

Imię i nazwisko studenta: Zofia Zinkowska

Nr albumu: 180925

Poziom kształcenia: Studia pierwszego stopnia

Forma studiów: stacjonarne

Kierunek studiów: Fizyka Techniczna

Specjalność: Informatyka Stosowana

**PRACA DYPLOMOWA INŻYNIERSKA**

Tytuł pracy w języku polskim: Aplikacja „ewidencja roślin”

Tytuł pracy w języku angielskim: The „plant register” application

Opiekun pracy: dr. inż. Bartosz Reichel



**OŚWIADCZENIE: dotyczące pracy dyplomowej zatytułowanej:**

**Aplikacja „ewidencja roślin”**

Imię i nazwisko: Zofia Zinkowska

Data i miejsce urodzenia: 20.07.2000, Słupsk

Nr albumu: 180925

Wydział: Wydział Fizyki Technicznej i Matematyki Stosowanej

Kierunek: fizyka techniczna

Poziom kształcenia: pierwszy

Forma studiów: stacjonarnie

Typ pracy: praca dyplomowa inżynierska

Świadomy(a) odpowiedzialności karnej z tytułu naruszenia przepisów ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz. U. z 2019 r. poz. 1231, z późn. zm.) i konsekwencji dyscyplinarnych określonych w ustawie z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (t.j. Dz. U. z 2020 r. poz. 85, z późn. zm.), 1. a także odpowiedzialności cywilnoprawnej oświadczam, że przedkładana praca dyplomowa została opracowana przeze mnie samodzielnie.

Niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadaniem tytułu zawodowego.

Wszystkie informacje umieszczone w ww. pracy dyplomowej uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami zgodnie z art. 34 ustawy o prawie autorskim i prawach pokrewnych.

29.03.2022, Zofia Zinkowska

Data i podpis lub uwierzytelnienie w portalu uczelnianym Moja PG

\*) Dokument został sporządzony w systemie teleinformatycznym, na podstawie §15 ust. 3b Rozporządzenia MNiSW z dnia 12 maja 2020 r. zmieniającego rozporządzenie w sprawie studiów (Dz.U. z 2020 r. poz. 853). Nie wymaga podpisu ani stempla

**STRESZCZENIE**

Praca dyplomowa przedstawia aplikację internetową, służącą do utrzymania / ewidencji roślin w ogrodzie botanicznym. W pierwszej oraz drugiej części pracy uwaga poświęcona jest wymaganiom jakim oprogramowanie musi sprostać. Trzecia część opisuje technologie oraz narzędzia programistyczne wykorzystane do implementacji. Kluczowym miejscem jest rozdział czwarty, który przedstawia projekt tworzonego oprogramowania. Końcowe rozdziały przedstawiają testy, podsumowanie, spis rysunków oraz bibliografię.

**Słowa kluczowe:**

Utrzymanie ewidencji roślin, ewidencja roślin, aplikacja internetowa, C#, TypeScript, ASP.NET Core, Vue.js.

**Dziedzina nauki i techniki z wymogami OECD:**

**ABSTRACT**

The thesis presents a web application for maintaining / registering plants in a botanical garden. In the first and second part of the work, the focus was on the requirements that the software should handle. The third part describes the technologies used for the implementation. The most significant part of the thesis is the fourth chapter, which presents the design of the software being developed. The final chapters present the tests, summary, list of figures, and a bibliography.

**Keywords:**

Plants maintennance registration, plants registration, web application, C#, TypeScript, ASP.NET Core, Vue.js.

Spis treści

[**1.** **Wstęp** 5](#_Toc116227658)

[***1.1.*** ***Aplikacja internetowa*** 5](#_Toc116227659)

[***1.2.*** ***Cel projektowania aplikacji*** 5](#_Toc116227660)

[***1.3.*** ***Podział pracy*** 6](#_Toc116227661)

[**2.** **Specyfikacja wymagań** 6](#_Toc116227662)

[***2.1.*** ***Dziedzinowy słownik pojęć*** 6](#_Toc116227663)

[***2.2.*** ***Interfejs graficzny aplikacji*** 6](#_Toc116227664)

[***2.3.*** ***Wymagania biznesowe*** 6](#_Toc116227665)

