



Dokumentacja projektu: Wodomierze Online

Wojciech adamski dla Smartbits Sp. z o.o.

09 grudnia 2023

Spis treści

1 Wprowadzenie	7
1.1 Wprowadzenie	7
1.2 Założenia projektowe	7
1.3 Użyte biblioteki i technologie	7
1.4 Struktura plików	8
1.5 Model UML aplikacji	9
2 Pliki	12
2.1 config.py	12
2.1.1 Opis zawartości	12
2.1.2 Uwagi	13
2.2 error_handlers.py	13
2.2.1 Opis zawartości	13
2.2.2 Uwagi	14
2.3 forms.py	14
2.3.1 Klasy Formularzy	14
2.3.2 Uwagi	15
2.4 models.py	16
2.4.1 Model User	16
2.4.2 Model Meter	17
2.4.3 Model MeterReading	17
2.4.4 Model Message	17
2.4.5 Model Event	18
2.4.6 Uwagi	18
2.5 routes.py	18
2.5.1 Spis funkcji	18
2.5.2 welcome()	20
2.5.3 home()	20
2.5.4 login()	21

2.5.5	logout()	21
2.5.6	upload_csv()	21
2.5.7	user_meters()	22
2.5.8	meter_details()	22
2.5.9	delete_meter()	22
2.5.10	clear_readings()	23
2.5.11	update_meter_name()	23
2.5.12	update_meter_address()	23
2.5.13	admin_panel()	24
2.5.14	add_user()	24
2.5.15	user_overview()	24
2.5.16	remove_meter()	25
2.5.17	add_meter()	26
2.5.18	edit_account()	27
2.5.19	assign_meter()	27
2.5.20	delete_meters()	28
2.5.21	delete_user()	29
2.5.22	deactivate_user()	29
2.5.23	update_user_notes()	30
2.5.24	messages()	31
2.5.25	message()	31
2.5.26	delete_message()	32
2.5.27	assign_meters_to_user()	32
2.5.28	summary()	33
2.5.29	assign_meter_to_superuser()	34
2.5.30	superuser_assign_meter()	34
2.5.31	superuser_user_overview()	35
2.5.32	superuser_panel()	35
2.5.33	assign_user_to_superuser()	36
2.5.34	remove_assigned_user()	37
2.5.35	generate_report()	37
2.5.36	display_report()	38
2.5.37	add_multiple_users()	38
2.6	Plik utils.py	39
2.6.1	admin_required	39
2.6.2	superuser_required	40
2.6.3	allowed_file	40
2.6.4	is_valid_link	40
2.6.5	process_csv_water	40

2.6.6	process_csv_heat	41
2.6.7	process_csv_events	41
2.6.8	create_report_data	41
3	Szablony	43
3.1	base.html	44
3.1.1	Nagłówek	44
3.1.2	Nawigacja	44
3.1.3	Główny Kontener Treści	44
3.1.4	Stopka	44
3.1.5	Skrypty JavaScript	44
3.1.6	Stylizacja	45
3.2	admin_panel.html	45
3.2.1	Sekcja Dodawania Użytkowników	45
3.2.2	Lista Użytkowników i Liczników	45
3.2.3	Formularze	45
3.2.4	Skrypty JavaScript	46
3.2.5	Stylizacja	46
3.2.6	Zastosowanie	46
3.3	display_report.html	46
3.3.1	Nagłówek Raportu	46
3.3.2	Przyciski Akcji	47
3.3.3	Tabela Raportu	47
3.3.4	Skrypty JavaScript	47
3.3.5	Stylizacja	47
3.3.6	Zastosowanie	47
3.4	edit_account.html	48
3.4.1	Przycisk Powrotu	48
3.4.2	Formularz Edycji Profilu	48
3.4.3	Walidacja Formularza	48
3.4.4	Stylizacja	48
3.4.5	Zastosowanie	49
3.5	generate_report.html	49
3.5.1	Przycisk Powrotu	49
3.5.2	Formularz Generowania Raportu	49
3.5.3	Interaktywność	49
3.5.4	Stylizacja	50
3.5.5	Zastosowanie	50
3.6	home.html	50

