



**WYDZIAŁ
ELEKTROTECHNIKI
I INFORMATYKI
POLITECHNIKI RZESZOWSKIEJ**

Evaluation Warning: The document was created with Spire.Doc for Python.

Piotr Grabias

Portal internetowy dla firmy

Praca dyplomowa inżynierska

Opiekun pracy:
dr inż. Mariusz Gamracki

Rzeszów, rok 2025

Evaluation Warning: The document was created with Spire.Doc for Python.

Spis treści

1.	Wprowadzenie.....	4
1.1.	Kontekst pracy.....	4
1.2.	Motywacja	4
1.3.	Opis pracy.....	5
2.	Metodyka tworzenia oprogramowania.....	5
2.1.	Czym jest oprogramowanie.....	5
2.2.	Pojęcie tworzenia oprogramowania	5
2.3.	Aplikacje.....	6
3.	Opis rozwiązań technologicznych.....	6
3.1.	Najpopularniejsze technologie w tworzeniu aplikacji webowych	6
3.2.	Vue.js.....	7
3.2.1.	Virutal DOM	7
3.2.2.	Struktura komponentu.....	7
3.2.3.	Dyrektywy	8
3.2.4.	Pinia store.....	9
3.2.5.	Vuetify.....	9
3.3.	Python.....	10
3.3.1.	Django	10
3.3.2.	Model, view, template.....	11
3.3.3.	Django admin	11
3.3.4.	REST Framework.....	12
4.	Baza danych	12
4.1.	System zarządzania bazą danych	13
5.	Hosting aplikacji	15
5.1.	Wprowadzenie do wdrażania aplikacji.....	15
5.1.	Docker	15
5.2.	Firebase – hosting aplikacji	17
5.3.	PythonAnywhere – hosting serwera.....	17
6.	Prezentacja projektu	18
6.1	Wygląd aplikacji - produkty.....	18
6.2.	Logowanie, rejestracja	21
6.2	Koszyk użytkownika, zamawianie przedmiotów	23
6.3.	Panel administratora.....	26
6.4.	Pozostałe podstrony	28

Evaluation Warning: The document was created with Spire.Doc for Python.

3.3.1.	Django	10
3.3.2.	Model, view, template.....	11
3.3.3.	Django admin	11
3.3.4.	REST Framework.....	12
4.	Baza danych	12
4.1.	System zarządzania bazą danych	13
5.	Hosting aplikacji	15
5.1.	Wprowadzenie do wdrażania aplikacji.....	15
5.1.	Docker	15
5.2.	Firebase – hosting aplikacji	17
5.3.	PythonAnywhere – hosting serwera.....	17
6.	Prezentacja projektu	18
6.1	Wygląd aplikacji - produkty.....	18
6.2.	Logowanie, rejestracja	21
6.2	Koszyk użytkownika, zamawianie przedmiotów	23
6.3.	Panel administratora.....	26
6.4.	Pozostałe podstrony	28

1. Wprowadzenie

1.1. Kontekst pracy

W dobie dynamicznego rozwoju technologii informatycznych oraz rosnącego znaczenia obecności w Internecie, coraz więcej firm decyduje się na tworzenie własnych portali internetowych. Tego rodzaju platformy nie tylko umożliwiają prezentację oferty firmy, ale także stanowią kluczowe narzędzie do interakcji z klientami, zarządzania procesami biznesowymi oraz poprawy efektywności operacyjnej. Współczesne portale internetowe muszą sprostać rosnącym wymaganiom użytkowników w zakresie dostępności, bezpieczeństwa oraz intuicyjności obsługi. Wprowadzenie takich rozwiązań może przyczynić się do zwiększenia konkurencyjności firmy na rynku oraz poprawy jej wizerunku w oczach klientów.

Celem niniejszego projektu inżynierskiego jest zaprojektowanie i implementacja portalu internetowego dla firmy, który spełni określone wymagania funkcjonalne jednocześnie zapewniając nowoczesne i efektywne narzędzie wspierające działalność firmy.

Evaluation Warning: The document was created with Spire.Doc for Python.

1.2. Motywacja

Motywacją do podjęcia tego projektu jest zainteresowanie nowoczesnymi technologiami wykorzystywanymi w kwestii tworzenia oprogramowania dla aplikacji internetowych. W dzisiejszych czasach posiadanie portalu internetowego dla znaczącej większości przedsiębiorstw jest koniecznym elementem strategii biznesowej.

