

**Akademia Finansów i Biznesu Vistula**

Wydział Inżynierski

Kierunek: Informatyka

**Viktor Rumiievskyi**

Nr. Albumu: 51061

**Projekt i implementacja sklepu internetowego  
systemów i elementów inteligentnego domu**

Praca inżynierska

napisana pod kierunkiem

dr inż. IwonaDolińska

Warszawa, 2023

## Spis treści

<b>WYKAZ TERMINÓW, SKRÓTÓW I SYMBOLI.....</b>	<b>4</b>
<b>1. Rozdział I. «ANALIZA I UZASADNIENIE TEMATU PROJEKTU DYPLOMOWEGO» .....</b>	<b>5</b>
1.1. Uzasadnienie tematu .....	5
1.2. Analiza produktów oprogramowania - analogie .....	5
1.3. Zalety sklepu internetowego.....	7
<b>2. Rozdział II «Koncepcja (Inteligentnego Domu)» .....</b>	<b>9</b>
2.1. Możliwości inteligentnego domu .....	9
2.2. Systemy i elementy "inteligentnego domu" .....	10
2.2.1. «Inteligentny dom» dla mieszkania .....	11
2.2.1.1. Jak zainstalować system w mieszkaniu .....	12
2.2.2. Zalety systemu .....	13
2.2.3. Który «Inteligentny dom» jest lepszy – główne parametry do porównania.....	14
2.2.3.1. Ajax.....	14
2.2.3.2. BroadLink .....	15
<b>3. Rozdział III. «Koncepcja (Sklepu internetowego)».....</b>	<b>16</b>
3.1. Klasyfikacja sklepów internetowych .....	16
3.2. Korzyści z handlu elektronicznego .....	17
3.3. Rodzaje handlu elektronicznego .....	18
3.4. Rodzaje platform e-commerce .....	18
3.5. Zasoby dotyczące tworzenia sklepu internetowego .....	18
3.5.1.1. Magento .....	18
3.5.1.2. OpenCart.....	19
3.5.1.3. osCommerce.....	19
<b>4. Rozdział IV «Pojęcie strony internetowej» .....</b>	<b>20</b>
4.1. Języki programowania stron internetowych.....	20
4.1.1. Technologie programowania stron internetowych.....	20
4.1.1.1. Projektowanie stron internetowych .....	20
4.1.1.2. Programowanie po stronie klienta .....	21
4.1.1.3. Programowanie po stronie serwera .....	21
4.1.2. HTML .....	22
4.1.2.1. Struktura dokumentu HTML .....	23
4.1.3. CSS.....	24
4.1.3.1. Zalety CSS.....	25
4.1.3.2. Funkcje CSS.....	25
4.1.4. JavaScript.....	26
4.1.4.1. Dlaczego JavaScript? .....	26
4.1.4.2. Co JavaScript może zrobić w przeglądarce? .....	27

4.1.4.3.	Czego nie może zrobić JavaScript w przeglądarce?	27
4.1.5.	Node.js	28
4.1.6.	React	29
4.1.6.1.	Komponent React	29
5.	Rozdział V «Opis struktury i funkcjonowania baz danych»	31
5.1.	Co to są bazy danych?	31
5.2.	Typy baz danych	31
5.3.	Systemy zarządzania bazami danych (DBMS)	32
5.4.	Programy bazodanowe	32
5.4.1.	Porównanie wiodącego oprogramowania bazodanowego Open Source	34
5.5.	Język zapytań SQL	34
5.5.1.	Kto używa języka SQL	34
5.5.2.	Podstawowe polecenia SQL	35
5.6.	MongoDB	35
5.6.1.	Najlepsze aplikacje do korzystania z MongoDB	36
5.6.2.	5 powodów, by wybrać MongoDB	38
5.6.3.	Małe porównanie SQL z MongoDB	39
6.	Rozdział VI «Kluczowe etapy tworzenia strony internetowej»	40
6.1.	Krok 1. Definiujemy cele	40
6.2.	Krok 2. Wybór domeny i hostingu	40
6.3.	Krok 3. Sporządzamy zadanie techniczne	41
6.4.	Krok 4. Pisanie tekstów	42
6.5.	Krok 5. Projekt	42
6.6.	Krok 6. Optymalizacja	42
6.7.	Krok 7. Układ strony	43
6.8.	Krok 8. Wypełnianie treści	44
6.9.	Krok 9. Testowanie	44
7.	Opis struktury aplikacji WWW	46
8.	Opis zawartości strony głównej	46
9.	Opis zawartości strony produktowej	47
10.	Opis strony autoryzacji użytkownika	50
	Wnioski	51
	Bibliografia	52
	Dodatek 1	53
	Dodatek 2	75

# WYKAZ TERMINÓW, SKRÓTÓW I SYMBOLI

**HTML** - HyperText Markup Language

**CSS** - Cascading Style Sheets

**JS** - JavaScript

**PHP** - PHP Hypertext Preprocessor

**SQL** - Structured Query Language

**CMS** - Content Management System

**GSM** - Global System for Mobile Communications

**Z-Wave** - jest protokołem komunikacji bezprzewodowej, który jest używany głównie dla automatyki domowej

**Jeweller** - protokół radiowy zapewniający dwukierunkową komunikację pomiędzy koncentratorem a urządzeniami systemu zabezpieczeń Ajax.

**TIS** - Internetowy system handle

**SSL** - Secure Sockets Layer

**B2B** - Business to business

**B2C** - Biznes dla klienta

**C2C** - Od klienta do klienta

**SASS** - Syntactically Awesome Style Sheets

**LESS** - Leaner Style Sheets

**HTTP** - Hypertext Transfer Protocol

**HTTPS** - Hypertext Transfer Protocol Secure

**NoSQL** - baza danych, która zapewnia mechanizm przechowywania i wyszukiwania danych różni się od podejścia do tabel w relacyjnych bazach danych.

# 1. Rozdział I. «ANALIZA I UZASADNIENIE TEMATU PROJEKTU DYPLOMOWEGO»

## 1.1. Uzasadnienie tematu

Celem tej pracy jest opracowanie sklepu internetowego sprzedającego systemy i elementy inteligentnego domu.

Żyjemy teraz w świecie, w którym większość codziennych zadań jest uproszczona lub zautomatyzowana, a trend ten rośnie z każdym rokiem. I ten temat projektu dyplomowego został wybrany, aby pokazać, w jaki sposób powstają i składają się sklepy internetowe. Stworzenie maszyny internetowej i umieszczenie jej w Internecie jest jedną z alternatywnych metod pozycjonowania firmy i informowania grupy docelowej.

Obecnie rozwój sklepów internetowych jest dość powszechną działalnością. Możesz tworzyć witryny za pomocą «HTML<sup>1</sup>, CSS,<sup>2</sup> JS<sup>3</sup>, PHP<sup>4</sup>, SQL<sup>5</sup>, Java» i innych obszarów programowania internetowego.

Teraz technologia internetowa rozwija się tak bardzo, że witryny można tworzyć na dowolnym poziomie złożoności bez znajomości HTML, CSS. To dzięki CMS<sup>6</sup>. CMS - oprogramowanie do organizowania stron internetowych lub innych zasobów informacyjnych w Internecie lub poszczególnych sieciach komputerowych.

Ten projekt dyplomowy zostanie stworzony przy użyciu "HTML, CSS, JavaScript, PHP, SQL"

Osoba ma takie możliwości, że nawet z innego kraju może wyłączyć światła i wodę w domu. Cóż, to są małe rzeczy, osoba może nawet podlewać kwiaty. A to wszystko dzięki zautomatyzowanemu systemowi "Smart Home".

## 1.2. Analiza produktów oprogramowania - analogie

Do analizy wybrano kilka witryn o podobnej tematyce i przeznaczeniu. Jedną z takich stron jest strona Secur

---

<sup>1</sup> **HTML** - HyperText Markup Language

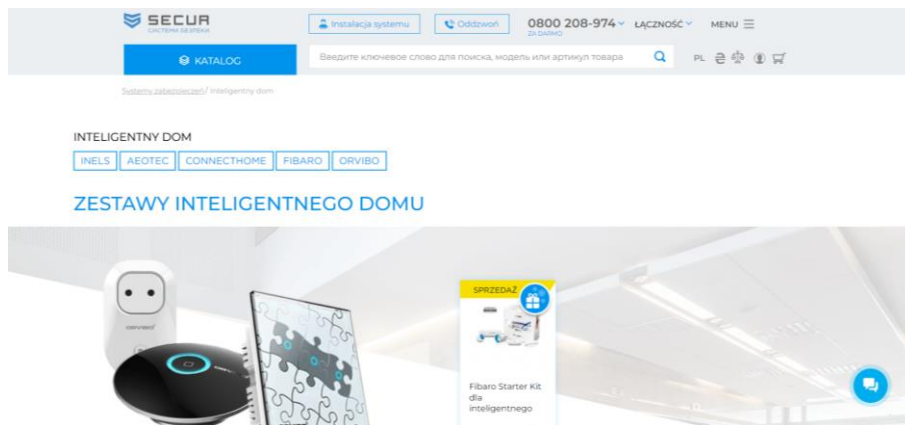
<sup>2</sup> **CSS** – Cascading Style Sheets

<sup>3</sup> **JS** - JavaScript

<sup>4</sup> **PHP** - PHP Hypertext Preprocessor

<sup>5</sup> **SQL** - Structured Query Language

<sup>6</sup> **CMS** - Content Management System



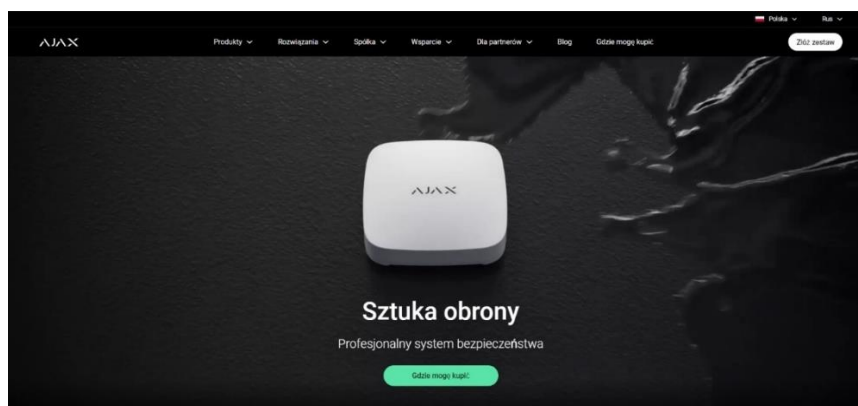
Rys. 1.1. – Strona główna strony Secur

Witryna sprzedaje komponenty do inteligentnego domu. Ten sklep oferuje usługi i akcesoria do inteligentnych domów innych firm. Takie jak «AJAX, TECSAR, ZKTECO, UNIVIEW»



Rys. 1.2. – Strona główna strony internetowej TECSAR

Sklep TECSAR zajmuje się tylko rozwojem kamer. Ma duży wybór «Kamer dla komornika».



Rys. 1.3. – Strona główna witryny Ajax

Sklep Ajax. Strona zajmuje się w pełni wyposażeniem domu: Huby, Czujniki wycieków, Czujniki ruchu, Czujniki pożarowe.

<b>Kryteria</b>	Aplikacja internetowa Secur	Aplikacja internetowa TECSAR	Aplikacja internetowa AJAX
Bez zbędnych reklam	-	+	+
Nieistotne informacje	+	-	-
Przestarzały projekt	+	-	-
Możliwość informacji zwrotnej	+	+	+
Łatwość obsługi	+	+	+
Ciekawy opis produktu	-	+	+

### 1.3. Zalety sklepu internetowego

#### 1. Oszczędności czas

##### *Szybkość samowystarczalności*

Głównym powodem jest niska inwestycja w rozpoczęcie projektu podczas tworzenia strony i szybkie rozpoczęcie sprzedaży. W tych warunkach w pierwszych tygodniach możesz pokryć wszystkie koszty i osiągnąć zysk.

#### 1. Skuteczność

##### *Analityka*

Ta zaleta sklepu internetowego zwiększa efektywność biznesową bez większych trudności, czasu i kosztów pracy. Konfiguracja usług analitycznych umożliwia zbieranie szczegółowych informacji o odwiedzających witrynę, cechach ich zachowania, transformacji w kupujących itp.

##### *Asortymentu*

W ramach projektu online praktycznie nie ma ograniczeń w poszerzaniu asortymentu produktów. Wszystkie koszty związane ze zmianami będą dotyczyły wyłącznie aktualizacji informacji na stronie.

#### 2. Oszczędności budżetowe

*Koszt wynajmu powierzchni handlowej*

Sklep internetowy nie wymaga powierzchni sprzedaży, biura sprzedaży, a nawet magazynu, jeśli firma stosuje metodę dostaw bezpośrednich (dropshipping). Potrzeba takich inwestycji uderza szczególnie w przedsiębiorcę na samym początku rozwoju.

*Próg wejścia*

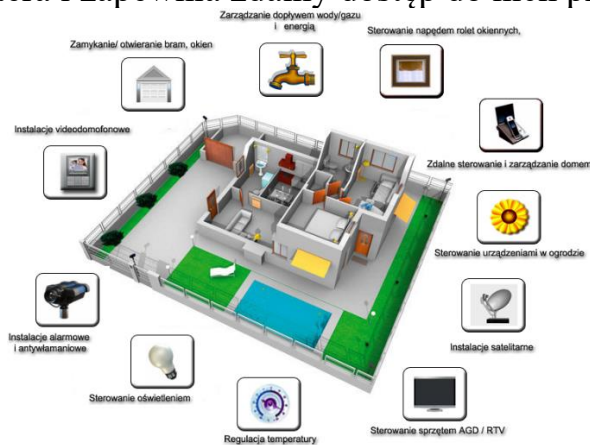
Uruchomienie witryny może kosztować skromną kwotę - do 1500 dolarów. W większości przypadków wystarczy to do utworzenia, wypełnienia i uruchomienia sklepu internetowego.



## 2. Rozdział II «Koncepcja (Inteligentnego Domu)»

**Inteligentny dom** - to system urządzeń domowych zdolnych do wykonywania czynności i rozwiązywania niektórych codziennych zadań bez interwencji człowieka.

Urządzenia mogą być podłączone do sieci komputerowej, co pozwala na sterowanie nimi za pomocą komputera i zapewnia zdalny dostęp do nich przez Internet.



Rys. 2.1. – Elementy inteligentnego domu

### 2.1. Możliwości inteligentnego domu

System Smart Home jest w stanie rozpoznać konkretne sytuacje zarówno w domu, jak i poza nim, i zgłosić to. Może symulować twoje zachowanie w domu (włączać i wyłączać światła, kontrolować okiennice, podlewać trawniki, "słuchać" muzyki) i kontrolować zachowanie innych, gdy jesteś daleko, to znaczy kontrolować dostęp do pomieszczeń i poszczególnych funkcji domu.

System Smart Home jest w stanie oszczędzać energię elektryczną, przełączając urządzenia elektryczne w tryb "ekonomiczny" i zapewnić optymalne warunki temperaturowe dla Twojego przyjazdu.

- zapewnić optymalną temperaturę w pomieszczeniu
- włączanie/wyłączanie systemu ogrzewania podłogi, stopni i wejścia do garażu
- włączanie/wyłączanie podlewania trawnika
- włączanie/wyłączanie, sterowanie i równomierne rozprowadzanie dźwięków audio i wideo
- zapewnić bezpieczeństwo domu pod nieobecność właściciela



Rys. 2.2. – demonstracja systemu inteligentnego domu

## 2.2. Systemy i elementy "inteligentnego domu"

Kompleks inteligentnego domu składa się z dużej liczby komponentów:

- centralny sterownik (hub), który zawiera algorytmy systemu;
- termostaty, czujniki, które odczytują otoczenie domu (niektóre są instalowane nawet w piwnicy budynku);
- panel sterowania do aktywacji, zarządzania sprzętem;
- grupa kontrolowanych urządzeń: sprzęt AGD, inteligentne gniazdka, dźwigi elektryczne, elektrozamki, żaluzje w oknach, automatyka inteligentnego ogrzewania, oświetlenie;
- baterie: przekaźniki, bezpieczniki, baterie.

Gdy system zostanie w pełni zainstalowany, gadżety mobilne, komputer domowy staną się jego częścią. Korzystanie z oprogramowania umożliwi zdalne sterowanie za pośrednictwem routera Wi-Fi lub sieci GSM<sup>7</sup>.

### Podstawowe możliwości inteligentnego domu

- 1) **Regulacja temperatury.** Automatyzacja klimatyzacji tworzy idealny mikroklimat osobno dla każdego pomieszczenia. Klimatyzacja reguluje optymalny poziom wilgotności w każdym pomieszczeniu. Istnieje kilka sposobów na zwiększenie temperatury: włącz inteligentne ogrzewanie podłogowe lub baterie grzewcze. Łatwo jest kontrolować ozonowanie, jonizację, filtrację, jeśli zainstalowane są odpowiednie klimatyzatory.

<sup>7</sup> GSM - Global System for Mobile Communications



Rys. 2.3. - Regulacja temperatury

- 2) **Zaopatrzenie w wodę.** Autonomiczne utrzymanie lokalnego obszaru występuje poprzez nawadnianie trawników, podlewanie kwiatów w ogrodzie, w zależności od intensywności opadów.

Kontrolowanie pracy dźwigów pomaga zidentyfikować uszkodzone segmenty rurociągu.

Oczyszczanie wody w basenie odbywa się zgodnie z zakodowanym scenariuszem. Ale osobno należy wspomnieć o zarządzaniu telewizją satelitarną, dostępem do Internetu, przesyłaniem obrazów z wewnętrznych kamer wideo do telewizorów.

- 3) **System sprzętu audio-wideo.** Scentralizowane nagłośnienie wszystkich pomieszczeń (w tym piwnicy) z indywidualnym poziomem głośności, komunikacja głosowa między pomieszczeniami, pojedyncza baza treści multimedialnych, automatyczne wyciszenie muzyki podczas przychodzących połączeń telefonicznych lub dzwonek do drzwi, wyświetlanie komunikatów ostrzegawczych czujników na monitorach telewizyjnych.

### 2.2.1. «Inteligentny dom» dla mieszkania

W mieszkaniu technika ta jest w stanie sprawować kontrolę w różnych aspektach.

#### 1. Ogrzewanie

Możliwość UD jest pożądana. Rzeczywiście, oprócz zapewnienia pożądanego mikroklimatu w pomieszczeniu, możliwość regulacji temperatury pomaga obniżyć koszty gazu lub energii elektrycznej. W pomieszczeniach, w których będzie analizowana temperatura, instalowane są czujniki termiczne, które znajdują się pod kontrolą modułu systemu «inteligentny dom».

