

AKADEMIA TECHNICZNO-HUMANISTYCZNA

W BIELSKU - BIAŁEJ

WYDZIAŁ BUDOWY MASZYN I INFORMATYKI

PRACA DYPLOMOWA

INŻYNIERSKA nr

Grzegorz Kastelik

Nr albumu: 051414

Kierunek: Informatyka

Specjalność: Inżynieria Oprogramowania i Systemy Informatyczne

Temat pracy: Platforma umożliwiająca generowanie dokumentów i umów na podstawie
interaktywnych szablonów

Zakres pracy:

1. Analiza problemu i określenie wymagań funkcjonalnych i pozafunkcjonalnych
2. Wybór technologii i narzędzi realizacji
3. Opracowanie systemu tworzenia i generowania umów
4. Utworzenie platformy internetowej implementującej opracowany system
5. Konfigurowanie komponentów użytych w platformie
6. Edycja pisemna pracy

Kategoria rodzaju pracy:

Katedra Informatyki i Automatyki / Zakład Informatyki

Promotor: **dr. Łukasz Więclaw**

.....

Podpis i pieczęć
Kierownika Katedry

Bielsko-Biała, rok akademicki

Spis treści

Wstęp	5
Cel i zakres pracy	7
1 Specyfikacja	8
1.1 Umowy	8
1.2 Platforma internetowa	9
1.3 Słownik pojęć	10
1.4 Wymagania biznesowe	11
1.5 Wymagania funkcjonalne	12
1.6 Wymagania niefunkcjonalne	13
1.7 Reguły biznesowe	14
2 Użyte technologie, narzędzia i usługi	15
2.1 Języki	15
2.1.1 JSON	15
2.1.2 SCSS	15
2.1.3 HTML	16
2.1.4 JavaScript	16
2.1.5 PHP 7.4	16

2.2	Główne frameworki i biblioteki	17
2.2.1	Laravel	17
2.2.2	Vue	18
2.2.3	VuetifyJs	18
2.2.4	DOMPDF	20
2.3	Narzędzia	21
2.3.1	PhpStorm	21
2.3.2	Git	21
2.3.3	Docker	22
2.4	Usługi	23
2.4.1	VPS DigitalOcean	23
2.4.2	Domena generatorumowy.pl	24
2.4.3	SSL	25
2.4.4	Forge	25
2.4.5	Jenkins	26
3	Projekt systemu tworzenia umów	28
3.1	Opis działania systemu	28
3.2	Schemat bazodanowy	28
4	Opis platformy implementującej system	29
4.1	Platforma	29
4.1.1	Implementacja	29
4.1.2	Instrukcja użytkowania	29
4.2	API	29
4.2.1	Implementacja	29

4.2.2	Instrukcja użytkowania	29
4.2.3	Testy	29
5	Podsumowanie oraz wnioski	30
	Spis rysunków	31
	Bibliografia	32

Wstęp

Rozwój technologii jaki obserwujemy na przestrzeni ostatnich lat, znacząco pozwala nam na stwierdzenie, iż pełna cyfryzacja usług to kierunek dla każdej nowoczesnej firmy, instytucji, czy też państw. Społeczeństwo, szczególnie młodsze, bardzo szybko adaptuje nowoczesne rozwiązania w dostarczaniu usług. Można powiedzieć, że kierunek cyfryzacji jest wręcz wymuszony, aby dana firma mogła przetrwać i sprawnie funkcjonować na runku.

Ludzie nauczyli się, iż taka forma świadczenia usług niesie za sobą szereg korzyści. Jest często tańsza, gdyż nie wymaga posiadania dodatkowych pracowników do zarządzania. Szybsza i wygodniejsza, gdyż pozwala na skorzystanie bez wychodzenia z domu. Może z niej skorzystać wiele osób w jednym czasie. Nie trzeba do tego stać w kolejkach znacząco przyczynia się do oszczędzania czasu jaki trzeba poświęcić.

Przedsiębiorstwa są nie tylko zmuszone do cyfryzacji, ale jednocześnie zachęcane. Pomimo kosztów początkowych mają one szansę na efektywne skalowanie swoich usług i dostarczenie ich większej ilości użytkowników niż dotychczas.

Taka sytuacja nie ominęła również prawników. Zmotywowani automatyzacją są gotowi i chętni część swojej wiedzy przenieść w świat wirtualny. Duża część pracy prawniczej jest schematyczna, szczególnie tworzenie umów dla klientów. Wykorzystując ten fakt, można z łatwością znaleźć zastosowania dla cyfryzacji w owym środowisku. Dzięki takiemu zabiegowi otwierają się na wielu nowych klientów, całkowicie globalnie, ponieważ jedyne co potrzebuje klient to połączenie z Internetem i przeglądarka. Usługi takie mogą świadczyć dużo taniej, gdyż nie skupiają swojego czasu na schematycznych działaniach i mogą skupić się na najważniejszych aspektach swojej pracy.

Platforma do tworzenia umów ma więc bardzo praktyczne zastosowanie i jest dla praw-

ników niezbędna, jeśli chcą wejść w świat usług wirtualnych. Jest ona stworzona, aby maksymalnie ułatwić i zautomatyzować ich pracę bez konieczności pomocy z zewnątrz, oraz specjalistycznej wiedzy z zakresu programowania i technologii internetowych.

Cel i zakres pracy

Celem pracy było zaprojektowanie systemu umożliwiającego tworzenia umów prawniczych zaimplementowanego jako platforma Internetowa.

System tworzenia umów w swojej podstawowej formie ma za zadanie dostarczyć platformę Internetową oraz API umożliwiającą pełną obsługę wyłączając z tego proce budowy szablonu umowy za pomocą bloków. Jego zadaniem jest umożliwić wszystkie działania związane z platformą, szczególnie dostarczać wszelkie informacje potrzebne do dynamicznego generatora szablonów tworzonych jako osobna część w platformie ze względu na poziom skomplikowania i potrzebę dopracowania elementów pod finalnych użytkowników.