1.3. Opis pracy

Praca składać się będzie z części: analityczno-projektowej, w której opiszę metody, techniki oraz narzędzia używane przy rozwiązywaniu zagadnień projektowych tego typu i uzasadnię swój wybór, wyniki, gdzie zademonstruję rezultaty swojej pracy nad projektem oraz dokonam praktycznej oceny uzyskanego rozwiązania. Ostatnią z nich będzie zakończenie, w którym podsumuję pracę i zawrę wnioski, podam możliwości dalszego rozwinięcia mojego projektu i wskażę obszar wykorzystania wyników mojej pracy.

2. Metodyka tworzenia oprogramowania

2.1. Czym jest oprogramowanie

Oprogramowanie to zestaw instrukcji lub programów wykorzystywanych do obsługi komputerów i wykonywania zadań. Jest kluczowym elementem każdego systemu komputerowego,

Evaluation Warning: The document was created with Spire.Doc for Python.
pośredniczącym między użytkownikiem a sprzętem. Oprogramowanie można podzielić na kilka głównych kategorii takich jak: oprogramowanie systemowe(system operacyjny, sterowniki), do zarządzania podstawowymi funkcjami komputera i pozwalają na współpracę między sprzętłem a aplikacjami użytkownika., oprogramowanie użytkowe(przeglądarki internetowe, gry, oprogramowanie do komunikacji), służące do wykonywania określonych zadań dla użytkowników końcowych, tak zwane aplikacje, którym poświęcone będą kolejne podrozdziały.

2.2. Pojęcie tworzenia oprogramowania

Tworzenie oprogramowania to działania poświęcone projektowaniu, programowaniu, komponowaniu dokumentacji oraz testowaniu aplikacji. Innymi słowy jest to proces pisania i utrzymywania kodu, a w przypadku tego projektu obejmuje tworzenie aplikacji webowych. Przeważnie jest ono podzielone na obszary specjalizacji programistów. Zazwyczaj ten proces rozpoczyna się od analizy potrzeb projektu, wykonania planów działania, po wykonaniu tych czynności rozpoczyna się etap tworzenia programowania. [2]

2.3. Aplikacje

Aplikacja to stworzony program działający w odpowiednim środowisku na komputerze bądź urządzeniach mobilnych. Zawierają zestaw funkcji czy metod, tworząc większy program. Podziału aplikacji można dokonać na natywne, desktopowe oraz internetowe.

Sektor, który zostanie przedstawiony w tej pracy to tworzenie stron internetowych tzw. web development. Jest to proces pisania strony internetowej działającej na serwerze i udostępnianej użytkownikowi przez sieć internetową. Wszystkie opierają się o trzy fundamenty, HTML, CSS oraz JavaScript. Istnieje również ostatnio bardzo popularny sektor aplikacji mobilnych, ze względu na rosnące wykorzystanie telefonów w życiu codziennym. Rozwój takich aplikacji polega na stworzeniu programów w środowiskach Android lub iOS, przy czym najczęściej stosowane technologie to Android, Swift, Flutter.

3. Opis rozwiązań technologicznych

Evaluation Warning: The document was created with Spire.Doc for Python.

3.1. Najpopularniejsze technologie w tworzeniu aplikacji webowych

Obecnie wykorzystuje się wiele różnych stosów technologicznych, z angielskiego „stack”, które określają obszar umiejętności programisty, można podzielić je na:

- 1) Front-end – służy do tworzenia interfejsu użytkownika, inaczej warstwa oprogramowania uruchamiana po stronie klienta w jego przeglądarce. Odpowiedzialny jest za wyświetlanie struktury strony oraz interakcji z użytkownikiem. Najpopularniejsze wykorzystywane technologie to Angular, Vue.js, React dla JavaScripta oraz TailwindCSS, Bootstrap czy bulma dla CSS
- 2) Back-end – odnosi się do warstwy aplikacji działającej po stronie serwera, zawiera logikę aplikacji, zarządza bazami danych oraz serwerami ,przy użyciu technologii takich jak Node.js, Python(Django), Ruby on Rails, PHP, Java Spring lub .NET

Wybór technologii to często najważniejsza decyzja do podjęcia podczas tworzenia aplikacji internetowych. Trzeba wziąć pod uwagę wiele różnych czynników, takich jak rodzaj pisanego programu użytkowego, funkcjonalności, wydajność w danych rozwiązańach, skalowalność oraz bezpieczeństwo. Zdarza się, że najważniejszym elementem jest znajomość danego rozwiązania i poziom umiejętności. W przypadku tego projektu wybrane zostały dla Front-endu Vue.js i na Back-end Python Django REST Framework. W następnych podrozdziałach uzasadnię swój wybór.