#### 2. Oświetlenie w obudowie

Nie tylko zwykłe lampy są często instalowane w domach. Istnieją różne «dzwonki i gwizdki», takie jak paski LED, żarówki i inne odmiany. Wygodne jest używanie jednego urządzenia do sterowania tymi urządzeniami. Możesz użyć gotowych opcji oświetlenia, a także dostosować własne, które są najbardziej odpowiednie dla różnych sytuacji życiowych.

### 3. Klimatyzacja i wentylacja

System kontroluje cyrkulację powietrza w pomieszczeniu i robi to osobno w każdym pomieszczeniu. Na przykład łazienka lub toaleta wymaga bardziej aktywnej wentylacji powietrza. W sypialni lub salonie ważna jest klimatyzacja i jej dostosowanie do określonych parametrów wilgotności i temperatury.

### 4. Multimedia w domu

Ulubiona funkcja właścicieli, którzy lubią słuchać muzyki, stale poruszając się po mieszkaniu. Dzięki konfiguracji systemu odtwarzane utwory przemieszczają się z pokoju do pokoju po Słuchaczu.

### 5. Lokale

Czujnik ruchu i kontrola dostępu ochronią mieszkanie przed penetracją rabusiów i niechcianych osób. Dla tych, którzy nie mają zakazu dostępu do obudowy, klucze magnetyczne lub specjalne przepustki są przewidziane, które umożliwiają swobodny dostęp.

### 6. Sytuacje awaryjne

W tym celu system "inteligentnego domu" wykorzystuje czujniki, które monitorują nieszczelności rur, wycieki gazu, pożary lub dym w pomieszczeniach i inne sytuacje awaryjne.



Rys. 2.4. - sposoby instalacji elementów sterujących inteligentnego domu w mieszkaniu

#### 2.2.1.1. Jak zainstalować system w mieszkaniu

Aby stworzyć pomysł na konfigurację systemu "inteligentnego domu", należy zapoznać się z procedurą instalacji wszystkich jego elementów.





Rys. 2.5. - pomiar schematu elektrycznego

Ten proces odbywa się w trzech krokach.

### 1. Projekt

Możesz "wyciągnąć" gotowy projekt z Internetu lub powierzyć projekt instalatorom, co będzie wysokiej jakości i skuteczne. W skrajnych przypadkach, jeśli istnieje doświadczenie i umiejętności w pracy z projektem, właściciel domu może spróbować ukończyć projekt samodzielnie.

### 2. Urządzenia i sprzęt

Mając projekt w ręku, łatwo jest zdecydować, jaki sprzęt kupić dla formacji. Dostępne są niedrogie i drogie opcje. Wybór należy do właściciela mieszkania.

### 3. Instalacja

Proces ten jest konwencjonalnie podzielony na zewnętrzny i wewnętrzny. Instalacja gniazd, przełączników, przełączników, lamp, kamer i czujników sygnalizacji pożaru jest zawarta w scenie zewnętrznej. Właściciel mieszkania jest w stanie wykonać taką instalację niezależnie.

Ale lepiej powierzyć wewnętrzną instalację "profesjonalistom" tej sprawy. W końcu konieczne jest ułożenie i podłączenie ekranów. Rozsądniej jest wykonywać prace wewnętrzne do czasu zakończenia napraw w pomieszczeniu, ponieważ niektóre zmiany są wymagane do wprowadzenia pewnych zmian, takich jak usunięcie starego licznika energii elektrycznej i zainstalowanie nowego "inteligentnego".

## 2.2.2. Zalety systemu

Możliwość instalacji systemu, nawet jeśli naprawy zostały już wykonane – ponieważ technologia Z-Wave<sup>8</sup> wykorzystuje energooszczędne i miniaturowe moduły częstotliwości radiowej, które są wbudowane w elektronikę użytkową i różne urządzenia, takie jak oświetlenie, ogrzewanie, kontrola dostępu, systemy rozrywki i sprzęt gospodarstwa domowego, można go zamontować na każdym etapie budowy bez hałasu, kurzu i układania przewodów

<sup>8</sup> **Z-Wave** - jest protokołem komunikacji bezprzewodowej, który jest używany głównie dla automatyki domowej

**Komfort** - sterowanie domem zdalnie, za pomocą paneli dotykowych, ekranu telewizora; jeden pilot do wszystkich urządzeń elektrycznych (jedno naciśnięcie = kilka akcji); różne scenariusze światła i ogrzewania; sterowanie sprzętem w oparciu o aktualne parametry (czas, temperatura, poziom oświetlenia, ruch).

### 2.2.3. Który «Inteligentny dom» jest lepszy – główne parametry do porównania

Zanim zaczniesz rozumieć zawilosci i funkcjonalność różnych systemów automatyki mieszkaniowej, dla wygody możemy wyróżnić główne kryteria ich oceny. Na przykład mogą to być następujące punkty:

- standardowe komponenty zestawu i możliwość skalowania (czyli dodawania sprzętu w celu rozszerzenia funkcjonalności);
- kolejność łączenia elementów systemu ze sobą (przewodowa bezprzewodowa);
- metoda sterowania (za pomocą komputera, smartfona, panelu sterowania);
- kanały komunikacji z użytkownikiem (Internet, GSM, kanał radiowy);

#### 2.2.3.1. Ajax



Rys. 2.6. - Produkty Ajax

Producent: Ukraina. W związku z tym interfejsy ukraiński i rosyjski są domyślnie obsługiwane.<sup>9</sup>

Ten system automatyki domowej w pełni radzi sobie z dwoma ważnymi zadaniami jednocześnie:

- zapewnia komfort i wygodę w zarządzaniu podtrzymywaniem życia w pomieszczeniu;
- gwarantuje bezpieczeństwo mieszkania w całości, kontrolując granice obiektu pod kątem włamania, a także elektrycznych, przeciwpożarowych, gazowych i innych możliwych zagrożeń dla domu.

Sprzęt Ajax Smart Home działa na bezpiecznie zaszyfrowanym i bezpiecznym dwukierunkowym połączeniu radiowym Jeweller<sup>10</sup> własnej konstrukcji, ma pełną autonomię od sieci dzięki rezerwowemu źródłu zasilania - koncentratorowi, charakteryzuje się stylowym wyglądem wszystkich swoich urządzeń.

#### Zalety:

<sup>9</sup> <https://ajax.systems/pl/>

<sup>10</sup> **Jeweller** - protokół radiowy zapewniający dwukierunkową komunikację pomiędzy koncentratorami a urządzeniami systemu zabezpieczeń Ajax.

- łatwa instalacja;
- kanał komunikacji bezprzewodowej pomiędzy elementami systemu;
- duży obszar pokrycia sygnałem (do 2000 m);
- podłączenie do 100 urządzeń;
- inteligentna wtyczka pokazuje zużycie energii elektrycznej (biorąc pod uwagę podłączone urządzenia), automatycznie wyłącza się, gdy napięcie spada;
- obecność przycisku napadowego na pilocie zdalnego sterowania (brelok);

**Wady:**

- działające tylko przy działaniu centralnego sterownika (Hub), czyli braku autonomii czujników;
- sterowanie tylko przez telefon, chociaż eliminuje to konieczność instalowania dodatkowych programów na komputerze.

### 2.2.3.2. BroadLink



Rys. 2.7. - Produkty BroadLink

Producent: Chiny. Domyślnie nie ma ukraińskiego interfejsu, w razie potrzeby można go znaleźć.<sup>11</sup>

Sprzęt BroadLink Smart Home to zestaw nowoczesnych urządzeń cyfrowych przeznaczonych do racjonalnego zarządzania sprzętem AGD, a także oświetleniem, energią, bezpieczeństwem i innymi systemami w domu. Każdy element takiego kompleksu może działać zarówno niezależnie, jak i wchodzić ze sobą w interakcje.

**Zalety:**

- szybko instaluje, łączy i konfiguruje;
- posiada szeroką gamę czujników (wilgotność, temperatura, oświetlenie, hałas, zanieczyszczenie powietrza);
- możesz łatwo dodawać i usuwać różne urządzenia; działa bez centralnego koncentratora (autonomiczna obsługa czujników);

**Wady:**

- krótki zasięg sygnału (do 50 m);
- brak zasilania awaryjnego do koncentratora;

<sup>11</sup> <https://www.ibroadlink.com/>

### 3. Rozdział III. «Koncepcja (Sklepu internetowego)»

**Sklep** internetowy to rodzaj strony internetowej z katalogiem towarów lub usług, która pozwala na sprzedaż i dokonywanie zakupów przez Internet. Proces zakupu najczęściej odbywa się poprzez dodanie wybranych pozycji do koszyka i złożenie zamówienia online.

W 2020 roku liczba sklepów internetowych w Polsce przekroczyła 42 tys. Branża e-commerce nabiera tempa, o jakim do niedawna mogliśmy tylko pomarzyć.

Konkurencja o uwagę e-konsumentów rośnie z dnia na dzień, a ci, którzy do tej pory korzystali z biznesu offline, chcą mieć udział w e-commerce.

Wraz z rozwojem branży e-commerce powstaje coraz więcej sklepów internetowych, które stale ewoluują, tworząc nie tylko usprawnienia tradycyjnej działalności gospodarczej, ale przede wszystkim sieć powiązań, w których użytkownicy mogą się nawzajem odnaleźć i zaspokoić swoje potrzeby.

#### 3.1. Klasyfikacja sklepów internetowych

Istnieje kilka systemów klasyfikacji sklepów internetowych:<sup>12</sup>

1. zgodnie z modelem biznesowym: sklep w pełni on-line i połączenie biznesu off-line z internetowym (gdy sklep internetowy tworzony jest na podstawie istniejącej rzeczywistej struktury handlowej);
2. w relacjach z dostawcami: sklepy, które posiadają własny magazyn (dostępność realnych zapasów); sklepy działające na podstawie umów z dostawcami (brak znaczących zapasów);
3. w zależności od stopnia automatyzacji wśród systemów transakcyjnych sklepów elektronicznych rozróżnia się Web-showcases, a właściwie sklepy internetowe i handlowe systemy internetowe (TIS).<sup>13</sup>

<sup>12</sup> <https://www.sempire.pl/sklep-internetowy-definicja-rodzaje-funkcjonalnosci.html>

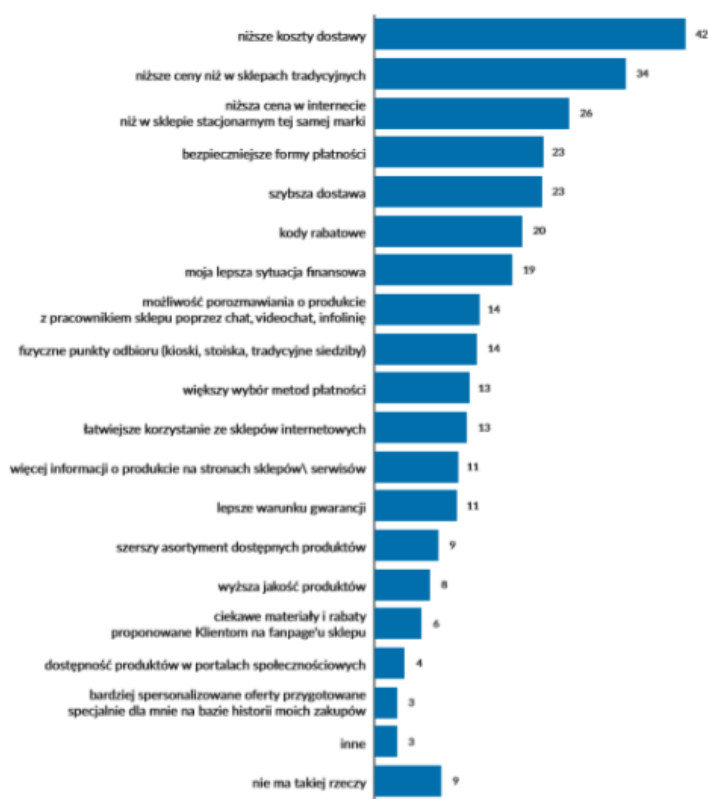
<sup>13</sup> TIS - Internetowy system handlu



## Potencjalne drivers zakupów online

Co skłoniłoby Cię do kupowania przez Internet?

Dane w %



Ryc. 3.1. - Czynniki motywacyjne do zakupów w sklepie internetowym

### 3.2. Korzyści z handlu elektronicznego

E-commerce oferuje wiele korzyści.

Po pierwsze, próg wejścia jest bardzo niski. Najtańsze oprogramowanie do tworzenia sklepów kosztuje około 20 zł miesięcznie. Więc, jeśli jesteś małym rzemieślnikiem, jest to łatwy sposób na sprzedaż swoich towarów. E-commerce to także niższe koszty niż w przypadku handlu tradycyjnego. Nie ma potrzeby wynajmowania pokoju w najdroższych miejscach. Posiadanie własnego produktu, wystarczającej ilości magazynu w dowolnym miejscu lub jeśli nie jest sprzedawany w dużych ilościach, możesz nawet przechowywać go w budynku na własną rękę terytorium. Wszystkie operacje związane z pracą sklepu można wykonywać z domu, nie ma potrzeby wynajmowania biura. Sklep internetowy to także możliwość sprzedaży 24 godziny na dobę. Może więc zarabiać pieniądze, kiedy dosłownie "śpi". Szerokie możliwości, jeśli chodzi o marketing takiego sklepu. Nowoczesne narzędzia pozwalają promować swój sklep na wiele sposobów, od pozycjonowania, przez reklamę w sieciach społecznościowych i innych miejscach, po marketing oparty na współpracy z twórcami różnorodnej treści online. Obszar nie jest przez nikogo ograniczony, jak ten tradycyjny.

### 3.3. Rodzaje handlu elektronicznego

Handel online nie zawsze musi być bezpośrednio związany ze sprzedażą klientowi indywidualnemu. Dwa najważniejsze rodzaje działalności to B2B i B2C, a także C2C - mniej oczywiste. Kim one są?

**B2B**<sup>14</sup> lub business for business to wszystkie transakcje zawierane między podmiotami gospodarczymi. Taka sprzedaż może odbywać się zarówno za pośrednictwem platform transakcyjnych, jak i za pośrednictwem handlowców, z którymi można się skontaktować przez Internet.

**B2C**<sup>15</sup>, czyli biznes dla konsumenta. Jest to pierwszy i najbardziej oczywisty handel internetowy. Są to transakcje między sprzedawcą, którym jest firma, a osobą fizyczną, czyli konsumentem.

**C2C**<sup>16</sup>, czyli klient jest klientem. Jest to sprzedaż między osobami fizycznymi. Dzieje się to za pośrednictwem sieci społecznościowych i różnych reklam, takich jak wspomniany wcześniej OLX. Taka sprzedaż może być również dokonywana na eBay lub Allegro.

### 3.4. Rodzaje platform e-commerce

Na rynku istnieje wiele platform e-commerce, które różnią się nieznacznie lub znacząco. Istnieje jednak jeden główny podział, który dzieli je wszystkie na otwarte i zamknięte platformy:

**Otwarte platformy e-commerce** – jeśli nie masz o nich większej wiedzy, są stosunkowo trudne do skonfigurowania i wdrożenia, ale mają dużą elastyczność, jeśli chodzi o dostosowywanie ich różnych mają one między innymi dodatkowe funkcje, takie jak subskrypcja lub rezerwacja.

**Zamknięte platformy e-Commerce** budowane są na potrzeby prostych rozwiązań e-commerce. rozwój wielkości i nowych funkcji sklepu internetowego.

### 3.5. Zasoby dotyczące tworzenia sklepu internetowego

#### 3.5.1.1. Magento

**Magento** - platforma eCommerce typu open source przeznaczona do tworzenia sklepów internetowych.

---

<sup>14</sup> **B2B** - Business to business

<sup>15</sup> **B2C** - Biznes dla klienta

<sup>16</sup> **C2C** - Od klienta do klienta



### 3.5.1.2. OpenCart

**OpenCart to** system zarządzania treścią open source przeznaczony do tworzenia sklepów internetowych. Rozpowszechniane na licencji GNU General Public License.

OpenCart można zainstalować na dowolnym serwerze WWW. Wokół OpenCart powstała duża społeczność (ponad 46000 członków), dzięki której powstało ponad 8500 darmowych rozszerzeń w postaci dodatkowych modułów.



### 3.5.1.3. osCommerce

**osCommerce** (Open Source Commerce) to system zarządzania treścią typu open source przeznaczony do tworzenia sklepów internetowych. Może być zainstalowany na dowolnym serwerze WWW z obsługą PHP i MySQL. Rozpowszechniane na licencji GNU General Public License



## 4. Rozdział IV «Pojęcie strony internetowej»

Strona internetowa lub strona internetowa to zbiór stron internetowych i zależnych treści dostępnych w Internecie, które są połączone zarówno w treści, jak i nawigacji pod jedną nazwą domeny.

Witryna jest również nazywana węzłem internetowym, komputerem, do którego przypisany jest unikalny adres IP, i ogólnie każdym obiektem w Internecie, który ma adres identyfikujący go w sieci (witryna FTP, witryna WWW itp.).

Zestaw połączonych ze sobą zasobów informacyjnych online przeznaczonych do przeglądania w sieci komputerowej za pomocą specjalnych programów — przeglądarek. Strona internetowa może być zestawem dokumentów elektronicznie, usługą online.

### 4.1. Języki programowania stron internetowych

Nie każdy, kto ma własną stronę internetową, musi znać technologie użyte do jej stworzenia. Niemniej jednak taka wiedza może być dla nas bardzo przydatna i niewątpliwie będzie cenna przy kontaktowaniu się z agencją IT, która stworzy taką stronę internetową.

#### 4.1.1. Technologie programowania stron internetowych

##### 4.1.1.1. Projektowanie stron internetowych

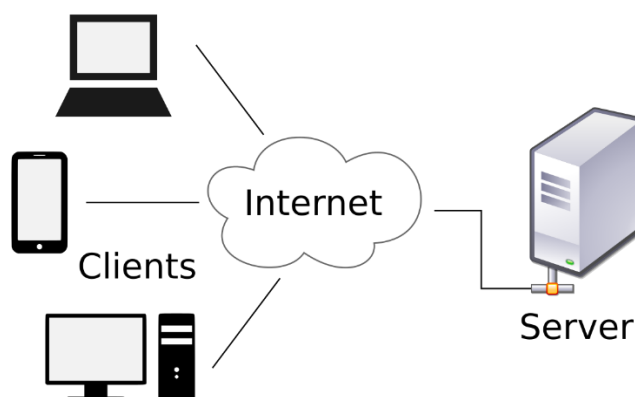
- Język znaczników HTML to kod używany do strukturyzowania i wyświetlania strony internetowej i jej zawartości
- CSS jest w rzeczywistości językiem stylów, który jest odpowiedzialny za wyświetlanie dokumentów HTML. CSS działa z czcionkami, z kolorami symboli i tła, z marginesami, z liniami, z wysokością i szerokością elementów wyświetlanych, z obrazami tła, z pozycjonowaniem elementów i wiele więcej
- SASS<sup>17</sup> to preprocesor CSS, którego można użyć do zmniejszenia ilości powtarzanego kodu i zaoszczędzenia czasu. Jest to bardziej stabilne rozszerzenie CSS, które jasno i strukturalnie opisuje style dokumentu.
- LESS<sup>18</sup> to dodatek do CSS. Ma wszystkie podstawowe możliwości preprocesorów, a nawet więcej, ale nie ma warunkowych konstrukcji i cykli w zwykłym dla nas znaczeniu. Główną zaletą jest jego prostota, niemal standardowa składnia css oraz możliwość rozszerzenia funkcjonalności poprzez system wtyczek.