3.6.1	Powitanie Użytkownika	50
3.6.2	Lista Przypisanych Liczników	50
3.6.3	Interaktywność	51
3.6.4	Stylizacja	51
3.6.5	Zastosowanie	51
3.7	login.html	51
3.7.1	Formularz Logowania	51
3.7.2	Link do Rejestracji	52
3.7.3	Stylizacja	52
3.7.4	Zastosowanie	52
3.8	message.html	52
3.8.1	Szczegóły Wiadomości	53
3.8.2	Przycisk Usunięcia Wiadomości	53
3.8.3	Przycisk Powrotu	53
3.8.4	Stylizacja i Układ	53
3.8.5	Zastosowanie	53
3.9	messages.html	54
3.9.1	Formularz Wysyłania Wiadomości	54
3.9.2	Lista Wiadomości	54
3.9.3	Filtrowanie Wiadomości	54
3.9.4	Skrypty JavaScript	54
3.9.5	Stylizacja i Układ	55
3.9.6	Zastosowanie	55
3.10	meter_details.html	55
3.10.1	Informacje o Liczniku	55
3.10.2	Formularz Edycji Adresu	55
3.10.3	Wykresy i Tabele Odczytów	55
3.10.4	Skrypty JavaScript	56
3.10.5	Stylizacja i Układ	56
3.10.6	Zastosowanie	56
3.11	send_message.html	56
3.11.1	Formularz Wysyłania Wiadomości	56
3.11.2	Stylizacja i Układ	57
3.11.3	Zastosowanie	57
3.12	superuser_panel.html	57
3.12.1	Panel Administratora Zasobów	57
3.12.2	Wyszukiwanie Użytkowników	58
3.12.3	Interakcje i Skrypty JavaScript	58
3.12.4	Zastosowanie	58

3.13	superuser_user_overview.html	58
3.13.1	Przegląd Użytkownika	58
3.13.2	Przypisane Liczniki	59
3.13.3	Interakcje i Skrypty JavaScript	59
3.13.4	Zastosowanie	59
3.14	upload_csv.html	59
3.14.1	Formularz Przesyłania Pliku	59
3.14.2	Zastosowanie	60
3.14.3	Stylizacja i Układ	60
3.15	user_meters.html	60
3.15.1	Lista Liczników	60
3.15.2	Linki do Szczegółów Licznika	61
3.15.3	Zastosowanie	61
3.15.4	Stylizacja i Układ	61
3.16	user_overview.html	61
3.16.1	Informacje o Użytkowniku	61
3.16.2	Notatki o Użytkowniku	62
3.16.3	Przypisane Liczniki	62
3.16.4	Wolne Liczniki	62
3.16.5	Przypisani Użytkownicy (dla Superużytkowników)	62
3.16.6	Usuwanie i Dezaktywacja Użytkownika	62
3.16.7	Skrypty JavaScript	62
3.16.8	Stylizacja i Układ	63
4	Style CSS	64
4.1	GeneralStyles.css	64
4.1.1	Opis	64
4.1.2	Zawartość	64
4.2	Layout.css	64
4.2.1	Opis	64
4.2.2	Zawartość	65
4.3	MediaQueries.css	65
4.3.1	Opis	65
4.3.2	Zawartość	65
4.4	ButtonsAndForms.css	65
4.4.1	Opis	65
4.4.2	Zawartość	65
4.5	UserList.css	66
4.5.1	Opis	66

4.5.2 Zawartość	66
5 Linki i kontakt	67
5.1 serwer SQL	67
5.2 Serwer aplikacji	68
5.3 Github	68
5.4 Kontakt	68

Rozdział 1

Wprowadzenie

1.1 Wprowadzenie

Projekt "Wodomierze Online" to aplikacja webowa służąca do zarządzania i monitorowania odczytów wodomierzy. Aplikacja umożliwia użytkownikom przeglądanie, dodawanie i eksportowanie danych odczytów, a także zarządzanie użytkownikami i licznikami.

1.2 Założenia projektowe

- Zarządzanie odczytami wodomierzy.
- Możliwość dodawania i edytowania danych użytkowników i liczników.
- Eksport danych do różnych formatów (CSV, PDF).
- Interfejs użytkownika dostosowany do różnych typów urządzeń.

1.3 Użyte biblioteki i technologie

- Flask - mikroframework do tworzenia aplikacji webowych w Pythonie.
- SQLAlchemy - narzędzie ORM do obsługi baz danych.
- Jinja2 - silnik szablonów dla Pythona.

- Bootstrap - framework CSS do tworzenia responsywnych interfejsów użytkownika.
- Plotly - biblioteka do tworzenia interaktywnych wykresów.