3.2. Vue.js

Vue.js to jeden z najpopularniejszych frameworków JavaScript, przeznaczony do budowy nowoczesnych, interaktywnych aplikacji webowych. Jego największą zaletą jest prostota, która sprawia, że jest łatwy do nauki i wdrożenia, a jednocześnie oferuje rozbudowaną funkcjonalność umożliwiającą tworzenie zaawansowanych projektów. Vue.js łączy w sobie najlepsze cechy dwóch innych czołowych frameworków – React i Angular. Z Reacta zapożycza komponentową architekturę, pozwalającą na tworzenie wielokrotnego użytku modułów aplikacji. Z Angulara przejmuje możliwość korzystania z dynamicznego dwukierunkowego wiązania danych (two-way data binding) oraz wbudowane funkcje służące do manipulacji DOM.

Evaluation Warning: The document was created with Spire.Doc for Python.

DOM (Document Object Model) to strukturalna reprezentacja dokumentu HTML lub XML, która pozwala programistom na interakcję i manipulację treścią strony internetowej za pomocą języków programowania, takich jak JavaScript. Kiedy strona ładuje się w przeglądarce, HTML nie jest wyświetlany bezpośrednio. Zamiast tego przeglądarka odczytuje kod HTML i tworzy jego odwzorowanie w postaci drzewa DOM. Każdy element HTML, taki jak tagi, atrybuty i tekst, staje się węzłem tego drzewa, który można modyfikować w czasie rzeczywistym. Manipulowanie rzeczywistym DOM (tzw. real DOM) jest kosztowne pod względem wydajności, szczególnie w przypadku dużych aplikacji. Dlatego wiele nowoczesnych frameworków, takich jak Vue.js czy React, wykorzystuje mechanizm Virtual DOM. Virtual DOM to wirtualne odwzorowanie drzewa DOM, które przechowywane jest w pamięci. Zamiast bezpośrednio manipulować rzeczywistym DOM, framework najpierw dokonuje zmian w Virtual DOM. [3]

3.2.2. Struktura komponentu

Komponenty to podstawowy element aplikacji w Vue.js, pozwalają na tworzenie wielokrotnie uzywalnych jednostek, które można łatwo wykorzystywać i komponować w bardziej złożone widoki. Składają się on zazwyczaj z trzech części: szablon(template), skrypt oraz styl.

W szablonie definiujemy strukturę HTML komponentu, używając zwykłej składni HTML, umieszczamy ją wewnątrz tagu <template>. Aby wykorzystać dane zadeklarowane w skrypcie możemy użyć podwójnych nawiasów klamrowych między tagami ‘{{ }}’.

Wewnątrz skryptu zawieramy logikę komponentu, jest on umieszczony wewnątrz tagu <script>. Eksportuje obiekty, zmienne, funkcje do szablonu.

Style definiują wygląd komponentu, umieszczone w tagu <style>, jeśli dodamy atrybut scoped to będą one ograniczone do danego komponentu, jeśli tego nie zrobimy będą one dzielone między różnymi komponentami, co może prowadzić do konfliktów.

3.2.3. Dyrektywy

Dyrektwy to specjalne atrybuty z przedrostkiem ‘v-’, ich wartością jest wyrażenie JavaScript. Ich zadaniem jest aktualizowanie DOM, zawsze gdy wartość wyrażenia się zmieni. Podstawowe dyrektywy to ‘v-if’, ‘v-for’, ‘v-on’ oraz ‘v-bind’.

Vue.js w wielu miejscach ułatwia programistom zadanie, upraszczając często używane elementy, w tym wypadku jest to dyrektywa ‘v-bind’, jest ona tak szeroko stosowana, że udostępniony został do niej skrócony zapis w postaci ‘:’. Pozwala ona na powiązanie atrybutu HTML ze skryptem.

Evaluation Warning: The document was created with Spire.Doc for Python.
Dyrektyna warunkowa ‘v-if’ pozwala na wyświetlenie danego bloku HTML, jeśli wartość wyrażenia tej dyrektywy jest prawdziwa. W sytuacji przeciwej możemy użyć ‘v-else’, aby wyświetlić coś w sytuacji niezgodnej z warunkiem. ‘V-for’ jest używane do renderowania listy elementów na podstawie tablicy lub obiektu znajdującego się w części skryptowej.