<sup>17</sup> SASS - Syntactically Awesome Style Sheets

<sup>18</sup> LESS - Leaner Style Sheets

#### 4.1.1.2. Programowanie po stronie klienta

- Język programowania JavaScript jest najpopularniejszym i obsługiwanym językiem programowania przez wszystkie nowoczesne przeglądarki internetowe. Jest on przeznaczony do tworzenia stron internetowych "na żywo". Skrypty w przeglądarce są połączone bezpośrednio z KODEM HTML i gdy tylko strona się załaduje, są natychmiast wykonywane
- React to biblioteka JavaScript używana do tworzenia interfejsu użytkownika
- VueJS to biblioteka JavaScript do tworzenia interfejsów internetowych przy użyciu szablonu architektury MVVM (Model-View-ViewModel).
- AngularJS to framework JavaScript przeznaczony do tworzenia jednostronicowych aplikacji internetowych
- jQuery to biblioteka JavaScript zaprojektowana w celu uproszczenia skryptów podczas pracy z węzłami elementów HTML w przeglądarce lub do pracy w przeglądarce bez interfejsu graficznego
- AJAX to biblioteka, która znacznie upraszcza i przyspiesza pisanie kodu JavaScript, a także pozwala na pracę ze wszystkimi przeglądarkami

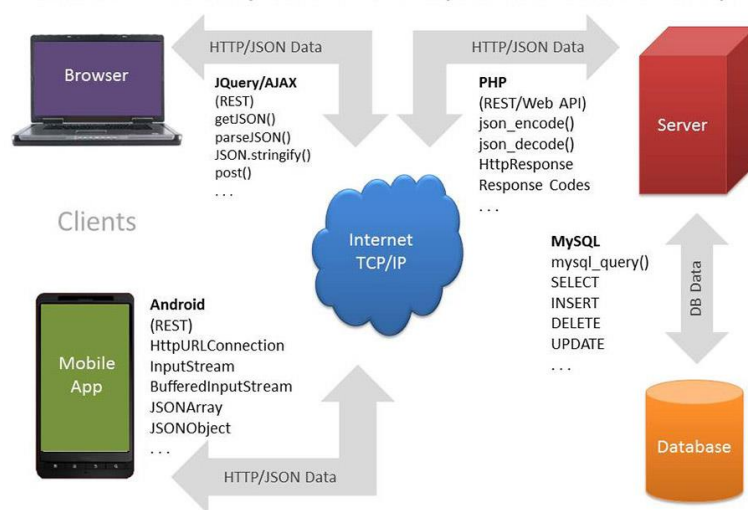


Rys. 4.4. - Model klient-serwer

#### 4.1.1.3. Programowanie po stronie serwera

- Język programowania PHP jest najpopularniejszym językiem programowania www po stronie serwera. PHP jest obsługiwany przez większość dostawców hostingu
- Node.js jest środowiskiem do wykonywania kodu JavaScript, jest to po prostu kolejny sposób wykonywania kodu na komputerze
- ASP.NET. — model tworzenia aplikacji internetowych wykorzystujący minimum kodu zawierającego usługi niezbędne do tworzenia aplikacji internetowych dla przedsiębiorstw

- Python to język programowania ogólnego przeznaczenia, którego celem jest przede wszystkim poprawa produktywności samego programisty.
- Ruby on Rails to internetowe środowisko programistyczne typu open source zoptymalizowane pod kątem łatwego programowania i zrównoważonej wydajności



Rys. 4.5. - Architektura klient-serwer

#### 4.1.2. HTML

**HTML** (z języka angielskiego. HyperText Markup Language to ustandaryzowany język znaczników hipertekstowych do przeglądania stron internetowych w przeglądarce. Przeglądarki internetowe odbierają dokument HTML z serwera przy użyciu protokołów HTTP<sup>19</sup>/HTTPS<sup>20</sup> lub otwierają go z dysku lokalnego, a następnie interpretują kod w interfejs, który będzie wyświetlany na ekranie monitora.<sup>21</sup>



Rys. 4.6. – Tak wygląda kod HTML i jego wynik

<sup>19</sup> **HTTP** - Hypertext Transfer Protocol

<sup>20</sup> **HTTPS** - Hypertext Transfer Protocol Secure

<sup>21</sup> **Chris Minnick, Ed Tittel** - HTML5 CSS3 Dla manekinów, 2013

#### 4.1.2.1. Struktura dokumentu HTML

**HTML** jest oznakowanym językiem znaczników dla dokumentów. Każdy dokument w HTML jest zbiorem elementów, a początek i koniec każdego elementu jest oznaczony specjalnymi znacznikami - **znacznikami**. elementy mogą być puste, to znaczy nie zawierają żadnego tekstu i innych danych. W tym przypadku znacznik zamykający zwykle nie jest wskazany (na przykład znacznik zawijania linii `<br>` - pojedynczy i nie trzeba go zamykać) . elementy mogą mieć atrybuty definiujące dowolne z ich właściwości (na przykład atrybut `href=` w odwołaniu). Atrybuty są określone w znaczniku otwierającym.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Document</title>
8  </head>
9  <body>
10  <
11  </body>
12 </html>

```

Rys. 4.7. – Przykład użycia

HTML używa znaczników do wyświetlania tekstu, obrazów i innej zawartości w przeglądarce internetowej. Znaczniki HTML zawierają specjalne " **elementy**", takie jak `<head>`, `<title>`, `<body>`, `<header>`, `<footer>`, `<article>`, `<section>`, `<p>`, `<div>`, `<img>`, `<side>`, `<audio>`, `<canvas>`, `<datalist>`, `<details>`, `<bed>`, `<nav>`, `<output>`, `<progress>`, `<video>` i wiele innych.

**Element HTML `<head>`** zawiera informacje o dokumencie nadające się do odczytu maszynowego (metadane), takie jak tytuł, skrypty i strony stylów.

```

1  <!doctype html>
2  <html>
3    <head>
4      <title>Tytuł strony</title>
5    </head>
6  </html>

```

Rys. 4.8. - Przykład użycia

**Element HTML nagłówek <tytuł>** ( Na rysunku 1. 9)definiuje tytuł dokumentu, który jest wyświetlany w tytule okna przeglądarki lub na karcie strony. Zawiera tylko tekst, a znaczniki wewnątrz elementu są ignorowane.

Element **<tytuł>** jest zawsze używany wewnątrz **bloku <głowa>**.

```

1  <head>
2  <title>Oszałamiający nagłówek strony</title>
3  </head>

```

Rys. 4.9. – Przykład użycia

**Element HTML <body>** ( Rysunek 1. 10 ) to treść (zawartość) dokumentu HTML. W dokumencie może znajdować się tylko jedna <ciało>.

```

1  <body>
2  <p>To jest akapit</p>
3  </body>

```

Rys. 4.10. - Przykład użycia

### 4.1.3. CSS

CSS to specjalny język stylów dla stron używanych do opisywania ich wyglądu. Same strony są napisane w językach znaczników danych.

```

1  body {
2  → margin: 0;
3  → font-family: 'Montserrat', sans-serif;
4
5  → font-size: 15px;
6  → line-height: 1.6;
7  → color: #333;
8  }
9
10 *
11 *:before,
12 *:after {
13 → box-sizing: border-box;
14 }
15
16 h1, h2, h3, h4, h5, h6 {
17 → margin: 0;
18 }
19

```

Rys. 4.11. - Pole z kodem CSS



### 4.1.3.1. Zalety CSS

- CSS oszczędza czas – możesz napisać CSS raz, a następnie ponownie użyć tego samego arkusza na wielu stronach HTML. Można zdefiniować styl dla każdego elementu HTML i zastosować go do jak największej liczby stron internetowych.
- Strony ładują się szybciej – jeśli używasz CSS, nie musisz za każdym razem pisać atrybutów tagów HTML. Wystarczy napisać jedną regułę znacznika CSS i zastosować ją do wszystkich przypadków tego znacznika. Tak więc mniej kodu oznacza szybsze ładowanie.
- Łatwość konserwacji - Aby dokonać globalnej zmiany, po prostu zmień styl, a wszystkie elementy na wszystkich stronach internetowych zostaną automatycznie zaktualizowane.
- Ulepszone style do HTML - CSS ma znacznie szerszy zestaw atrybutów niż HTML, dzięki czemu możesz dać znacznie lepszy widok strony HTML w porównaniu do atrybutów HTML.
- Zgodność z wieloma urządzeniami — arkusze stylów umożliwiają optymalizację zawartości dla więcej niż jednego typu urządzenia. Korzystając z tego samego dokumentu HTML, różne wersje strony internetowej można przysyłać na urządzenia przenośne, takie jak PDA i telefony komórkowe, lub do drukowania.<sup>22</sup>

### 4.1.3.2. Funkcje CSS

Dzięki CSS możesz wykonywać ogromną (i stale rosnącą) liczbę zadań.

- określ typ czcionki, rozmiar, kolor i użyte efekty;
- wybierz kolory tła i obrazy;
- kontroluj wiele aspektów pozycjonowania tekstu, w tym wyrównanie i odstępy;
- ustawianie pól i obramowań;
- kontrolować wyświetlanie list;
- określić strukturę tabeli i jej wyświetlanie;
- automatyczne generowanie treści dla standardowych elementów strony, takich jak liczniki i stopki;
- kontrolować wyświetlanie kursora;
- tworzenie przejść;
- animowanie wartości właściwości CSS za pomocą klatek kluczowych animacji;
- tworzenie struktur multi-collier;
- używaj czcionek pochodnych spośród tysięcy dostępnych na stronach internetowych.
- definiowanie dźwiękowych arkuszy stylów dla urządzeń.<sup>23</sup>

<sup>22</sup> Chris Minnick, Ed Tittel - HTML5 CSS3 Dla manekinów, 2013

<sup>23</sup> Chris Minnick, Ed Tittel - HTML5 CSS3 Dla manekinów, 2013

### 4.1.4. JavaScript

JavaScript to język programowania dla sieci. Zdecydowana większość stron internetowych korzysta z JavaScript, a wszystkie nowoczesne przeglądarki internetowe – na komputery stacjonarne, konsole do gier, tablety elektroniczne i smartfony – zawierają interpreter JavaScript, dzięki czemu JavaScript jest najczęściej używanym językiem programowania w historii. JavaScript jest jedną z trzech technologii, które powinien znać każdy twórca stron internetowych: język znaczników HTML, który pozwala definiować zawartość stron internetowych, język stylów CSS, który pozwala definiować wygląd stron internetowych oraz programowanie JavaScript język, który pozwala definiować zachowanie stron internetowych.<sup>24</sup>



```
$(function () {
    var header = $("#header"),
        introH = $("#intro").innerHeight(),
        scrollOffset = $(window).scrollTop();

    /* Fixed Header */
    checkScroll(scrollOffset);

    $(window).on("scroll", function () {
        scrollOffset = $(this).scrollTop();
        checkScroll(scrollOffset);
    });
});
```

Rys. 4.12. - Pole z kodem JS

#### 4.1.4.1. Dlaczego JavaScript?

Kiedy Tworzono JavaScript, nosił on inną nazwę – "**LiveScript**", jednak język Java był wówczas bardzo popularny i zdecydowano, że pozycjonowanie JavaScript jako "młodszego brata" Javy będzie przydatne.

Z biegiem czasu JavaScript stał się całkowicie niezależnym językiem z własną specyfikacją o nazwie **ECMAScript**, a teraz nie ma nic wspólnego z Javą.

Obecnie JavaScript może być wykonywany nie tylko w przeglądarce, ale także na serwerze lub na dowolnym innym urządzeniu, które ma specjalny program o nazwie "silnik" JavaScript.

Przeglądarka ma swój własny silnik, który jest czasami nazywany "maszyną wirtualną JavaScript".

Różne silniki mają różne "nazwy kodowe". Na przykład:

- V8 – w Chrome, Opera i Edge.
- SpiderMonkey – w Firefoksie.

<sup>24</sup> **David** Flanagan - JavaScript. Ostateczny przewodnik, wydanie 6

Te tytuły są przydatne, ponieważ są często używane w artykułach dla programistów. Będziemy z nich również korzystać. Na przykład, jeśli "Funkcja X jest obsługiwana przez V8", to "X" najprawdopodobniej działa w Chrome, Operze i Edge<sup>25</sup>

#### 4.1.4.2. Co JavaScript może zrobić w przeglądarce?

Nowoczesny JavaScript jest "bezpiecznym" językiem programowania. Nie zapewnia dostępu niskiego poziomu do pamięci ani procesora, ponieważ został pierwotnie stworzony dla przeglądarek, które go nie wymagają.

Możliwości JavaScript są w dużym stopniu zależne od środowiska, w którym działa. Na przykład Node.JS obsługuje funkcje odczytu / zapisu dowolnych plików, wykonywania żądań sieciowych itp.

Przeglądarka JavaScript ma wiele wspólnego z manipulowaniem stronami internetowymi, interakcją użytkownika i interakcjami z serwerem WWW.

Na przykład w przeglądarce Java można:

- Dodaj nowy kod HTML do strony, zmień istniejącą zawartość, zmodyfikuj style.
- Reaguj na akcje użytkownika, klikając myszą, przesuwając wskaźnik, naciskając.
- Wysyłaj żądania sieciowe do zdalnych serwerów, przesyłaj i pobieraj pliki (technologie AJAX i COMET).
- Odbieraj i ustawiaj pliki cookie, zadawaj pytania odwiedzającemu, pokazuj wiadomości.
- Zapamiętywanie danych dla klienta ("pamięć lokalna").

#### 4.1.4.3. Czego nie może zrobić JavaScript w przeglądarce?

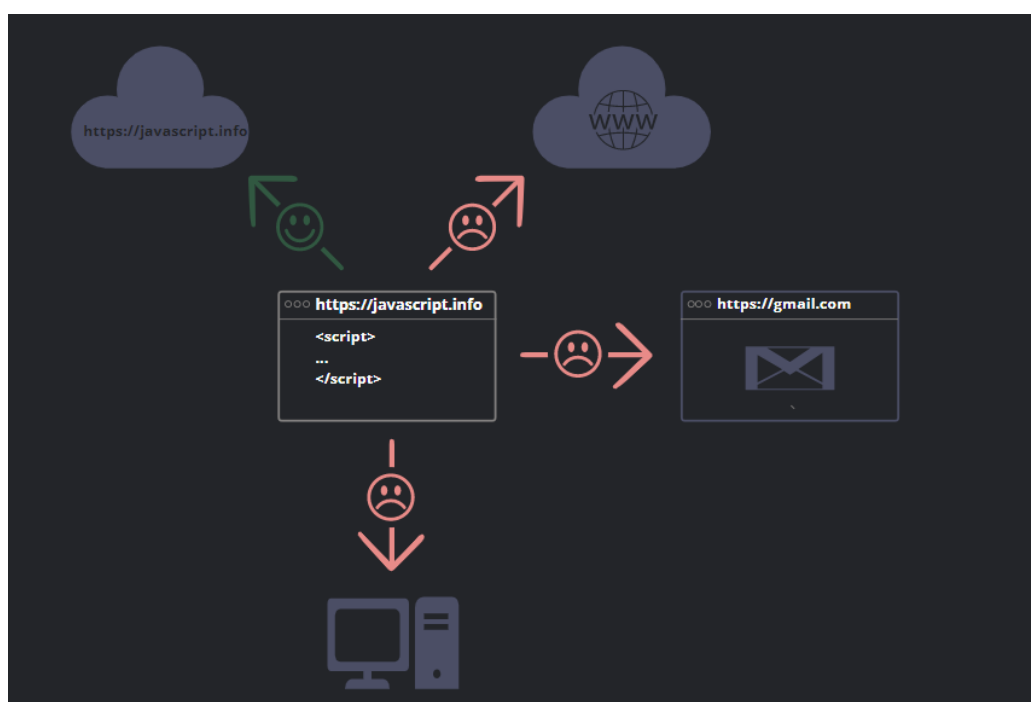
Możliwości JavaScript w przeglądarce są ograniczone ze względu na bezpieczeństwo użytkownika. Celem jest zapobieżenie dostępowi pozbawionej skrupułów stronie internetowej do danych osobowych lub spowodowaniu uszkodzenia danych użytkownika.

Przykłady takich ograniczeń obejmują:

- JavaScript na stronie internetowej nie może odczytywać/zapisywać dowolnych plików na dysku twardym, kopiować ani uruchamiać programów. Nie ma bezpośredniego dostępu do funkcji systemu operacyjnego.
- Nowoczesne przeglądarki pozwalają mu pracować z plikami, ale z ograniczonym dostępem, i zapewniają go tylko wtedy, gdy użytkownik wykonuje określone czynności, takie jak przeciąganie pliku do okna przeglądarki lub wybieranie go za pomocą znacznika `<input>`.
- Różne okna/karty nie wiedzą o sobie nawzajem. Czasami jedno okno, używając JavaScript, otwiera inne okno. Mimo to JavaScript z jednej strony nie ma dostępu do innej, jeśli pochodzi z różnych witryn (z innej domeny, protokołu lub portu).

<sup>25</sup> <https://geek.justjoin.it/wszystko-co-musicie-wiedziec-o-javascript-co-to-dla-kogo-i-ile-zarobimy/>

- Nazywa się to Polityką tego samego pochodzenia. Aby obejść to ograniczenie, obie strony muszą wyrazić na to zgodę i zawierać kod JavaScript, który w szczególności wymienia dane.
- To ograniczenie jest konieczne, ponownie, dla bezpieczeństwa użytkownika. Strona <https://anysite.com> otwarta przez użytkownika nie powinna mieć dostępu do innej karty przeglądarki z adresem URL <https://gmail.com> i kradzieży informacji stamtąd.
- JavaScript może łatwo wchodzić w interakcje z serwerem, z którego pochodzi bieżąca strona. Ale jego zdolność do wyodrębniania danych z innych witryn / domen jest ograniczona. Chociaż jest to możliwe w zasadzie, co wymaga wyraźnej zgody (wyrażonej w nagłówkach HTTP) ze stroną zdaną. Ponownie, jest to ograniczenie bezpieczeństwa.



Rys. 4.13. - Z czym JavaScript nie działa w Internecie

#### 4.1.5. Node.js

Node.js to litera N w MEAN. To, że zajmuje ostatnie miejsce, nie oznacza, że jest mniej ważny od pozostałych komponentów - w rzeczywistości jest to szkielet stosu!

W dużym skrócie: Node.js to platforma programowa, która pozwala na stworzenie własnego serwera WWW i budowanie na nim aplikacji internetowych. Sam Node.js nie jest serwerem internetowym ani językiem programowania. Zawiera wbudowaną bibliotekę serwera HTTP, stąd nie trzeba uruchamiać osobnego serwera WWW, takiego jak **Apache** czy **Internet Information Services (IIS)**. Ostatecznie zyskujesz większą kontrolę nad swoim serwerem internetowym, czyniąc go bardziej

skomplikowanym do skonfigurowania i uruchomienia, zwłaszcza w środowisku produkcyjnym.<sup>26</sup>

