**dokumentacja techniczna aplikacji webowej docsee**

Spis treści

[1. WPROWADZENIE 2](#_Toc154744257)

[1.1 Cel Dokumentacji 2](#_Toc154744258)

[1.2 Cel Aplikacji 2](#_Toc154744259)

[1.3 Zakres Dokumentacji 2](#_Toc154744260)

[2. ARCHITEKTURA SYSTEMU 3](#_Toc154744261)

[2.1 Działanie Aplikacji Docsee 3](#_Toc154744262)

[3. INSTRUKCJA INSTALACJI 4](#_Toc154744263)

[3.1 Wymagania Systemowe 4](#_Toc154744264)

[3.2 Instrukcje Instalacji 4](#_Toc154744265)

[4. STRUKTURA KATALOGÓW 5](#_Toc154744266)

[4.1 Opis Struktury Katalogów Projektu 5](#_Toc154744267)

[5. FRONTEND 5](#_Toc154744268)

[5.1 Struktura Aplikacji 5](#_Toc154744269)

[5.2 Wykorzystane Biblioteki/Frameworki 6](#_Toc154744270)

**Kraków 2023 r.**

## WPROWADZENIE

### Cel Dokumentacji

Celem niniejszej dokumentacji jest dostarczenie kompleksowej informacji dotyczącej struktury, funkcji, i zasady działania aplikacji webowej Docsee.

(Reszta dokumentacji zawiera szczegółowe opisy poszczególnych sekcji i komponentów aplikacji.)

### Cel Aplikacji

Aplikacja Docsee została stworzona jako projekt końcowy studiów podyplomowych o kierunku Programowanie aplikacji webowych na Akademii Górniczo Hutniczej AGH w Krakowie.

Niektóre funkcje aplikacji są w wersji demo, czyli nie działają tak jak w normalnej aplikacji webowej stworzonej dla większego odbioru.

Docsee to aplikacja internetowa zaprojektowana w celu usprawnienia przechowywania i organizacji kluczowych dokumentów osobistych. Dzięki Docsee użytkownicy mogą bezpiecznie przesyłać i przechowywać różne ważne dokumenty, w tym skany dowodów osobistych, dyplomów, certyfikatów, akt zatrudnienia, życiorysów i innych. Platforma kładzie nacisk na przyjazną dla użytkownika nawigację i oferuje intuicyjne funkcje kategoryzacji, dzięki czemu użytkownicy mogą łatwo zarządzać niezbędną dokumentacją i uzyskiwać do niej dostęp w jednym scentralizowanym miejscu. Docsee ma na celu uproszczenie zarządzania dokumentami i zwiększenie dostępności, zapewniając użytkownikom wygodny i bezpieczny sposób śledzenia ich najważniejszych zapisów.

### Zakres Dokumentacji

Dokumentacja techniczna ma na celu dostarczenie kompleksowej wiedzy na temat aplikacji, a jej zakres obejmuje następujące obszary:

#### Opis funkcji aplikacji

W tej sekcji opisane są główne funkcje i możliwości aplikacji. Każda funkcja jest szczegółowo przedstawiona, wskazując, jakie problemy rozwiązuje dla użytkownika końcowego i jakie korzyści przynosi.

#### Instrukcja instalacji

Szczegółowo opisuje kroki niezbędne do poprawnej instalacji i konfiguracji aplikacji na środowisku lokalnym.

#### Struktura katalogów

Opisuje strukturę katalogów projektu, pomagając programistom i administratorom w zlokalizowaniu konkretnych elementów aplikacji. Jest to szczególnie istotne dla zespołów pracujących nad projektem.

#### Frontend

Opisuje strukturę i organizację warstwy frontendowej, w tym wykorzystane biblioteki, frameworki, zarządzanie stanem, komunikację z backendem, komponenty i inne istotne elementy interfejsu użytkownika.

