

# **SYSTEM STEROWANIA ALARMEM OPARTY NA RASPBERRY PI**

SEBASTIAN SZYMAŃSKI

CEZARY PIERNIKOWSKI

KAMIL ŁANIA

TOMASZ SIUDMIAK

# CEL PROJEKTU

- OPRACOWANIE INTELIGENTNEGO SYSTEMU STEROWANIA ALARMEM
- INTEGRACJA URZĄDZEŃ ALARMOWYCH IOT ZA POMOCĄ ZIGBEE2MQTT
- MOŻLIWOŚĆ ZARZĄDZANIA SYSTEMEM POPRZECZ APLIKACJĘ WEBOWĄ
- POŁĄCZNIENIE RASPBERRY PI Z BRAMKĄ ZIGBEE2MQTT

# ARCHITEKTURA SYSTEMU

- RASPBERRY PI JAKO JEDNOSTKA CENTRALNA
- BRAMKA ZIGBEE2MQTT DO KOMUNIKACJI Z URZĄDZENIAMI
- APLIKACJA WEBOWA DO ZARZĄDZANIA I MONITOROWANIA
- URZĄDZENIA ZIGBEE (CZUJNIKI RUCHU, ALARMY, KAMERY)

# ROLA RASPBERRY PI - KONTROLER

- ODPOWIADA ZA PRZETWARZANIE SYGNAŁÓW Z CZUJNIKÓW
- URUCHAMIA OPROGRAMOWANIE STERUJĄCE ALARMEM
- KOMUNIKUJE SIĘ Z BRAMKĄ ZIGBEE2MQTT

# ZIGBEE2MQTT

## MECHANIZM DZIAŁANIA

- POZWALA NA POŁĄCZENIE RASPBERRY PI Z CZUJNIKAMI I ALARMEM.
- OBSŁUGUJE SZEROKI ZAKRES URZĄDZEŃ ZIGBEE.
- PRZETWARZA DANE I PRZEKAZUJE JE DO APLIKACJI WEBOWEJ.

# FUNKCJONALNOŚCI APLIKACJI WEBOWEJ

- DODAWANIE URZĄDZEŃ – KONFIGURACJA NOWYCH CZUJNIKÓW.
- MONITOROWANIE ZUŻYCIA ENERGII – ANALIZA PRACY SYSTEMU.
- LOGI SYSTEMOWE – ŚLEDZENIE HISTORII ZDARZEŃ I ALERTÓW.
- ZDALNE STEROWANIE ALARMEM – AKTYWACJA/DEZAKTYWACJA.

# TECHNOLOGIE W PROJEKCIE

- BACKEND: PYTHON (FLASK/DJANGO)
- FRONTEND: REACT/HTML+CSS+JS
- KOMUNIKACJA: MQTT, ZIGBEE2MQTT
- BAZA DANYCH: POSTGRESQL / SQLITE

# PRZYKŁADOWY PRZEPŁYW DANYCH

- CZUJNIK WYKRYWA RUCH I PRZESYŁA SYGNAŁ ZIGBEE.
- ZIGBEE2MQTT PRZEKAZUJE DANE DO RASPBERRY PI.
- RASPBERRY PI PRZETWARZA DANE I WYSYŁA KOMUNIKAT DO APLIKACJI.
- UŻYTKOWNIK OTRZYMUJE POWIADOMIENIE I MOŻE PODJĄĆ AKCJĘ.



# NOWOŚCI

- BOT TELEGRAM UMOŻLIWIAJĄCY ZDALNE STEROWANIE ALARMEM ZA POMOCĄ WIADOMOŚCI
- SYSTEM ZARZĄDZANIA UŻYTKOWNIKAMI
- MODUŁ ROZPOZNAWANIA TWARZY, KTÓRY POZWOLI NA AUTORYZACJĘ UŻYTKOWNIKÓW
- PRZENIESIENIE INFRASTRUKTURY NA KONTENERY DOCKEROWE

