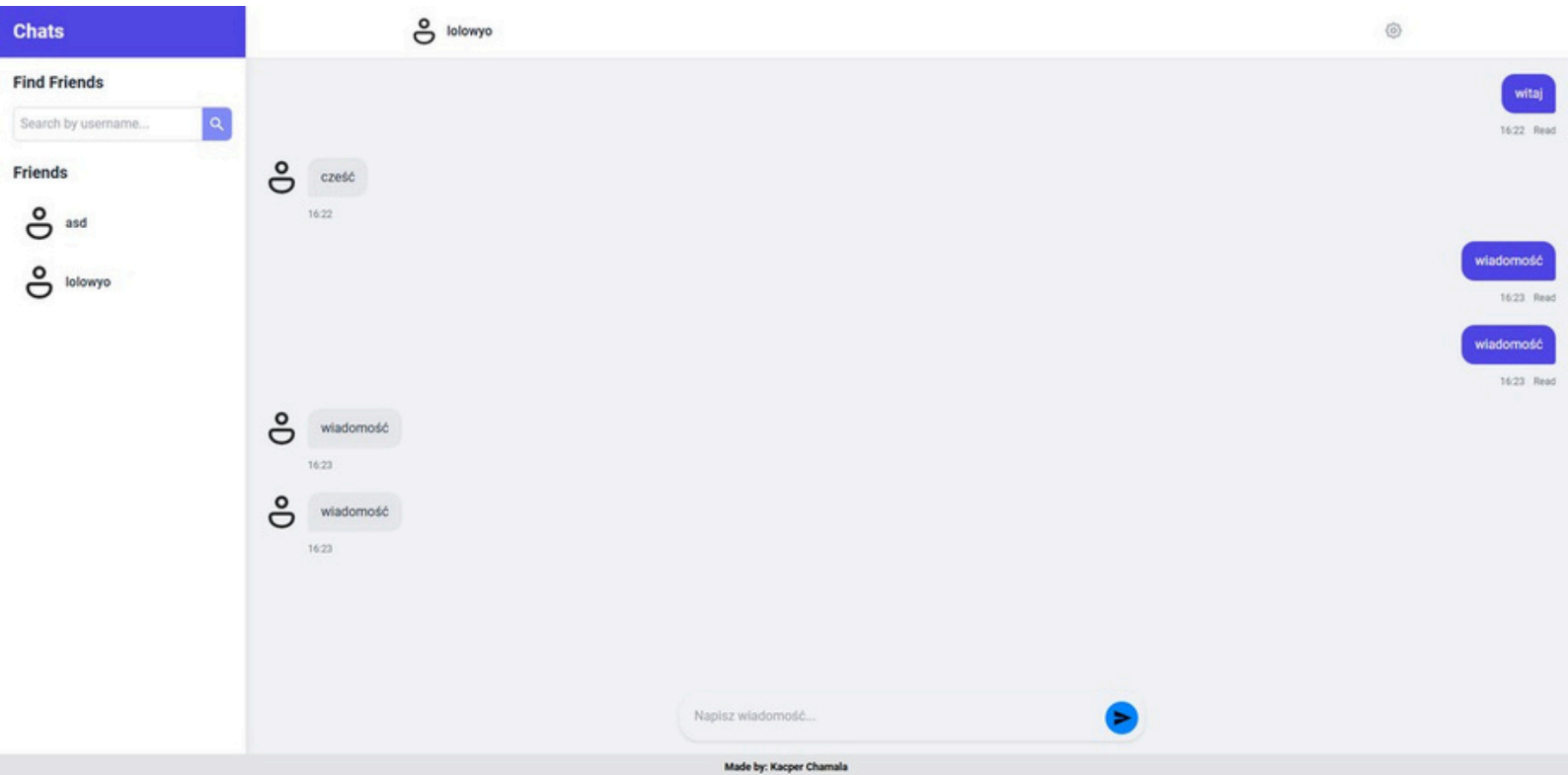


# Dokumentacja projektu

## "Komunikator Online"



Projekt wykonał: Kacper Chamala



## Spis treści

1. Opis programu
2. Wymagania systemowe
3. Sposób instalacji
4. Baza Danych
5. Modele BazoDanowe
6. Diagram dla operacji wysyłania wiadomości



# 1. Opis programu

Aplikacja umożliwia użytkownikom komunikację w czasie rzeczywistym z innymi zarejestrowanymi użytkownikami.