W zakres pracy wchodzi więc utworzenie API które umożliwi pełne zarządzanie całym systemem oraz platformę która będzie mogła korzystać z API w celu wykonywania akcji związanych z tworzeniem umów, wypełnianiem, oraz zarządzaniem całym systemem.

Rozdział 1

Specyfikacja

1.1 Umowy

Na początek warto przytoczyć podstawową definicję pojęcia umowy, jest to:

„Wzajemne uzgodnienie dwóch lub więcej stron mająca na celu obopólne dobro oraz określająca wzajemne obowiązki oraz prawa stron.” [1]

Umowy zawierać można na wiele sposobów, każdy kraj dodaje swoje obostrzenia jednak w większości możliwe jest utworzenie umowy w wersji pisemnej z podpisem i egzemplarzem dla każdej ze stron.

Każda umowa to odrębna sytuacja jednak wiele z nich podchodzi pod schemat, dzięki któremu możemy zautomatyzować czynności. Bazując na tym platforma umożliwia tworzenie takich oto schematów, aby mogły być w automatyczny sposób dostarczane do wielu klientów i przez wielu na raz wypełniane.

Wypełnienie wiąże się z procesem z góry określonym przez który musi przejść użytkownik, aby uzyskać umowę, którą będzie musiał jedynie wypełnić. Proces ten w większości przypadków to zebranie takich danych jakie potrzeba, aby idealnie przygotować umowę pod daną osobę.

Umowy występują jednak pod różnymi poziomami skomplikowania. Część to proste jednostronne umowy z prostym systemem. Kolejne to umowy kilkunastu stronicowe precyzyjnie określające wszystkie możliwe akcje. Wiąże się to z dużą ilością danych jakie trzeba zebrać, przetworzyć i w zgodzie z nimi zbudować umowę końcową.

1.2 Platforma internetowa

Platforma internetowa to dzisiaj jeden z najczęściej spotykanych typów aplikacji. Jej wszechstronność i dostępność znacząco odbiegają od innych znanych nam sposobów dostarczania i tworzenia programów.

Aplikacje internetowe, platformy, dają niezmierne możliwości. Jedną z najważniejszych jest wcześniej wspomniana dostępność. Aplikacja taka jest hostowana przez serwer i jedynie co potrzebne, aby skorzystać z takiego programu to przeglądarka internetowa, oraz stały dostęp do Internetu chociaż i to w dzisiejszych czasach niekoniecznie.

Kolejną zaletą aplikacji hostowanych jest fakt, iż użytkownik zawsze korzysta z aktualnej wersji oprogramowani, bez potrzeby ręcznej aktualizacji. Zaletą jest również fakt, iż aplikacja taka dostępna jest z każdego urządzenia, czy mówimy tutaj o komputerze stacjonarnym, smartphonie, tablecie. Każde z tych urządzeń będzie w stanie uruchomić aplikację i korzystać z niej.

Komunikacja odbywa się poprzez sieć. Użytkownik nie potrzebuje zapisywać danych u siebie co wiąże się z następną zaletą, a mianowicie – jeśli uszkodzone zostanie urządzenie, dane nie przepadną i dalej będą dostępne.

Aktualnie aplikacje Internetowe są najpopularniejsze zarówno, jeśli chodzi o developerów jak i o użytkowników. Bezproblemowo widać to w dzisiejszych czasach, gdy na topie są takie aplikacje jak Facebook, Instagram, Gmail, Twitter

1.3 Słownik pojęć

Szablon umowy (ang. Contract)

Jest to encja opisująca precyzyjnie jak ma wyglądać formularz umowy dla użytkownika, jak ma wyglądać wygenerowana umowa końcowa. Opisuje wszystkie zależności oraz uwarunkowania.

Część szablonu (ang. Contract part)

Jest to fragment szablonu umowy, przechowywany do ponownego użycia. Najczęściej używany do elementów które powtarzają się w wielu umowach.

Formularz umowy (ang. Contract form)

Jest to formularz jaki musi wypełnić użytkownik, jeśli chce uzyskać wygenerowaną umowę. Wygląd i zawartość formularza definiują szablon umowy.

Panel

Cześć platformy odpowiedzialny za nawigację użytkownika po opcjach systemu. Panel pełni również opcję informacyjną, przedstawiając szereg najważniejszych danych dla użytkownika w zależności od posiadanych uprawnień.

Uprawnienie (ang. Permission)

Jest to encja precyzująca dostęp do danych akcji w systemie na przykład istnieje uprawnienie pod nazwą manage-users, które określa czy dany użytkownik może zarządzać użytkownikami w systemie.

Konto (ang. Account)

Jedno konto określa jednego użytkownika w systemie. Konto posiada określoną rolę i może wykonywać określone akcje.

Rola (ang. Role)

Jest to nazwany zbiór uprawnień w systemie.

Użytkownik (ang. User)

Podstawowy użytkownik, w systemie którego uprawnienia pozwalają jedynie na wypełnianie umów i używanie funkcji z tym związanych.

Administrator

Osoba posiadające pełne uprawnienia w systemie.

Moduł

Zbiór kodu po stronie backendowej jak i frontendowej rozszerzający podstawowe funkcjonalności systemu

Generowanie umowy

Zakolejkowany proces, podczas którego dane z formularza umowy użyte zostają w procesie budowy umowy na podstawie szablonu umowy. Finalnym efektem działania generatora jest plik PDF będącym gotową umową. Generator po wygenerowaniu umowy oddaje ją do modułu dostarczania.

API (ang. Application Programming Interface)

„Interfejs programowania aplikacji, czyli określony zestaw reguł, dzięki którym programy komputerowe mogą między sobą udostępniać dane – porozumiewać się ze sobą. Dzięki API możliwe jest korzystanie z funkcjonalności innych aplikacji w naszej aplikacji. API dostarcza specyfikacje podprogramów, struktur danych protokołów komunikacyjnych i klas obiektów.”ⁱⁱ

1.4 Wymagania biznesowe

WB1.