# BOT TELEGRAM

- MODUŁ TELEGRAM BOT STANOWI ZDALNY INTERFEJS UŻYTKOWNIKA DLA SYSTEMU ALARMOWEGO SMART HOME
- TELEGRAM BOT DZIAŁA JAKO ODDZIELNY MIKROSERWIS, URUCHAMIANY W KONTENERZE DOCKER. KOMUNIKUJE SIĘ ON Z BACKENDEM SYSTEMU ALARMOWEGO POPRZECZ HTTP API.
- UŻYTKOWNIK WYSYŁA KOMENDĘ → BOT WERYFIKUJE TOKEN POPRZECZ ENDPOINT BACKENDU → PO POPRAWNEJ WALIDACJI UŻYTKOWNIK ZOSTAJE DODANY DO LISTY SUBSKRYBENTÓW



# FACE ID

- MODUŁ FACE ID UMOŻLIWIA LOGOWANIE UŻYTKOWNIKÓW NA PODSTAWIE ROZPOZNAWANIA TWARZY Z KAMERY, STANOWIĄC ALTERNATYWĘ DLA TRADYCYJNEGO LOGINU I HASŁA.
- SYSTEM WYKORZYSTUJE LOKALNĄ BAZĘ ZDJĘĆ UŻYTKOWNIKÓW ORAZ ALGORYTMY PRZETWARZANIA OBRAZU OPENCV DO IDENTYFIKACJI TWARZY
- PO POPRAWNYM ROZPOZNANIU UŻYTKOWNIK JEST WERYFIKOWANY W BAZIE DANYCH I LOGOWANY DO SYSTEMU Z ZACHOWANIEM PRZYPISANYCH UPRAWNIE