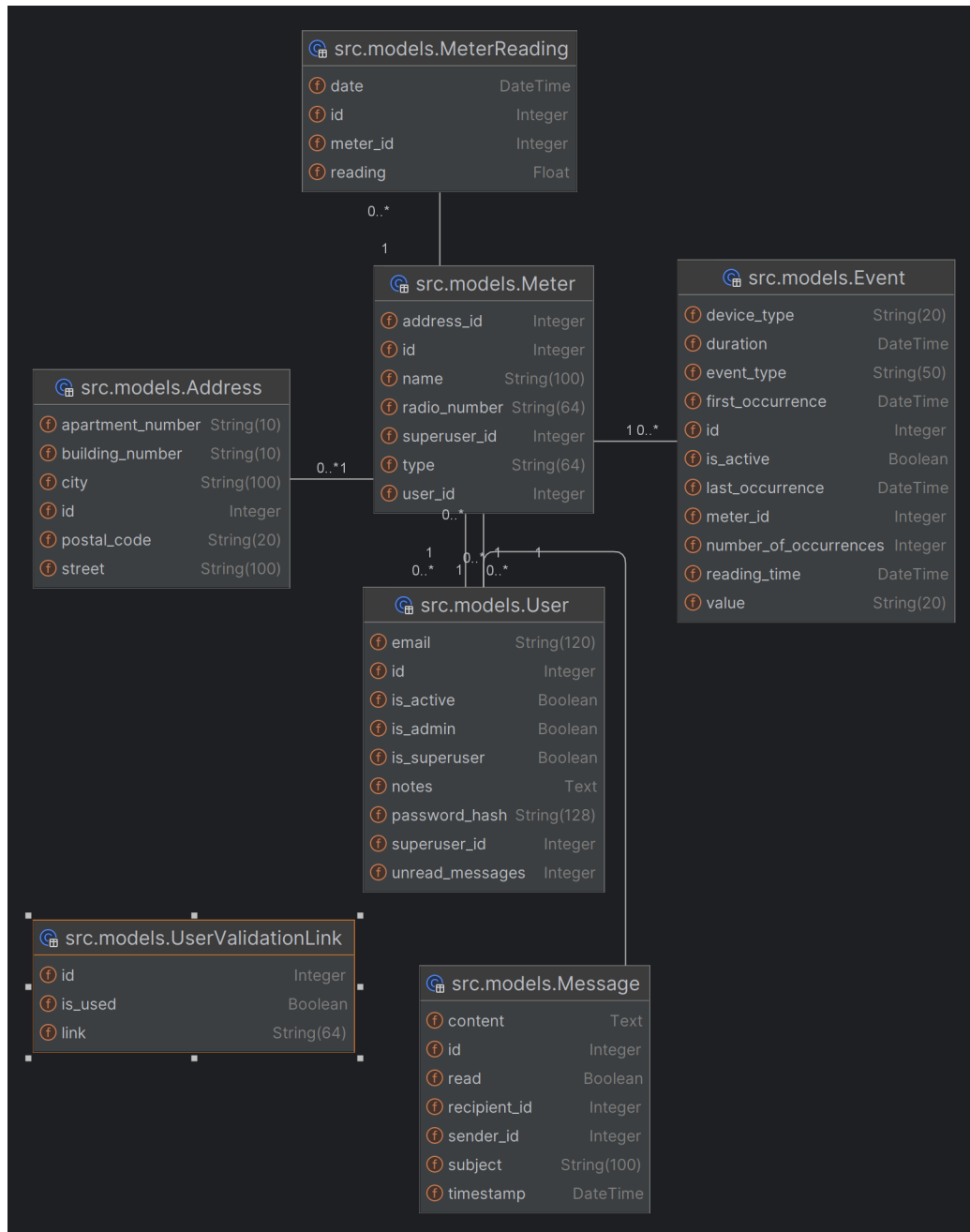
1.4 Struktura plików

Projekt składa się z następujących plików i folderów:

- `app.py` - Główny plik aplikacji, zawierający konfigurację i uruchomienie serwera.
- `README.md` - Plik z opisem projektu.
- `requirements.txt` - Lista zależności projektu.
- **migrations/** - Folder zawierający migracje bazy danych.
 - `alembic.ini` - Konfiguracja Alembic.
 - `env.py` - Środowisko migracji.
 - `README` - Opis migracji.
 - `script.py.mako` - Szablon skryptu migracji.
 - **versions/** - Folder z wersjami migracji.
- **src/** - Folder z kodem źródłowym aplikacji.
 - `config.py` - Konfiguracja aplikacji.
 - `error_handlers.py` - Obsługa błędów.
 - `forms.py` - Formularze Flask-WTF.
 - `models.py` - Modele bazy danych.
 - `routes.py` - Definicje tras aplikacji.
 - `tests.py` - Testy jednostkowe.
 - `utils.py` - Funkcje pomocnicze.
- **static/** - Folder z plikami statycznymi.
 - `style.css` - Arkusz stylów CSS.
 - **img/** - Obrazy używane w aplikacji.

- **js/** - Skrypty JavaScript.
- Szablony - Folder z szablonami Jinja2.
 - add_meter.html, add_reading.html, add_user.html, admin_panel.html, assign_meter.html, base.html, display_report.html, edit_account.html, generate_report.html, home.html, login.html, message.html, messages.html, meter_details.html, meter_details_beta.html, register.html, send_message.html, summary.html, superuser_panel.html, superuser_user_overview.html, upload_csv.html, user_meters.html, user_overview.html, user_summary.html.

1.5 Model UML aplikacji



Rysunek 1.1: Modele aplikacji

Rozdział 2

Pliki

2.1 config.py

Plik `config.py` zawiera konfigurację aplikacji Flask. Jest to kluczowy plik, który definiuje różne ustawienia i parametry niezbędne do prawidłowego działania aplikacji.