Platforma musi być dostępna przez przeglądarkę.

WB2.

Platforma musi umożliwiać zarządzanie schematami umów

WB3.

Platforma musi umożliwiać autoryzację użytkowników

WB4.

Platforma musi umożliwiać zarządzanie użytkownikami

WB5.

Platforma musi umożliwiać zarządzanie rolami i powiązanymi z nimi rolami

WB6.

Musi być dostępne API w celu późniejszych integracji

WB7.

Najważniejsze encje w systemie muszą mieć zaimplementowany mechanizm tak zwanego usuwania łagodnego (ang. soft delete). Który ma zapewnić dostęp do danych nawet po ich usunięciu.

1.5 Wymagania funkcjonalne

W1.

Użytkownik musi mieć możliwość rejestracji. Zarejestrować mogą się jedynie osoby, które podadzą dane oraz zaakceptują regulamin i politykę RODO.

W2.

Użytkownik musi mieć możliwość logowania za pomocą adresu email oraz hasła.

W3.

Hasło użytkownika musi być bezpieczne przechowywane w bazie danych za pomocą nowoczesnej metody szyfrowania.

W4.

Użytkownik musi mieć możliwość zmiany imienia, nazwiska oraz hasła.

W5.

Użytkownik musi mieć możliwość wypełniania formularzy umów do których ma dostęp.

W6.

Użytkownik musi mieć dostęp do wygenerowanych umów i mieć możliwość ich późniejszego pobrania.

W7.

Administrator musi mieć możliwość do zarządzania użytkownikami. Zarządzanie oznacza możliwość usuwania, dodawania nowych, edycji danych użytkownika w tym roli jakie są do niego przypisane.

W8.

Administrator musi mieć możliwość tworzenia ról dla użytkowników oraz przyporządkowywania im uprawnień istniejących w systemie.

W9.

Administrator musi mieć dostęp do tworzenia szablonów umów ich edycji oraz usuwania.

W10.

Użytkownik musi mieć możliwość filtrowania dostępnych umów po nazwie oraz po kategorii przypisanej do danej umowy.

1.6 Wymagania niefunkcjonalne

W1.

Aplikacja powinna działać poprawnie na przeglądarkach:

- Edge – v81
- Mozilla FireFox – v72
- Google Chrome – v79

W2.

Aplikacja nie powinna być związana ściśle z jednym systemem operacyjnym

W3.

Aplikacja powinna posiadać system kolejkowania. Który w późniejszym rozwoju i rozszerzeniu działalności powinien być zaimplementowany dla wysyłania maili, oraz renderowania umów.

1.7 Reguły biznesowe

RB1.

Każdy użytkownik po zarejestrowaniu przez formularz powinien uzyskać rolę USER określającą minimalne uprawnienia w systemie.

RB2.

Usunięcie użytkownika, schematu umowy lub wypełnionego formularza od użytkownika powoduje usunięcie łagodne (umożliwiające proces przywrócenia w razie potrzeby)

Rozdział 2

Użyte technologie, narzędzia i usługi

Do budowy platformy użyty został szereg technologii umożliwiających tworzenie dynamicznych stron internetowych oraz pełnoprawnego API. Jednocześnie użyte zostało wiele narzędzi umożliwiających codzienną pracę w zespole, szybką publikację bez utraty działania strony, testowanie, proces ogólnie zwany continuous integrations.

2.1 Języki

2.1.1 JSON

Język, który bardzo często wykorzystywany jest jako język opisujący konfigurację. Tak też wykorzystywany jest w systemie.

JSON (ang. JavaScript Object Notation) jest to format wymiany danych. Prosty do czytania i zrozumienia przez ludzi. Znacząco czytelniejszy niż jego przestarzały konkurent XML

2.1.2 SCSS

Język ten rozszerza możliwości CSS o szereg funkcjonalności ułatwiających pracę ze stylami. Odpowiada ze opisanie wyglądu szablonu napisanego za pomocą HTML

SCSS jest w pełni kompatybilny z css dzięki czemu możemy bez problemu zmienić rozszerzenia plików .css na .scss i skompilować taki kod.

Kompilowaniem do .css zajmuje się tutaj SASS (ang. Syntactically Awesome Stylesheets)

2.1.3 HTML

Jako główny język tworzący szablony stron. Opisuje on strukturę, która następnie obudowywana jest stylami i dynamicznie zarządzana dzięki skryptom. Jest to znacznikowy język programowania.

Najnowszą wersją jest HTML5 będący jednocześnie standardem. Duży nacisk kładzie się w tej odmianie na semantykę. Dodane zostały również nowe typy pól, atrybuty oraz tagi.

2.1.4 JavaScript

Język ten wykorzystywany jest jako fundament dla nowoczesnych frameworków frontendowych. Pozwala pisać dynamiczne strony internetowe. Jest to skryptowy język programowania. Ma możliwość działania po stronie klienta jak i serwera. Najczęściej jednak używany jest po stronie klienta w przeglądarkach internetowych.

Dzięki kompilatorom takim jak Webpack, Babel możliwe jest pisanie w najnowszej wersji, standardzie ECMAScript. Kod następnie kompilowany jest to starszych wersji, aby mógł działać na większości dzisiejszych przeglądarek. Niestety przeglądarki wciąż pozostają w tyle z implementacją nowych funkcjonalności i takie działanie jest nieuniknione, jeśli chcemy zachowywać standardy i funkcjonalność.

Najnowsza wersja to ECMAScript 2018, przeglądarki natomiast zatrzymały się na obsłudze wersji z przed 2 lat i to jedynie częściowo.

2.1.5 PHP 7.4

Główna część systemu zaimplementowana jest przy użyciu języka PHP. Dzięki najnowszej wersji można liczyć na bardzo dobrą wydajność (szczególnie dzięki rozszerzonemu wsparciu

mechanizmu ładowania wstępnego) oraz szereg rozbudowanych funkcjonalności.

