

Do działania formularza kontaktowego niezbędne jest jednocześnie uruchomienie naszej reactowej aplikacji (npm start) oraz w drugiej karcie terminala uruchomienie serwera do wysyłania wiadomości, z poziomu components/pages - (node server.js)

W razie jakichkolwiek problemów, na dowód działania wysyłania maili z formularza kontaktowego, czy tworzenie nowych kont użytkownika zamieszczamy poniższe zrzuty ekranów.

React App
localhost:3000/contact

CaRent

Kontakt Koszyk Logowanie

Komunikat ze strony localhost:3000
Message Sent

W razie jakichkolwiek problemów, na dowód działania wysyłania maili z formularza kontaktowego, czy tworzenie nowych kont użytkownika zamieszczamy poniższe zrzuty ekranów.

Imię:

Michał Olejnik

Adres e-mail:

michal.p.olejnik@wp.pl

Wiadomość:

Dzień dobry, wiadomość testowa

Sending...

Lub telefonicznie, na numer +48 111 111 111

Wiadomości co prawda trafiają do folderu SPAM, jednak docierają

Spam - olejnik1999@wp.pl

Contact Form Submission - X

Pobierz Napisz Komunikator Adresy Etykieta

Od olejniksmuszkiewicz@gmail.com ☆

Temat **Contact Form Submission**

Do Ja <olejnik1999@wp.pl> ☆

Name: Michał Olejnik

Email: michal.p.olejnik@wp.pl

Message: Dzień dobry, wiadomość testowa

Walidacje w naszym projekcie w procesie rejestracji:

1. Czy nazwa użytkownika – mail, potencjalnie istnieje? Czy nie zawiera polskich znaków:

The screenshot shows the CaRent registration page. The title is "Rejestracja E-mail". The email field contains "MichalOlejnikBartoszSmuszkiewicz@wp.pl". A validation error message is displayed: "Część przed znakiem .@ nie może zawierać symbolu „.”". The password and confirm password fields are empty. The "Rejestracja" button is disabled, and the "Log In" link is visible.

2. Czy pola nie są puste?

The screenshot shows the CaRent registration page. The title is "Rejestracja E-mail". The email field is empty. A validation error message is displayed: "Wypełnij to pole.". The password and confirm password fields are also empty. The "Rejestracja" button is disabled, and the "Log In" link is visible.

3. Czy hasła są identyczne?

The screenshot shows the CaRent registration page. The title is "Rejestracja". Below the title, it says "Hasła nie są identyczne!". The email field contains "test@test234.com". The password and confirm password fields are empty. The "Rejestracja" button is disabled, and the "Log In" link is visible.

4. Czy użytkownik już istnieje?

Po pomyślnej rejestracji, informacja o nowym użytkowniku przychodzi na stronę Firebase, co ważne informacje te uzyskać może jedynie Administrator.

The image shows two parts of a web application. The top part is a registration form for 'CaRent' with fields for E-mail, Hasło (Password), and Powtórz hasło (Repeat password), and a 'Rejestracja' (Registration) button. The bottom part is a screenshot of the Firebase Authentication console showing a list of users, including the user 'test@test1234.com'.

CaRent Registration Form:

- Header: CaRent
- Navigation: Home, Oferta, Kontakt, Logowanie
- Form Title: Rejestracja E-mail
- Fields: E-mail (test@test1234.com), Hasło, Powtórz hasło
- Buttons: Rejestracja, Masz już konto? Log In

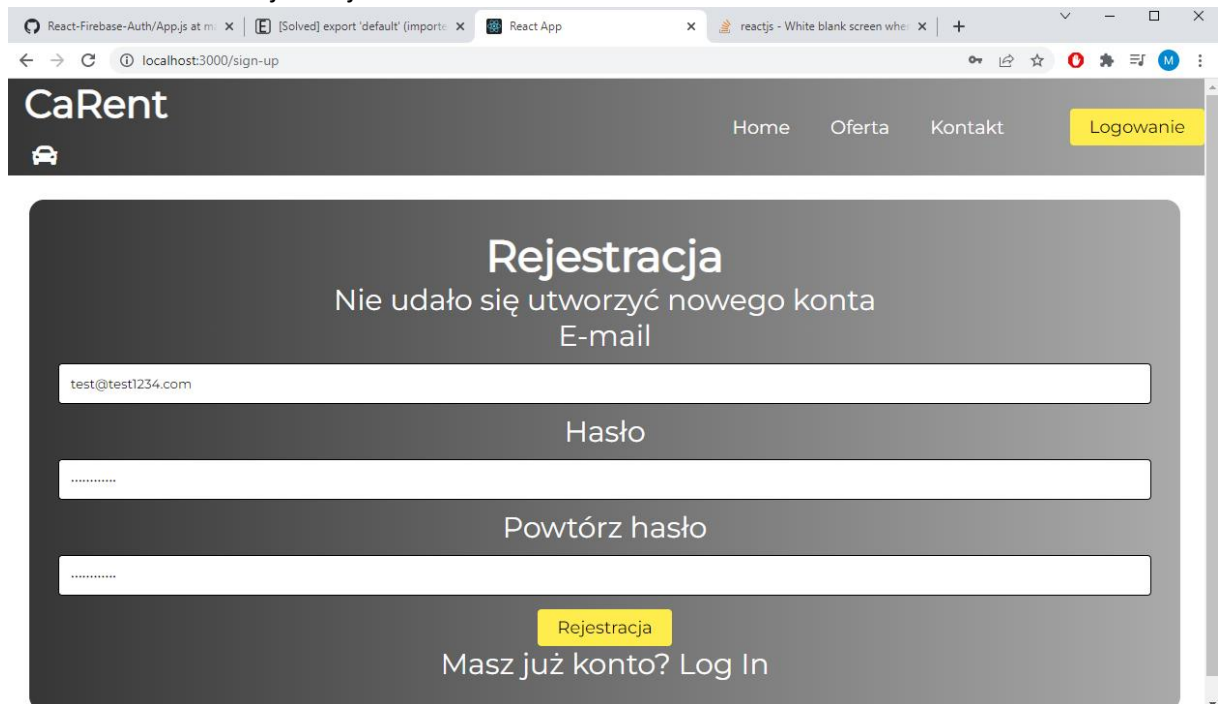
Firebase Authentication Console:

- Project: Auth-Development
- Section: Authentication
- Sub-section: Users
- Search: Search by email address, phone number, or user UID
- Buttons: Add user, Get started
- Table:

Identifier	Providers	Created	Signed In	User UID
test@test1234.com		Feb 14, 2022	Feb 14, 2022	Pw63XxiAAAnMUh3GWI7F3C50557...

Rows per page: 50 1 - 1 of 1

Gdy konto o podanej nazwie test@test1234.com istnieje, strona nie pozwoli na ponowne utworzenie konta o tej samej nazwie



The screenshot shows a web browser window with the URL `localhost:3000/sign-up`. The page has a dark header with the 'CaRent' logo and navigation links: 'Home', 'Oferta', 'Kontakt', and a yellow 'Logowanie' button. The main content area is a light gray box with the title 'Rejestracja' and the message 'Nie udało się utworzyć nowego konta E-mail'. Below this are three input fields: the first contains 'test@test1234.com', the second is labeled 'Hasło' and contains '*****', and the third is labeled 'Powtórz hasło' and also contains '*****'. A yellow 'Rejestracja' button is positioned below the third field. At the bottom of the box, it says 'Masz już konto? Log In'.

Dlatego również został zaimplementowany Loading, w przypadku gdy użytkownik przypadkiem wciśnie 2x szybko przycisk „rejestracja”, nic się nie stanie, a konto zostanie utworzone

5. Nie można zalogować się na konto, gdy ktoś jest już zalogowany. Firebase automatycznie sprawdza czy ktoś jest już zalogowany na konto.
6. Przycisk „Profil” w przypadku, gdy użytkownik nie jest zalogowany, przenosi na stronę logowania. Po zalogowaniu zmienia swoją funkcjonalność. Zasada ta została określona w pliku PrivateRoute.js

Funkcjonalności:

Użytkownik może się zarejestrować, zalogować, wylogować, zresetować hasło, zaktualizować swoje dane (mail i hasło)

Może przeglądać naszą ofertę kart podarunkowych, dodawać je do koszyka. Następnie informacje o zamówieniu docierają do nas drogą mailową.

(Docelowo pomiędzy koszykiem a dostarczeniem informacji o zamówieniu, pojawiałby się system płatności, z możliwością realizacji zamówienia poprzez kartę kredytową, blik, czy PayPal. Jednak ta funkcjonalność wykraczała poza wymagania projektowe oraz nasze umiejętności.)

Strona zawiera formularz kontaktowy, z podpętym adresem mailowym, stworzonym na potrzeby naszego projektu. Maile są dostarczane na określony w pliku (server.js) adres

mailowy (W tym przypadku olejnik1999@wp.pl). Wiadomości docierają, choć trafiają do folderu SPAM.

Do stworzenia naszej aplikacji użyliśmy:

Do stworzenia naszego **panelu nawigacji** – importowanie ikon. Spośród imponującej bazy ikon skorzystaliśmy z:

Font Awesome – <https://fontawesome.com/>

- fa-car – ikona samochodu widoczna przy naszym logo
- fa-times oraz fa-bars – ikony do obsługi panelu nawigacyjnego w trybie mobilnym

WAŻNE! – aplikacja posiada „tryb mobilny” – design panelu nawigacyjnego zmienia się gdy szerokość otwartego okna przeglądarki spadnie poniżej 750px

Do stworzenia **formularza kontaktowego**

-Nodemailer - <https://nodemailer.com/about/>

- npm install express cors nodemailer

Ale również inspirowaliśmy się rozwiązaniami z: <https://w3collective.com/react-contact-form/>

Do obsługi oraz tworzenia nowych kont użytkowników:

Firebase - <https://console.firebase.google.com/>

Do stworzenia stylu naszej aplikacji posłużyły pliki CSS, natomiast eksperymentowaliśmy również z bootstrapem

npm i bootstrap react-bootstrap,

A efekty tych eksperymentów można obserwować poprzez uruchomienia skomentowanej linijki:

```
import "bootstrap/dist/css/bootstrap.min.css"
```

w pliku index.js

Tworząc naszą aplikację inspirowaliśmy się:

https://www.youtube.com/watch?v=fL8cFghTHwA&t=2s&ab_channel=BrianDesign

<https://w3collective.com/react-contact-form/>

https://www.youtube.com/watch?v=AmldY1Eb8tY&ab_channel=CodingwithBasir

<https://dev.to/iamandrewluca/private-route-in-react-router-v6-lg5>

https://www.youtube.com/watch?v=PKwu15ldZ7k&ab_channel=WebDevSimplified

<https://stackoverflow.com/questions/62861269/attempted-import-error-usehistory-is-not-exported-from-react-router-dom>

https://www.w3schools.com/react/react_router.asp

<https://stackoverflow.com/questions/63124161/attempted-import-error-switch-is-not-exported-from-react-router-dom>