```
var http = require('http'); // Pobierz moduł http

// Utwórz serwer internetowy i określ funkcję przetwarzania żądań
var server = http.createServer(function (req, res) {
  console.log('Rozpoczęcie przetwarzania wniosku');
  // Przekazanie kodu odpowiedzi i nagłówków
  res.writeHead(200, {
    'Content-Type': 'text/plain; charset=UTF-8'
  });
  res.end('Hello world!');
});

// Uruchomienie serwera WWW
server.listen(1991, "127.0.0.1", function () {
  console.log('Serwer działa pod adresem. http://127.0.0.1:1991/');
});
```

Rys. 4.14. - Przykład programu, który uruchamia serwer WWW wyświetla komunikat w konsoli i odpowiada na każde żądanie HTTP komunikatem "Hello World"

## 4.1.6. React

React to narzędzie do budowania interfejsów użytkownika. Jego głównym zadaniem jest zapewnienie, aby to, co można zobaczyć na stronach internetowych, zostało wyświetlone na ekranie. React znacznie ułatwia tworzenie interfejsów, dzieląc każdą stronę na małe fragmenty. Te fragmenty nazywamy komponentami.<sup>27</sup>



Rys. 4.15. - przykład podziału strony na komponenty

### 4.1.6.1. Komponent React

Komponent React to, mówiąc najprościej, fragment kodu, który jest częścią strony internetowej. Każdy komponent jest funkcją JavaScript, która zwraca fragment kodu reprezentujący fragment strony.

<sup>26</sup> Simon Holmes - Getting MEAN with Mongo, Express, Angular 2017

<sup>27</sup> <https://global4net.com/ecommerce/czym-jest-react-i-jakie-ma-zalety/>

Aby utworzyć stronę, wywołujemy te funkcje w określonej kolejności, zbieramy wyniki wywołań i pokazujemy je użytkownikowi.  
Zapiszmy komponent w tagu `<script>` pliku `index.html` z typem ustawionym na „text/babel”

```
<script type="text/babel">
  function OurFirstComponent() {
    return (
      // Oto kod, który reprezentuje element interfejsu użytkownika
    );
  }
</script>
```

Rys. 4.16. – Kod na React

Kiedy wywołujemy funkcję `OurFirstComponent()`, zwracany jest fragment strony. React używa języka programowania o nazwie JSX, który jest podobny do HTML, ale działa wewnątrz JavaScript, co odróżnia go od HTML. Możesz tutaj dodać zwykły kod HTML, aby wprowadzić go do interfejsu użytkownika:

