

Krótki Raport

demo:

<https://euvicdemo-kolodziej.netlify.app>

Spis treści:

1. Techstack
2. Struktura projektu
3. Wykonanie

1. Tech stack:

- React.js (główne środowisko)
- css (tworzenie stylów i widoków)
- axios (obsługa zapytań HTTP)
- firebase (baza danych do przechowywania wysłanych wiadomości)
- React google recaptcha (zabezpieczenie antybotowe Google)
- React router dom (zmiana url podczas nawigacji po stronie)
- React icons (importowanie ikon)

2. Struktura projektu

Pliki css znajdują się w folderach bezpośrednio przy plikach zwracających JSX które stylizują.

Projekt podzielony został na foldery:

- UI (komponenty interfejsu, które są uniwersalne (tutaj spinner ładowania)
- Images (zdjęcia, obrazki, pliki svg)
- hoc (Higher Order Components) (Komponenty służące obsłudze innych komponentów)
- Containers (Kontenery na treść pokazywaną na danej stronie)
 - folder [page_name] (podfoldery w zależności od strony)
- Components (Komponenty używane w kontenerach)

- .atoms - zbiór najmniejszych komponentów, które nie są pierwotnie renderowane na stronie
- .molecules - zbiór większych komponentów, które nie są pierwotnie renderowane na stronie
- folder [container_name] - podfoldery w zależności od strony lub funkcjonalności

3. Wykonanie

W kontenerze ContactPage zaimplementowana została strona składająca się z komponentów z których wyróżnić można: 2 komponenty layoutowe (Footer i NavBar) oraz 2 komponenty przedstawiające zawartość strony. W komponencie ContactForm znajdują się pola input oraz przycisk służące do wysłania wiadomości do bazy danych. Wysłanie jest obsługiwane tylko w przypadku poprawności wprowadzonych informacji tj. w każdym polu znajduje się tekst oraz adres email posiada poprawną składnię. W przypadku braku poprawności któregoś z pól input wyświetlana jest informacja poniżej oraz ikona w prawej części pola. Po wysłaniu post API wyświetlana jest informacja czy wiadomość została wysłana poprawnie oraz ewentualny kod błędu.

Na poniższym zdjęciu przedstawione zostały informacje zawarte w bazie danych po uprzednim wysłaniu żądania http.