2.1.1 Opis zawartości

- `ALLOWED_EXTENSIONS` - Zbiór rozszerzeń plików, które są dozwolone do przesyłania na serwer. W tym przypadku dozwolone są pliki `csv` i `json`.
- `UPLOAD_FOLDER` - Nazwa folderu, do którego będą przesyłane pliki. W tym przypadku jest to folder `'uploads'`.
- `EMAIL_KEY` - Klucz wykorzystywany do weryfikacji adresów email. Jest to klucz w formacie bajtowym.
- **Klasa `Config`** - Główna klasa konfiguracyjna zawierająca ustawienia aplikacji.
 - `SECRET_KEY` - Tajny klucz aplikacji, wykorzystywany do zabezpieczania danych aplikacji, takich jak sesje. Pobierany jest ze zmiennej środowiskowej.
 - `DB_USER`, `DB_PASS`, `DB_SERVER`, `DB_NAME` - Parametry konfiguracyjne bazy danych, w tym nazwa użytkownika, hasło, serwer i nazwa bazy danych. Są one pobierane ze zmiennych

środowiskowych, z opcjonalnymi wartościami domyślnymi dla `DB_SERVER` i `DB_NAME`.

- `SQLALCHEMY_DATABASE_URI` - URI bazy danych wykorzystywane przez SQLAlchemy do połączenia z bazą danych. Jest to skonstruowany ciąg połączenia z bazą danych SQL Server, wykorzystujący wcześniej zdefiniowane parametry.

2.1.2 Uwagi

- Plik ten jest kluczowy dla bezpieczeństwa i konfiguracji aplikacji. Należy zachować ostrożność, aby nie ujawnić wrażliwych danych, takich jak `SECRET_KEY` czy dane dostępu do bazy danych.
- Zmienne środowiskowe są preferowanym sposobem przechowywania wrażliwych danych, szczególnie w środowisku produkcyjnym.
- Plik ten powinien być dostosowany do specyficznych wymagań i ustawień środowiska, w którym działa aplikacja.

2.2 `error_handlers.py`

Plik `error_handlers.py` zawiera funkcje obsługi błędów dla aplikacji Flask. Te funkcje są wykorzystywane do obsługi typowych błędów HTTP, takich jak 404 (nie znaleziono), 500 (błąd wewnętrzny serwera), 403 (dostęp zabroniony) i 401 (brak autoryzacji).

2.2.1 Opis zawartości

- **`handle_internal_server_error(e)`** - Funkcja obsługi błędu 500. Wyświetla komunikat o błędzie i przekierowuje użytkownika na stronę główną.
- **`handle_not_found_error(e)`** - Funkcja obsługi błędu 404. Informuje użytkownika, że strona nie została znaleziona i przekierowuje na stronę główną.
- **`handle_forbidden_error(e)`** - Funkcja obsługi błędu 403. Informuje użytkownika o braku dostępu do żądanej strony i przekierowuje na stronę główną.

- **handle_unauthorized_error(e)** - Funkcja obsługi błędu 401. Informuje użytkownika o braku autoryzacji do dostępu do żądanej strony i przekierowuje na stronę główną.

2.2.2 Uwagi

- Funkcje te są wykorzystywane do obsługi typowych błędów HTTP w sposób przyjazny dla użytkownika, zapewniając jednocześnie, że użytkownik jest przekierowywany na bezpieczną stronę (stronę główną).
- Każda funkcja obsługi błędów przyjmuje jako argument obiekt błędu (e), który może być wykorzystany do uzyskania dodatkowych informacji o błędzie.
- Użycie funkcji flash pozwala na wyświetlenie komunikatu o błędzie na stronie głównej, co jest pomocne w informowaniu użytkownika o napotkanym problemie.
- Warto rozważyć rozszerzenie obsługi błędów o dodatkowe informacje diagnostyczne lub logowanie, co może być pomocne w rozwiązywaniu problemów.

2.3 forms.py

Plik `forms.py` zawiera definicje klas formularzy używanych w aplikacji Flask. Te formularze są wykorzystywane do interakcji z użytkownikiem, takich jak logowanie, rejestracja, dodawanie i edycja danych.

2.3.1 Klasy Formularzy

- **LoginForm** - Formularz logowania użytkownika. Zawiera pola dla adresu email, hasła oraz opcji zapamiętania użytkownika.
- **MeterForm** - Formularz dodawania nowego licznika. Zawiera pola dla numeru radiowego, typu licznika oraz identyfikatora użytkownika.