Do nasłuchiwanego zdarzeń takich jak kliknięcie przycisku i wywoływanie odpowiednich metod wykorzystywana jest dyrektywa ‘v-on’ albo w skrócie ‘@’. Poniżej znajduje się przykład wykorzystania dyrektyw, na bardzo prostym przykładzie przycisku kontrolującego wyświetlenie listy osób:

```
<template>

<button @click="changeVisibility()">Zmien widoczność</button>

<div v-if="isVisible">

  <p v-for="person in people" :key="person.name">
    {{ person.name }} {{ person.lastname }}
  </p> </div>

<div v-else>Niewidoczne</div>
```

```

</template>

<script setup>

import { ref } from "vue";

const isVisible = ref(false);

const people = ref([
  { name: "Jan", lastname: "Kowalski" },
  { name: "Piotr", lastname: "Nowak" },
]);

function changeVisibility() {
  isVisible.value = !isVisible.value;
}
</script>

```

3.2.4. Pinia store

Evaluation Warning: The document was created with Spire.Doc for Python.

W większości nowoczesnych aplikacji trzeba wykorzystać mechanizm, który zajmie się przechowywaniem danych, najczęściej jest to tzw. store – magazyn. W przypadku Vue.js są dwie główne możliwości: Vuex oraz Pinia, w ostatnim czasie Pinia stała się oficjalnym rozwiązaniem dla aplikacji Vue do zarządzania stanem danych. Pinia lepsza jest dla aplikacji małych oraz średnich, ze względu na prostą strukturę oraz łatwą implementację. Ze względu na rozmiar rozwiązania, w tym projekcie wybrano Pinia do dysponowania danymi produktów, użytkowników, transakcji,

3.2.5. Vuetify

Częstym wyborem do projektowania interfejsu aplikacji są aktualnie obecne na rynku frameworki CSS, Tailwind czy Bootstrap, natomiast w Vue znajduje się jeszcze dodatkowa opcja którą jest Vuetify. To rozwiązanie jest oparte na stworzonym przez Google stylu graficznym, Material Design. Działa na zasadzie udostępnienia nowych elementów HTML z przedrostkiem ‘v-’ takich jak ‘v-btn’ z możliwością zmiany ich parametrów(z angielskiego ‘props’), co zwykle daje duże pole do dostosowania wyglądu do potrzeb aplikacji. Eliminuje całkowicie potrzebę pisania stylów CSS, co jest dużym ułatwieniem. Praca z tym frameworkiem nie sprawia żadnych problemów, ponieważ dokumentacja została świetnie wykonana, a samo zastosowanie stosunkowo proste.

3.3. Python

Python to wysokopoziomowy, wszechstronny język programowania, który zdobył ogromną popularność wśród programistów na całym świecie. Jego prostota, przejrzysta składnia oraz ogromna liczba dostępnych bibliotek sprawiają, że znajduje zastosowanie w różnych dziedzinach informatyki – od analizy danych i sztucznej inteligencji, przez automatyzację zadań, po tworzenie aplikacji internetowych. Jednym z kluczowych celów, jakie przyświecały twórcom języka, była czytelność kodu, co czyni Pythona intuicyjnym i łatwym do nauki, nawet dla początkujących programistów. Składnia Pythona jest mocno zbliżona do języka angielskiego, co jeszcze bardziej ułatwia pracę z tym językiem.

Jednym z największych atutów Pythona jest bogaty ekosystem bibliotek i narzędzi, które umożliwiają szybkie prototypowanie i realizację nawet skomplikowanych projektów. W dziedzinie aplikacji webowych powstało wiele dedykowanych frameworków, które pozwalają programistom tworzyć rozbudowane aplikacje internetowe. Streamlit – przeznaczony głównie do budowy interaktywnych aplikacji do analizy danych i wizualizacji wyników. Flask – lekki framework idealny do mniejszych projektów lub aplikacji wymagających większej elastyczności. Django – najbardziej kompleksowy i zaawansowany framework, który oferuje pełen zestaw narzędzi do tworzenia aplikacji webowych, takich jak system uwierzytelniania, zarządzanie bazami danych czy obsługa routingu.

Właśnie Django został wybrany do realizacji tego projektu, ze względu na jego wszechstronność, stabilność oraz szeroką dokumentację i aktywną społeczność. Django umożliwia szybkie tworzenie aplikacji o dużej złożoności, a jednocześnie dba o bezpieczeństwo oraz łatwość rozwoju i skalowania kodu.[4]

3.3.1. Django

Django to zaawansowany framework, który znacząco upraszcza proces tworzenia stron internetowych za pomocą języka Python. Jego główną zaletą jest dostarczenie gotowych narzędzi i rozwiązań, które przyspieszają pracę nad aplikacjami webowymi i eliminują konieczność budowania wielu elementów od podstaw. Django działa w oparciu o wzorzec architektoniczny MVT (Model-View-Template), który jest zbliżony do popularnego wzorca MVC, stosowanego w większości frameworków webowych.