PHP jest językiem przystosowanym do tworzenia stron internetowych, portali, API jak i zaawansowanych aplikacji webowych. Jednocześnie jest w tym segmencie liderem, znacząco zostawiając w tyle konkurencję. Stale utrzymuje pod sobą około 79% rynku (rys. 1).

	2019 1 Jan	2019 1 Feb	2019 1 Mar	2019 1 Apr	2019 1 May	2019 1 Jun	2019 1 Jul	2019 1 Aug	2019 1 Sep	2019 1 Oct	2019 1 Nov	2019 1 Dec	2020 1 Jan	2020 18 Jan
PHP	78.9%	78.9%	78.9%	79.0%	79.1%	79.0%	79.0%	79.1%	79.1%	79.0%	79.0%	78.9%	78.9%	79.2%
ASP.NET	11.8%	11.6%	11.5%	11.4%	11.3%	11.2%	11.1%	11.0%	10.9%	10.9%	10.8%	10.7%	10.6%	10.6%
Java	4.0%	4.0%	4.0%	4.0%	4.0%	4.0%	4.0%	3.9%	3.8%	3.8%	3.8%	3.8%	3.7%	3.4%
Ruby	2.4%	2.4%	2.5%	2.5%	2.5%	2.5%	2.6%	2.6%	2.7%	2.7%	2.8%	2.9%	3.0%	3.0%
static files	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.0%	2.0%	2.0%	2.0%	1.9%	1.8%	1.8%
Scala	1.2%	1.3%	1.3%	1.3%	1.3%	1.3%	1.4%	1.4%	1.5%	1.5%	1.6%	1.6%	1.6%	1.6%
Python	1.1%	1.1%	1.1%	1.1%	1.1%	1.1%	1.1%	1.2%	1.2%	1.2%	1.2%	1.3%	1.3%	1.3%
JavaScript	0.7%	0.7%	0.7%	0.7%	0.7%	0.7%	0.7%	0.7%	0.8%	0.8%	0.8%	0.8%	0.8%	0.8%
ColdFusion	0.5%	0.5%	0.5%	0.5%	0.5%	0.5%	0.5%	0.5%	0.5%	0.5%	0.5%	0.5%	0.5%	0.5%
Perl	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.2%	0.2%
Erlang	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%

Rysunek 1: Przedstawienie użycia języków po stronie serwera na podstawie strony https://w3techs.com/technologies/history_overview/programming_language na dzień 12.01.2020

2.2 Główne frameworki i biblioteki

2.2.1 Laravel

Jest to najpopularniejszy framework do tworzenia aplikacji internetowych w języku PHP. Zostawił w tyle większość konkurencji, umożliwiając bezkompromisową pracę. Jest zbudowany na fundamencie wzorca MVC (ang. Model View Controller). Słynie z efektywnej i bardzo eleganckiej składni. Jedną z głównych cech jest czystość kodu i struktury jaką oddaje nam do użytku Laravel. Pozostałe cechy które warto wymienić to:

- Prosty i szybki system routingu
- Gotowy kontener wstrzykiwania zależności z ogromem funkcji dodatkowych i konfiguracyjnych

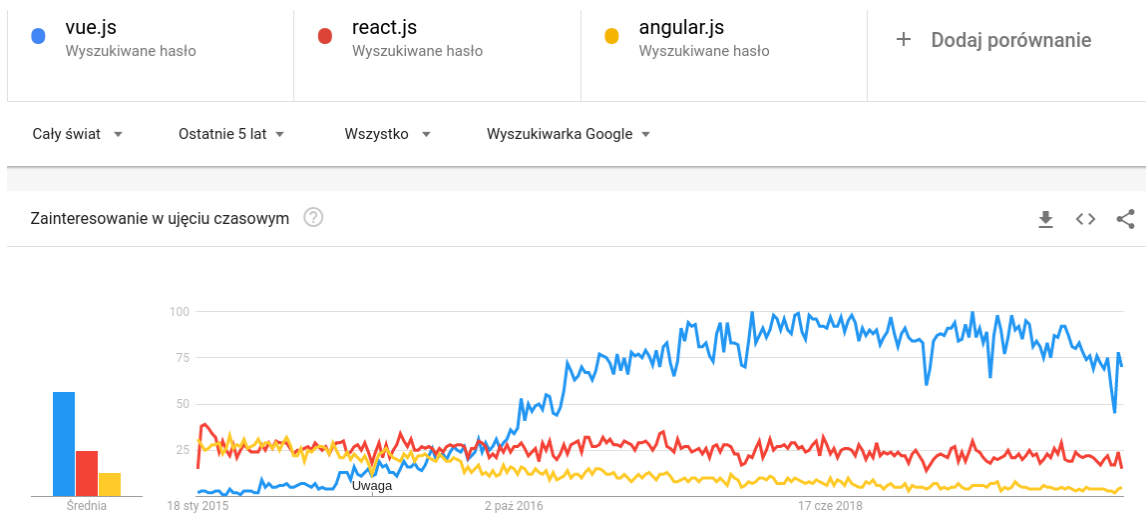
- Intuicyjny system ORM dla praktycznie wszystkich najnowszych baz danych
- Precyzyjny system migracji do bazy danych
- Potężny system do obsługi wypełniania bazy danych
- Zadania w tle
- Obsługa i możliwość tworzenia komend
- Mechanizm kolejkowania zadań
- System mailingowy
- System notyfikacji
- Implementacja wielu wzorców projektowych do konkretnych zadań
- System eventowy
- Broadcasting

2.2.2 Vue

Jest to biblioteka oparta o język JavaScript. Pozwala na tworzenie złożonych aplikacji webowych składających się z szerokiej listy komponentów. Oparta jest o architekturę MVVM(ModelView View Model). Znaczącym plusem jaki posiada ta biblioteka jest jej intuicyjność i łatwość w dostarczeniu gotowych rozwiązań. Stale zyskuje na popularności poszerzając listę swoich zwolenników (rys. 2), oraz ilość bibliotek rozszerzających.

