytuacji może być tak, że nie mamy rozdzielenia Front-endu od Back-endu. W każdym razie, zawsze na początku trzeba się zastanowić nad tym, co ma wejść w skład kontenera.

Dość oczywistym jest fakt, że chcielibyśmy zamknąć w kontenerze Front-end. Powstanie wówczas możliwość uruchomienia danego serwisu na wielu instancjach jednocześnie, a przez to możemy zacząć load balancować ruch.

Pojawia się pytanie, czy chcemy w kontenerze umieścić także bazę danych? To zależy od scenariusza użycia. Na pewno wstawienie bazy danych do kontenera ma swoje zalety, ale posiada także dwa duże ograniczenia. Pierwszym z nich jest problem z synchronizacją danych pomiędzyinstancjami bazy danych w przypadku uruchomienia aplikacji w środowisku balansowanym. Kolejnym problemem może być przenaszałość danych w momencie wystąpienia awarii. Zatem jeśli jest to prosta strona z WordPress to można się pokusić o umieszczenie jej w jednym kontenerze z bazą danych oraz zadbanie o odpowiednio częste kopie zapasowe danych. W innych przypadkach bazę danych warto ustawić centralnie poza kontenerem.

Co w takim razie z Back-end'em – jeśli takowy występuje? Tutaj także podejście jest różne. W najprostszym ujęciu można zapakować go wraz z Front-endem. Będzie to działało, ale może uniemożliwić nam uruchomienie balansera. Wszystko zależy będzie od tego, jak została zaimplementowana nasza aplikacja. W innych sytuacjach Back-end warto zapakować do oddzielnego kontenera.

W naszym przykładzie posiadamy tylko Front-end, bez bazy danych oraz oddzielnego Back-endu, ale nie to nie zmienia nic w samym sposobie budowania obrazu Dockera.

Ważne jest jedynie to, aby ustalić co będzie niezbędne do uruchomienia naszej aplikacji. Na pewno będzie to system Linux, a do tego niezbędne środowisko uruchomieniowe, czyli serwer aplikacji oraz odpowiednie biblioteki i interpretery. W przypadku Node.js możemy pójść na skróty i skorzystać z obrazu Dockerowego przygotowanego już z pełnym środowiskiem Node. Jest on dostępny tutaj: [https://hub.docker.com/\\_/node](https://hub.docker.com/_/node). Należy sprawdzić, czy w naszym przypadku także nie znajdzie się gotowy obraz w repozytorium Dockera.

## Definicja kontenera

Jesteśmy gotowi już na przygotowanie kontenera. W tym celu w folderze z naszą aplikacją webową tworzymy plik **Dockerfile**:

```
touch Dockerfile
```

**Dockerfile** jest plikiem z przepisem jak utworzyć obraz kontenera. To tutaj zapisujemy poszczególne polecenia jakie ma wykonać Docker, celem utworzenia obrazu.

Definicję rozpoczynamy od wskazania obrazu bazowego, na jakim ma zostać skompilowana nasza aplikacja. W naszym przypadku jest to już wcześniej wspomniany obraz Node z oficjalnego repozytorium dockera. W tym celu dodajemy wpis do pliku **Dockerfile**:

```
FROM node:10
```

Następnie tworzymy folder na źródła naszej aplikacji. W to miejsce wgramy wszystkie niezbędne pliki do uruchomienia serwisu:

```
# Create app directory
RUN mkdir -p /opt/app
```

Następnie zmieniamy folder roboczy na nowo utworzony folder:

```
WORKDIR /opt/app
```

W kolejnym kroku tworzymy użytkownika, pod którym będziemy uruchamiać naszą aplikację. Domyślnie wszystko uruchamia się na prawach użytkownika root.

```
# Add user app
RUN adduser --disabled-password app
```

Teraz przyszedł czas na wgranie plików źródłowych aplikacji do obrazu:

```
# Copy source files to WORKDIR
COPY . .
```

Musimy zmienić właściciela plików na użytkownika, który będzie uruchamiał naszą aplikację:

```
RUN chown -R app:app /opt/app
```

Teraz możemy określić, że dalszy kod będzie uruchamiał się z poziomu użytkownika app:

```
# Change execution to user app
USER app
```

Instalujemy wszystkie zależności na obraz poleceniem:

```
# Install all dependecies
RUN npm install
```

Dajemy dostęp do portu 3000 z kontenera:

```
# Allow access to port 3000 on container  
EXPOSE 3000
```

Na koniec startujemy serwer aplikacji **npm**:

```
# Run command - start npm server  
CMD [ "npm", "start" ]
```

Mamy już gotowy plik **Dockerfile**. Zanim skompilujemy obraz musimy zdefiniować jeszcze plik z opisem czego docker ma nie wgrywać do obrazu. W tym celu tworzymy plik o nazwie **.dockerignore** w tym samym miejscu co plik **Dockerfile** oraz wpisujemy do niego:

```
node_modules  
npm-debug.log
```

W ten oto sposób nie będziemy wgrywali naszych lokalnych kopii bibliotek, gdyż zainstalują się one same w trakcie uruchamiania kontenera.