[***2.4.*** ***Wymagania funkcjonalne użytkownika*** 6](#_Toc116227666)

[***2.5.*** ***Wymagania niefunkcjonalne użytkownika*** 6](#_Toc116227667)

[***2.6.*** ***Wymagania niefunkcjonalne systemowe*** 7](#_Toc116227668)

[**3.** **Przygotowanie do tworzenia aplikacji** 7](#_Toc116227669)

[***3.1.*** ***Wykorzystane narzędzia programistyczne*** 7](#_Toc116227670)

[***3.1.1.*** ***Git*** 7](#_Toc116227671)

[***3.1.2.*** ***Bootstrap*** 8](#_Toc116227672)

[***3.1.2.*** ***Baza danych LiteDB*** 8](#_Toc116227673)

[**4.** **Projekt aplikacji - implementacja** 8](#_Toc116227674)

[***4.1.*** ***Diagram*** 8](#_Toc116227675)

[***4.2.*** ***Wyjątki*** 8](#_Toc116227676)

[***4.3.*** ***Adresowanie*** 8](#_Toc116227677)

[**5.** **Testy** 8](#_Toc116227678)

[**6.** **Podsumowanie** 9](#_Toc116227679)

[***6.1.*** ***Efekt końcowy pracy*** 9](#_Toc116227680)

[**7.** **Spis rysunków** 9](#_Toc116227681)

[Bibliografia 9](#_Toc116227682)

# **1. Wstęp**

Ze względu na rosnące zapotrzebowanie jakościowe oraz konkurencję na rynku informatyki, konieczny jest ciągły rozwój. Dlatego pojawiają się nowe rozwiązania w zakresie utrzymania / ewidencji roślin ułatwiające codzienną pracę.

## ***1.1. Aplikacja internetowa***

Aplikacja internetowa jest to program komputerowy pracujący na serwerze. Komunikuje się poprzez sieć komputerową z hostem użytkownika komputera, przy użyciu przeglądarki internetowej, będącej interaktywnym klientem aplikacji.

Aplikacje internetowe są jednym z najpopularniejszych typów programów komputerowych. Służą nie tylko do rozrywki, ale również jako systemy obsługi klientów. Obecnie najbardziej rozpoznawalne aplikacje internetowe to Gmail, YouTube oraz Facebook.

Podstawową zaletą tworzenia oprogramowania tego typu jest fakt, że aplikacje internetowe są kompatybilne wieloplatformowo, gdyż nie zależą od systemu operacyjnego, a jedynym wymaganiem do sprawnego korzystania z programu jest posiadanie zainstalowanej przeglądarki internetowej z dostępem do internetu. Następną zaletą jest to, że użytkownik posiada dostęp do swoich danych nawet w momencie, kiedy następuje awaria urządzenia, ponieważ dane są gromadzone na serwerach dostawcy oprogramowania. Aplikacje webowe nie wymagają instalacji oraz aktualizacji (wdrażania i utrzymania) po stronie klienta, ponieważ zawsze serwowane są w najnowszej wersji serwera. Nie potrzebują weryfikacji / certyfikacji od poszczególnych producentów (np. Google Store, Apple Store).

Aplikacje internetowe oprócz zalet posiadają również wady. Jedną z nich jest całkowita zależność od serwera przez co użytkownik nie ma możliwości dokonywania operacji bez połączenia z siecią. Oprócz tego dostawca oprogramowania jest w stanie śledzić niemal każdy ruch użytkownika podczas jego korzystania z aplikacji. W dzisiejszych czasach mało, które aplikacje, nawet aplikacje nie internetowe, potrafią działać bez dostępu do internetu (np. aplikacja desktopowa nadal składowałaby dane na centralnym serwerze). W takiej architekturze utrata urządzenia byłaby jednoznaczna z utratą wszystkich danych.

## ***1.2. Cel projektowania aplikacji***

Na co dzień aplikacje internetowe ułatwiają nam zarządzanie pocztą, pozwalają robić zakupy i opłacać rachunki. Aplikacja ma na celu rozwiązanie problemu przestarzałego oprogramowania służącego do utrzymania / ewidencji roślin w ogrodzie botanicznym. Przedsięwzięcie dąży do wypełnienia brakujących potrzeb w systemie. Dzięki możliwości skanowania kodów etykiet oraz drukowaniu etykiet użytkownik będzie wiedział dokładnie jakie rośliny znajdują się w ogrodzie botanicznym. Nie tylko usprawni to pracę użytkownikowi, jak również użytkownik zyska na czasie, ponieważ automatyzacja procesów wcześniej wykonywanych w sposób manualny, w znaczny sposób wykorzystywała czas osób administrujących.