```
<script type="text/babel">
  function OurFirstComponent() {
    return (
      <h1>Hello, I am a React Component!</h1>
    );
  }
</script>
```

Rys. 4.17. – Kod z funkcji `OurFirstComponent()`.

## 5. Rozdział V «Opis struktury i funkcjonowania baz danych»

### 5.1. Co to są bazy danych?

Baza danych (bazy danych) – zbiór danych zorganizowany zgodnie z koncepcją, który opisuje charakterystykę tych danych i relacje między ich elementami; ten zestaw obsługuje co najmniej jedną z aplikacji (zgodnie z normą ISO/IEC 2382:2015). Ogólnie rzecz biorąc, baza danych zawiera schematy, tabele, widoki, procedury składowane i inne obiekty. Dane w bazie danych są zorganizowane zgodnie z modelem organizacji danych. Tak więc nowoczesna baza danych, oprócz samych danych, zawiera ich opis i może zawierać środki do ich przetwarzania.

### 5.2. Typy baz danych

Istnieje wiele różnych typów baz danych. Wybór optymalnej bazy danych dla Twojego przedsiębiorstwa zależy od tego, w jaki sposób Twoja firma będzie wykorzystywać dane.

- **Relacyjne bazy danych.** Relacyjne bazy danych stały się niezwykle popularne w 1980 roku. Elementy w relacyjnej bazie danych są zorganizowane jako zestaw tabel zawierających kolumny i wiersze. Technologia relacyjnych baz danych zapewnia najbardziej wydajny i elastyczny sposób dostępu do ustrukturyzowanych informacji.
- **Obiektowe bazy danych.** Informacje w obiektowych bazach danych prezentowane są w postaci obiektów, podobnie jak w oprogramowaniu obiektowym.
- **Rozproszone bazy danych.** Rozproszona baza danych składa się z dwóch lub więcej plików w różnych lokalizacjach. różne sieci.
- **Hurtownie danych.** Centralna hurtownia danych - hurtownia danych to rodzaj bazy danych przeznaczonej głównie do szybkiego wykonywania zapytań i analiz.
- **Baza danych NoSQL<sup>28</sup>** lub nierelacyjna baza danych umożliwia przechowywanie i przetwarzanie danych nieustrukturyzowanych i częściowo ustrukturyzowanych (w przeciwieństwie do relacyjnych baz danych, które definiują, jak zorganizowane są wszystkie dane wprowadzone do bazy danych). Bazy danych NoSQL zyskały na popularności wraz z rozprzestrzenianiem się i rosnącą złożonością aplikacji internetowych.

---

<sup>28</sup> **NoSQL** - baza danych, która zapewnia mechanizm przechowywania i wyszukiwania danych różni się od podejścia do tabel relacyjnych baz danych.

- **Bazy danych wykresów.** Baza danych wykresów przechowuje dane w postaci encji i relacji encji.
- **Bazy danych OLTP.** Oltp Database to szybka analityczna baza danych przeznaczona do wykonywania dużej liczby transakcji przez wielu użytkowników.

### 5.3. Systemy zarządzania bazami danych (DBMS)

W kontekście baz danych warto zastanowić się nad koncepcją DBMS — systemu zarządzania bazami danych. treść i wyświetlanie informacji. Najczęstsze DBMS to MySQL, PostgreSQL, Oracle, Microsoft SQL Server. Wynajem hostingu współdzielonego od HOSTiQ.ua wymaga użycia MariaDB - odgałęzienia MySQL DBMS, a także PostgreSQL. Jeśli na przykład planujesz zakup VPS lub serwera w Europie lub USA, możesz sam wybrać oprogramowanie dla swoich baz danych.

### 5.4. Programy bazodanowe

#### Altibase

Platforma: Linux

Języki: C, C ++, PHP, wszystkie języki obsługujące ODBC lub JDBC.

Wersja w chmurze: Tak



Altibase to wysokowydajna i relacyjna baza danych typu open source. Altibase ma ponad 650 klientów korporacyjnych, w tym 8 firm z listy Fortune Global 500, i wdrożyła ponad 6000 krytycznych przypadków użycia w różnych branżach.

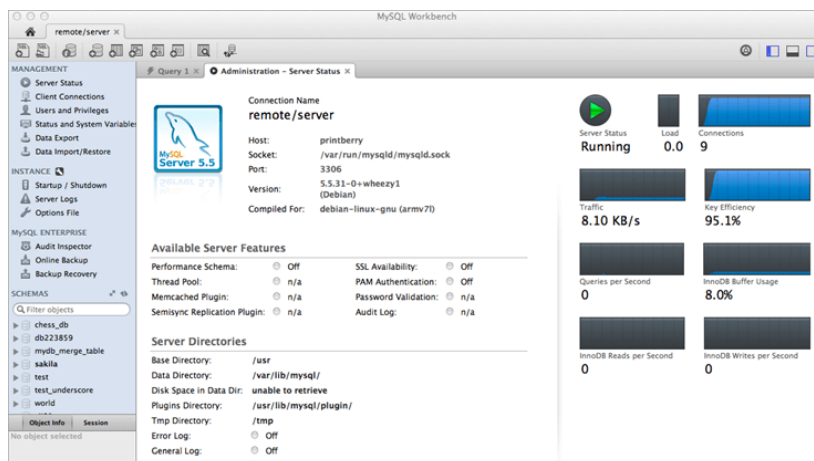
#### MySQL

Platforma: Windows, Linux i Mac.



Języki: SQL i C, C ++, Java, Perl, PHP, Python i Tcl do programowania klienta.

Wersja w chmurze: Tak



Rys.. 5.1. – Strona główna aplikacji «MySQL»

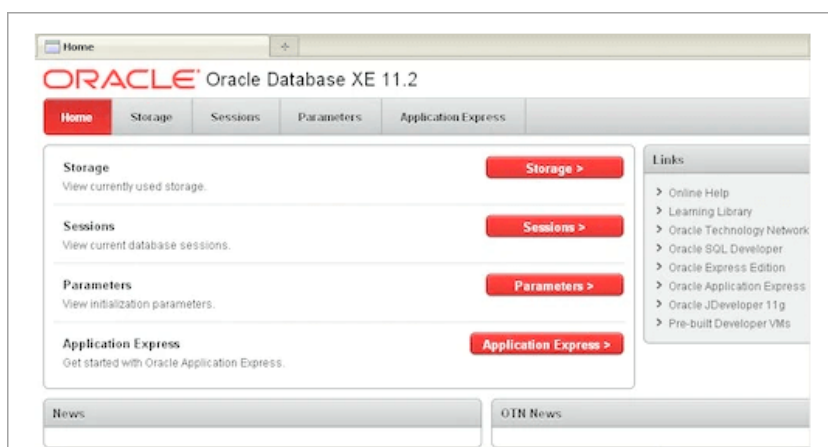
MySQL zwróci wysoką wydajność i pomoże Ci zbudować aplikacje na dużą skalę bez danych. Ta baza danych open source ma różne wersje, takie jak Enterprise Edition, Standard Edition i Classic Edition, MySQL oferuje różne funkcje dla każdego z nich. Zapewnia również usługę chmurową Oracle MySQL, która jest opłacalnym rozwiązaniem dla usługi bazy danych klasy korporacyjnej.

## Oracle

Platforma: Windows i Linux

Języki: C, C++, Java, COBOL, PL/SQL i Visual Basic.

Wersja w chmurze? Tak



Rys.. 5.1. – Strona główna aplikacji «Oracle»

Oracle oferuje zarówno rozwiązania lokalne, jak i chmurowe do zarządzania bazami danych. Oracle zapewnia również infrastrukturę chmury z elastycznymi konfiguracjami.

### 5.4.1. Porównanie wiodącego oprogramowania bazodanowego Open Source

Darmowe oprogramowanie bazodanowe	Podest	Ograniczony	Łatwość obsługi	Wersja w chmurze
Altibase	Linuksa	Bez ograniczeń	Bardzo łatwe	Tak
MySQL	Windows, Linux, Mac.	Bez ograniczeń	Łatwo	Tak
Wyrocznia	Windows, Linux	1 GB pamięci RAM 11 GB bazy danych 1procesor.	Średni	Tak

## 5.5. Język zapytań SQL

SQL. Język zapytań strukturalnych jest deklaratywnym językiem programowania do interakcji użytkownika z bazami danych, używanym do tworzenia zapytań, aktualizowania i zarządzania relacyjnymi bazami danych, tworzenia schematu bazy danych i modyfikowania go oraz kontrolowania systemów dostępu do bazy danych. Sam SQL nie jest ani systemem zarządzania bazami danych, ani oddzielnym oprogramowaniem. W przeciwieństwie do prawdziwych języków programowania (C lub Pascal), SQL może tworzyć interaktywne zapytania lub, będąc wbudowanym w programy aplikacyjne, działać jako instrukcje zarządzania danymi. Ponadto standard SQL zawiera funkcje definiowania zmian, weryfikacji i ochrony danych<sup>29</sup>

### 5.5.1. Kto używa języka SQL

SQL jest używany w różnych dziedzinach: bankowości, handlu, taksówkach i transporcie towarowym - wszędzie tam, gdzie gromadzi się wiele informacji i trzeba z nimi pracować.

SQL jest używany przez Ubera, YouTube, Netflix, Airbnb, Facebooka, Google, Amazona i innych gigantów. i oczywiście potrzebują pracowników do pracy z bazami danych: tworzenia i rozwijania baz danych, pisania zapytań SQL, analizowania wyników, grupowania i sortowania danych, ustawiania zadań dla programistów i testowania wyników.

<sup>29</sup> <https://www.oracle.com/pl/database/what-is-database/>

## 5.5.2. Podstawowe polecenia SQL

### POKAŹ BAZY DANYCH

Polecenie SQL jest odpowiedzialne za przeglądanie dostępnych baz danych.

### TWORZENIE BAZY DANYCH

Polecenie do utworzenia nowej bazy danych

### POKAŹ TABELĘ

Za pomocą tego prostego polecenia możesz zobaczyć wszystkie tabele dostępne w bazie danych.

### TWORZENIE TABELI

Polecenie SQL do utworzenia nowej tabeli

## 5.6. MongoDB

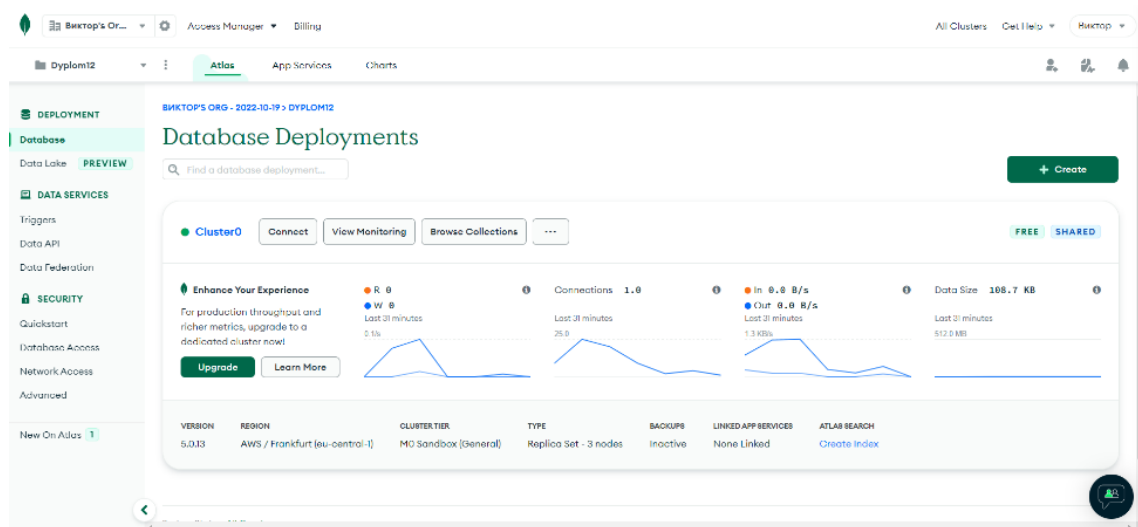
Możliwość przechowywania i wykorzystywania danych ma kluczowe znaczenie dla większości aplikacji. Wybraną bazą danych dla stosu MEAN jest MongoDB – litera M w MEAN. MongoDB wyjątkowo dobrze nadaje się do tego stosu. Mongo zostało stworzone w 2007 roku przez programistów jednego z działów Google, kiedy stanęli przed zadaniem, które trudno było rozwiązać za pomocą dostępnych im narzędzi. DBMS jest obecnie używany przez eBay, Adobe, Squarespace, Verizon i wiele innych korporacji<sup>30</sup>

---

<sup>30</sup> Simon Holmes - Getting MEAN with Mongo, Express, Angular 2017

## 5.6.1. Najlepsze aplikacje do korzystania z MongoDB

### MongoDB Compass

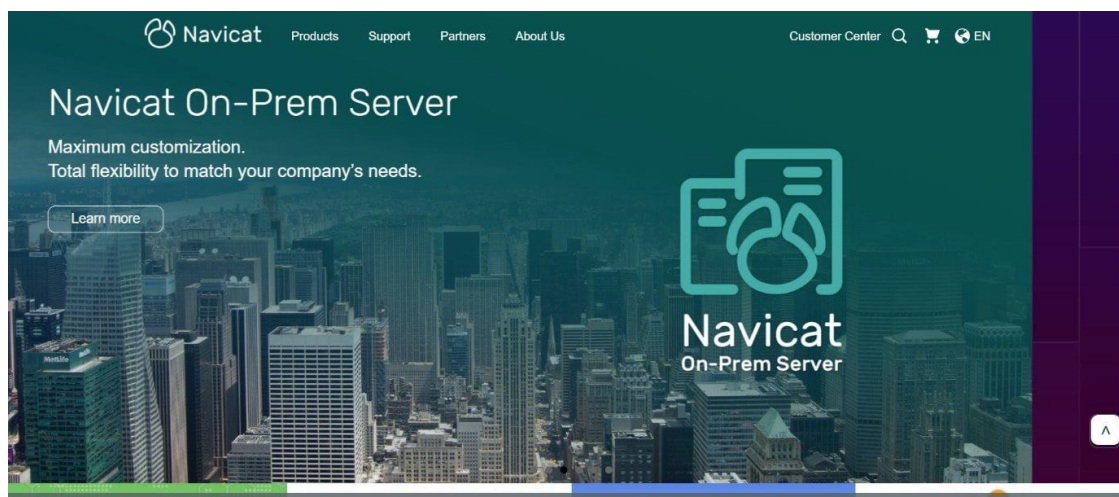


Rys.. 5.1. – Strona główna aplikacji «MongoDB Compass»

Ponieważ MongoDB jest właścicielem MongoDB Compass, jest zawsze aktualizowany o najnowsze modyfikacje MongoDB. Jest to jeden z najlepszych darmowych programów GUI MongoDB z następującymi funkcjami:

- Compass to interaktywny klient front-end MongoDB, który umożliwia interaktywne wyszukiwanie i eksplorację danych w tabelach MongoDB.
- Jest to najlepszy GUI MongoDB dla zespołów, aby znaleźć bezpłatne rozwiązanie, które może wysyłać zapytania i agregować dane tabeli MongoDB oraz je analizować.
- Oferuje wizualnie wyjaśnione strategie zapytań do monitorowania i dostrajania wydajności MongoDB.
- Inteligentnie zintegrowany edytor sprawdza poprawność schematu JSON.
- Oferuje obsługę wtyczek w celu zwiększenia funkcjonalności
- Metryki serwera i bazy danych w czasie rzeczywistym pomagają zapewnić płynne działanie bazy danych.
- MongoDB Compass jest całkowicie darmowy — głównie do wdrożenia lokalnego
- MongoDB Atlas jest oferowany w kilku planach dla w pełni zarządzanego rozwiązania chmurowego hostowanego na AWS lub Google Cloud.
- W przypadku aplikacji bezserwerowych koszt wynosi 0,30 USD. Milion odczytów w USA, podczas gdy dedykowana instancja MongoDB w chmurze kosztuje 57 USD. USA miesięcznie za 10 GB przestrzeni dyskowej i 2 GB pamięci RAM

## Navicat



Rys.. 5.2. – Strona główna aplikacji «Navicat»

**Navicat Premium** to rozwiązanie do tworzenia baz danych, które pomaga firmom każdej wielkości budować, zarządzać i utrzymywać bazy danych.

- Użytkownicy mogą łączyć się z MySQL, MariaDB, MongoDB, Microsoft Azure i Amazon Redshift, a także innymi lokalnymi i chmurowymi bazami danych.
- Funkcje obejmują wizualizację danych, fragment kodu i uzupełnianie, planowanie tworzenia kopii zapasowych, klastrowanie wirtualne i analizę schematu.
- Navicat Premium jest oferowany dla sektora komercyjnego i niekomercyjnego na okresy miesięczne, roczne lub jednorazowe.
- Program umożliwia programistom przenoszenie i synchronizację danych w wielu bazach danych przy użyciu wytycznych.
- Posiada zintegrowaną funkcję edycji, która pozwala profesjonalistom dodawać, modyfikować lub usuwać rekordy oraz przeglądać zaktualizowane dane w formie drzewa, siatki lub JSON.
- Navicat Premium pozwala administratorom baz danych używać kreatora zapytań do projektowania, modyfikowania i wykonywania zapytań SQL.
- Oferuje również generowanie kodów i identyfikowanie problemów z kodowaniem PL/SQL za pomocą komponentów do debugowania.

## 5.6.2. 5 powodów, by wybrać MongoDB

### SZYBKI ROZWÓJ

MongoDB to dobry wybór dla aplikacji MVP. Ważne jest, aby startup szybko pojawił się na rynku, aby zaczął pozyskiwać informacje o użytkownikach zainteresowanych produktem i go rozwijać. W tym celu uruchamiane są aplikacje MVP z minimalną liczbą funkcji. Muszą działać szybko i stabilnie, ale jednocześnie spokojnie tolerują nawet duże zmiany w strukturze i skali. Dzięki Mongo możesz zaoszczędzić czas na ścisłym formatowaniu dokumentów i uniknąć związanych z tym problemów.

### MODEL ZORIENTOWANY NA DOKUMENTY

MongoDB to najpopularniejsza niekomercyjna baza danych NoSQL. Dane w nim przechowywane są nie w ściśle powiązanych tabelach, ale w dokumentach JSON z informacjami, które niekoniecznie muszą mieć jednakową strukturę. System przechowuje dane bez wyraźnej hierarchii i pozwala na szybki dostęp do potrzebnych informacji, a co najważniejsze pogrupowanie ich w sposób niezbędny dla Twojego projektu. W MySQL i innych relacyjnych bazach danych wszelkie niestandardowe rozwiązania muszą być projektowane ręcznie, co trwa dłużej i może wiązać się ze znaczną liczbą błędów. Mongo ułatwia rejestrowanie i przechowywanie zdarzeń oraz danych operacyjnych. Nie ma znaczenia, czy uzyskujesz informacje z czujników lub komentarze i oceny odwiedzających, czy może przechowujesz unikalne dane od tysięcy użytkowników — we wszystkich przypadkach korzystanie z mongodb jest korzystne.

### ŁATWIEJSZY DO ZROZUMIENIA I NAUKI

Zamiast klasycznego języka zapytań SQL, MongoDB wykorzystuje CRUD, który ułatwia konstruowanie prostych zapytań. Ponadto jego składnia jest podobna do składni JavaScript, do której przyzwyczajeni są programiści aplikacji internetowych. Aby pracować ze złożonymi zadaniami, programiści będą mieli Aggregation Framework, specjalny interfejs obsługujący duży zestaw funkcji. Wbudowane narzędzie „Kompas” pomaga szybciej znaleźć potrzebne dane i podpowiada, jak rozwiązać zadanie.

### SKALOWANIE

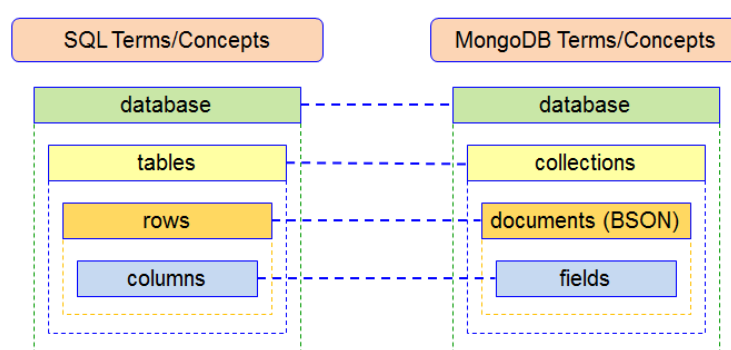
W przypadku mongodb korzyści zaczynają się od możliwości zwiększania liczby użytkowników aplikacji bez limitu. Nie ma znaczenia, czy chcesz przenieść wszystkie dokumenty z określonymi danymi na inny serwer, czy tylko część swojej bazy danych - z Mongo jest to dostępne od razu i nie wymaga konfigurowania skomplikowanych wtyczek. Dlatego używa się go w grach i startupach – jeśli projekt, trzeba będzie go szybko skalować, a z Mongo jest to najłatwiejsze. Możliwość

dowolnego zwiększania liczby węzłów w bazie danych, co częściowo jest osiągnięte dzięki zmniejszonej wydajności - aplikacja będzie zużywać więcej zasobów niż MySQL i Postgres do rozwiązywania tych samych zadań.

## FUNKCJONALNOŚĆ I ANALITYKA

W szczególności NoSQL DBMS i Mongo zawsze wygrywają pod względem funkcjonalności z relacyjnymi bazami danych, ponieważ zostały pierwotnie zaprojektowane do rozwiązywania określonych zadań z tym modelem danych, a nie jako uniwersalne narzędzie. API, transakcje ACID, złożone systemy wyszukiwania to tylko niewielka część funkcji, które od razu będą dostępne dla Twoich programistów i nie będą wymagały od nich pisania własnych rozwiązań. Analityka w MongoDB jest również dostępna od razu. BI Connector umożliwia dodawanie danych Mongo do dowolnego narzędzia Business Intelligence, które korzysta z protokołu MySQL. Możesz więc uzyskać pełną informację o danych, które Cię interesują i powiązaniach między nimi, niezależnie od tego, jak i gdzie są przechowywane. Użytkownicy MongoDB Atlas, zastrzeżonej pamięci masowej w chmurze Mongo, mają również wbudowane narzędzia analityczne, które mogą wizualizować dane w czasie rzeczywistym i nie tylko

### 5.6.3. Małe porównanie SQL z MongoDB



## 6. Rozdział VI «Kluczowe etapy tworzenia strony internetowej»

Tworzenie strony internetowej składa się z wielu kroków. Każdy z nich wymaga wiedzy i specjalisty o określonym profilu. Rozwój strony przypomina nieco budowę domu. Budowa podzielona jest na różne etapy – wylewanie fundamentu, budowanie ścian nośnych, zadaszenie itp., Jeśli jeden ze stopni zostanie pominięty, dom nie może zostać zbudowany, a jeśli zostanie zbudowany, nikt nie będzie mógł w nim mieszkać. Jeśli weźmiemy pod uwagę główne etapy rozwoju witryny, wszystko działa na tej samej zasadzie. Jeśli przynajmniej jeden z etapów jego tworzenia zostanie potraktowany nieuważnie lub nawet pominięty, cały pomysł traci swoje znaczenie, ponieważ zasób nie będzie w stanie normalnie pracować i wykonywać swojego zadania.

Klient musi zrozumieć, jakie etapy kryją się za tajemniczym terminem "tworzenie witryny" i dlaczego każdy z nich jest ważny.

### 6.1. Krok 1. Definiujemy cele

Przede wszystkim musisz zdecydować, do czego potrzebujesz witryny. Od tego będzie zależeć struktura i zawartość zasobu, a także sposoby promocji.

Zbadaj swoich odbiorców: kim są Twoi klienci i jak szukają informacji o produkcie lub usłudze. To zależy od tego, której witryny potrzebujesz. Dla tych, którzy mają już wielu klientów offline, zalecamy przeprowadzenie ankiety wśród klientów i pracowników.

W przypadku sklepów internetowych zalecamy rozpoczęcie badania handlu elektronicznego jako całości, aby zrozumieć, kto i jakie zamówienia w Internecie. I, oczywiście, nie zapomnij o konkurentach, którzy już stworzyli witryny. Zbadaj, w jaki sposób przyciągają klientów, jakie mają możliwości lub jakie błędy popełniają.

### 6.2. Krok 2. Wybór domeny i hostingu

Takie czynniki jak zapamiętywalność i obrazowanie nazwy domeny, strefa rejestracji domeny, obciążenie semantyczne nazwy domeny – wszystko to determinuje pierwsze wrażenie odwiedzającego i czasami może stać się momentem definiującym dla potencjalnego klienta w pytaniu "Czy jest to, czego potrzebuję na tej stronie?". Ale subiektywna ocena nazwy domeny witryny przez odwiedzającego to nie wszystko, o czym powinieneś pomyśleć przy wyborze domeny. Odpowiednio dobrana nazwa witryny może również pomóc w wyszukiwarce

Wybierając nazwę domeny, zawsze pamiętaj, że

- Nazwa powinna być łatwa do zapamiętania.
- Nazwa nie powinna być długa – im krótszy adres URL, tym łatwiej jest osobie wpisać go na klawiaturze i zapamiętać.



- Nazwa powinna mieć sens i kojarzyć się z Twoim biznesem – jeśli nazwa domeny jest związana z treścią strony, będzie zrozumiała dla odwiedzających i łatwiejsza do zapamiętania.
- Nazwa powinna dobrze wyglądać na materiałach reklamowych, wizytówkach i wszelkich innych nośnikach reklamowych, których użyjesz do promowania witryny.
- Wskazane jest, aby nazwa domeny zawierała słowa kluczowe (lub słowo) witryny, ponieważ guru SEO uczą, że niektóre wyszukiwarki zwracają na to uwagę, tasując wyniki wyszukiwania.
- Jeśli to możliwe, musisz wybrać nazwę w "najlepszej" strefie domeny, czyli w strefie, którą podświadomie przeciętna osoba próbuje wpisać po okresie. Jeśli weźmiemy Internet jako całość, to tutaj, oczywiście, powinieneś skupić się na strefie .COM. Jeśli twoja potencjalna publiczność mieszka głównie na Ukrainie, to .UA lub .COM. UA nie jest gorszy, a może nawet lepszy.
- Należy unikać używania cyfr w nazwie domeny, ponieważ wiele osób może kojarzyć cyfrę z literami, zwłaszcza jeśli po raz pierwszy nazwa domeny została odebrana przez ucho.
- Należy unikać używania łączników w nazwie domeny, aby nie kierować niepotrzebnego ruchu do witryny o tej samej nazwie, ale bez łącznika; jest mało prawdopodobne, że będziesz w stanie mieć taki sam ruch jak z zasobu bez łącznika.
- Staraj się nie używać skrótów w nazwie domeny, ponieważ są one trudne do zapamiętania. Oczywiście, jeśli twój skrót to GM, GE lub DG, możesz bez wahania śmiało i radośnie go używać.
- Upewnij się, że Twoja przyszła nazwa domeny nie narusza czyjegoes zarejestrowanego znaku towarowego, a nawet można go wybrać.

### 6.3. Krok 3. Sporządzamy zadanie techniczne

Zakres uprawnień, jak można się domyślić z nazwy, jest dokumentem, w którym zostaną zebrane wszystkie wymagania dotyczące przyszłej witryny. Stopień szczegółowości zależy od listy życzeń i złożoności zasobu. Wyróżnij główne sekcje:

- Opis celu i przeznaczenia witryny, a także krótka informacja o grupie docelowej. Aby twórcy witryny byli z tobą na tej samej długości fali, a jednocześnie pomogli ci wybrać format zasobu.
- Wymagania dotyczące projektu witryny. Ogólnie rzecz biorąc, opisz, w jaki sposób wyobrażasz sobie projekt przyszłego zasobu, wskaż pożądany schemat kolorów i, jeśli w ogóle, kolory korporacyjne. Również w tej sekcji musisz określić logo lub wskazać, co powinno być.
- Lista narzędzi do przeglądania witryn. Wymień przeglądarki i ich wersje, z których strona powinna być wyświetlana poprawnie, zwłaszcza jeśli specyfika biznesu wymaga wsparcia dla starszych wersji (na przykład dla sektora

bankowego). I nie zapomnij o optymalizacji pod kątem urządzeń mobilnych, aby Twoja witryna wyglądała schludnie na dowolnym urządzeniu.

- Wymagania dotyczące systemu zarządzania treścią. Określ listę podstawowych funkcji dla administratora witryny, na przykład dodając zdjęcia do portfolio lub zmieniając teksty na stronach, tworząc nowe sekcje lub zmieniając strukturę itp.
- Opis struktury strony. Przemyśl główne sekcje, w oparciu o Twoje produkty lub usługi, i wskaż, co w przybliżeniu znajdzie się na stronach (lub stronie w przypadku strony docelowej). Nie zapomnij przy planowaniu struktury uwzględnić żądania użytkowników, aby strona była łatwiejsza do promowania w wyszukiwaniu.

## **6.4. Krok 4. Pisanie tekstów**

Po ostatecznym uzgodnieniu struktury przyszłej witryny, musisz przygotować teksty na główne strony witryny:

- Strona główna. Pamiętaj, aby przygotować ten tekst do rozwoju witryny, aby zobaczyć, jak będzie wyglądać obok innych bloków. Informacje mogą wymagać uzupełnienia elementami graficznymi lub podzielenia na części.
- Strona usługi lub karta produktu. Napisz tekst dla jednej typowej strony, a następnie przez analogię wypełnij pozostałe.
- Strona o firmie. Warto wcześniej przygotować tekst i zastanowić się nad akcentami wizualnymi.

## **6.5. Krok 5. Projekt**

Ale teraz możesz już przemyśleć projekt projektu. Wielu błędnie zaczyna od tego etapu, więc w końcu dostają dobre, ale niepotrzebne strony.

Przed skontaktowaniem się z projektantem zalecamy dokonanie mini-wyboru witryn, które lubisz, ale nie zapomnij o cechach przedmiotu działalności. Skoncentruj się na konkurentach i zapoznaj się z trendami projektowymi, aby przemyśleć unikalny i nowoczesny design. Wybierając firmę lub freelancera, koniecznie spójrz na portfolio. Powinien ci się spodobać i, idealnie, zawierać przykłady na swój temat.

## **6.6. Krok 6. Optymalizacja**

Wspomnieliśmy już o optymalizacji tekstów, teraz przejdźmy do optymalizacji całej witryny. Podstawowa lista prac do wykonania:

- Tworzenie mapy trafności. Na podstawie wcześniej zebranego rdzenia semantycznego tworzymy plan rozmieszczenia słów kluczowych pomiędzy stronami serwisu.

- Zarejestruj swoją witrynę w Google Search Console, aby skonfigurować wyświetlanie informacji o witrynie dla wyszukiwarek.
- Dodanie mapy witryny (mapy witryny.xml), która pomoże przyspieszyć indeksowanie witryny.
- Tworzenie robotów.txt. Plik z instrukcjami dla robotów wyszukiwania, które strony wymagają lub nie muszą być indeksowane.
- Konfiguracja 404 stron. Pomoże to przekierować użytkowników z pustych sekcji na działające strony witryny.
- Podłączenie certyfikatu SSL, jeśli planujesz konto osobiste i / lub formularz zamówienia na swojej stronie internetowej.
- Pisownia znaczników i metatagów. Niezbędne, aby strony wyglądały atrakcyjnie w wyszukiwaniu.

Najlepiej jest połączyć specjalistę seo do wykonania tych prac, który później będzie zaangażowany w promocję strony internetowej.

## **6.7. Krok 7. Układ strony**

Etap kodowania to etap, na którym tworzona jest rzeczywista strona internetowa . W tej chwili programista stron internetowych kontroluje wszystkie prototypowe grafiki, frameworki i mapę witryny i wykorzystuje je do stworzenia funkcjonalnej strony internetowej.

Podczas pisania kodu strony internetowej wykorzystywane są technologie wymienione we wstępie do tego artykułu, takie jak HTML, CSS i JS. Proces ten podzielony jest na dwie części: frontend i backend.

Frontend , jak sama nazwa wskazuje, to rozwój strony internetowej, którą użytkownicy widzą w wyszukiwarce (na przykład animacja obrazu). Backend z kolei odnosi się do rozwoju bazy danych i strony żądania do serwera (na przykład wyświetlania ceny produktu).

[illegible]

Rys. 6.1. - Przykładowy kod witryny

## 6.8. Krok 8. Wypełnianie treści

Menedżer treści i copywriter są odpowiedzialni za treść w zespole programistycznym. Copywriter tworzy materiał tekstowy, a menedżer treści publikuje go. Oprócz tekstów mogą to być obrazy i filmy.

Ten etap może być równoległy z innymi etapami tworzenia. Najważniejsze jest, aby ustawić odpowiednie zadania dla wykonawców. Dzięki równoległemu wykonywaniu zadań można wykryć błędy i niespójności (na przykład nieprawidłowe wyświetlanie obrazów, przesunięcie tekstu itp.).

Ze względu na zawartość sklep wygląda na pełnoprawny i przyciąga więcej klientów. Opis produktu ujawnia cechy produktu i jego zastosowanie. Wysokiej jakości obrazy pomagają odwiedzającym zapoznać się z produktem. Poprzez opis sekcji użytkownicy poznają informacje o markach. Dobra treść pomaga sklepowi sprzedawać produkt

## 6.9. Krok 9. Testowanie

Na tym etapie tworzenia sklepu internetowego możesz zidentyfikować zarówno drobne błędy, jak i poważne naruszenia. Specjalista ds. Kontroli jakości sporządza plan testów, a także wykorzystuje listy kontrolne do wyraźnej identyfikacji błędów. Sprawdzana jest zarówno zgodność techniczna i poprawność strony, jak i percepcja wizualna i zgodność z wymaganiami zakresu uprawnień.

Testowanie jest wymagane do następujących celów:

- sprawdzenie wyświetlania sklepu internetowego na wszystkich urządzeniach określonych w OWU;

- sprawdzanie wyświetlacza we wszystkich przeglądarkach i dostępnych systemach operacyjnych;
- zgodność z parametrami określonymi w SIWZ (szybkość ładowania strony, wydajność itp.);
- analiza funkcjonalnego zachowania użytkownika (czy odpowiada ona zadaniom ustalonym przy opracowywaniu projektu);
- kontrola użyteczności;
- kontrola pracy usług stron trzecich (połączenie CRM);
- wyszukiwanie błędów technicznych.

## 7. Opis struktury aplikacji WWW

Aplikacja webowa do zakupu komponentów do inteligentnego domu. Składa się z jednej strony niezbędnej do zamówienia towarów do domu.

Strony aplikacji internetowych:

Na stronie początkowej (przy uruchamianiu aplikacji) znajduje się opis produktów, które można kupić, ile osób korzysta z naszej strony;

- strona autoryzacji użytkownika;
- «header», który zawiera linki do stron autoryzacyjnych i do koszyka
- strona o produktach
- «footer» z treścią tekstową, stworzony w celu lepszej prezentacji aplikacji;

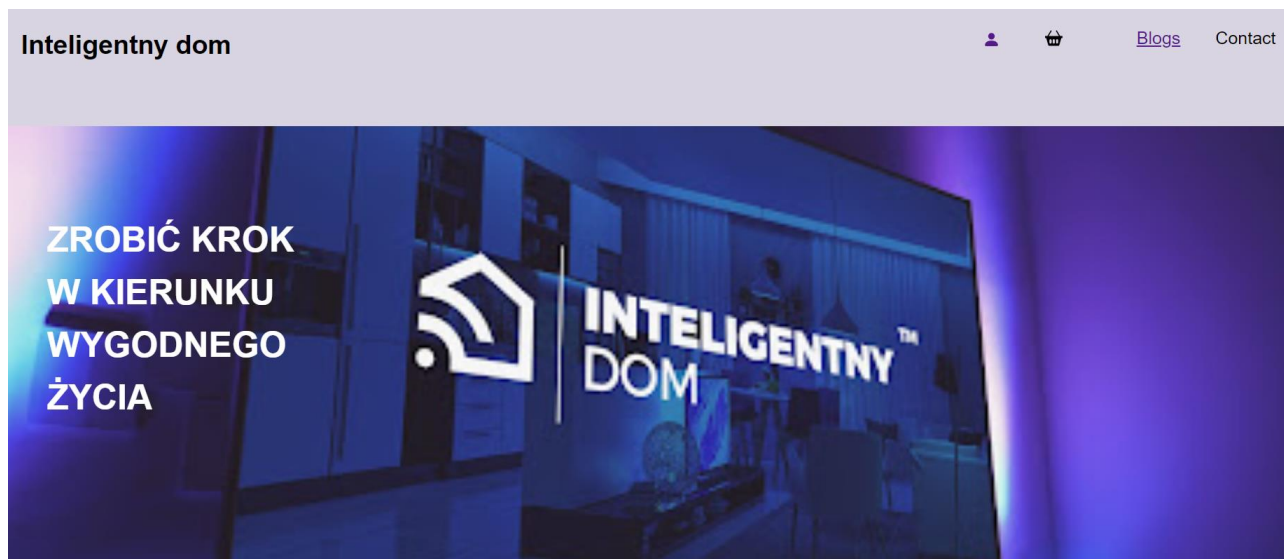
Każda strona zawiera nagłówki i stopki, które umożliwiają użytkownikowi nawigację do wszystkich dostępnych stron aplikacji z dowolnej strony.

## 8. Opis zawartości strony głównej

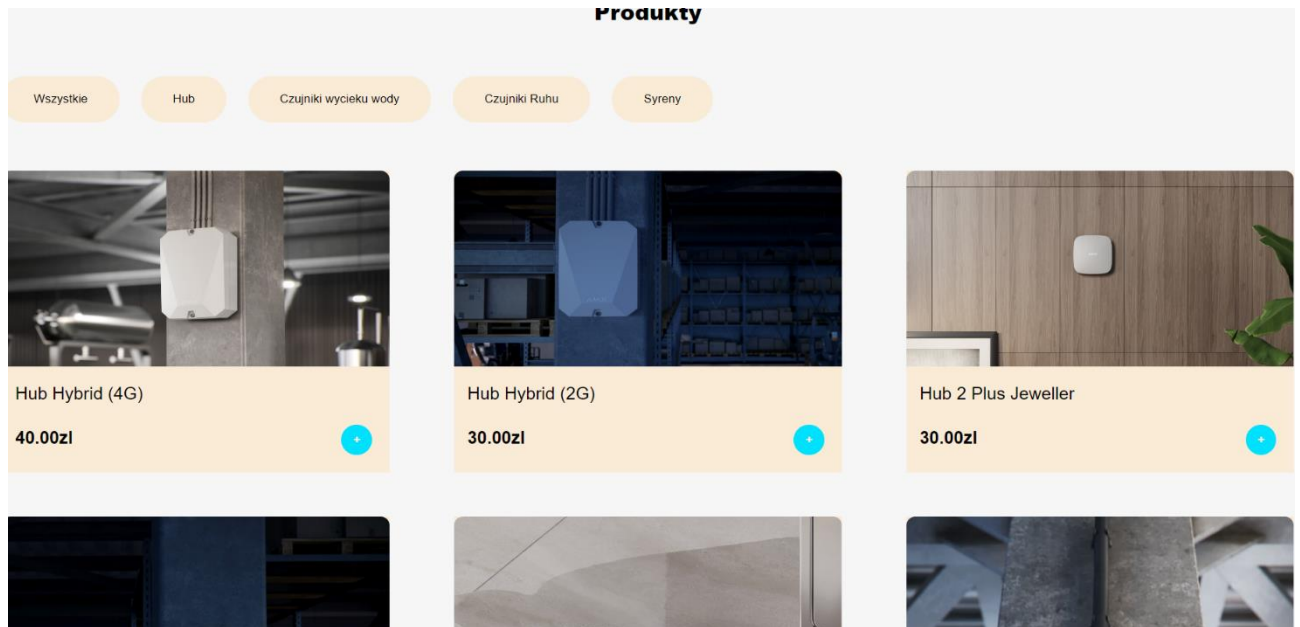
Strona główna aplikacji.

Strona główna zawiera nazwę sklepu. Również z tej strony użytkownik może przejść do autoryzacji oraz do koszyka. Jest też wyliczenie liczb "LUDZIE KORZYSTAJĄCY Z SYSTEMU INTELIGENTNEGO DOMU". ; "KLIENCI W KLUBIE" ; „ZAMÓWIENIE ZREALIZOWANE”.

Możesz zobaczyć jakie mamy produkty



Rys. 1. - Header strony głównej



Rys. 2. - Strona produktów



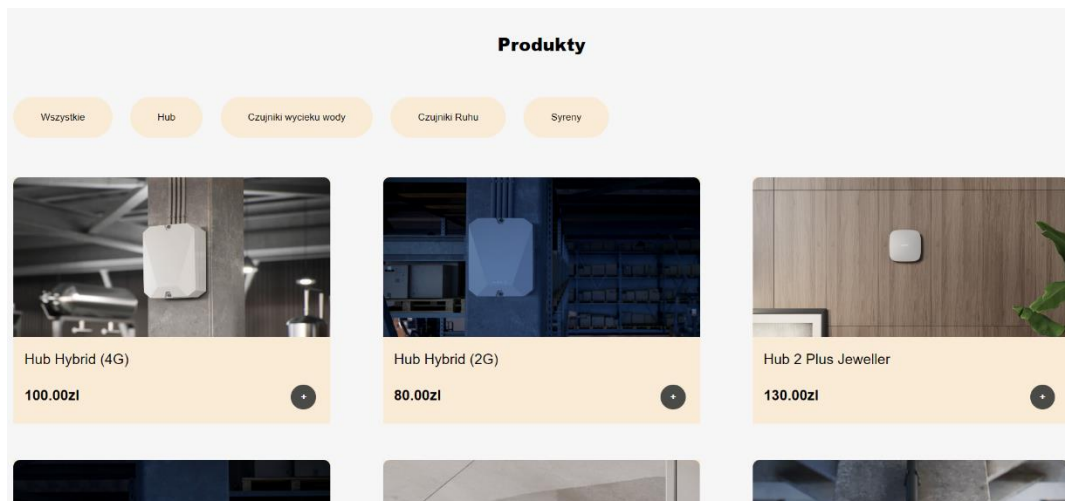
Rys. 3. - «Footer» aplikacja

## 9. Opis zawartości strony produktowej

Mamy 5 kategorii (rys. 4) („Wszystkie”; „Hub”; „Czujniki przepływu wody”; „Czujniki ruchu”; „Syreny”;). Jeśli klikniesz na „Czujniki wycieku wody”, pojawią się tylko „Czujniki wycieku wody”, a wszystkie inne znikną (Rys. 6).

Każdy produkt posiada opis oraz cenę (rys. 5). Jeśli chcemy zobaczyć jego opis, musimy kliknąć w zdjęcie. I pojawia się pełny opis produktu.

Produkt możemy dodać do koszyka. Jeśli w koszyku nic nie ma, pojawia się komunikat „Brak dostępnych towarów” (rys. 7). W koszyku możemy wyjąć towar, obliczyć ile kosztują wszystkie towary w koszyku (rys. 8).

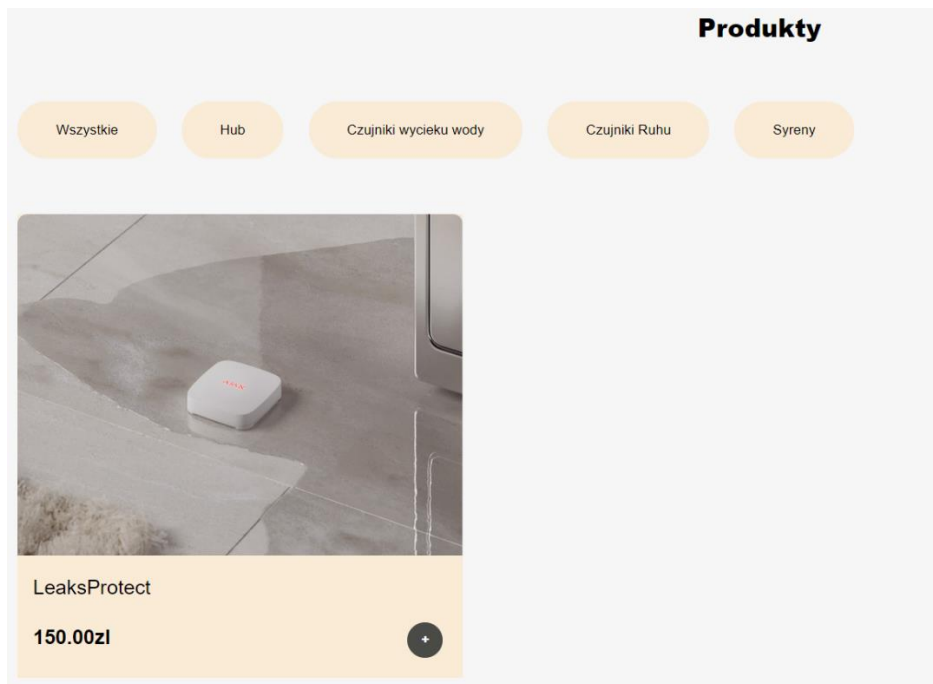


Rys. 4. – Kategoria produktów

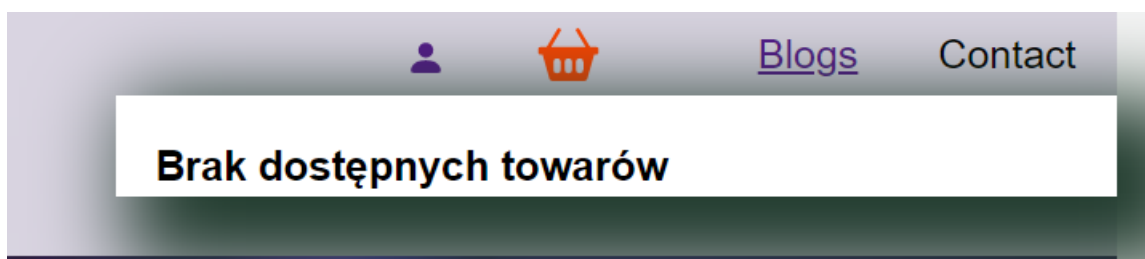


Rys. 5. – Opis produktów

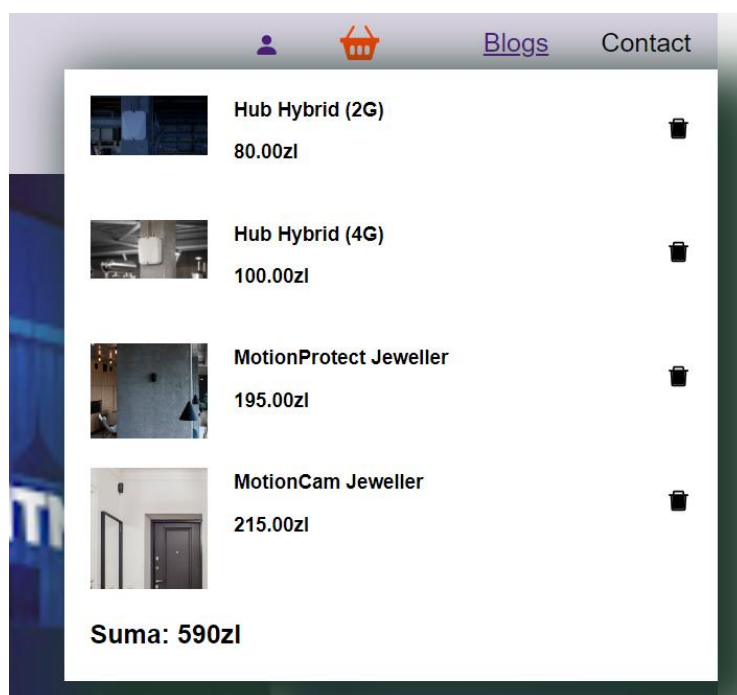




Rys. 6. – Jeśli klikniesz na kategorię "Czujniki wycieku wody"



Rys. 7. – Pusty koszyk



Rys. 8. – Koszyk zakupów

## 10. Opis strony autoryzacji użytkownika

Po kliknięciu autoryzacji (rys. 1) użytkownik przechodzi do strony autoryzacji (rys. 9). W celu zalogowania się użytkownik musi podać następujące dane: e-mail: hasło. Jeżeli użytkownik wpisał email niepoprawnie, konsola wyświetli problem (rys. 10). Jeśli użytkownik wprowadził błędne hasło, problem jest podświetlany w konsoli (rys. 13). A jeśli użytkownik zrobi wszystko poprawnie, to otrzymamy konsolę, jego dane oraz accessToken i odświeżenie Token (rys. 11) oraz Cookies (rys. 12).

localhost:3000/login

## Login/Registration

Email

Password

LOGIN

[Don't have an account? Sign Up here.](#)

Rys. 9. - Strona autoryzacyjna

