

Projekt Bazy Danych 2

Zespół 24

Krzysztof Barankiewicz, Marek Dziecioł, Jakub Jurczak, Wiktor Kruk, Jakub Mieczkowski

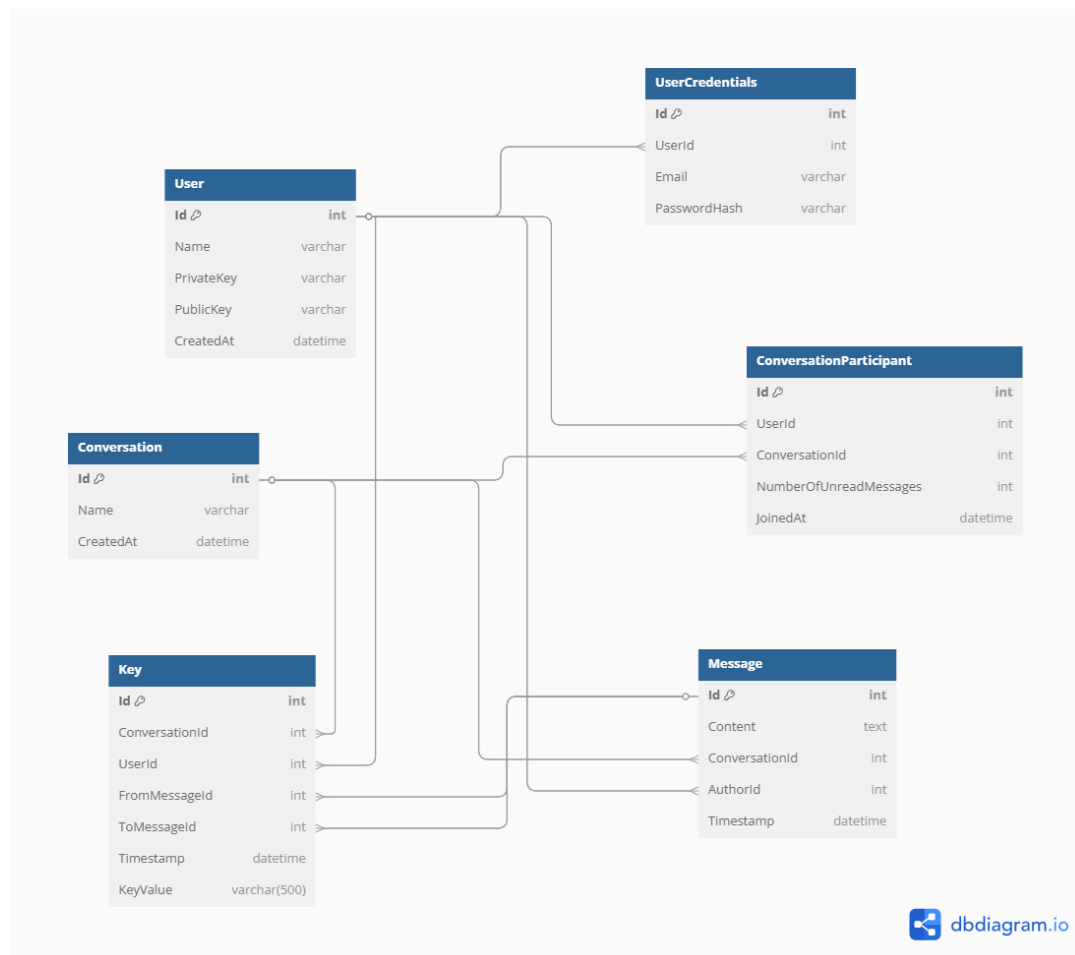
Temat projektu

Tematem projektu jest komunikator umożliwiający czatowanie ze znajomymi przez internet. Dodatkowa funkcjonalność to enkrypcja wiadomości end-to-end.