W osiągnięciu celu pomogą nowe technologie programistyczne tj. C# - ASP.NET Core, TypeScript, Vue.js.

## ***1.3. Podział pracy***

Celem projektowanej aplikacji będzie ułatwienie odczytu / utrzymania ewidencji roślin, poprzez możliwość skanowania, drukowania etykiet, szybki wgląd w aplikacji. Podział na rodzaje roślin pozwoli łatwo i skutecznie sprawdzić stan roślin.

# **2. Specyfikacja wymagań**

Rozdział przedstawia wymagania stawiane aplikacji internetowej służącej do utrzymania / ewidencji roślin. W rozdziale omawiane są kolejno wymagania biznesowe, funkcjonalne użytkownika oraz wymagania niefunkcjonalne użytkownika i systemowe.

## ***2.1. Dziedzinowy słownik pojęć***

Użytkownik systemu – osoba korzystająca z aplikacji do utrzymania / ewidencji roślin.

CSS - Cascading Style Sheets - Kaskadowe Arkusze Stylów.

🡪 🡪 dopisać 🡨 🡨

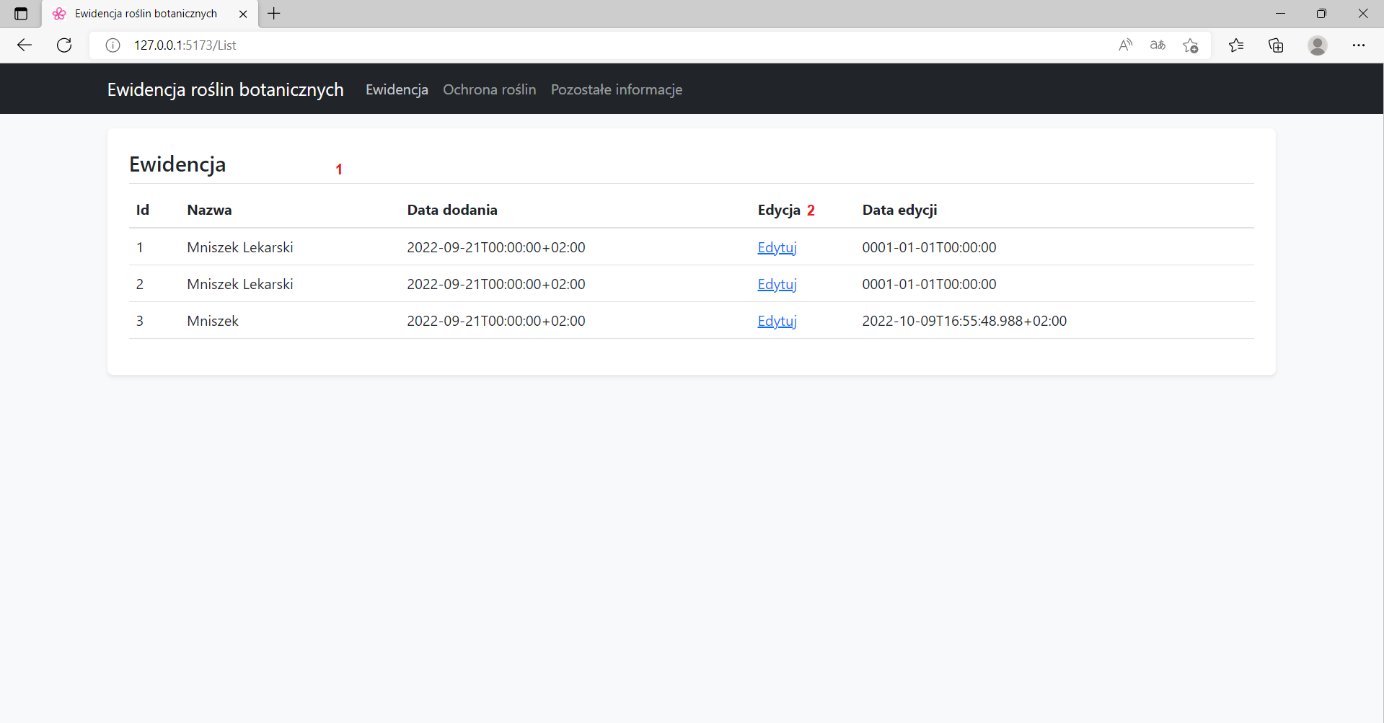
## ***2.2. Interfejs graficzny aplikacji***