2.2.3 VuetifyJs

Vuetify jest frameworkiem na licencji MIT służącym do budowy interfejsów użytkownika w aplikacjach webowych. Projekt ten jest wspierany przez wolontariuszy i sponsorów z całej społeczności Vue. Posiada spójny cykl aktualizacji, oraz możliwość wykupienia wsparcia dla



Rysunek 2: Popularność 3 największy frameworków JavaScript na podstawie Google Trends







użytku biznesowego. Co tydzień wypuszczany jest również pakiet poprawek złaszanych przez społeczność

Framework ten obsługuje wszystkie główne przeglądarki. Przeglądarki starsze również będą działać jednak wymagają zastosowania pewnych kroków które można bez problemu znaleźć w dokumentacji.

Vuetify dostarcza również swoje płatne rozwiązania takie jak szablony lub też wcześniej wspomniane wsparcie.

Główną zaletą jest łatwość z jaką tworzy się interfejs przy pomocy gotowych komponentów. Ich mnogość pozwala na tworzenie dowolnych konfiguracji prze potrzeby tworzenia własnych rozwiązań. Dodatkowo dokumentacja jest przejrzysta i intuicyjna. Zarówno osoba początkująca jak i zaawansowana doceni plusy użytkowania owego frameworku.

Jest to obecnie jedno z lepszych rozwiązań na rynku co poświadcza ilość funkcjonalności dostarczanych przez Vuetify (rys. 3)

Vue Framework Comparison 2020						
Features	 Vuetify	 Vue Bootstrap	 Buefy	 Element UI	 Vue Material	 Quasar Framework
Total components*	82+	45+	41+	86+	56+	81+
Full Accessibility and Section 508 support	✓	✓	✓			
Server Side Rendering support	✓	✓	✓	✓	✓	✓
Long-term Support	✓					
Release cadence**	Weekly	Bi-Weekly	Bi-Monthly	Bi-Weekly	Yearly	Bi-Weekly
Treeshaking	Automatic	Manual	Manual	Manual	Manual	Automatic
RTL support	✓	✓		✓		✓
Premium Themes	✓	✓			✓	
Business and Enterprise support	✓					

Rysunek 3: Porównanie możliwości najbardziej znanych frameworków do Vue

2.2.4 DOMPDF

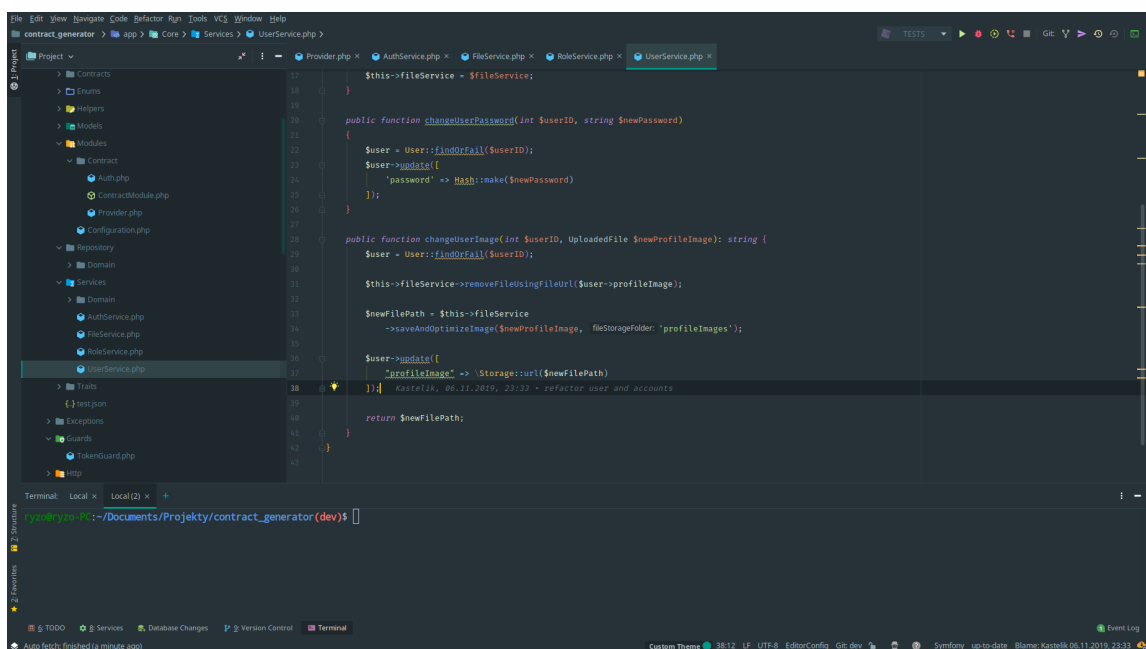
Jest to darmowa biblioteka która umożliwia konwertowanie języka HTML do plików PDF. Filarem DOMPDF jest jego zgodność ze standardem HTML oraz silnik renderujący utworzony w języku PHP. Biblioteka z łatwością otworzy zewnętrzne arkusze stylów, znaczniki inline oraz atrybuty poszczególnych elementów. Render to pliku PDF możliwy jest dzięki bibliotece PDFLib której posiadanie jest wymogiem do działania. Główne funkcje jakie dostarcza nam biblioteka to:

- obsługa standardu CSS 2.1 oraz częściowa CSS3
- obsługa większości tagów prezentacyjnych ze standardu HTML 4.0
- obsługa zewnętrznych arkuszy stylów, nawet poprzez dociąganie z serwerów FTP i innych
- obsługa rozbudowanych tabel
- obsługa większości popularnych formatów zdjęć
- brak zewnętrznych zależności
- obsługa języka PHP wewnątrz

2.3 Narzędzia

2.3.1 PhpStorm

W pełni funkcjonalne, wieloplatformowe PHP IDE zbudowane na bazie IntelliJ IDEA firmy JetBrains. Został on zaprojektowany specjalnie w celu ułatwienia rozwoju aplikacji internetowych napisanych w PHP. Dostarcza wszystkie narzędzia i funkcje dla PHP i obsługuje technologie frontendowe. Znakomicie radzi sobie z dużymi projektami, podpowiadaniem składni, oraz debugowaniem. Na dzień dzisiejszy środowisko to jest bezkonkurencyjne. Jest to rozwiązanie płatne jednak z wieloma zniżkami czy też darmowymi wersjami dla studentów.