- **UploadForm** - Formularz do przesyłania plików CSV. Obsługuje różne typy plików (wodomierze, ciepłomierze, zdarzenia dla wodomierzy i ciepłomierzy).
- **UserForm** - Formularz do dodawania nowego użytkownika. Zawiera pola dla adresu email, hasła, potwierdzenia hasła oraz opcji administratora i superużytkownika.
- **EditAccountForm** - Formularz do zmiany hasła użytkownika. Zawiera pola dla obecnego hasła, nowego hasła oraz potwierdzenia nowego hasła.
- **AddUserForm** - Formularz do dodawania nowego użytkownika z potwierdzeniem adresu email.
- **SetPasswordForm** - Formularz do ustawienia nowego hasła z potwierdzeniem.
- **UserNotesForm** - Formularz do dodawania notatek do profilu użytkownika.
- **UserOverviewForm** - Formularz do wyświetlania notatek w przeglądzie użytkownika.
- **MessageForm** - Formularz do wysyłania wiadomości. Zawiera pola dla tematu, treści, odbiorcy oraz opcji wysłania do wszystkich użytkowników.
- **AssignMeterToSuperuserForm** - Formularz do przypisywania licznika do superużytkownika.
- **AssignMeterToUserForm** - Formularz do przypisywania licznika do użytkownika.

2.3.2 Uwagi

- Formularze wykorzystują bibliotekę Flask-WTF oraz WTForms do tworzenia i walidacji pól formularza.
- Walidatory, takie jak DataRequired, Email, EqualTo, zapewniają podstawową walidację danych wejściowych.

- W niektórych formularzach, takich jak `AssignMeterToSuperuserForm` i `AssignMeterToUserForm`, dynamicznie generowane są opcje wyboru na podstawie danych z bazy danych.
- Formularze są integralną częścią interfejsu użytkownika, umożliwiając łatwą i bezpieczną interakcję z aplikacją.

2.4 models.py

Plik `models.py` zawiera definicje modeli danych używanych w aplikacji Flask. Modele te są mapowane na tabele w bazie danych i służą do reprezentacji różnych encji, takich jak użytkownicy, liczniki, odczyty liczników itp.

2.4.1 Model User

Model `User` reprezentuje użytkownika aplikacji. Zawiera następujące pola:

- **id** - Unikalny identyfikator użytkownika (klucz główny).
- **email** - Adres email użytkownika. Służy jako unikalny identyfikator użytkownika w systemie.
- **password_hash** - Zaszyfrowane hasło użytkownika.
- **is_admin** - Flaga określająca, czy użytkownik posiada uprawnienia administratora.
- **is_superuser** - Flaga określająca, czy użytkownik jest superużytkownikiem.
- **meters** - Relacja z modelem `Meter`, reprezentująca liczniki przypisane do użytkownika.
- **messages_sent** - Relacja z modelem `Message`, reprezentująca wiadomości wysłane przez użytkownika.
- **messages_received** - Relacja z modelem `Message`, reprezentująca wiadomości otrzymane przez użytkownika.

2.4.2 Model Meter

Model Meter reprezentuje licznik. Zawiera następujące pola:

- **id** - Unikalny identyfikator licznika (klucz główny).
- **radio_number** - Numer radiowy licznika.
- **type** - Typ licznika (np. wodomierz, ciepłomierz).
- **user_id** - Identyfikator użytkownika, do którego przypisany jest licznik.
- **readings** - Relacja z modelem MeterReading, reprezentująca odczyty danego licznika.

2.4.3 Model MeterReading

Model MeterReading reprezentuje odczyt licznika. Zawiera następujące pola:

- **id** - Unikalny identyfikator odczytu (klucz główny).
- **date** - Data odczytu.
- **reading** - Wartość odczytu.
- **meter_id** - Identyfikator licznika, do którego należy dany odczyt.

2.4.4 Model Message

Model Message reprezentuje wiadomość. Zawiera następujące pola:

- **id** - Unikalny identyfikator wiadomości (klucz główny).
- **subject** - Temat wiadomości.
- **content** - Treść wiadomości.
- **sender_id** - Identyfikator nadawcy wiadomości.
- **recipient_id** - Identyfikator odbiorcy wiadomości.
- **read** - Flaga określająca, czy wiadomość została przeczytana.

2.4.5 Model Event

Model Event reprezentuje zdarzenie. Zawiera następujące pola:

- **id** - Unikalny identyfikator zdarzenia (klucz główny).
- **type** - Typ zdarzenia.
- **date** - Data zdarzenia.
- **meter_id** - Identyfikator licznika, z którym związane jest zdarzenie.
- **details** - Szczegółowe informacje o zdarzeniu.

2.4.6 Uwagi

- Modele wykorzystują bibliotekę SQLAlchemy do mapowania obiektowo-relacyjnego (ORM).
- W modelu User zaimplementowano metody do ustawiania i weryfikacji hasła, wykorzystując funkcje haszujące.
- Modele zawierają również metody pomocnicze, takie jak generowanie tokenów dla resetowania hasła.
- Modele są kluczowym elementem aplikacji, umożliwiającym efektywne zarządzanie danymi i interakcję z bazą danych.