Interfejs graficzny został wykonany przy użyciu biblioteki Bootstrap-Vue 3. Używane są również style kaskadowe CSS.

Obraz zawierający tekst

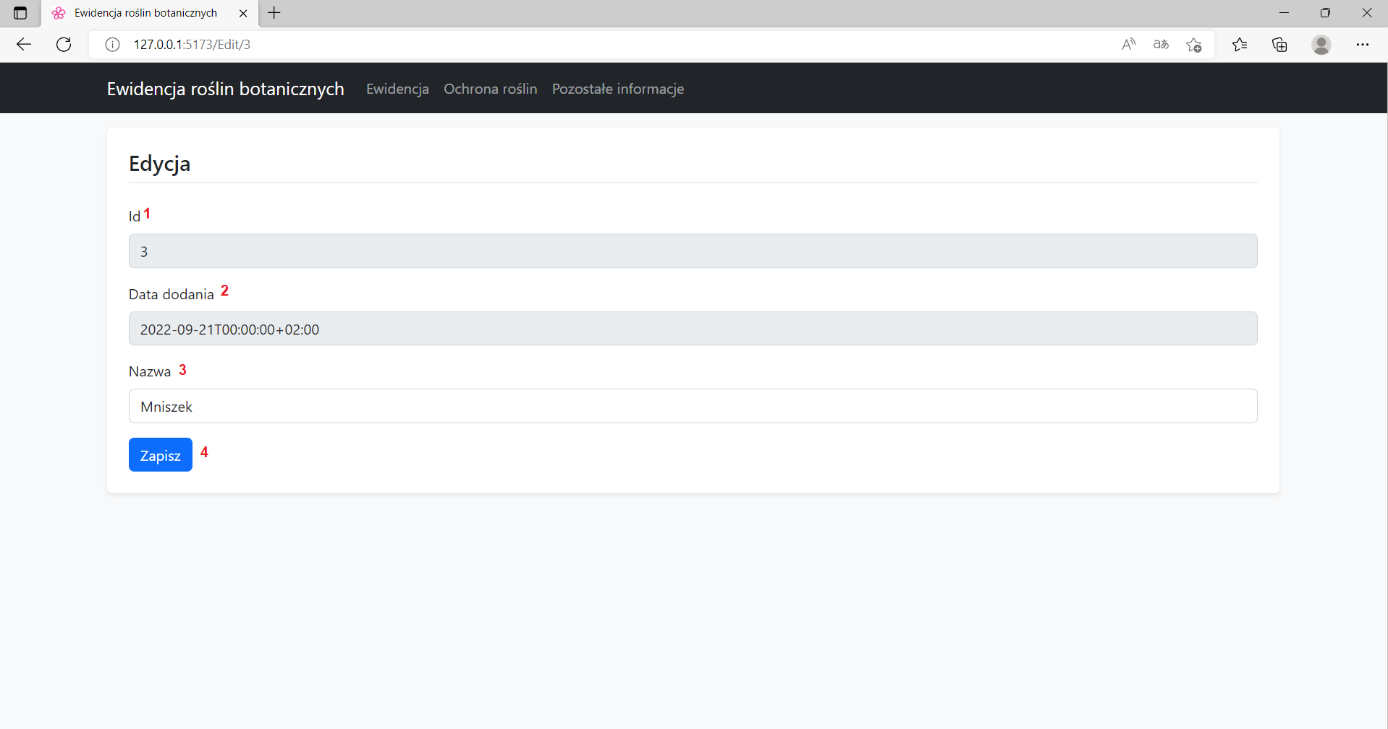
Opis wygenerowany automatycznie

Rysunek Interfejs graficzny – strona główna



Rysunek Interfejs graficzny – ewidencja roślin

1. Lista roślin w ewidencji.
2. Przycisk otwierający stronę do edycji informacji o roślinie.



Rysunek Interfejs graficzny – formularz edycji pól z ewidencji roślin

1. Pole tekstowe z numerem identyfikacyjnym rośliny.
2. Pole z datą dodania rośliny do ewidencji.
3. Pole tekstowe do wprowadzenia nazwy rośliny.
4. Przycisk wysyłający dane z formularza do serwera.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek Interfejs graficzny – ochrona roślin