```
{message: "Nie znaleziono użytkownika z tym adresem e-mail", errors: []}
  errors: []
  message: "Nie znaleziono użytkownika z tym adresem e-mail"
```

Rys. 10. – Użytkownik wprowadził błędny adres e-mail

```
[...]  
accessToken: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXCVJ9.eyJ1bmFpbCt6b3Q0NnBnbWpfc5jb28lClpZCTE6IjYzODMjMDk5MTZlbnVkbkNlcTlwNDcwWCIsIm1zQGw0aXZhZGVkIjpmeWxzZW5iaWF0IjoxNWJmYjQyOTMlClcHa10jE2NyAyl  
refreshToken: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXCVJ9.eyJ1bmFpbCt6b3Q0NnBnbWpfc5jb28lClpZCTE6IjYzODMjMDk5MTZlbnVkbkNlcTlwNDcwWCIsIm1zQGw0aXZhZGVkIjpmeWxzZW5iaWF0IjoxNWJmYjQyOTMlClcHa10jE2NyAyl  
vuser: {email: "rost45@gmail.com", id: "6380c298099e23bed7204700", isActive: false}  
    email: "rost45@gmail.com"  
    id: "6380c298099e23bed7204700"  
    isActive: false
```

Rys. 11. - Użytkownik wpisał prawidłowy e-mail

Name	Value	Domain	Path	Expires / ...	Size	HttpOnly	Secure	SameSite	SameParty	Partition ...	Priority
refreshToken	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFPbCl6InV3C3Q0...	localhost	/	30.0 days	320	✓					Medium

## Rys. 12. - Cookies

```
{message: "Nieprawidłowe hasło", errors: []}
  errors: []
  message: "Nieprawidłowe hasło"
```

Rys. 13. - Użytkownik wprowadził nieprawidłowe hasło

# Wnioski

W wyniku prac powstała aplikacja webowa do prostego zamawiania produktów «Inteligentny Dom».

Trafność wybranego tematu została zbadana poprzez przeprowadzenie ankiety i analizę korzystania z podobnych usług oraz ich popularności.

Przeprowadzono analizę istniejących rozwiązań programistycznych, porównując zalety i wady podobnych aplikacji dla różnych systemów i celów. Następnie wyciągnięto wniosek dotyczący stworzenia rozwiązania programistycznego w postaci aplikacji internetowej i wybrano metodę opracowania rozwiązania programistycznego.

Po przeanalizowaniu możliwych sposobów wdrożenia zdecydowano się na stworzenie aplikacji webowej w oparciu o Node.js i React.js w języku programowania JavaScript, do stworzenia części server oraz front-end. Użyłem MongoDB do stworzenia bazy danych. Użył aplikacji «Insomnia» do przetestowania żądań POST i GET.

Opracowane oprogramowanie posiada następujące funkcje:

- Autoryzacja użytkownika za pomocą adresu e-mail i password;
- Będzie można wybrać za pomocą filtra, wybrać produkty, których chce użytkownik;
- Dodawanie produktów do koszyka;

# Bibliografia

1. Eva Holland, Chris Minnick - Kodowanie za pomocą JavaScript for Dummies, 2015
2. Simon Holmes - Getting MEAN with Mongo, Express, Ang 2017
3. Chris Minnick, Ed Tittel - HTML5 CSS3 Dla manekinów, 2013
4. Davida Flanagan - JavaScript. Ostateczny przewodnik, wydanie 6, 2012
5. Inteligentny dom: [Zasób elektroniczny]. – 2023. – Tryb dostępu do zasobów:  
<https://www.sempire.pl/sklep-internetowy-definicja-rodzaje-funkcjonalnosci.html>
6. Inteligentny dom:: [Zasób elektroniczny]. – 2023. – Tryb dostępu do zasobów:  
[https://www.cstore.pl/blog/e-commerce-co-to-jest/#zalety\\_ecommerce](https://www.cstore.pl/blog/e-commerce-co-to-jest/#zalety_ecommerce)
7. Inteligentny dom:: [Zasób elektroniczny]. – 2023. – Tryb dostępu do zasobów:  
<https://www.x-kom.pl/poradniki/4913-inteligentny-dom-jaki-system-wybrac-czym-sie-kierowac.html>
8. BroadLink: [Zasób elektroniczny]. – 2023. – Tryb dostępu do zasobów:  
<https://www.ibroadlink.com/>
9. Ajax System: [Zasób elektroniczny]. – 2023. – Tryb dostępu do zasobów:  
<https://ajax.systems/pl/>
10. Sklep internetowy: [Zasób elektroniczny]. – 2023. – Tryb dostępu do zasobów:  
<https://thedigitalmarketing.pl/co-to-jest/e-commerce-definicja-co-to-jest/>
11. Stronie internetowej: [Zasób elektroniczny]. – 2023. – Tryb dostępu do zasobów: <https://pomoc.home.pl/baza-wiedzy/co-to-jest-strona-internetowa>
12. Stronie internetowej: [Zasób elektroniczny]. – 2023. – Tryb dostępu do zasobów: <https://www.programistkaikot.pl/2019/01/statyczne-czy-dynamiczne-strony.html>
13. JavaScript: [Zasób elektroniczny]. – 2023. – Tryb dostępu do zasobów:  
<https://www.w3schools.com/js/>
14. JavaScript: [Zasób elektroniczny]. – 2023. – Tryb dostępu do zasobów:  
<https://pl.reactjs.org/>
15. Node.js: [Zasób elektroniczny]. – 2023. – Tryb dostępu do zasobów:  
[https://www.w3schools.com/nodejs/nodejs\\_intro.asp](https://www.w3schools.com/nodejs/nodejs_intro.asp)
16. React: [Zasób elektroniczny]. – 2023. – Tryb dostępu do zasobów:  
<https://www.udemy.com/course/react-ru/>
17. Baza danych: [Zasób elektroniczny]. – 2023. – Tryb dostępu do zasobów:  
<https://www.oracle.com/pl/database/what-is-database/>
18. SQL: [Zasób elektroniczny]. – 2023. – Tryb dostępu do zasobów:  
<https://www.oracle.com/pl/database/what-is-database/>
19. SQL: [Zasób elektroniczny]. – 2023. – Tryb dostępu do zasobów:  
<https://uk.wikipedia.org/wiki/SQL>
20. MongoDB: [Zasób elektroniczny]. – 2023. – Tryb dostępu do zasobów:  
<https://uk.wikipedia.org/wiki/MongoDB>
21. MongoDB: [Zasób elektroniczny]. – 2023. – Tryb dostępu do zasobów:  
<https://codeguida.com/post/519>

# **Dodatek 1**

## **Listy programów**

# Serwer

## Stworzenie serwera na Node.js + połączenie bazy danych (MongoDB)