#### Backend

W tej sekcji znajdują się informacje na temat architektury backendu, wykorzystanych technologii, obsługi bazy danych, autentykacji i innych kluczowych elementów związanych z logiką dzialania

## ARCHITEKTURA SYSTEMU

### Działanie Aplikacji Docsee

Apikacja została napisana w języku JavaScript, wykorzystując framework React. Treść całej aplikacji jest napisana w języku angielskim. Strona główna Docsee informuje użytkownika w 2-3 zdaniach o tym do czego ona pokrótce służy, z przyciskiem przekierowującym do głównej części aplikacji (jeśli użytkownik jest zalogowany, jeżeli nie jest zostanie przekierowany do formularza z logowaniem). Znajduje się tutaj również nawigacja do kart logo Docsee, '**PRICING**','**ABOUT US**',**'LOG IN'**. Logo Docsee przenosi nas do strony startowej aplikacji.

Karta 'PRICING' przedstawia 3 oferty planu dzialania aplikacji (wybór planu ma charakter demonstracyjny, gdyż przyciski tych ofert przekierowują na stronę główną).

Karta 'ABOUT US' zawiera rozszerzony opis działania i funkcjonowania aplikacji Docsee. Opis ten dotyczy tego, jak miałaby działać i zachowywać się aplikacja wydana docelowo do powszechnego działania.

Karta 'LOG IN' przenosi użytkownika bezpośrednio do formularza logowania. Jeżeli użytkownik nie posiada konta w aplikacji, może stworzyć je za pomocą formularza rejestracji. Gdy użytkownik się zaloguję link ‘LOG IN’ zniknie.

Jeżeli logowanie/rejestracja przebiegły pomyślnie użytkownik dostaje dostęp do głównej częśći aplikacji. W wersji desktopowej są dwa panele lewy do nawigacji i prawy do wyświetalnia dokumentów oraz plików w danych kategoriach. W wersji mobilnej panel z nawigacją znajduję się u góry ekranu, natomiast panel główny pod nim.

Panel nawigacji składa się z przycisków ‘**DOCS**’ oraz ‘**CATEGORIES**’. Dzięki zastosowaniu odpowiedniej nawigacji z pomocą biblioteki react-router, przycisk DOCS jest domyślnie aktywny. Powoduję to ,że poniżej zostają wyświetlone nazwy dokumentów przesłanych przez użytkownika na serwer. Jeżeli użytkownik nie posiada żadnych plików na serwerze wyświetlany jest napis zachęcający do dodania pierwszego dokumentu. Każdy dokument po dodaniu można usunać. Pod tą listą jest przycisk otwierający formularz dodwania nowego dokumentu. Pozwala on na wybranie dowolnego pliku w formacie .jpg, .jpeg, .png, .pdf. Zastosowałem tu walidację formularza przez co nie ma możliwośći wysłania formularza bez uprzedniego dodania pliku oraz wyboru odpowiadającej kategorii, którą można wybrać z listy (jeżeli jakaś z kategorii istnieje) lub stworzyć samemu. Po dodaniu nowego dokumentu zostanie wyświetlona jego nazwa w powyższej liście. Każdy element listy jest klikalny i przekierowywuje użytkownika do nowej karty, jednak nie przeładowywuje strony co jest zasługą odowiednich bibliotek reacta.Gdy element z listy zostanie kliknięty, do adresu URL zostanie dopisane id danego dokumentu oraz jego nazwa np: „**app/docs/84?name=38493f78-50d3-42ef-81cd-eaac757b4aee-cv.pdf**”. Dzięki temu możliwe jest namierzenie id danego dokumentu poprzez odczytanie adresu URL i wyświetlenie go na głównym panelu.Pozwala to na swobodne przeglądanie danych dokumentów oraz ich pobieranie z serwera. Na panelu nawigacji natomiast zamiast wcześniejszej listy, wyświetlane są teraz główne informacje o danym dokumencie, takich jak jego nazwa, kategoria, typ, rozmiar oraz data ostaniej modyfikacji. Pod listą jest przycisk ‘**BACK**’, który cofa nas do wcześniejszego miejsca.