Technologie użyte

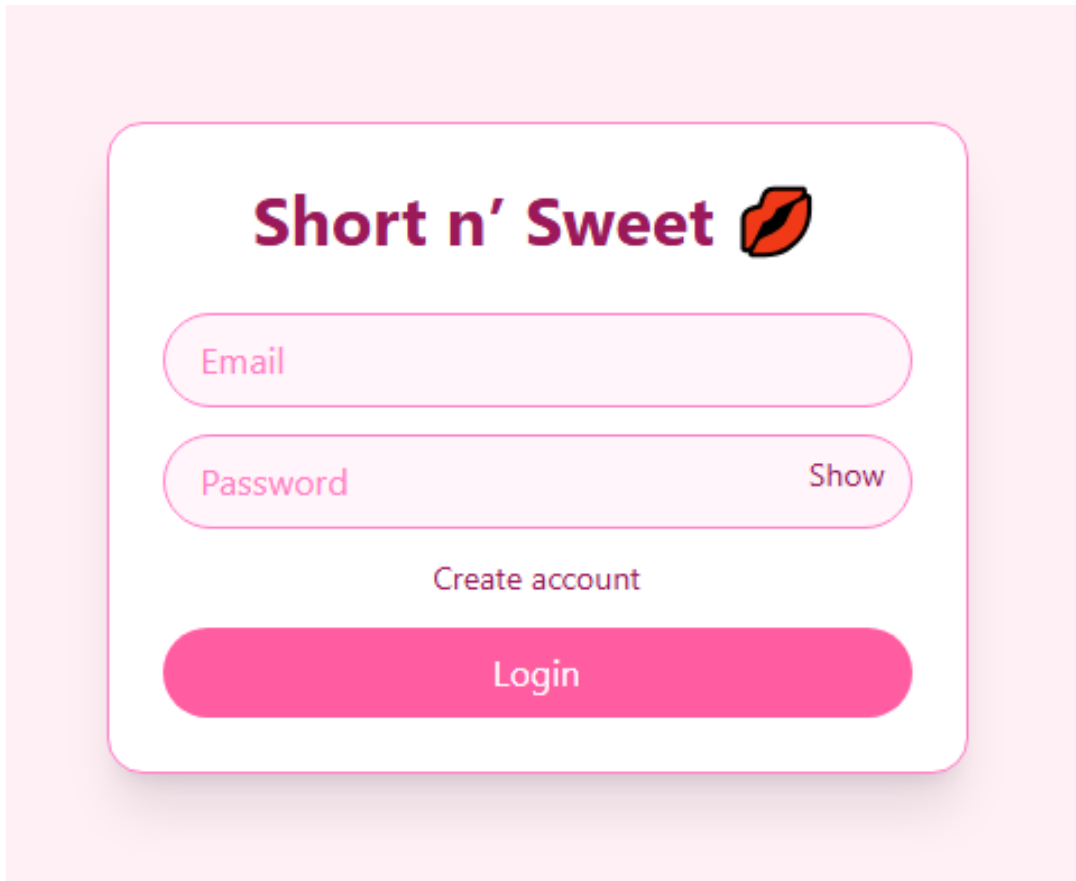
- React (Frontend)
- NodeJS (WebSocket)
- ASP.net (Backend)
- PostgreSQL (Baza danych)

Model ER bazy danych




Opis funkcji aplikacji webowej

1. Logowanie



The image shows a login form for an application named "Short n' Sweet". The form is centered on a light pink background. It features a title "Short n' Sweet" in a bold, dark pink font, followed by a red lips icon. Below the title are two input fields: "Email" and "Password". The "Password" field has a "Show" button next to it. Below the input fields is a "Create account" link and a large, rounded "Login" button.