System wspiera funkcje logowania i rejestracji nowych użytkowników, dodawania znajomych oraz wysyłania i odbierania wiadomości.

Jest to aplikacja webowa stworzona z użyciem architektury MVC (Model-View-Controller), co zapewnia przejrzystość kodu, skalowalność i łatwość utrzymania.



## 2. Wymagania systemowe

- Serwer http: Apache 2.4.58
- PHP: 8.2.12
- Baza Danych MariaDB: 10.4.32
- Node JS: v20.10.0



## 3. Sposób instalacji

- Pliki projektu umieszczamy na przygotowanym serwerze apache który spełnia wymagania z sekcji 2.
- Zaimportuj plik komunikator.db.sql do swojej bazy danych
- Zmieniamy konfigurację w pliku config.php:

```
1 $config = [  
2     'db' => [  
3         'host' => "localhost",  
4         'user' => "root",  
5         'pass' => "",  
6         'name' => "komunikator",  
7         'handle' => null,  
8     ],  
9     'proto' => "http",  
10    'url' => "localhost/projekt1",  
11    'title' => "Komunikator online",  
12    'template' => "default",  
13    'default_page' => "/Web/Auth",  
14    'css' => ['style.css'],  
15    'js' => ['https://cdn.tailwindcss.com', 'tailwind.js'],  
16 ];
```

Musimy zmienić wartości zmiennych w sekcji 'db':

'host' - Adres serwera bazy danych (np. localhost),

'user' - Użytkownik bazy danych,

'pass' - Hasło do bazy danych,

'name' - Nazwa bazy danych, w której znajdują się tabele aplikacji

Oraz zmienić wartości zmiennych proto i url:

'proto' - Protokół używany przez aplikację (http lub https)

'url' - Adres URL aplikacji (np. nazwa folderu na serwerze)

- Zmieniamy konfigurację w pliku .htaccess:

A terminal window with a dark background and three colored window control buttons (red, yellow, green) at the top left. It displays four lines of configuration code for the .htaccess file.

```
1 RewriteEngine On
2 RewriteBase /projekt1
3 RewriteRule (.*)\.html index.php?cmd=$1 [L]
4 ErrorDocument 404 "brak strony!"
```

Musimy zmienić wartości 'RewriteBase' na nazwa folderu na serwerze w którym znajduje się nasza aplikacja

- Gdy wszystko jest gotowe musimy uruchomić WebSocket  
Będąc w folderze bazowym projektu w terminalu wpisujemy komendę