2.5 routes.py

Plik `routes.py` zawiera definicje tras (endpointów) aplikacji Flask. Każda trasa jest powiązana z określoną funkcją, która obsługuje żądania HTTP do danego endpointu. Poniżej znajduje się szczegółowy opis każdej funkcji wraz z jej przeznaczeniem, parametrami, zwracanymi wartościami oraz odwołaniami do innych elementów aplikacji.

2.5.1 Spis funkcji

- `welcome()`
- `home()`

- login()
- logout()
- upload_csv()
- user_meters()
- meter_details()
- delete_meter()
- clear_readings()
- update_meter_name()
- update_meter_address()
- admin_panel()
- add_user()
- user_overview()
- remove_meter()
- add_meter()
- edit_account()
- assign_meter()
- delete_meters()
- delete_user()
- deactivate_user()
- update_user_notes()
- messages()
- message()
- delete_message()
- assign_meters_to_user()

- `summary()`
- `assign_meter_to_superuser()`
- `superuser_assign_meter()`
- `superuser_user_overview()`
- `superuser_panel()`
- `assign_user_to_superuser()`
- `remove_assigned_user()`
- `generate_report()`
- `display_report()`
- `add_multiple_users()`

2.5.2 `welcome()`

Funkcja `welcome()` przekierowuje użytkownika do odpowiedniej strony w zależności od tego, czy jest zalogowany. Jeśli użytkownik jest zalogowany, przekierowuje do strony głównej (`home`). W przeciwnym razie przekierowuje do strony logowania (`login`).

Parametry: Brak.

Zwracane wartości: Przekierowanie do odpowiedniej strony.

Używane zmienne: `current_user` - obecnie zalogowany użytkownik.

Odwołania: `url_for` - funkcja Flask do generowania URL.

2.5.3 `home()`

Funkcja `home()` służy do wyświetlania strony głównej aplikacji. W zależności od roli zalogowanego użytkownika, przekierowuje do odpowiedniego panelu (administratora lub superużytkownika) lub wyświetla stronę główną dla zwykłego użytkownika.

Parametry: Brak.

Zwracane wartości: Renderowany szablon `home.html` dla zwykłego użytkownika lub przekierowanie do panelu administratora/superużytkownika.

Używane zmienne: `current_user` - obecnie zalogowany użytkownik.

Odwołania: `url_for`, `redirect` - funkcje Flask do generowania URL i przekierowań.

2.5.4 `login()`

Funkcja `login()` obsługuje proces logowania użytkownika. Jeśli użytkownik jest już zalogowany, przekierowuje go do strony głównej. W przeciwnym razie, obsługuje formularz logowania i weryfikuje dane użytkownika.

Parametry: Brak.

Zwracane wartości: Renderowany szablon `login.html` lub przekierowanie do strony głównej po pomyślnym zalogowaniu.

Używane zmienne: `LoginForm` - formularz logowania, `current_user` - obecnie zalogowany użytkownik.

Odwołania: `url_for`, `redirect`, `flash` - funkcje Flask do generowania URL, przekierowań i wyświetlania komunikatów.

2.5.5 `logout()`

Funkcja `logout()` służy do wylogowywania użytkownika. Po wylogowaniu przekierowuje użytkownika do strony głównej.

Parametry: Brak.

Zwracane wartości: Przekierowanie do strony głównej (home).

Używane zmienne: Brak.

Odwołania: `logout_user`, `url_for`, `redirect` - funkcje Flask do wylogowywania użytkownika i przekierowań.

2.5.6 `upload_csv()`

Funkcja `upload_csv()` obsługuje przesyłanie plików CSV przez administratora. Umożliwia przesłanie danych dotyczących wodomierzy, ciepłomierzy oraz zdarzeń związanych z tymi urządzeniami.

Parametry: Brak.

Zwracane wartości: Renderowany szablon `upload_csv.html` lub przekierowanie do strony głównej po pomyślnym przesłaniu pliku.

Używane zmienne: `UploadForm` - formularz przesyłania plików, `UPLOAD_FOLDER` - ścieżka do folderu z plikami.

Odwołania: `url_for`, `redirect`, `flash`, `process_csv_water`, `process_csv_heat`, `process_csv_events` - funkcje Flask do generowania URL, przekierowań, wyświetlania komunikatów i przetwarzania plików CSV.

2.5.7 `user_meters()`

Funkcja `user_meters()` wyświetla listę liczników przypisanych do zalogowanego użytkownika.

Parametry: Brak.

Zwracane wartości: Renderowany szablon `user_meters.html` z listą liczników.

Używane zmienne: `current_user` - obecnie zalogowany użytkownik.

Odwołania: `Meter` - model licznika z bazy danych.

2.5.8 `meter_details()`

Funkcja `meter_details()` wyświetla szczegółowe informacje o wybranym liczniku, w tym odczyty i zdarzenia związane z tym licznikiem. Dostęp do tej funkcji mają tylko właściciel licznika, administrator lub superużytkownik z odpowiednimi uprawnieniami.