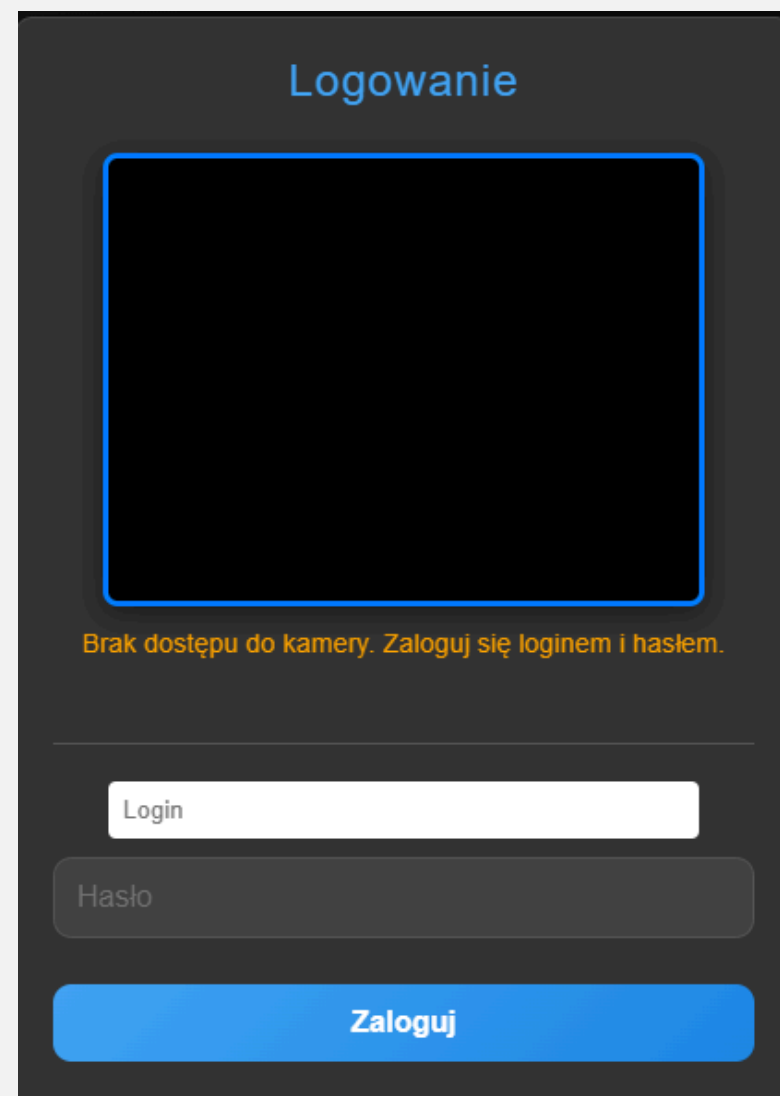
# ZARZĄDZANIE UŻYTKOWNIKAMI

- SYSTEM POSIADA CENTRALNY MECHANIZM REJESTROWANIA UŻYTKOWNIKÓW OPARTY NA BAZIE DANYCH SQLITE.
- JEDYNNIE KONTO ADMINISTRATORA MA UPRAWNIENIA DO TWORZENIA NOWYCH UŻYTKOWNIKÓW ORAZ ZARZĄDZANIA ICH DANymi, TAKIMI JAK LOGIN, HASŁO I ROLA.
- DLA KAŻDEGO UŻYTKOWNIKA PRZYPADA INNE FACE ID DO LOGOWANIA

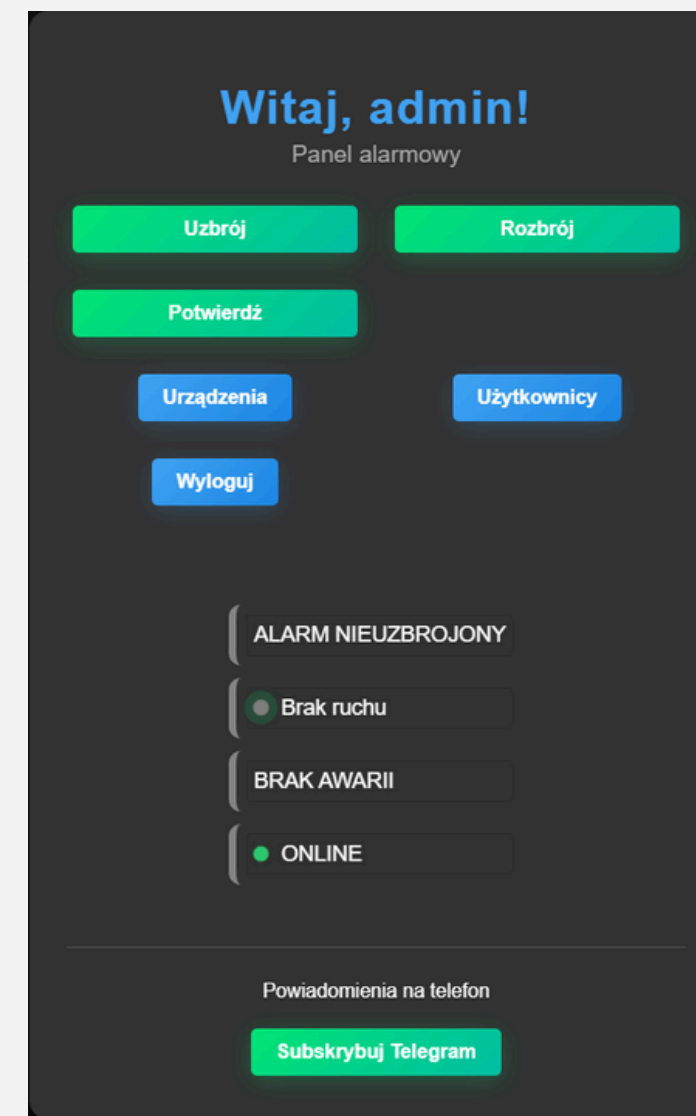
# INFRASTRUKTURA DOCKER

- PROJEKT ZOSTAŁ OPARTY NA ARCHITEKTURZE KONTENEROWEJ Z WYKORZYSTANIEM DOCKERA I DOCKER COMPOSE.
- KAŻDY MODUŁ SYSTEMU, TAKI JAK BACKEND ALARMU, PANEL WEBOWY, BAZA DANYCH CZY BOT TELEGRAMA, DZIAŁA JAKO OSOBNY KONTENER.
- UŁATWIA ZARZĄDZANIE USŁUGAMI, ICH URUCHAMIANIE ORAZ IZOLACJĘ POSZCZEGÓLNYCH KOMPONENTÓW. KONTENERYZACJA ZAPEWNIĄ PRZENOŚNOŚĆ, ŁATWĄ KONFIGURACJĘ I STABILNOŚĆ CAŁEGO SYSTEMU.