Przycisk ‘**CATEGORIES’** wyświetla wszystkie kategorie które zostały przez nasz utworzone, w prostokątnych bloczkach wraz z informacją o ilośći dokumentów należących do danych kartegorii. Również jest to element klikalny przenoszący nas do nowej karty (nowego URL i bez przeładowywania strony).Każdą kategorię można usunąć tylko wtedy gdy liczba dokumentów, które się w niej znajdują będzie równa 0, w przeciwnym razie przycisk do usuwania kategori nie bedzię widoczny. Po kliknięciu w dany bloczek kategorii uruchamiana jest nowa lista w panelu nawigacji zawierająca nazwe kategorii, a poniżej wszystkie nazwy dokumentów, które ona zawiera. Na panelu głównym natomiast zostają wysietlone bloki z nazwą dokumentu, datą utworzenia, ostatnią modyfikacją oraz linkiem do dokumentu, który zostanie wyświetlony w nowej karcie.

Dodatkowo w części głównej aplikacji znajduję się stopka z informacją o prawach autorskich. W górnym prawym rogu jest wyświetlony e-mail obecnie zalgowanego użytkownika oraz przycisk służący do wylogowania się z aplikacji.

## INSTRUKCJA INSTALACJI

### Wymagania Systemowe

Do uruchomienia aplikacji lokalnie zaleca się korzystanie z programu Visual Studio Code. Dodatkwo system musi spełniać następujące wymagania:

* Node.js (wersja >= v18.13.0)
* Npm (wersja >= 8.19.3)

Wymagane jest zainstalowanie bądź zaktualizowanie bierzących programów w celu uruchomienia aplikacji w środowisku VS Code.

### Instrukcje Instalacji

Aplikacje w wersji LIVE można bezpośrednio zobaczyć pod adresem <https://docsee.netlify.app>. Dzięki serwisowi Netlify.com aplikacja została umieszczona na serwerze.

W celu uruchomienia aplikacji lokalnie na komputrze należy postąpić następująco:

1. Uruchom program VS Code, następnie otwórz folder główny z aplikacją Docsee lub sklonuj repozytorium: `git clone <https://github.com/Rockwood98/docsee>
2. Proszę włączyć terminal, a następnie przejść do katalogu projektu poleceniem:

‘cd nazwa\_katalogu’ (np. ‘cd docsee’)

1. Zainstaluj zależności ‘npm install’ lub’ npm i’
2. Uruchom aplikację poleceniem ‘npm run dev’

## STRUKTURA KATALOGÓW

### Opis Struktury Katalogów Projektu

Struktura katalogów projektu została zorganizowana zgodnie z wytycznymi przedstawionymi w warunkach zaliczenia na kierunku Programowanie Aplikacji Webowych na AGH.

Główny katalog to ‘docsee/’. W nim znajduję się ‘/src’ zawierający wszystkie źródła projektu. Katalog ‘/auth’ zawiera pliki służące do zarządzania i konfigurowania kontem użytkownika, oraz zapewniania odpowiedniej autoryzacji.

Katalog ‘/components’ zawiera dosłownie wszystkie komponenty znajdujące się w aplikacji, przykładowo komponent PageNav.jsx oraz PageNav.module.css (plik z CSS) odpowiedzialny za nawigacje w aplikacji, Button.jsx + Button.module.css tworzący wszystkie przyciski znajdujące się na stronie.

Folder ‘/pages’ przechowuje kod opowiedzialny za budowę wszystkich podstron w aplikacji, przykładowo Homepage.jsx/Homepage.module.css jest to kod strony głównej, witającej w Docsee, lub Login.jsx/Login.module.css odpowiadający stronie z formularzem logowania.

Katalog ‘/services’obejmuję dokumenty w formacie JavaScrpit, które odpwiadają za wysyłanie odpowiednich zapytań/komend do bazy danych w serwisie Supabase.com

Folder ‘/imgs’ zawiera wszystkie obrazy w formacie .png i .jpg znajdujące się na stronie, natomiast katalog ‘/assets’ zachowuje w sobie domyślne SVG dostarczone przez sam react.

Poza katalogami w folderze ‘/src’ są pliki takie jak App.js, jest to kod budujący całą aplikację w jedną całość. Index.css składa się z głównych styli odpowiadających całemu programowi oraz main.jsx, który dotyczy importowania biblioteki react.

Poza tym w katalogu głównym znajdują się pliki konfiguracyjne wymagane do uruchomienia aplikacji tj. .eslintrc- do konfiguracji narzędzia ESLint służącego do statystycznego analizowania kodu źródłowego w języku JavaScript, .gitignore dla serwisu Github, index.html, package.json/package-lock.json do odpowiedniego skonfigurowania środowiska w czasie instalacji, vite.confg- konfiguracja Vite, README.md oraz katalog ‘/docs’ z dokumentacją techniczną.