```
require('dotenv').config()
const express = require('express');
const cors = require('cors');
const cookieParser = require('cookie-parser')
const mongoose = require('mongoose');
const router = require('./router/index')
const errorMiddleware = require('./middlewares/error-middleware');

const PORT = process.env.PORT || 5000;
const app = express()

app.use(express.json());
app.use(cookieParser());
app.use(cors({}));
app.use('/api', router);
app.use(errorMiddleware);

const start = async () => {
  try {
    await mongoose.connect(process.env.DB_URL, {
      useNewUrlParser: true,
      useUnifiedTopology: true
    })
    app.listen(PORT, () => console.log(`Server started on PORT = ${PORT}`))
  } catch (e) {
    console.log(e);
  }
}

start ()
```

## ENV

```
PORT=51061
DB_URL=mongodb+srv://vityok:jk1mbfsdn545520@cluster0.klitdj4.mongodb.net/?retryWrites=true&w=majority
JWT_ACCESS_SECRET=jwt-secret-key
JWT_REFRESH_SECRET=jwt-refresh-secret-key
```

## Stworzenie «token-service»

```
const jwt = require('jsonwebtoken');
const tokenModel = require('../models/token-model');

class TokenService {
  generateTokens(payload) {
    const accessToken = jwt.sign(payload, process.env.JWT_ACCESS_SECRET,
{expiresIn: '15s'})
    const refreshToken = jwt.sign(payload, process.env.JWT_REFRESH_SECRET,
{expiresIn: '30s'})
    return {
      accessToken,
      refreshToken
    }
  }

  validateAccessToken(token){
    try {
      const userData = jwt.verify(token, process.env.JWT_ACCESS_SECRET);
      return userData;
    } catch (e) {
      return null;
    }
  }

  validateRefreshToken(token){
    try {
      const userData = jwt.verify(token, process.env.JWT_REFRESH_SECRET);
      return userData;
    } catch (e) {
```

```

        return null;
    }
}

//SaveToken
async saveToken(userId, refreshToken) {
    const tokenData = await tokenModel.findOne({user: userId})
    if (tokenData) {
        tokenData.refreshToken = refreshToken;
        return tokenData.save();
    }
    const token = await tokenModel.create({user: userId, refreshToken})
    return token;
}

//RemoveToken
async removeToken(refreshToken) {
    const tokenData = await tokenModel.deleteOne({refreshToken})
    return tokenData;
}

//FindToken
async findToken(refreshToken) {
    const tokenData = await tokenModel.findOne({refreshToken})
    return tokenData;
}
}

module.exports = new TokenService();

```

## Stworzenie «user-service»

```

const UserModel = require('../models/user-model');
const bcrypt = require('bcrypt');
const uuid = require('uuid');
const mailService = require('../mail-service');
const tokenService = require('../token-service');
const UserDto = require('../dtos/user-dto');

```



```

const ApiError = require('../exceptions/api-error');

class UserService {

  //Registration
  async registration(email, password) {
    const candidate = await UserModel.findOne({email})
    if (candidate) {
      throw ApiError.BadRequest(`Użytkownik z adresem pocztowym ${email} już istnieje`)
    }
    const hashPassword = await bcrypt.hash(password, 3);
    const activationLink = uuid.v4(); // v34fa-asfasf-142saf-sa-asf

    const user = await UserModel.create({email, password: hashPassword, activationLink})
    await mailService.sendActivationMail(email, `${process.env.API_URL}/api/activate/${activationLink}`);

    const userDto = new UserDto(user); // id, email, isActivated
    const tokens = tokenService.generateTokens({...userDto});
    await tokenService.saveToken(userDto.id, tokens.refreshToken);

    return {...tokens, user: userDto}
  }

  //Login
  async login(email, password) {
    const user = await UserModel.findOne({email})
    if (!user) {
      throw ApiError.BadRequest('Nie znaleziono użytkownika z tym adresem e-mail')
    }
    const isPassEquals = await bcrypt.compare(password, user.password);
    if (!isPassEquals) {
      throw ApiError.BadRequest('Nieprawidłowe hasło');
    }
    const userDto = new UserDto(user);

```

```

    const tokens = tokenService.generateTokens({...userDto});

    await tokenService.saveToken(userDto.id, tokens.refreshToken);
    return {...tokens, user: userDto}
  }

//Logout
  async logout(refreshToken) {
    const token = await tokenService.removeToken(refreshToken);
    return token;
  }

//Refresh
  async refresh(refreshToken) {
    if (!refreshToken) {
      throw ApiError.UnauthorizedError();
    }
    const userData = tokenService.validateRefreshToken(refreshToken);
    const tokenFromDb = await tokenService.findToken(refreshToken);
    if (!userData || !tokenFromDb) {
      throw ApiError.UnauthorizedError();
    }
    const user = await UserModel.findById(userData.id);
    const userDto = new UserDto(user);
    const tokens = tokenService.generateTokens({...userDto});

    await tokenService.saveToken(userDto.id, tokens.refreshToken);
    return {...tokens, user: userDto}
  }
}

module.exports = new UserService();

```

## token-model

```

const {Schema, model} = require('mongoose');

const TokenSchema = new Schema({

```

```

    user: {type: Schema.Types.ObjectId, ref: 'User'},
    refreshToken: {type: String, required: true},
  })

module.exports = model('Token', TokenSchema);

```

## user-model

```

const {Schema, model} = require('mongoose');

const UserSchema = new Schema({
  email: {type: String, unique: true, required: true},
  password: {type: String, required: true},
  isActivated: {type: Boolean, default: false},
  activationLink: {type: String},
})

module.exports = model('User', UserSchema);

```

## user-controller

```

const userService = require('../service/user-service');
const {validationResult} = require('express-validator');
const apiError = require('../exceptions/api-error');
const ApiError = require('../exceptions/api-error');

class UserController {

  //Registration
  async registration(req, res, next) {
    try {
      const errors = validationResult(req);
      if(!errors.isEmpty()) {
        return next(ApiError.BadRequest('Błąd walidacji', errors.array()))
      }
      const {email, password} = req.body;
      const userData = await userService.registration(email, password);

```

```

        res.cookie('refreshToken', userData.refreshToken, {maxAge: 30 * 24
* 60 * 60 * 1000, httpOnly: true})
        return res.json(userData);
    } catch (e) {
        next(e);
    }
}

//Login
async login(req, res, next) {
    try {
        const {email, password} = req.body;
        const userData = await userService.login(email, password);
        res.cookie('refreshToken', userData.refreshToken, {maxAge: 30 * 24 *
60 * 60 * 1000, httpOnly: true})
        return res.json(userData);
    } catch (e) {
        next(e);
    }
}

//Logout
async logout(req, res, next ) {
    try {
        const {refreshToken} = req.cookies;
        const token = await userService.logout(refreshToken);
        res.clearCookie('refreshToken');
        return res.json(token);
    } catch (e) {
        next(e);
    }
}

async activatet(req, res, next ) {
    try {

    } catch (e) {
        next(e);
    }
}

```

```

}

//Refresh
async refresh(req, res, next ) {
  try {
    const {refreshToken} = req.cookies;
    const userData = await userService.refresh(refreshToken);
    res.cookie('refreshToken', userData.refreshToken, {maxAge: 30 * 24
* 60 * 60 * 1000, httpOnly: true})
    return res.json(userData);
  }catch (e) {
    next(e);
  }
}

}

module.exports = new UserController();

```

## Frontend

### Index.js

```

import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import authReducer from "././state";
import { configureStore } from '@reduxjs/toolkit';
import { Provider } from 'react-redux';
import {
  persistStore,
  persistReducer,
  FLUSH,
  REHYDRATE,
  PAUSE,

```

```

    PERSIST,
    PURGE,
    REGISTER,
  } from "redux-persist";
import storage from 'redux-persist/lib/storage';
import { PersistGate } from 'redux-persist/integration/react';

const persistConfig = { key: "root", storage, version: 1 };
const persistedReducer = persistReducer(persistConfig, authReducer);
const store = configureStore ({
  reducer: persistedReducer,
  middleware: (getDefaultMiddleware) =>
    getDefaultMiddleware({
      serializableCheck: {
        ignoreActions: [FLUSH, REHYDRATE, PAUSE, PERSIST, PURGE, REGISTER]
      },
    }),
});

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <Provider store={store}>
      <PersistGate loading={null} persistor={persistStore(store)}>
        <App />
      </PersistGate>
    </Provider>

  </React.StrictMode>
);

```

## App.js

```

import React from 'react';
import Shop from './components/Shop/App'
import {BrowserRouter, Routes, Route} from 'react-router-dom';
import LoginPage from './scenes/loginPage';

```

```

import { useMemo } from "react";
import { useSelector } from 'react-redux';
import { CssBaseline, ThemeProvider } from "@mui/material";
import { createTheme } from "@mui/material/styles";
import { themeSettings } from "../theme";
import Blogs from '../components/Blogs';

function App() {
  const mode = useSelector((state) => state.mode);
  const theme = useMemo(() => createTheme(themeSettings(mode)), [mode]);
  return (
    <div className="app">
      <BrowserRouter>
      <ThemeProvider theme={theme}>
        <CssBaseline/>
        <Routes>
          <Route path="/" element={<Shop />} />
          <Route path="/blog" element={<Blogs />} />
          <Route path="/login" element={<LoginPage />} />
        </Routes>
      </ThemeProvider>
    </BrowserRouter>
  </div>
  );
}

export default App;

```

## LoginPages.index.jsx

```

import { Box, Typography, useTheme, useMediaQuery } from "@mui/material";
import Form from "../Form";
import React from 'react';

const LoginPage = () => {
  const theme = useTheme();
  const isNonMobileScreens = useMediaQuery("(min-width: 1000px)");

```

```

return (
  <Box>
    <Box
      width="100%"
      backgroundColor={theme.palette.background.alt}
      p="1rem 6%"
      textAlign="center"
    >
      <Typography fontWeight="bold" fontSize="32px" color="primary">
        Login/Registration
      </Typography>
      <Box
        width={isNonMobileScreens ? "50%" : "93%"}
        p="2rem"
        m="2rem auto"
        borderRadius="1.5rem"
        backgroundColor={theme.palette.background.alt}
      >
        <Typography fontWeight="500" variant="h5" sx={{ mb: "1.5rem" }}>
        </Typography>
        <Form />
      </Box>
    </Box>
  </Box>
);
};

export default LoginPage;

```

## Shop. App.js

```

import React from 'react'
import Footer from '../Footer/index'
import Header from '../Header'
import Figures from '../Figures'
import {Carousel} from '../Carousel/Carousel'
import Text from '../Text'

```



```

import Items from '../Items'
import Text_Items from '../Text_Items'
import Categories from '../Categories'
import ShowFullItma from '../ShowFullItma'

class App extends React.Component {
  constructor (props) {
    super(props)
    this.state = {
      orders: [],
      currentItems: [],
      items: [
        {
          id: 1,
          title: 'Hub Hybrid (4G)',
          img: 'hh4g_m@1x.jpg',
          desc: 'Panel sterowania hybrydowego systemu alarmowego
umożliwiający fotograficzną weryfikację alarmów.',
          desc1: 'Zgodność z urządzeniami Jeweller i Fibra',
          desc2: 'Do 100 urządzeń',
          desc3: 'Ethernet i 2 karty SIM (2G/3G/4G)',
          desc4: 'Do 25 kamer lub urządzeń DVR',
          desc5: 'Do 5 podwajaczy zasięgu sygnału radiowego',
          desc6: 'Do 50 użytkowników',
          desc7: 'Do 9 grup bezpieczeństwa',
          desc8: 'Do 32 scenariuszy automatyzacji',
          category: 'Hub' ,
          price: '40.00',
        },
        {
          id: 2,
          title: 'Hub Hybrid (2G)',
          img: 'hh2g_m@1x.jpg',
          desc: 'Panel sterowania hybrydowego systemu alarmowego
umożliwiający fotograficzną weryfikację alarmów',
          desc1: 'Zgodność z urządzeniami Jeweller i Fibra',
          desc2: 'Do 100 urządzeń',
          desc3: 'Ethernet i 2 karty SIM (2G)',

```

```

    desc4: 'Do 25 kamer lub urządzeń DVR',
    desc5: 'Do 5 podwajaczy zasięgu sygnału radiowego',
    desc6: 'Do 50 użytkowników',
    desc7: 'Do 9 grup bezpieczeństwa',
    desc8: 'Do 32 scenariuszy automatyzacji',
    category: 'Hub' ,
    price: '30.00',
  },
  {
    id: 3,
    title: 'Hub 2 Plus Jeweller',
    img: 'hub_1_1_m@1x.jpg',
    desc: 'Centrala systemu alarmowego z obsługą fotograficznej
weryfikacji alarmów',
    desc1: 'Zgodność z urządzeniami Jeweller',
    desc2: 'Do 200 urządzeń',
    desc3: 'Ethernet, Wi-Fi, 2 karty SIM (2G/3G/4G)',
    desc4: 'Do 100 kamer lub urządzeń DVR',
    desc5: 'Do 5 podwajaczy zasięgu sygnału radiowego',
    desc6: 'Do 200 użytkowników',
    desc7: 'Do 25 grup bezpieczeństwa',
    desc8: 'Do 64 scenariuszy automatyzacji',
    category: 'Hub' ,
    price: '30.00',
  },
  {
    id: 4,
    title: 'Hub Plus Jeweller',
    img: 'hub_1_3_xl@1x.jpg',
    desc: 'Centrala systemu alarmowego',
    desc1: 'Zgodność z urządzeniami Jeweller',
    desc2: 'Do 150 urządzeń',
    desc3: 'Ethernet, Wi-Fi, 2 karty SIM (2G/3G)',
    desc4: 'Do 50 kamer lub urządzeń DVR',
    desc5: 'Do 5 podwajaczy zasięgu sygnału radiowego',
    desc6: 'Do 99 użytkowników',
    desc7: 'Do 25 grup bezpieczeństwa',
    desc8: 'Do 64 scenariuszy automatyzacji',
    category: 'Hubr' ,

```

```

    price: '30.00',
  },
  {
    id: 5,
    title: 'LeaksProtect',
    img: 'leaks_1_1_xl@1x.jpg',
    desc: 'Bezprzewodowy czujnik wycieku wody',
    desc1: 'Instalacja bez narzędzi',
    desc2: 'Informuje o wyschnięciu wody',
    desc3: 'Wodoodporna obudowa',
    desc4: 'Do 5 lat pracy na baterii',
    desc5: 'Komunikacja z hubem na odległość do 1 300 m',
    category: 'Czujniki wycieku wody' ,
    price: '30.00',
  },
  {
    id: 6,
    title: 'MotionCam (PhOD) Fibra',
    img: 'mc_ffibra_xl@1x.jpg',
    desc: 'Przewodowy czujnik ruchu z aparatem fotograficznym',
    desc1: 'Zdjęcie z alarmu',
    desc2: 'Zdjęcie na żądanie',
    desc3: 'Zdjęcie za scenariusza',
    desc4: 'Dostarczenie zdjęcia w 9 sekund',
    desc5: 'Podświetlenie podczerwone do zdjęć w ciemności',
    desc6: 'Filtrowanie fałszywych alarmów spowodowanych ruchem',
    desc7: 'Kompensacja temperatury',
    desc8: 'Odporność na zwierzęta',
    desc9: 'Pobór prądu do 100 µA z linii Fibra',
    desc10: 'Do 2000 m dwukierunkowej komunikacji radiowej',
    category: 'Czujniki Ruhu' ,
    price: '30.00',
  },
  {
    id: 7,
    title: 'MotionCam Fibra',
    img: 'mc_f_xl@1x.jpg',
    desc: 'Przewodowy czujnik ruchu z aparatem fotograficzny',
    desc1: 'Zdjęcie z alarmu',

```

```

    desc2: 'Dostarczenie zdjęcia w 9 sekund',
    desc3: 'Podświetlenie podczerwone do zdjęć w ciemności',
    desc4: 'Filtrowanie fałszywych alarmów spowodowanych ruchem',
    desc5: 'Kompensacja temperatury',
    desc6: 'Odporność na zwierzęta',
    desc7: 'Pobór prądu do 100 µA z linii Fibra',
    desc8: 'Do 2000 m dwukierunkowej komunikacji radiowej',
    category: 'Czujniki Ruhu' ,
    price: '30.00',
  },
  {
    id: 8,
    title: 'MotionProtect Fibra',
    img: 'mp_f_xl@1x.jpg',
    desc: 'Przewodowy czujnik ruchu',
    desc1: 'Filtrowanie fałszywych alarmów spowodowanych ruchem',
    desc2: 'Kompensacja temperatury',
    desc3: 'Odporność na zwierzęta',
    desc4: 'Pobór prądu do 50 µA z linii Fibra',
    desc5: 'Do 2000 m dwukierunkowej komunikacji radiowej',
    category: 'Czujniki Ruhu' ,
    price: '30.00',
  },
  {
    id: 9,
    title: 'MotionCam (PhOD) Jeweller',
    img: 'mc_phod_xl@1x.jpg',
    desc: "Bezprzewodowy czujnik ruchu z kamerą",
    desc1: "Zdjęcie alarmowe",
    desc2: "Zdjęcie na życzenie",
    desc3: "Zdjęcie według scenariusza",
    desc4: "Dostawa zdjęć w 9 sekund",
    desc5: "Oświetlenie w podczerwieni do zdjęć w ciemności",
    desc6: "Filtrowanie fałszywych alarmów spowodowanych ruchem",
    desc7: "Kompensacja temperatury",
    desc8: "Odporność na zwierzęta",
    desc9: "Do 4 lat pracy na baterii",
    desc10: "Dwukierunkowa łączność radiowa do 1700 m",
    category: 'Czujniki Ruhu' ,
  }

```