Rysunek 4: Wygląd IDE w projekcie

2.3.2 Git

Git jest bardzo popularnym, potężnym i wydajnym systemem kontroli wersji Open Source, który śledzi treści takie jak pliki i katalogi. Jego głównym celem jest zarządzanie projektem lub zestawem plików w miarę ich zmian w czasie.

System kontroli wersji, czyli VCS, jak to się powszechnie określa, jest systemem, który śledzi historię zmian, gdy ludzie i zespoły współpracują ze sobą nad projektami. W miarę jak projekt ewoluuje, zespoły mają możliwość przeprowadzania testów, usuwania błędów i tworzenia nowego kodu z pewnością, że każda wersja może zostać odzyskana w dowolnym momencie.



Rysunek 5: Przykładowy układ na zasadzie git flow ze strony <https://barloblog.wordpress.com>

2.3.3 Docker

Docker to narzędzie zaprojektowane w celu ułatwienia tworzenia, wdrażania i uruchamiania aplikacji przy użyciu kontenerów. Kontenery pozwalają programiście na spakowanie aplikacji ze wszystkimi częściami, których potrzebuje, takimi jak biblioteki i inne zależności, i wysłanie jej w jednym pakiecie. W ten sposób, dzięki kontenerowi, deweloper może mieć pewność, że aplikacja będzie działać na każdej innej maszynie, bez względu na wszelkie niestandardowe ustawienia, które maszyna może mieć, a które mogą się różnić od tych, które są używane do pisania i testowania kodu, a także od serwera produkcyjnego.

W pewnym sensie, Docker jest trochę jak wirtualna maszyna. Ale w przeciwieństwie do maszyny wirtualnej, zamiast tworzyć cały wirtualny system operacyjny, Docker pozwala aplikacjom korzystać z tego samego jądra Linuksa co system, na którym są uruchomione i wymaga tylko dostarczenia aplikacji z rzeczami, które nie są jeszcze uruchomione na komputerze hosta. Daje to znaczący wzrost wydajności i zmniejsza rozmiar aplikacji.

2.4 Usługi

2.4.1 VPS DigitalOcean

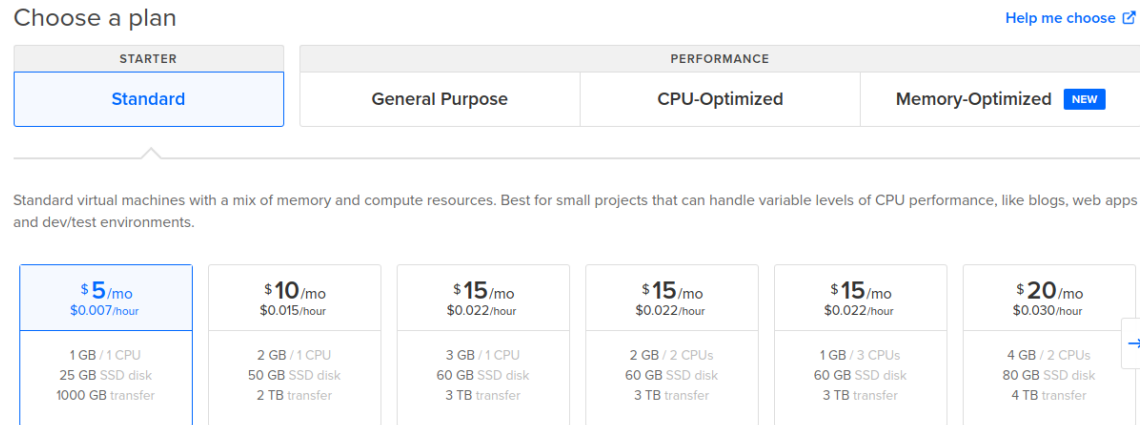
DigitalOcean jest znanym dostawcą chmury obliczeniowej który oferuje infrastrukturę zgodną z zasadą IaaS (ang. Infrastructure as a Service) dla twórców oprogramowania. Jest bardzo popularny i znakomicie konkuruje z AWS (ang. Amazon Web Services) czy też Google Compute Engine

W celu uruchomienia środowiska programiści uruchamiają instancję prywatnej maszyny wirtualnej (VM) która w DigitalOcean nosi nazwę droplet(kropelka). Użytkownik może wybrać wielkość kropelki, region geograficzny i centrum danych w którym będzie ona działać oraz system operacyjny z rodziny Linux. Można jednocześnie użyć swoich obrazów do postawienia gotowego środowiska uruchomieniowego.

Aktualnie dostawca oferuje czternaście rozmiarów kropli. Najmniejszy rozmiar zaczyna się od 1GB pamięci RAM z 1 procesorem i 25GB pamięci SSD, taki układ to koszt 5\$ miesięcznie (rys. 6). Wraz ze wzrostem specyfikacji rośnie również cena, aby maksymalnie osiągnąć 960\$ za 192GB pamięci RAM z 32 procesorami i 12TB pamięci SSD.

Deweloperzy używają systemu DigitalOcean do zarządzania i monitorowania kropli za pomocą panelu sterowania i API open source. Panel kontrolny umożliwia deweloperom skalowanie i przebudowę kropli na podstawie zmian w obciążeniu pracą oraz wykonywanie kopii zapasowych i przekierowywanie ruchu sieciowego pomiędzy kropelkami.