Short n' Sweet 

Email

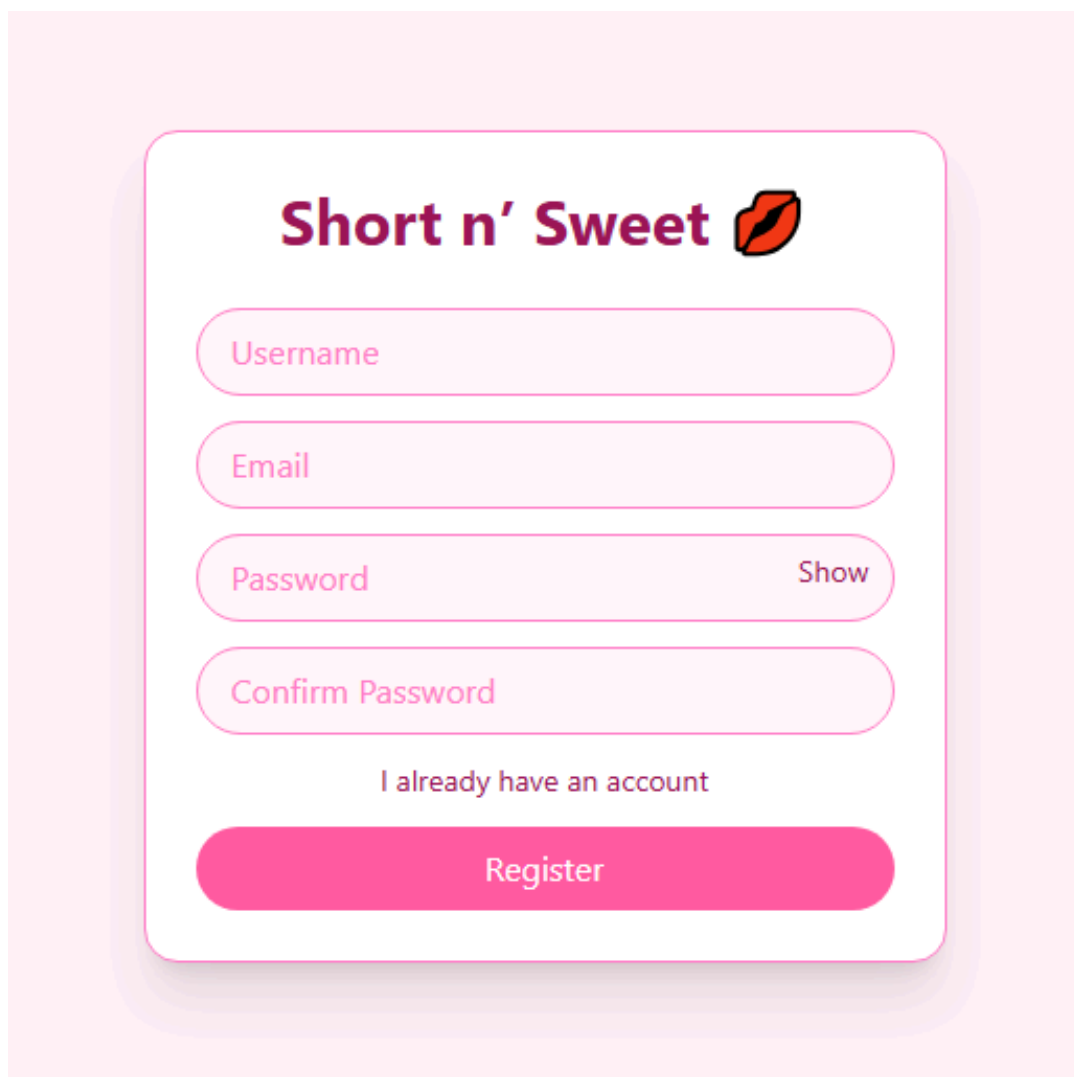
Password [Show](#)

[Create account](#)


Login

Po otwarciu witryny wyświetla się opcja zalogowania użytkownika, należy wpisać email oraz hasło.

2. Rejestracja użytkownika



The registration form is titled "Short n' Sweet" with a red lip icon. It contains four input fields: "Username", "Email", "Password", and "Confirm Password". The "Password" field has a "Show" toggle. Below the fields is a link "I already have an account" and a large red "Register" button.

Short n' Sweet 

Username

Email

Password Show

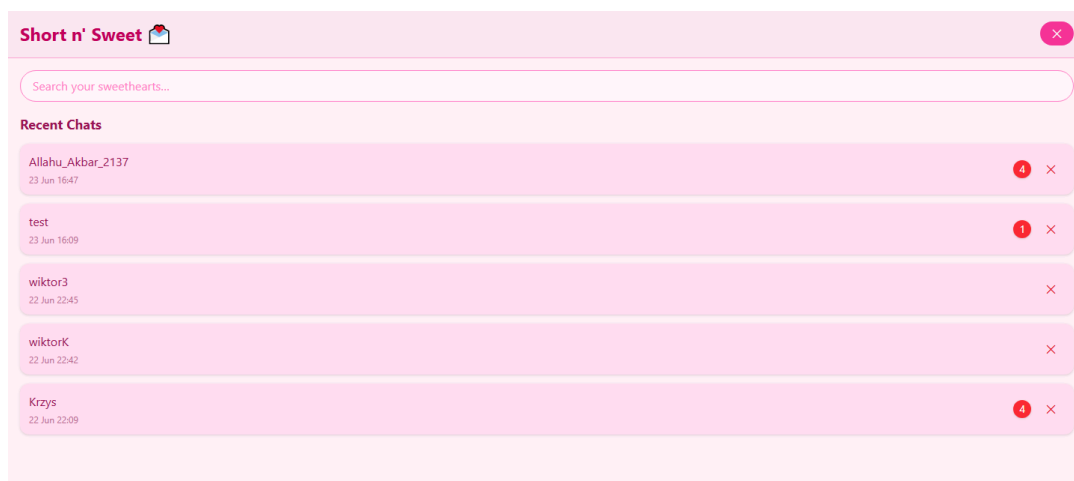
Confirm Password

[I already have an account](#)