## ***2.3. Wymagania biznesowe***

Nazwa aplikacji to Ewidencja Roślin. Oprogramowanie ma za zadanie ułatwić utrzymanie / ewidencję roślin. W rozdziale omawiane są wymagania biznesowe.

## ***2.4. Wymagania funkcjonalne użytkownika***

* Rejestracja konta użytkownika – wpisanie nazwy użytkownika, adresu e-mail oraz hasła do bazy danych użytkowników w celu stworzenia konta.
* Logowanie do aplikacji – proces uwierzytelnienia i autoryzacji użytkownika aplikacji, polegający na podaniu adresu e-mail i hasła uwierzytelniającego w celu dostępu do konta. / Wylogowanie z aplikacji – rezygnacja z uprzednio uzyskanego w skute logowania dostępu do konta.
* Usuwanie konta użytkownika – usunięcie informacji o użytkowniku z bazy danych.
* Zmiana hasła do konta – wpisanie nowego hasła użytkownikowi w celu zmiany danych logowania do konta.

## ***2.5. Wymagania niefunkcjonalne użytkownika***

* Aplikacja powinna odpowiadać aktualnym standardom sieciowym i wytycznym dla dostępności treści internetowych – ma prawidłowo wyświetlać się w różnych przeglądarkach:
* Google Chrome
* Microsoft Edge
* Mozilla FireFox

## ***2.6. Wymagania niefunkcjonalne systemowe***

* Aplikacja powinna korzystać z dostępnych darmowych rozwiązań systemów informatycznych.
* Aplikacja powinna być wieloplatformowa.
* Interfejs aplikacji – system powinien być łatwy w obsłudze poprzez przejrzysty wygląd interfejsu, zrozumiały dla niedoświadczonego użytkownika. Aplikacja powinna być zaopatrzona w regulamin usługi, zrozumiałe menu oraz przyciski na stronie.
* Niezawodność aplikacji – przeprowadzanie testów poszczególnych modułów aplikacji. Optymalne dobranie architektury sprzętowo-programowej na jakiej będzie działać aplikacja oraz serwer bazodanowy.
* Bezpieczeństwo aplikacji – ograniczenie wszystkim użytkownikom praw dostępu do bazy danych. Proces logowania jest realizowany poprzez podanie nazwy użytkownika oraz hasła użytkownika w formularzu logowania. Użycie protokołu HTTPS, który chroni integralność i poufność danych przesyłanych również będzie wchodził w skład bezpieczeństwa aplikacji.

# **3. Przygotowanie do tworzenia aplikacji**

Poniżej zostały przedstawione użyte technologie podczas implementacji aplikacji. Przy wyborze technologii autorce zależało na wykorzystaniu prostych i szybkich rozwiązań, darmowych oraz ogólnodostępnych. Dzięki odpowiedniemu doborze narzędzi rozwiązanie ewentualnych problemów było szybsze.

## ***3.1. Wykorzystane narzędzia programistyczne***

W tym rozdziale zostały przedstawione wykorzystane narzędzia programistyczne, które były wykorzystane podczas tworzenia projektu.

## ***3.1.1. Git***

Git jest rozproszonym systemem kontroli wersji. Został stworzony przez Linusa Torvaldsa w 2005 roku.

System z założenia miał podtrzymywać wielkie rozproszone projekty programistyczne, takie jak np. Linux.