Parametry: `meter_id` - identyfikator licznika.

Zwracane wartości: Renderowany szablon `meter_details.html` ze szczegółami licznika.

Używane zmienne: `Meter`, `MeterReading`, `current_user` - model licznika, model odczytów i obecnie zalogowany użytkownik.

Odwołania: `url_for`, `redirect`, `flash` - funkcje Flask do generowania URL, przekierowań i wyświetlania komunikatów.

2.5.9 `delete_meter()`

Funkcja `delete_meter()` umożliwia administratorowi usunięcie wybranego licznika z bazy danych.

Parametry: `meter_id` - identyfikator licznika do usunięcia.

Zwracane wartości: Przekierowanie do panelu administratora po usunięciu licznika.

Używane zmienne: `Meter` - model licznika, `db` - instancja bazy danych.

Odwołania: `url_for`, `redirect`, `flash` - funkcje Flask do generowania URL, przekierowań i wyświetlania komunikatów.

2.5.10 `clear_readings()`

Funkcja `clear_readings()` służy do wyczyszczenia wszystkich odczytów dla wybranego licznika. Dostępna tylko dla administratora.

Parametry: `meter_id` - identyfikator licznika, którego odczyty mają zostać wyczyszczone.

Zwracane wartości: Przekierowanie do szczegółów licznika po wyczyszczeniu odczytów.

Używane zmienne: `Meter` - model licznika, `db` - instancja bazy danych.

Odwołania: `url_for`, `redirect`, `flash` - funkcje Flask do generowania URL, przekierowań i wyświetlania komunikatów.

2.5.11 `update_meter_name()`

Funkcja `update_meter_name()` pozwala zalogowanemu użytkownikowi na zmianę nazwy przypisanego do niego licznika.

Parametry: `meter_id` - identyfikator licznika, którego nazwa ma zostać zmieniona.

Zwracane wartości: Przekierowanie do szczegółów licznika po zmianie nazwy.

Używane zmienne: `Meter` - model licznika, `current_user` - obecnie zalogowany użytkownik, `db` - instancja bazy danych.

Odwołania: `url_for`, `redirect`, `flash`, `request.form` - funkcje Flask do generowania URL, przekierowań, wyświetlania komunikatów i obsługi danych formularza.

2.5.12 `update_meter_address()`

Funkcja `update_meter_address()` umożliwia zalogowanemu użytkownikowi aktualizację adresu przypisanego do jego licznika.

Parametry: `meter_id` - identyfikator licznika, którego adres ma zostać zaktualizowany.

Zwracane wartości: Przekierowanie do szczegółów licznika po aktualizacji adresu.

Używane zmienne: Meter - model licznika, Address - model adresu, db - instancja bazy danych, request.form - dane formularza.

Odwołania: url_for, redirect - funkcje Flask do generowania URL i przekierowań.

2.5.13 admin_panel()

Funkcja admin_panel() wyświetla panel administratora, umożliwiając zarządzanie użytkownikami i licznikami.

Parametry: Brak.

Zwracane wartości: Renderowanie szablonu panelu administratora.

Używane zmienne: UserForm - formularz użytkownika, User - model użytkownika, Meter - model licznika, db - instancja bazy danych.

Odwołania: render_template, url_for, redirect, flash - funkcje Flask do renderowania szablonów, generowania URL, przekierowań i wyświetlania komunikatów.

2.5.14 add_user()

Funkcja add_user() umożliwia administratorowi dodanie nowego użytkownika do systemu.

Parametry: Brak.

Zwracane wartości: Przekierowanie do panelu administratora po dodaniu użytkownika.

Używane zmienne: UserForm - formularz użytkownika, User - model użytkownika, db - instancja bazy danych.

Odwołania: render_template, url_for, redirect, flash - funkcje Flask do renderowania szablonów, generowania URL, przekierowań i wyświetlania komunikatów.

2.5.15 user_overview()

Funkcja user_overview() służy do wyświetlania szczegółów użytkownika w panelu administratora, w tym przypisywania mu liczników oraz zarządzania jego notatkami.

Parametry: user_id - identyfikator użytkownika, którego szczegóły mają zostać wyświetlone.

Zwracane wartości: Renderowanie szablonu z widokiem szczegółów użytkownika.

Używane zmienne:

- User - model użytkownika.
- Meter - model licznika.
- UserForm - formularz użytkownika.
- UserNotesForm - formularz notatek użytkownika.
- db - instancja bazy danych.
- request.form - dane formularza.

Odwołania:

- render_template - funkcja Flask do renderowania szablonów.
- flash - funkcja Flask do wyświetlania komunikatów.
- redirect - funkcja Flask do przekierowań.
- url_for - funkcja Flask do generowania URL.

