

SlideBuilder – Dokumentacja README

Cel aplikacji: SlideBuilder to proste, jednoplikowe narzędzie (HTML + CSS + JS) ułatwiające handlowcom i marketerom szybkie budowanie spersonalizowanych prezentacji PDF na bazie przygotowanych wcześniej slajdów projektów, oferty usług i danych kontaktowych.

Spis treści

1. Schemat danych w Supabase
2. Zewnętrzne biblioteki
3. Przepływ aplikacji
4. Zmienne globalne
5. Opis funkcji
6. Warstwa wizualna (CSS)
7. Rozbudowa i utrzymanie
8. Licencja

Schemat danych w Supabase

Tabela	Przykładowe kolumny	Opis
slides	id (PK, int), investor (text), region (text), industry (text), area (int), file_url (text)	Każdy wiersz to <i>jeden slajd</i> prezentacji projektu.
services	id, service_type (text), file_url	Pliki PDF ofert (np. „Zaprojektuj i Wybuduj”).
contact	id, name (text), file_url	Slajdy z danymi handlowca.

Warto wiedzieć: Aplikacja pobiera całe pliki PDF dopiero w momencie generowania końcowego dokumentu, więc zwykłe podglądy nie obciążają transferu.

Zewnętrzne biblioteki

Biblioteka	Rola
Supabase JS (@supabase/supabase-js)	Pobieranie danych i plików PDF ze storage.
SortableJS	Drag-and-drop i ręczne sortowanie wybranych slajdów.
PDF-Lib	Łączenie wielu PDF-ów w jeden (usługa + slajdy + kontakt).
PDF.js	Renderowanie podglądu plików PDF w oknie modalnym.
Font Awesome	Ikony interfejsu (lupa, budynek, linijka...).

Wszystkie zasoby są osadzone z CDN – aplikacja nie wymaga Node.js ani bundlera.

Przepływ aplikacji

1. **Render strony** – po zdarzeniu `DOMContentLoaded`:
 - a. Uzupelnienie selektorów filtrów (`populateFilter`).
 - b. Pobranie początkowej listy slajdów (`updateSlidesTable`).
 - c. Wczytanie usług i kontaktów.
 - d. Inicjalizacja `SortableJS` dla listy wybranych slajdów.
2. **Filtrowanie** – użytkownik wprowadza kryteria, klika „Zastosuj filtry”. Funkcja `updateSlidesTable`:
 - a. Zbiera wartości pól.
 - b. Woła `fetchSlides` (buduje zapytanie Supabase z warunkami).
 - c. Renderuje tabelę, dzieli wyniki na strony i tworzy paginację.
3. **Wybór slajdów** – każdy wiersz ma przycisk „Wybierz”. `addProject` dodaje slajd do listy, aktualizuje numerację.
4. **Podgląd inwestycji** – kliknięcie w nazwę w tabeli uruchamia `showInvestmentPreview`, która w oknie modalnym renderuje wszystkie strony PDF konkretnego projektu.
5. **Generowanie PDF-a** – przycisk „Generuj PDF” uruchamia `generatePDFDocument`:
 - a. Weryfikuje, czy wybrano usługę, slajdy i kontakt.
 - b. Ładuje kolejno pliki PDF (`Fetch` → `ArrayBuffer` → `PDF-Lib`).
 - c. Kopiuje strony i łączy je w jednym dokumencie.
 - d. Tworzy URL do bloba i otwiera go w nowej karcie.

Zmienne globalne

Zmienna	Typ	Przeznaczenie
supabaseUrl	string	Adres instancji Supabase.
supabaseKey	string	Klucz publiczny „anon” – w produkcji przenieś do .env!
supabaseClient	object	Instancja klienta Supabase JS do zapytań i storage.
allSlides	array	Wyniki aktualnego zapytania przed paginacją.
currentPage	number	Numer widocznej strony tabeli (1 lub 2).
pageSize	number	Ilość wierszy na stronę (stała = 5).

Opis funkcji

Poniżej „kuchnia” aplikacji – funkcje są krótkie i realizują pojedyncze zadania.

Funkcja	Co robi?	Kluczowe parametry
toTitleCase(str)	Zmienia tekst z uwzględnieniem polskich znaków na „Pierwsza Duża Litera”.	str – dowolny ciąg znaków.
populateFilter(columnName, selectId)	Wypełnia select unikalnymi wartościami z podanej kolumny tabeli slides.	columnName, selectId (id pola).
fetchSlides(filters)	Czyta slajdy z tabeli slides, buduje zapytanie z warunkami (od-do, branża, region...).	filters – obiekt kryteriów.
updateSlidesTable()	Spina cały proces filtrowania: zbiera dane formularza, zleca fetchSlides, resetuje paginację.	–
renderSlidesTable()	Renderuje wiersze tabeli + przycisk „Wybierz” dla każdego slajdu.	–
updatePagination() / changePage()	Liczenie, czy pokazać kółka stron; zmiana currentPage.	–
fetchServices / updateServices	Pobiera ofertę usług i renderuje kafelki (jeden zawsze domyślnie aktywny).	–
fetchContacts / updateContacts	Analogicznie do usług, tylko dla osób kontaktowych.	–
addProject(id, fileUrl, investor)	Dodaje slajd do listy wybranych (ul#selected-list).	id, fileUrl, investor.

removeProject(button)	Usuwa slajd z listy po kliknięciu „Usuń”.	button (this).
updateSlidesOrder()	Po każdym drag-and-drop nadaje kolejny numer „Slajd n: ...”.	–
resetFilters()	Czyści pola formularza i przywraca pełną listę slajdów.	–
generatePDFDocument()	Łączy pliki PDF w podanej kolejności i otwiera wynik.	–
showInvestmentPreview(name)	Wyszukuje wszystkie slajdy danej inwestycji i wywołuje PDF.js dla każdej strony.	name inwestora.
closeInvestmentPreview()	Ukrywa modal z podglądem.	–

TIP: Funkcje asynchroniczne (async/await) upraszczają obsługę promes i poprawiają czytelność kodu.

Warstwa wizualna (CSS)

- **Design tokens** – zmienne --primary-color, --dark-text itd. ułatwiają szybką zmianę motywu.
- **Układ** – flex i grid; maksymalna szerokość kontenera: 1000 px.
- **Klasy narzędziowe:** .btn, .section, .filter-item. Paginacja kółkami .page-circle.
- **Tryb modalny** – przeźroczysty overlay + position: fixed.

Za wygląd ikon odpowiada CDN Font Awesome, a delikatne animacje – właściwość transition.

Rozbudowa i utrzymanie

1. **Bezpieczeństwo klucza API** – wstaw go z pliku .env i wczytaj dynamicznie (np. Vite, Netlify Env).
2. **Więcej niż 10 slajdów?** Zwiększ pageSize lub dodaj dynamiczną paginację.
3. **Obsługa wielu filtrów** – dodaj nowe pola i rozszerz populateFilter oraz fetchSlides o kolejną kolumnę.
4. **Lazy-loading PDF-ów** – przy dużych plikach rozważ pobieranie i łączenie po stronie serwera (np. funkcja Edge).