Najważniejsze cechy:

* Programowanie rozproszone – system dostarcza programiście pracującemu nad projektem lokalną kopię rozwoju aplikacji.
* Zgodność z protokołami i systemami – repozytoria mogą być publikowane za pomocą HTTP, FTP, etc.
* Programowanie nieliniowe – wsparcie dla nieliniowego wytwarzania oprogramowania

## ***3.1.2. Bootstrap***

Bootstrap – biblioteka CSS, rozwijana przez programistów Twittera, wydawany na licencji. Zawiera zestaw przydatnych narzędzi ułatwiających tworzenie interfejsu graficznego stron oraz aplikacji internetowych. Bazuje głównie na gotowych rozwiązaniach HTML oraz CSS i może być stosowany m.in. do stylizacji takich elementów jak teksty, formularze, przyciski, wykresy, nawigacje i innych komponentów wyświetlanych na stronie.

## ***3.1.2. Baza danych LiteDB***

LiteDB to szybka, prosta, typu open source z zerową konfiguracją, wbudowana baza danych NoSQL dla .NET w pełni napisana w języku C#. LiteDB organizuje dokumenty w magazynach dokumentów znanych jako kolekcje. Co oznacza, że każda kolekcja jest identyfikowana przez unikatową nazwę i zawiera jeden lub więcej dokumentów, które mają ten sam schemat. Ponieważ LiteDB jest bazą danych bez serwera, nie musimy instalować jej w swoim systemie. Wystarczy dodać odwołanie do pliku LiteDB.dll w swoim projekcie. Alternatywnie można zainstalować LiteDB za pomocą Menadżera pakietów NuGet w programie Visual Studio.

# **4. Projekt aplikacji - implementacja**

Rozdział przedstawia wymagania stawiane aplikacji internetowej służącej do utrzymania / ewidencji roślin. W rozdziale omawiane są kolejno wymagania biznesowe, funkcjonalne użytkownika oraz wymagania niefunkcjonalne użytkownika i systemowe.

## ***4.1. Diagram***

Nazwa aplikacji to Ewidencja Roślin. Oprogramowanie ma za zadanie ułatwić utrzymanie / ewidencję roślin. W rozdziale omawiane są wymagania biznesowe, wymagania funkcjonalne i niefunkcjonalne – użytkownika i systemowe.

## ***4.2. Wyjątki***

Nazwa aplikacji to Ewidencja Roślin. Oprogramowanie ma za zadanie ułatwić utrzymanie / ewidencję roślin. W rozdziale omawiane są wymagania biznesowe, wymagania funkcjonalne i niefunkcjonalne – użytkownika i systemowe.

## ***4.3. Adresowanie***

Nazwa aplikacji to Ewidencja Roślin. Oprogramowanie ma za zadanie ułatwić utrzymanie / ewidencję roślin. W rozdziale omawiane są wymagania biznesowe, wymagania funkcjonalne i niefunkcjonalne – użytkownika i systemowe.

# **5. Testy**

Rozdział przedstawia wymagania stawiane aplikacji internetowej służącej do utrzymania / ewidencji roślin. W rozdziale omawiane są kolejno wymagania biznesowe, funkcjonalne użytkownika oraz wymagania niefunkcjonalne użytkownika i systemowe.

# **6. Podsumowanie**

Rozdział przedstawia wymagania stawiane aplikacji internetowej służącej do utrzymania / ewidencji roślin. W rozdziale omawiane są kolejno wymagania biznesowe, funkcjonalne użytkownika oraz wymagania niefunkcjonalne użytkownika i systemowe.

## ***6.1. Efekt końcowy pracy***

Nazwa aplikacji to Ewidencja Roślin. Oprogramowanie ma za zadanie ułatwić utrzymanie / ewidencję roślin. W rozdziale omawiane są wymagania biznesowe, wymagania funkcjonalne i niefunkcjonalne – użytkownika i systemowe.

# **7. Spis rysunków**

[Rysunek 1 Interfejs graficzny – strona główna 6](#_Toc116229037)

[Rysunek 2 Interfejs graficzny – ewidencja roślin 7](#_Toc116229038)

[Rysunek 3 Interfejs graficzny – formularz edycji pól z ewidencji roślin 7](#_Toc116229039)

[Rysunek 4 Interfejs graficzny – ochrona roślin 8](#_Toc116229040)

# Bibliografia

*[1] Dokumentacja bibilioteki Bootstrap-vue*  https://bootstrap-vue.org/docs#documentation-sections

*[2] Dokumentacja biblioteki Bootstrap* https://pl.wikipedia.org/wiki/Bootstrap\_(framework)

https://getbootstrap.com/docs/5.2/getting-started/introduction/