Register

Kliknięcie w przycisk "Create account" przełącza widok z logowania na rejestrację. Aby utworzyć konto wystarczy uzupełnić powyższe pola.

3. Lista czatów / menu główne



The main chat list interface shows a header with the app name and a search bar. Below is a "Recent Chats" section with a list of chat items. Each item displays the contact name, the last message timestamp, and a red circle with a white number indicating unread messages. A close button (X) is visible on the right of each chat item.

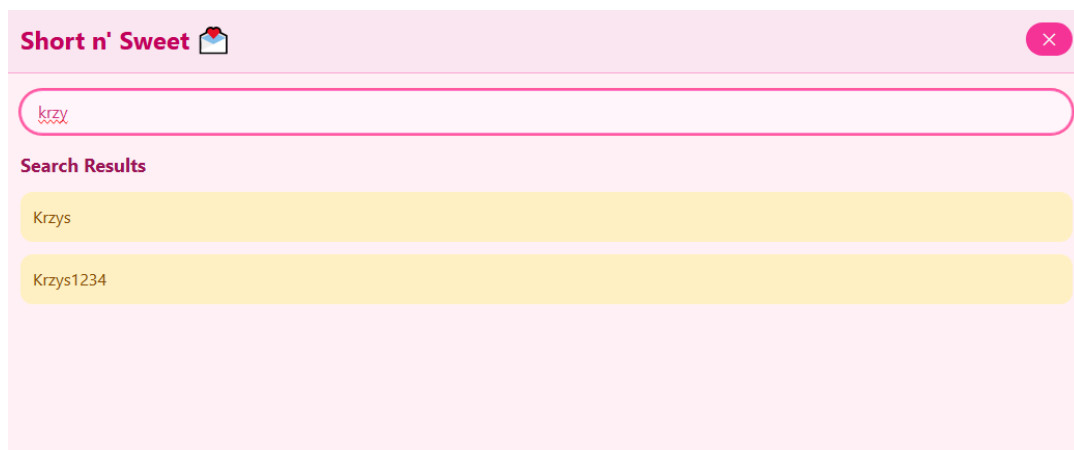
Short n' Sweet 

Search your sweethearts...