# ZRZUTY EKRANU Z APLIKACJI



Wygląd ekranu logowania

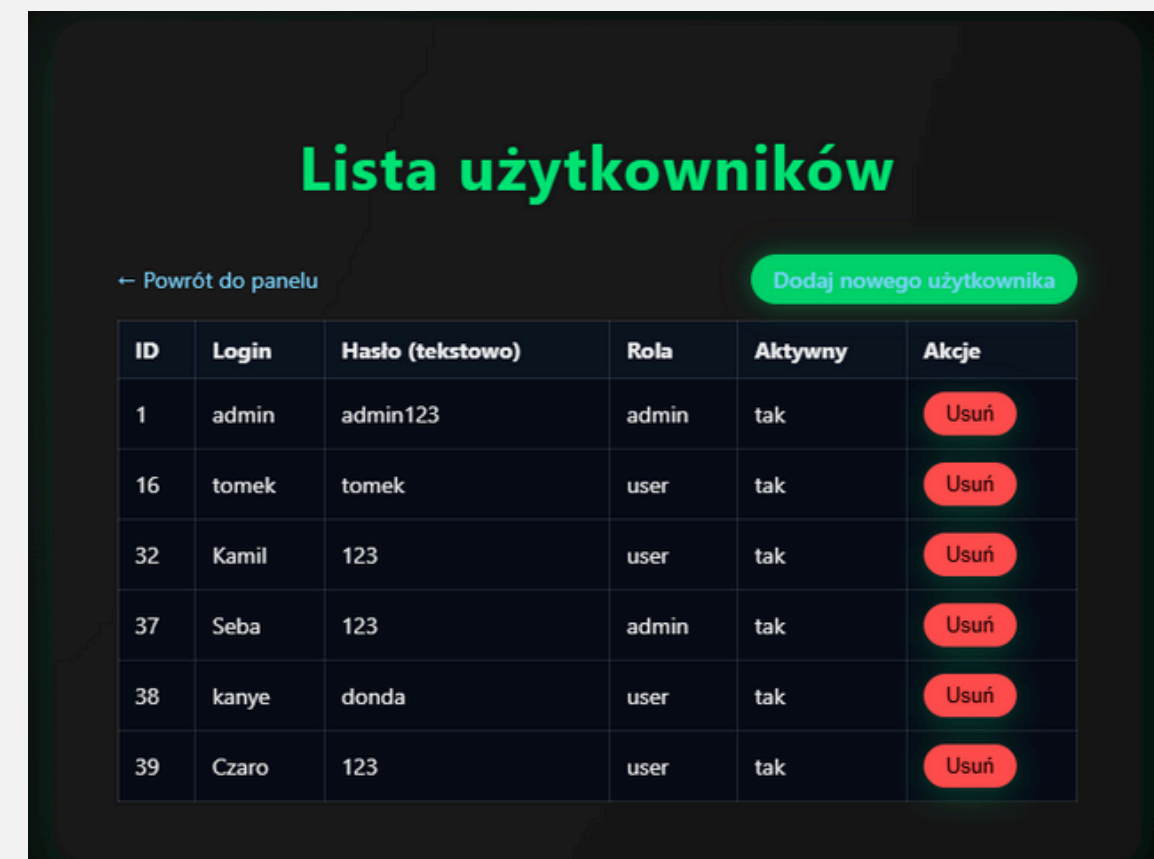


Wygląd głównego panelu sterowania

# ZRZUTY EKRANU Z APLIKACJI



Wygląd zakładki urządzenia



Lista użytkowników, dostępna tylko z konta admina



**DZIĘKUJEMY ZA UWAGĘ!**