## Budowanie obrazu kontenera

Jesteśmy już gotowi do zbudowania obrazu. W tym celu wydajemy polecenie będąc w folderze z plikiem Dockerfile:

```
docker build -t express-webapp .
```

Powyższe polecenie zbuduje obraz o nazwie express-webapp korzystając z pliku Dockerfile w aktualnej lokalizacji. Proces tworzenia kontenera wygląda następująco:

```
Sending build context to Docker daemon 46.08kB  
Step 1/10 : FROM node:10  
--> 1cc99e24ab2d  
Step 2/10 : RUN mkdir -p /opt/app  
--> Using cache  
--> 17d3154b4351  
Step 3/10 : WORKDIR /opt/app  
--> Using cache  
--> 87ae27888e60  
Step 4/10 : RUN adduser --disabled-password app  
--> Using cache  
--> 3f01f12bcb67  
Step 5/10 : COPY . .  
--> 04bf0e967713
```

```
Step 6/10 : RUN chown -R app:app /opt/app
--> Running in 9bc1c57ab5c5
Removing intermediate container 9bc1c57ab5c5
--> 77974c1ac2ad
Step 7/10 : USER app
--> Running in c7c16600779a
Removing intermediate container c7c16600779a
--> 2b1f191cf6db
Step 8/10 : RUN npm install
--> Running in b9beb40315bc
added 100 packages from 139 contributors and audited 101 packages in 3.042s
found 4 vulnerabilities (3 low, 1 critical)
  run `npm audit fix` to fix them, or `npm audit` for details
Removing intermediate container b9beb40315bc
--> be1dd0b8ac22
Step 9/10 : EXPOSE 3000
--> Running in b776c6659fa6
Removing intermediate container b776c6659fa6
--> 7bbba9b6e49d
Step 10/10 : CMD [ "npm", "start" ]
--> Running in 550f26a2745e
Removing intermediate container 550f26a2745e
--> f54d14db144a
Successfully built f54d14db144a
Successfully tagged express-webapp:latest
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All files and directories added to build context will have '-rwxr-xr-' permissions. It is recommended to double check and reset permissions for sensitive files and directories.
```

Jeśli wszystko się udało, to wydając poniższe polecenie zobaczymy nowo utworzony obraz kontenera:

```
docker images
```

C:\Prv\OneDrive\Dokumenty\PWr\Dydaktyka\ZTI\ProjektPower\Lab14\exampleWebApp\helloworld> docker images	REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
	express-webapp	latest	f54d14db144a	15 minutes ago	920MB

### Uruchomienie kontenera z aplikacją

Przyszedł czas na uruchomienie kontenera. W tym celu wykonujemy poniższe polecenie:

```
docker run -it -p 80:3000 express-webapp
```

Warto tutaj zwrócić uwagę na wartość przekazaną do parametru -p. Wartość 80:3000 oznacza mapowanie portu naszego hosta (w tym wypadku 80) na port kontenera (tutaj 3000). Jak pamiętamy w pliku Docker file, ustaliliśmy dostęp do tego portu odpowiednim poleceniem EXPOSE. Jak widać w trakcie uruchamiania mamy możliwość na dowolne przemapowanie portu wewnętrz kontenera na port pod jakim chcemy, aby aplikacja była widoczna na rzeczywistym hoście. Jak wiadomo port 80 jest domyślnym portem HTTP, toteż po wpisaniu adresu <http://localhost> powinna uruchomić się aplikacja uruchomiona z kontenera Docker.

### Jak debugować

A co jeśli coś nie działa? To najprościej zalogować się do kontenera poprzez uruchomienie shella z wnętrza. Uzyskujemy dostęp do wewnętrznego systemu operacyjnego. Dalej możemy wykorzystać to do sprawdzania czy wszystko poprawnie się zainstalowało, sprawdzić co zawierają logi aplikacji, itp.

Aby uruchomić wewnętrzny shell wystarczy uruchomić polecenie:

```
docker run -it express-webapp /bin/bash
```

Otrzymujemy znak zachęty:

```
app@def333eb3ad:/opt/app$
```



Fundusze  
Europejskie  
Wiedza Edukacja Rozwój



Politechnika Wrocławskiego

Unia Europejska  
Europejski Fundusz Społeczny



„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

**Politechnika Wrocławskiego**

**Wydział Informatyki i Zarządzania**



Politechnika  
Wrocławskiego

# Zaawansowane Technologie Webowe

Laboratorium

Temat: Projekt grupowy – Prezentacja wyników pracy grupowej

Opracował: mgr inż. Piotr Jóźwiak

Data: lipiec 2020

Liczba godzin: 2 godziny

## Table of Contents

Wstęp.....	3
------------	---

## Wstęp

Dotarliśmy do ostatnich zajęć w semestrze. Jak to zazwyczaj bywa jest to czas prezentowania rezultatów semestralnej pracy nad projektem. Dlatego podczas zajęć każda z grup projektowych powinna przedstawić rezultaty swojej pracy w postaci prezentacji osobistej oraz krótkiej prezentacji w Power Pointcie. W prezentacji należy przedstawić temat, jaki dana grupa realizowała wraz założeniami. Pokrótko należy przedstawić stos technologiczny wykorzystany do implementacji projektu. Każda z grup będzie miała na zaprezentowanie swojego projektu 15 minut. Dlatego należy tak dostosować ilość slajdów oraz czas prezentacji systemu, aby bezwzględnie zmieścić się w przewidzianym czasie.

Po zaprezentowaniu rezultatów praca podlega ocenie końcowej, która wraz z ocenami zdobytymi z zadań z pierwszej części semestru określa ocenę z przedmiotu.