Recent Chats

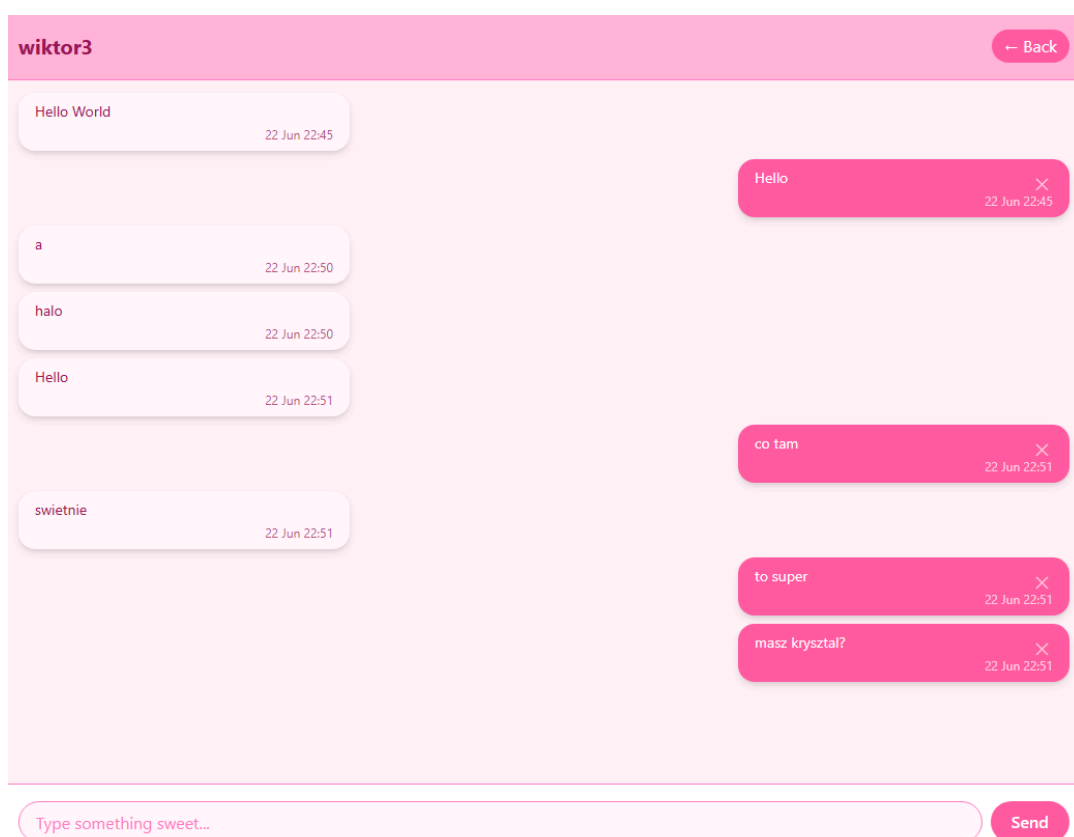
- Allahu_Akbar_2137
23 Jun 16:47 4 ×
- test
23 Jun 16:09 1 ×
- wiktor3
22 Jun 22:45 ×
- wiktorK
22 Jun 22:42 ×
- Krzys
22 Jun 22:09 4 ×

Po zalogowaniu/utworzeniu konta wyświetla się lista ostatnich czatów. Po prawej stronie wyświetla się ilość nieprzeczytanych wiadomości oraz możliwość usunięcia czatu.



Można również wyszukać innych użytkowników do czatowania.

4. Prywatne czaty



Funkcjonalność jest podobna do popularnych komunikatorów, np. Messenger lub WhatsApp. Za pomocą "X" można usuwać wiadomości po obu stronach.

Szyfrowanie

Każdy z użytkowników w trakcie rejestracji otrzymuje dwa klucze do szyfrowania hybrydowego. Obejmuje ono enkrypcję danych konwersacji - dopiero przy wyświetlaniu na ekranie są one odszyfrowywane.

`"id" : 4, "name" : "ajax", "publicKey" : " — — — — — BEGIN RSA PUBLIC KEY —`

Otrzymywane informacje o zalogowanym użytkowniku.

```
{ "id": 225, "content": "8nEvgYjc/FJDimdm+opcbrbv2KFclI/e/tLGeJhtHL4=",  
  "conversationId": 19, "conversationName": "wiktor3 ajax", "author": { "id": 4, "name": "ajax",  
    "publicKey": "", "createdAt": "2025-06-22T20:08:25.149492Z" }, "timestamp": "2025-06-  
22T20:52:30.497351Z" }, { "id": 224, "content":  
  "pHUBITREKz3jJm5PmQOVWTweBH3M5cgHyIc2OQaZHou61+rtPXGf5gexSQ==",  
  "conversationId": 19, "conversationName": "wiktor3 ajax", "author": { "id": 6, "name":  
    "wiktor3", "publicKey": "", "createdAt": "2025-06-22T20:44:53.296901Z" }, "timestamp":  
  "2025-06-22T20:51:55.546531Z" },
```

Przykładowe zaszyfrowane **bloki** z konwersacji

Frontend

Do szyfrowania wykorzystana została biblioteka *node-forge*. Ma on dwie strony *login* oraz *chatapp*.

Chatapp składa się z *ChatList* oraz *ChatRoom*.

Wiadomości w *ChatRoom* wykorzystują lazy loading (na start tylko 50 wiadomości, a jak przewinie się wyżej to ładuje więcej).

Styl i wygląd jest zrobiony za pomocą *tailwindcss*.

WebSocket

Prosty serwer służący do obsługi websocketa do wysyłania wiadomości. Dostaje informacje od Frontedu, że ktoś wysłał wiadomość do kogoś i jak ktoś inny obecnie ogląda wiadomości to dostaje informacje z websocketa, że powinien pobrać wiadomości.

Służy do szybszego bardziej efektywnego obsługiwania odświeżania wiadomości.