```

    price: '30.00',
  },
  {
    id: 10,
    title: 'MotionCam Jeweller',
    img: 'mc_xl@1x.jpg',
    desc: "Bezprzewodowy czujnik ruchu z kamerą",
    desc1: "Zdjęcie alarmowe",
    desc2: "Dostawa zdjęć w 9 sekund",
    desc3: "Oświetlenie w podczerwieni do zdjęć w ciemności",
    desc4: "Filtrowanie fałszywych alarmów spowodowanych ruchem",
    desc5: "Kompensacja temperatury",
    desc6: "Odporność zwierząt domowych",
    desc7: "Do 4 lat pracy na baterii",
    desc8: "Dwukierunkowa łączność radiowa do 1700 m",
    category: 'Czujniki Ruhu' ,
    price: '30.00',
  },
  {
    id: 11,
    title: 'MotionProtect Jeweller',
    img: 'motion_detect_block1_2_xl@1x.jpg',
    desc: "Bezprzewodowy czujnik ruchu",
    desc1: "Filtrowanie fałszywych alarmów spowodowanych ruchem",
    desc2: "Kompensacja temperatury",
    desc3: "Odporność zwierząt domowych",
    desc4: "Do 5 lat pracy na baterii",
    desc5: "Dwukierunkowa łączność radiowa do 1700 m",
    category: 'Czujniki Ruhu' ,
    price: '30.00',
  },
  {
    id: 12,
    title: 'HomeSiren Fibra',
    img: 'hsf_xl@1x.jpg',
    desc: "Kompaktowa przewodowa syrena domowa",
    desc1: "Głośność 81 do 105 dB",
    desc2: "Czas trwania sygnału od 3 do 180 sekund",
    desc3: "Zacisk dla dodatkowej diody LED",
  }

```

```

    desc4: "Wskazanie trybów bezpieczeństwa",
    desc5: "Pobór prądu do 50 µA z linii Fibra",
    desc6: "Komunikacja z hubem na odległość do 2000 m",
    category: 'Syreny' ,
    price: '30.00',
  },
  {
    id: 13,
    title: 'StreetSiren Fibra',
    img: 'ssf_xl@1x.jpg',
    desc: "Przewodowa syrena zewnętrzna",
    desc1: "Głośność 85 do 113 dB",
    desc2: "Czas trwania sygnału od 3 do 180 sekund",
    desc3: "Ramka LED do sygnalizacji alarmów i stanu systemu",
    desc4: "Klasa ochrony obudowy IP54",
    desc5: "Kompensacja temperatury",
    desc6: "Odporność na zwierzęta",
    desc7: "Do 4 lat pracy na baterii",
    category: 'Syreny' ,
    price: '30.00',
  },
  {
    id: 14,
    title: 'HomeSiren Jeweller',
    img: 'hub_1_3_xl@1x.jpg',
    desc: "Kompaktowa bezprzewodowa syrena domowa",
    desc1: "Głośność 81 do 105 dB",
    desc2: "Czas trwania sygnału od 3 do 180 sekund",
    desc3: "Zacisk dla dodatkowej diody LED",
    desc4: "Wskazanie trybów bezpieczeństwa",
    desc5: "Do 5 lat pracy na baterii",
    desc6: "Komunikacja z hubem na odległość do 2000 m",
    category: 'Syreny' ,
    price: '30.00',
  },
  {
    id: 15,
    title: 'StreetSiren Jeweller',
    img: 'homesiren_xl@1x.jpg',

```

```

        desc: "Bezprzewodowa syrena zewnętrzna",
        desc1: "Głośność 85 do 113 dB",
        desc2: "Czas trwania sygnału od 3 do 180 sekund",
        desc3: "Ramka LED do sygnalizacji alarmów i stanu systemu",
        desc4: "Klasa ochrony obudowy IP54",
        desc5: "Do 5 lat pracy na baterii",
        desc6: "Podłączenie zewnętrznego zasilacza 12 V=",
        desc7: "Komunikacja z hubem na odległość do 1500 m",
        category: 'Syreny' ,
        price: '30.00',
      },
    ],
    showFullItama: false,
    fullItama: {}
  }
  this.state.currentItems = this.state.items
  this.addToOrder = this.addToOrder.bind(this)
  this.deleteOrder = this.deleteOrder.bind(this)
  this.chooseCategory = this.chooseCategory.bind(this)
  this.onShowItama = this.onShowItama.bind(this)
}

render() {
  return (
    <div className='wrapper'>
      <Header orders={this.state.orders} onDelete={this.deleteOrder} />
      <Figures />
      <Text_Items />
      <Categories chooseCategory={this.chooseCategory} />
      <Items onShowItama={this.onShowItama}
items={this.state.currentItems} onAdd={this.addToOrder}/>
      {this.state.showFullItama && <ShowFullItama onAdd={this.addToOrder}
onShowItama={this.onShowItama} itama={this.state.fullItama} />}
      <Text />
      <Carousel>
        <div className='item item-1'>hgjhtjghjhghjgh</div>
        <div className='item item-2'>item 2</div>

```

```

        <div className='item item-3'>item 3</div>
      </Carousel>
    <Footer />
  </div>
)
}

onShowItema(itema) {
  this.setState({fullItema: itema})
  this.setState({showFullItema: !this.state.showFullItema})
}

chooseCategory(category) {
  if(category === 'all') {
    this.setState({ currentItems: this.state.items})
    return
  }
  this.setState({
    currentItems: this.state.items.filter(el => el.category === category)
  })
}

deleteOrder(id) {
  this.setState({ orders: this.state.orders.filter(el => el.id !==id)})
}

addToOrder(itema) {
  let isInArray = false
  this.state.orders.forEach(el => {
    if(el.id === itema.id)
      isInArray = true
  })
  if(!isInArray)
    this.setState({ orders: [...this.state.orders, itema] })
}
}

```



```
export default Ap
```

## Header.js

```
import React, { useState } from 'react'
import { BsFillBasket2Fill } from "react-icons/bs";
import { BsFillPersonFill } from "react-icons/bs";
import { NavLink } from 'react-router-dom';
import Order from './Order';

const showOrders = (props) => {
  let summa = 0
  props.orders.forEach(el => summa += Number.parseFloat(el.price))
  return (<div>
    {props.orders.map(el => (
      <Order onDelete={props.onDelete} key={el.id} itema={el} />
    ))}
    <p className='summa'>Suma: {new
Intl.NumberFormat().format(summa)}zł</p>
  </div>)
}

const showNothing = () => {
  return (<div className='empty'>
    <h2>Brak dostępnych towarów</h2>
  </div>)
}































export default function Header (props) {
  let [cartOpen, setCartOpen] = useState (false)
  return (
    <header>
      <div>
        <span className='logo'>Inteligentny dom</span>
        <ul className='nav'>
          <li><NavLink to ="/blog">Blogs</NavLink></li>
          <li>Contact</li>
        </ul>
      </div>
    </header>
  )
}
```

```
      <BsFillBasket2Fill onClick={() => setCartOpen(cartOpen = !cartOpen)}  
      className={`shop-cart-basket ${cartOpen && 'active'}`} />  
      <NavLink to ="/login"><BsFillPersonFill className='shop-cart-  
client' /></NavLink>  
  
      {cartOpen && (  
        <div className='shop-cart'>  
          {props.orders.length > 0 ?  
            showOrders(props) : showNothing()}  
        </div>  
      )}  
    </div>  
    <div className='presentation'></div>  
  </header>  
)  
}
```

## **Dodatek 2**

# **Konfiguracja bazy danych**



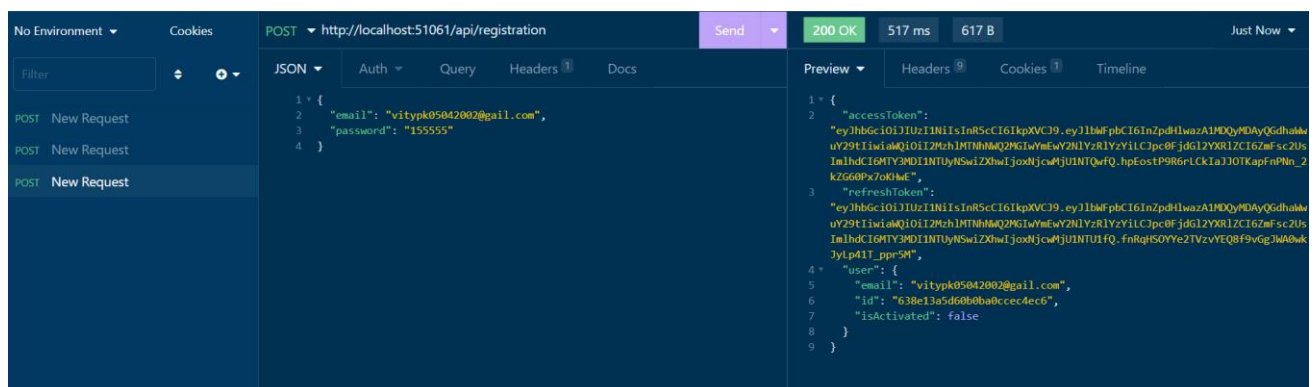
tokens					
	_id ObjectId	user ObjectId	refreshToken String	_v Int32	
1	ObjectId('6378bc88b8a35f36e46...	ObjectId('6378bc88b8a35f36e46...	"eyJhbGciOiJIUzI1NiIsInR5cCI6I...	0	  
2	ObjectId('6378def5ef757c872aa...	ObjectId('6378def5ef757c872aa...	"eyJhbGciOiJIUzI1NiIsInR5cCI6I...	0	  
3	ObjectId('637e0a6fe55ea195d2b...	ObjectId('637e0a6fe55ea195d2b...	"eyJhbGciOiJIUzI1NiIsInR5cCI6I...	0	  
4	ObjectId('6380c3e3099e23bed72...	ObjectId('6380c298099e23bed72...	"eyJhbGciOiJIUzI1NiIsInR5cCI6I...	0	  
5	ObjectId('6380c8ff099e23bed72...	ObjectId('6380c8ff099e23bed72...	"eyJhbGciOiJIUzI1NiIsInR5cCI6I...	0	  
6	ObjectId('6380c905099e23bed72...	ObjectId('6380c905099e23bed72...	"eyJhbGciOiJIUzI1NiIsInR5cCI6I...	0	  
7	ObjectId('6380caeb099e23bed72...	ObjectId('6380caeb099e23bed72...	"eyJhbGciOiJIUzI1NiIsInR5cCI6I...	0	  
8	ObjectId('6380ccaf099e23bed72...	ObjectId('6380ccaf099e23bed72...	"eyJhbGciOiJIUzI1NiIsInR5cCI6I...	0	  
9	ObjectId('6384efc4f79e99287da...	ObjectId('6384efc4f79e99287da...	"eyJhbGciOiJIUzI1NiIsInR5cCI6I...	0	  
10	ObjectId('6387319650d34ccdc5...	ObjectId('6387319650d34ccdc5...	"eyJhbGciOiJIUzI1NiIsInR5cCI6I...	0	  

Rys. 2. - refreshToken użytkownika

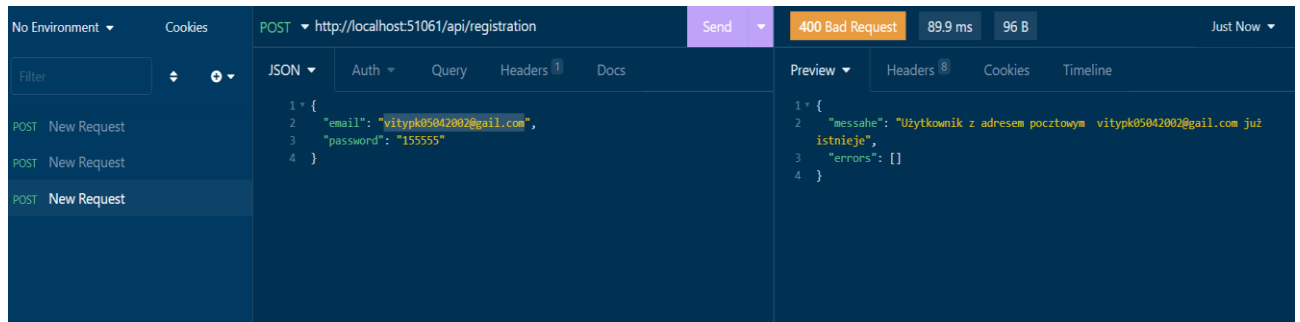
## Tworzenie użytkownika w «Insomnia»

Wykorzystaliśmy żądania typu POST. W pierwszej kolejności należy uruchomić serwer. Następnie przejdź do aplikacji i rozpocznij pracę. Od początku rejestrujemy użytkownika, wpisujemy e-mail i hasło (rys. 3). Jeśli wszystko jest w porządku, otrzymujemy "accessToken" i "refreshToken" (rys. 3). Jeśli chcemy zarejestrować ten sam e-mail, to pojawi się problem «**Użytkownik z adresem pocztowym vitypk0504200@gail.com już istnieje**» (rys. 4).

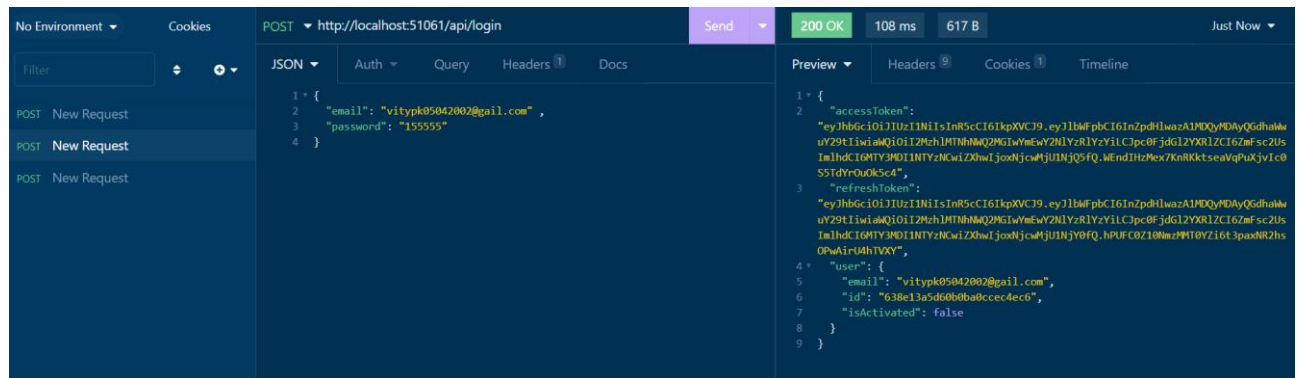
Następnie przystępujemy do autoryzacji. Wpisujemy e-mail i hasło, które właśnie zarejestrowaliśmy, jeśli wszystko jest w porządku, otrzymamy informacje o użytkowniku «accessToken» i «refreshToken» (rys. 5). Jeśli wpiszemy zły e-mail i hasło, otrzymamy błąd «**Nie znaleziono użytkownika z tym adresem e-mail**», «**Nieprawidłowe hasło**» (Rys. 6,7). Następnie przechodzimy do «MongoDBCompass» i sprawdzamy czy użytkownik został dodany do bazy danych



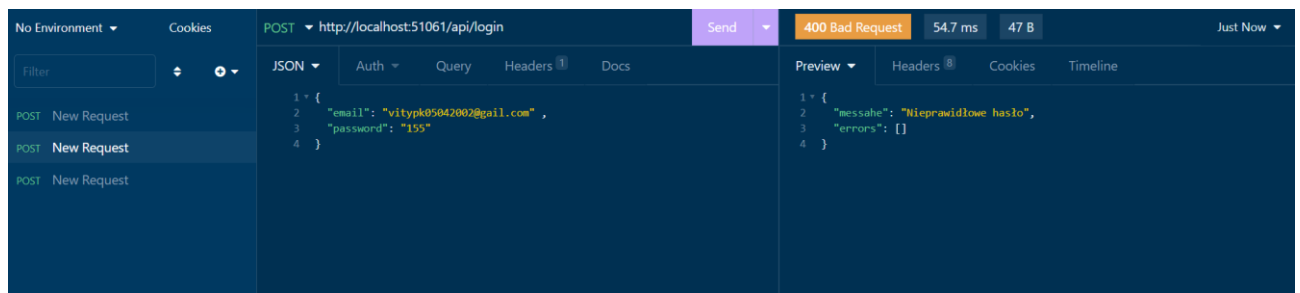
Rys. 3. - Zarejestrowany użytkownika



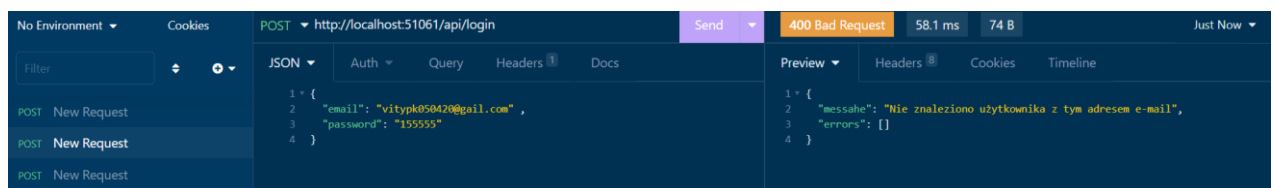
Rys. 4. - Wprowadził istniejącą wiadomość e-mail



Rys. 5. - Sprawdzona autoryzacja



Rys. 6. - Wprowadziłeś nieprawidłowy adres e-mail



Rys. 7. - Wprowadziłeś nieprawidłowe hasło

```
_id: ObjectId('638e13a5d60b0ba0ccec4ec6')
email: "vitypk05042002@gail.com"
password: "$2b$04$5n1Su4fNlLueYgWaeRc53.mpfM0q68YYL0gH7ftCpjJGVilK37/Ku"
isActivated: false
activationLink: "dd8b20ca-7531-4cde-820b-a7799028ac1e"
__v: 0
```

Rys. 8. - Sprawdzono połączenie z bazą danych