## FRONTEND

### Struktura Aplikacji

Docsee zbudowana jest za pomocą biblioteki react. Wszystkie elementy tej aplikacji podzielone są na oddzielne pliki tzw. moduły. W głównym plku App.jsx importowane są wszystkie strony budujące aplikacje i składane w całość, te strony pobierają natomiast wszystkie napisane wcześniej oddzielne komponenty, które je budują. Kod HTML jest tak naprawdę kodem JavaScript pisanym w języku JSX, jest to składnia, która umożliwia pisanie struktur UI w sposób podobny do HTML, jednak jest to nadal JavaScript. JSX jest bardziej deklarytywny i zwięzły niż czysty JavaScript, co sprawia ,że kod jest łatwiejszy do zrozumienia, umożliwa również korzystanie z wyrażeń JavaScript bezpośrednio wewnątrz struktury JSX, co ułatwia dynamiczne renderowanie treści. Jedną z dodatkowych funkcji jest obsługa zdarzeń, umożliwiająca bardziej rozbudowane interakcje w interfejsie użytkownika. Dzięki tworzeniu komponentów ‘welorazowego użytku’ możemy umieścić je na stronie w wielu miejsach, np w przypadku komonentu Button, wystaczy w danym miejscu w aplikacji zaimportować ten komonent następnie określić jego typ wyznaczony poprzez style CSS i nie tworzymy nowych zbędnych linijek kodu.

Jeżeli chodzi o część wizualną aplikacji jest ona pisana w ‘czystym’ CSS bez użycia zewnętrzych bibliotek typu Bootstrap czy Tailwind. W projekcie style zostały stworzone poprzez CSS Modules, jest to technika modularyzacji stylów w projektach internetowych. Pomaga ona unikać konfilktów nazw klas, zwiększa izolację komponentów oraz ułatwaa zarządzanie stylami w dużych projektach. CSS Modules wprowadzają lokalny zasięg dla klas. Każdy moduł stylów jest traktowany jako osobny zakres, co oznacza ,że nazwy klas są unikalne w obrębie danego modułu. Rozwiązanie to eliminuje globalne zasięgi, co jest szczególnie istotne w dużych projektach, gdzie konflikty nazw klas mogą prowadzić do nieprzewidywalnych stylizacyjnych skutków ubocznych. Stylizacja jest importowana bezpośrednio do plików JavaScript/JSX, co ułatwia przypisanie klas do elementów w kodzie komponentu. Iportowanie odbywa się przy użyciu specjalnej składni przykładowo: ‘import styles from ‘./nazwa\_komponentu.module.css’. Dużym plusem jest możliwość korzystania z dynamicznych klas w zależności od kontekstu.

### Wykorzystane Biblioteki/Frameworki

Frontend aplikacji webowej korzytsa z biblioteki react + vite. React to biblioteka stworzona przez Facebooka, służąca do budowy interfejsów użytkownika. Oparty jest na koncepcji komponentów, co oznacza ,że interfejs użytkownika jest rozbijany na mniejsze, samodzielne elementy zwane komponentami. Każdy komponent może posiadać swój własny stan (state) oraz metody, co zapewnia reużywalność i skalowalność kodu. React używa wirtualnego DOM-u, co pozwala na minimalizację liczby manipulacji w jego rzeczywistą strukturę, co przekłada się na lepszą wydajność. Dane przekazywane są w jednym kierunku, ułatwia to debugowanie i zarządzanie stanem.

Vite natomiast dotarcza środowisko do budowy aplikacji opartych na JavaScript, TypeScript, React oraz Vue. Zapewnia szybkość budowy, wbudowany serwer deweloperski, brak zbędnych konfiguracji np. ESLint oraz lepszą optymalizację.

Oba te narzędzia możemy łączyć w celu budowy nowoczesnych i wydajnych aplikacji internetowych. React dostarcza potężną bibliotekę do budowy interfejsów użytkownika, a Vite oferuje środowisko do szybkiego rozwoju projektów JavaScript.