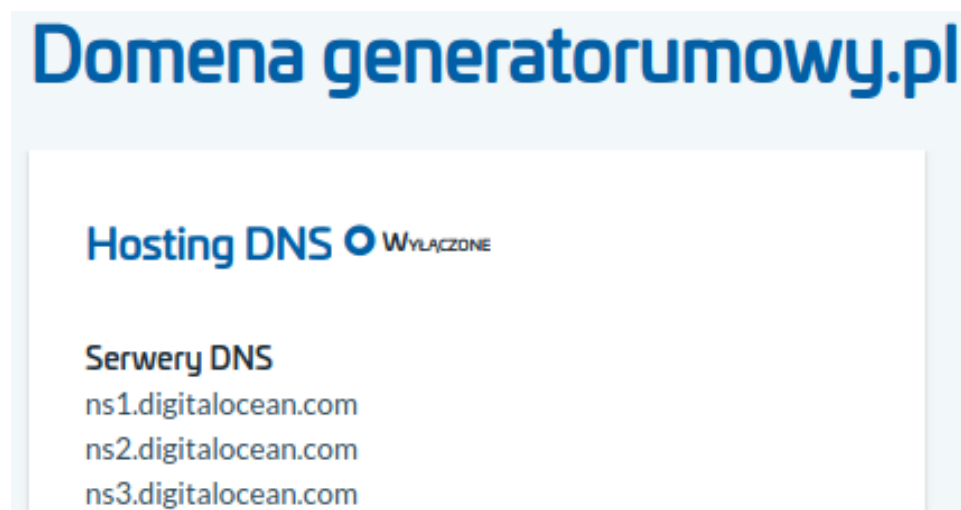
DigitalOcean został założony w 2011 roku przez Bena Ureckiego, Moiseya Ureckiego, Mitcha Wainera, Jeffa Carra i Aleca Hartmana. Siedziba firmy znajduje się w Nowym Jorku.



Rysunek 6: Wybór możliwych konfiguracji na stronie <https://cloud.digitalocean.com>

2.4.2 Domena generatorumowy.pl

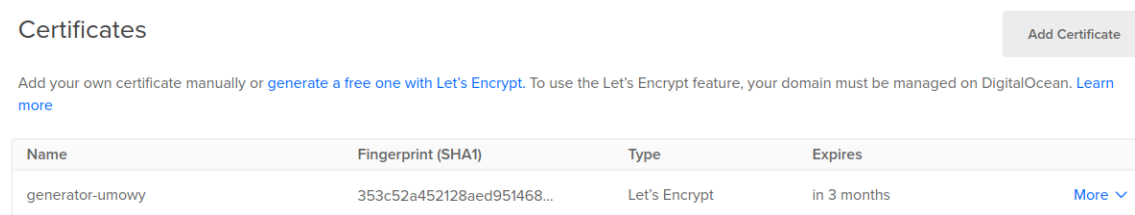
W celu utworzenia projektu koniecznym było wystawienie go dla osób trzecich, aby umożliwić testowanie przez osoby niezwiązane z pracą. Domena jest to podstawowy inedytfikator tożsamości w internecie. Jest to element adresu internetowego aplikacji webowych. Każdy może zakupić wolną domenę dla swoich celów. Należy w tym celu znaleźć odpowiadającego nam dostawcę i wykupić domenę o podanej nazwie. Po tym procesie możemy bez problemu przekierować ją na żądany hosting lub też maszynę serwerową (rys. 7).



Rysunek 7: Przykładowa konfiguracja u dostawy az.pl

2.4.3 SSL

Certyfikaty SSL to niewielkie pliki danych, które w sposób cyfrowy wiążą klucz kryptograficzny z danymi organizacji. Po zainstalowaniu na serwerze WWW, aktywuje on kłódkę i protokół https oraz umożliwia bezpieczne połączenia z serwera WWW do przeglądarki. Zazwyczaj SSL jest używany do zabezpieczania transakcji kartami kredytowymi, przesyłania danych i logowania, a ostatnio staje się normą przy zabezpieczaniu przeglądania stron serwisów społecznościowych. Do naszych celów certyfikat został wygenerowany dzięki darmowemu urzędowi certyfikacji Let's Encrypt (rys. 8)



The screenshot shows the 'Certificates' section of the DigitalOcean dashboard. At the top right is a button labeled 'Add Certificate'. Below it is a text instruction: 'Add your own certificate manually or [generate a free one with Let's Encrypt](#). To use the Let's Encrypt feature, your domain must be managed on DigitalOcean. [Learn more](#)'. Below this is a table with the following data:

Name	Fingerprint (SHA1)	Type	Expires	
generator-umowy	353c52a452128aed951468...	Let's Encrypt	in 3 months	More ▾





Rysunek 8: Przykład dodania certyfikatu na stronie <https://cloud.digitalocean.com>


2.4.4 Forge

Laravel Forge jest narzędziem do wdrażania i konfigurowania aplikacji internetowych. Został opracowany przez twórców szkieletu Laravela, ale może być użyty do zautomatyzowania wdrożenia dowolnej aplikacji internetowej, która korzysta z serwera PHP.

Tworzenie w pełni funkcjonalnego serwera WWW zazwyczaj wymaga instalacji wielu komponentów, takich jak NGINX, MySQL i PHP. Laravel Forge automatyzuje wszystkie niezbędne kroki instalacji i konfiguracji, co pozwala na szybkie uruchomienie witryny.

Po utworzeniu serwera, wdrażanie aktualizacji staje się przejrzyste i bezbolesne. Ponadto, można łatwo zarządzać konfiguracją swojej strony internetowej za pomocą interfejsu WWW. Wreszcie, Forge automatycznie udostępnia zaawansowane funkcje bezpieczeństwa, takie jak darmowe certyfikaty SSL (poprzez Let's Encrypt) i automatyczną konfigurację firewalla.

