

Aplikacja Bankowa

Bernard Cesarz

Strona powitalna

[Sign in](#)

Welcome in our bank!

Logowanie

[Sign in](#)

Step 1: Enter Email

Email

[Sign in](#)

Step 2: Enter Password

Password

Fill in following letters of the password: 4 5 11 15

If your password is shorter than the numbers of characters here, just put "-" in the place of this characters.

Logowanie

Mamy dwa ekrany logowania, pierwszy pyta o email, drugi o wybrane litery hasła. Niezależnie od tego czy email istnieje w bazie, przy drugim kroku numery liter są zawsze losowane, aby nie można było sprawdzić czy email jest używany.

Strona główna i tworzenie przelewu

Sign Out 🗑️
Account 🧑
Home 🏠
Secret info 🗝️
Your money: 999997000.0
Your email: bernard1@example.com

Transfers

[New transfer](#)

Sign Out 🗑️
Account 🧑
Home 🏠
Secret info 🗝️
Your money: 999997000.0
Your email: bernard1@example.com

New transfer

Title
Receiver name
Amount
Receiver code

[Back to transfers](#)

Sign Out 🗑️
Account 🧑
Home 🏠
Secret info 🗝️
Your money: 999996916.0
Your email: bernard1@example.com
Transfer was successfully created.

Transfer was successfully created.

Title: Hello there

Receiver name: General Kenobi

Amount: 42.0

Sender: 00001111022223333

Receiver: 0000111122223333

[Back to transfers](#)

Sekretne informacije

Sign Out 🏠

Account 🏠

Home 🏠

Secret info 🏠

Your money: 999996916.0

Your email: bernard1@example.com

Your secret information

Make sure no one is behind you, because this is very important info for real

Here is yours credit card number:

4123456789012340

Here is yours personal id number:

123-45-6780

Sekretnne informacje

Niestety nie udało mi się zaimplementować szyfrowania danych przed zapisaniem do bazy danych, ale tylko ich właściciel może je zobaczyć użytkując aplikację.

Edycja konta

Sign Out 🏠

Account 🏠

Home 🏠

Secret info 🏠

Your money: 999996916.0

Your email: bernard1@example.com

Current Email |

New Email |

Password (leave blank if you don't want to change it)

Password confirmation

Current password (we need your current password to confirm your changes)

Current Logins

User Agent	IP Address	Signed In At	Sign Out
Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:121.0) Gecko/20100101 Firefox/121.0	172.20.0.1	2024-01-22 01:17:17 UTC	<input type="button" value="Sign Out"/>
Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:121.0) Gecko/20100101 Firefox/121.0	172.20.0.1	2024-01-22 01:17:17 UTC	<input type="button" value="Sign Out"/>

Edycja konta

Możliwość zobaczenia aktywnie zalogowanych sesji, oraz ich zdalnego wylogowania. Istnieje także możliwość zmiany hasła. Niestety przy robieniu prezentacji zauważyłem, że przy udostępnianiu aplikacji przez nginx zniknęła funkcjonalność dynamicznego sprawdzania bezpieczeństwa hasła, która na bieżąco informowała użytkownika o tym jak dobre jest hasło.

Zapomniałem hasła

[Sign in](#)

Reset Password

Zapomniałem hasła

Przy logowaniu można wybrać opcję oznaczającą zapomnienie hasła, która pozwala po podaniu emaila dostać specjalnie wygenerowany adres, który pozwala zresetować hasło. Podobna funkcja istnieje też gdy wykonamy zbyt wiele prób (maksymalna dozwolona ilość to 7), wtedy nawet po poprawnym zalogowaniu, musimy najpierw podać swój adres mailowy i skorzystać z linku, który zostanie na niego wysłany.

Bibliografia

Działanie `has_secure_password`

<https://medium.com/swlh/understanding-whats-happening-under-the-hood-with-source-codes-and-existing-methods-3bca47cb5757>

Zaimplementowany mechanizm uwierzytelniania

<https://stevepolito.design/blog/rails-authentication-from-scratch>

Rozważany mechanizm częściowego hasła (ostatecznie zaimplementowany został autorski)

<https://magicofsecurity.com/partial-passwords-done-right/>

Przykładowa konfiguracja nginx, którą się inspirowałem

<https://desilva.io/posts/howto-setting-up-https-ssl-tls-with-nginx-rails>

Kod źródłowy dla metody `has_secure_password`

https://github.com/rails/rails/blob/main/activemodel/lib/active_model/secure_password.rb