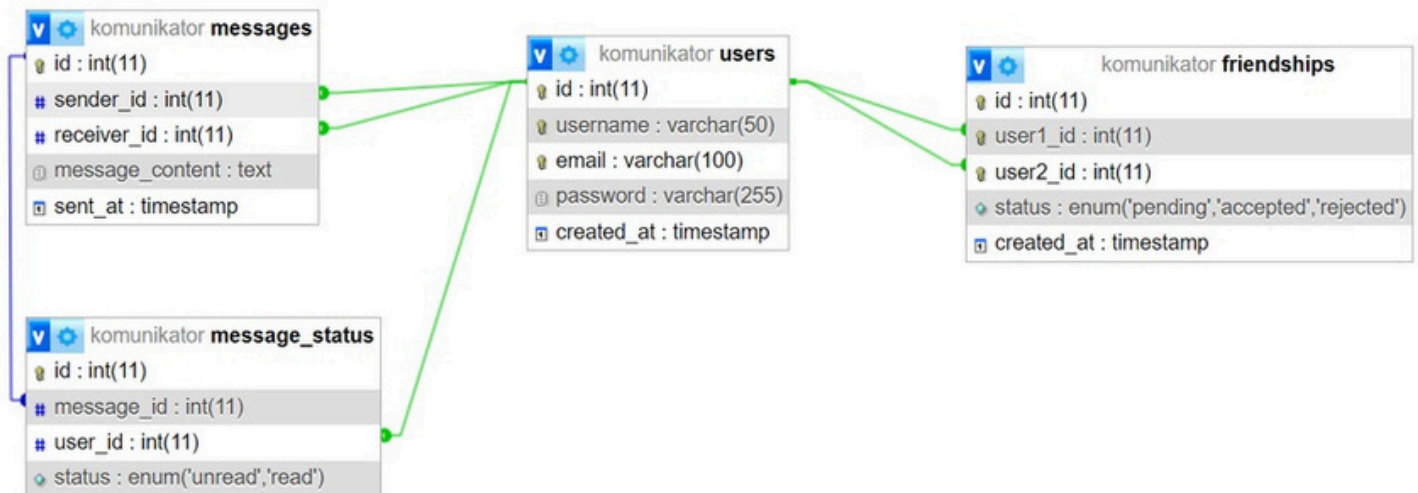
***node .\classes\WebSocket\ws.js***



## 4. Baza Danych

System zarządzania bazą danych: MariaDB 10.4.32

System kodowania: „utf8mb4\_general\_ci”







## 5. Modele BazoDanowe

### Status.php

createStatus - "Tworzy status wiadomości"  
updateStatus - "Aktualizuje status wiadomości"  
getStatus - "Pobiera status wiadomości dla użytkownika"  
updateMessageStatus - "Aktualizuje status wiadomości na 'read'"

createStatus(message\_id: int, user\_id: int, status: string = 'unread') : bool  
updateStatus(message\_id: int, user\_id: int, status: string) : bool  
getStatus(message\_id: int, user\_id: int) : array  
updateMessageStatus(message\_id: int, user\_id: int) : bool

### User.php

register - "Rejestruje nowego użytkownika"  
login - "Loguje użytkownika"  
findByEmail - "Wyszukuje użytkownika po adresie email"  
findByUsername - "Wyszukuje użytkownika po nazwie użytkownika"  
findById - "Wyszukuje użytkownika po ID"

register(username: string, email: string, password: string) : bool  
login(login: string, password: string) : array|bool  
findByEmail(email: string) : array|bool  
findByUsername(username: string) : array|bool  
findById(id: int) : array|bool

### Message.php

sendMessage - "Wysyła wiadomość od użytkownika"  
getMessages - "Pobiera wiadomości między użytkownikami"  
getLastMessageId - "Pobiera ID ostatniej wiadomości między użytkownikami"

sendMessage(sender\_id: int, receiver\_id: int, content: string) : bool  
getMessages(user\_id: int, contact\_id: int) : array  
getLastMessageId(user\_id: int, contact\_id: int) : int



## Friendship.php

sendRequest - "Wysyła zaproszenie do znajomych"  
updateRequest - "Aktualizuje status zaproszenia"  
getFriends - "Pobiera listę znajomych użytkownika"  
areFriends - "Sprawdza, czy użytkownicy są znajomymi"  
searchAvailableFriends - "Wyszukuje dostępnych znajomych"

sendRequest(user1\_id: int, user2\_id: int) : bool  
updateRequest(user1\_id: int, user2\_id: int, status: string) : bool  
getFriends(user\_id: int, status: string = 'accepted') : array  
areFriends(user1\_id: int, user2\_id: int) : array|bool  
searchAvailableFriends(user\_id: int, searchTerm: string) : array



## 6. Diagram wysyłania wiadomości