Active Servers All Servers ▾				
Name	IP Address	PHP Version	Status	Connection
 red-mist	46.101.231.42	7.3	 Active	 Successful 

Recent Events			
Server	Event	When	Output
red-mist	Switching To PHP 7.3.	January 16th, 7:00:00 PM	

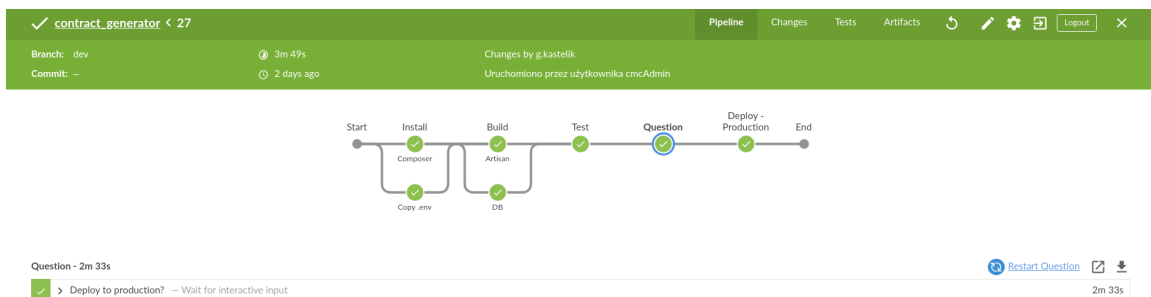
Rysunek 9: Przykład dodania certyfikatu na stronie <https://cloud.digitalocean.com>

2.4.5 Jenkins

Jenkins jest narzędziem automatyki open source napisanym w języku Java z wtyczkami zbudowanymi dla celów ciągłej integracji (ang. Continuous Integration). Jenkins jest używany do budowania i testowania projektów oprogramowania w sposób ciągły, ułatwiając programistom integrowanie zmian w projekcie i ułatwiając użytkownikom uzyskanie nowej konstrukcji. Pozwala również na ciągłe dostarczanie oprogramowania poprzez integrację z dużą liczbą technologii testowych i wdrożeniowych.

Dzięki Jenkinsowi, organizacje mogą przyspieszyć proces tworzenia oprogramowania poprzez automatyzację. Jenkins integruje wszelkiego rodzaju procesy cyklu życia oprogramowania, w tym tworzenie, dokumentowanie, testowanie, pakowanie, etapowanie, wdrażanie, analizę statyczną i wiele innych.

W naszym przypadku Jenkins odpowiada za proces testowania oraz wdrożeniowy. Udane testy wysyłają zapytanie o chęć wdrożenia najnowszej zmiany. Po pozytywnym rozpatrzeniu wysyłane jest odpowiednie zdarzenie do wcześniej przedstawionej platformy forge która zajmuje się wgraniem na serwer najnowszej wersji oprogramowania (rys. 10).



Rysunek 10: Proces testowania i wdrażania aplikacji

The screenshot shows the Jenkins pipeline interface for 'contract_generator' with the 'Tests' tab selected. A message states 'All tests are passing' and 'Nice one! All 16 tests for this pipeline are passing.' Below this, a table lists 16 passed tests, each with a duration of 1s.

Test Name	Duration
> testLoginUser - Tests.Unit.Services.AuthServiceTest	<1s
> testLoginUserThrowExceptionWhenBadEmail - Tests.Unit.Services.AuthServiceTest	<1s
> testLoginUserThrowExceptionWhenBadPassword - Tests.Unit.Services.AuthServiceTest	<1s
> testGetListOfAttributes - Tests.Unit.Services.Domain.AttributeServiceTest	<1s
> testGetListOfBlocks - Tests.Unit.Services.Domain.BlockServiceTest	<1s
> testGetListOfConditionals - Tests.Unit.Services.Domain.ConditionalServiceTest	<1s
> testAddContract - Tests.Unit.Services.Domain.ContractServiceTest	<1s
> testRemoveContract - Tests.Unit.Services.Domain.ContractServiceTest	<1s
> testRemoveContractThrowExceptionWhenNotFound - Tests.Unit.Services.Domain.ContractServiceTest	<1s
> testSaveAndOptimizeFile - Tests.Unit.Services.FileServiceTest	<1s
> testRemoveFileUsingFileUri - Tests.Unit.Services.FileServiceTest	<1s
> testRemoveFileUsingFileUriWithDefaultString - Tests.Unit.Services.FileServiceTest	<1s
> testThrowExceptionWhenTryToRemoveFileUsingBadUri - Tests.Unit.Services.FileServiceTest	<1s
> testChangeUserImage - Tests.Unit.Services.UserServiceTest	<1s
> testChangeUserImageThrowExceptionWhenUserNotFound - Tests.Unit.Services.UserServiceTest	<1s
> testExample - Tests.Feature.UserTest	<1s

Rysunek 11: Raport na temat testów przeprowadzonych na aplikacji

Rozdział 3

Projekt systemu tworzenia umów

3.1 Opis działania systemu

3.2 Schemat bazodanowy

Rozdział 4

Opis platformy implementującej system

4.1 Platforma

4.1.1 Implementacja

4.1.2 Instrukcja użytkowania

4.2 API

4.2.1 Implementacja

4.2.2 Instrukcja użytkowania

4.2.3 Testy

Rozdział 5

Podsumowanie oraz wnioski

Spis rysunków

1	Przedstawienie użycia języków po stronie serwera na podstawie strony https://w3techs.com/technologies/history_overview/programming_language na dzień 12.01.2020 . . .	17
2	Popularność 3 największy frameworków JavaScript na podstawie Google Trends 19	
3	Porównanie możliwości najbardziej znanych frameworków do Vue	20
4	Wygląd IDE w projekcie	21
5	Przykładowy układ na zasadzie git flow ze strony https://barloblog.wordpress.com	22
6	Wybór możliwych konfiguracji na stronie https://cloud.digitalocean.com . .	24
7	Przykładowa konfiguracja u dostawy az.pl	24
8	Przykład dodania certyfikatu na stronie https://cloud.digitalocean.com . . .	25
9	Przykład dodania certyfikatu na stronie https://cloud.digitalocean.com . . .	26
10	Proces testowania i wdrażania aplikacji	27
11	Raport na temat testów przeprowadzonych na aplikacji	27

Bibliografia

- [1] Strona internetowa <https://mfiles.pl/pl/index.php/Umowa> - Encyklopedia Zarządzania
- Dostępna na dzień 12.01.2020