Opis działania: Funkcja pobiera informacje o użytkowniku na podstawie user_id. Następnie, w zależności od tego, czy użytkownik jest superużytkownikiem czy administratorem, pobiera listę dostępnych i nieprzypisanych liczników oraz użytkowników. Umożliwia przypisanie liczników do użytkownika oraz aktualizację jego notatek. W przypadku przypisania licznika, funkcja zapisuje zmiany w bazie danych i wyświetla stosowny komunikat.

2.5.16 remove_meter()

Funkcja remove_meter() służy do odłączania licznika od użytkownika.

Parametry: meter_id - identyfikator licznika do odłączenia.

Zwracane wartości: Przekierowanie do widoku szczegółów superużytkownika lub użytkownika, w zależności od uprawnień bieżącego użytkownika.

Używane zmienne:

- Meter - model licznika.
- db - instancja bazy danych.

Odwołania:

- flash - funkcja Flask do wyświetlania komunikatów.
- redirect - funkcja Flask do przekierowań.
- url_for - funkcja Flask do generowania URL.

Opis działania: Funkcja odłącza licznik od użytkownika, usuwając powiązania z użytkownikiem i superużytkownikiem. Następnie zapisuje zmiany w bazie danych i wyświetla stosowny komunikat. W przypadku braku uprawnień do odłączenia licznika, wyświetla komunikat o błędzie.

2.5.17 add_meter()

Funkcja `add_meter()` służy do dodawania nowego licznika w panelu administratora.

Zwracane wartości: Renderowanie formularza dodawania licznika lub przekierowanie do panelu administratora po pomyślnym dodaniu licznika.

Używane zmienne:

- MeterForm - formularz dodawania licznika.
- Meter - model licznika.
- db - instancja bazy danych.

Odwołania:

- render_template - funkcja Flask do renderowania szablonów.
- flash - funkcja Flask do wyświetlania komunikatów.
- redirect - funkcja Flask do przekierowań.
- url_for - funkcja Flask do generowania URL.

Opis działania: Funkcja wyświetla formularz dodawania licznika. Po wypełnieniu formularza i jego zatwierdzeniu, dodaje nowy licznik do bazy danych i wyświetla komunikat o pomyślnym dodaniu.

2.5.18 edit_account()

Funkcja `edit_account()` służy do zmiany hasła użytkownika.

Zwracane wartości: Renderowanie formularza zmiany hasła lub przekierowanie do strony głównej po pomyślnej zmianie hasła.

Używane zmienne:

- `EditAccountForm` - formularz zmiany hasła.
- `db` - instancja bazy danych.

Odwołania:

- `render_template` - funkcja Flask do renderowania szablonów.
- `flash` - funkcja Flask do wyświetlania komunikatów.
- `redirect` - funkcja Flask do przekierowań.
- `url_for` - funkcja Flask do generowania URL.

Opis działania: Funkcja wyświetla formularz zmiany hasła. Po wypełnieniu formularza i jego zatwierdzeniu, zmienia hasło bieżącego użytkownika w bazie danych i wyświetla komunikat o pomyślnej zmianie hasła.

2.5.19 assign_meter()

Funkcja `assign_meter()` służy do przypisywania licznika do użytkownika.

Parametry:

- `user_id` - identyfikator użytkownika, do którego ma być przypisany licznik.
- `meter_id` - identyfikator licznika, który ma być przypisany.

Zwracane wartości: Przekierowanie do widoku szczegółów super-użytkownika lub użytkownika, w zależności od uprawnień bieżącego użytkownika.

Używane zmienne:

- `User` - model użytkownika.

- Meter - model licznika.
- db - instancja bazy danych.

Odwołania:

- flash - funkcja Flask do wyświetlania komunikatów.
- redirect - funkcja Flask do przekierowań.
- url_for - funkcja Flask do generowania URL.

Opis działania: Funkcja przypisuje licznik do użytkownika, zapisując zmiany w bazie danych. W przypadku braku uprawnień do przypisania licznika, wyświetla komunikat o błędzie.

2.5.20 delete_meters()

Funkcja delete_meters() służy do usuwania wszystkich liczników i odczytów z bazy danych.

Zwracane wartości: Przekierowanie do panelu administratora.

Używane zmienne:

- MeterReading - model odczytów licznika.
- Meter - model licznika.
- Address - model adresu.
- db - instancja bazy danych.

Odwołania:

- flash - funkcja Flask do wyświetlania komunikatów.
- redirect - funkcja Flask do przekierowań.
- url_for - funkcja Flask do generowania URL.

Opis działania: Funkcja usuwa wszystkie liczniki, odczyty i adresy z bazy danych. W przypadku wystąpienia błędu, wycofuje zmiany i wyświetla komunikat o błędzie.

2.5.21 delete_user()

Funkcja `delete_user()` służy do usuwania użytkownika z bazy danych.

Parametry: `user_id` - identyfikator użytkownika do usunięcia.

Zwracane wartości: Przekierowanie do panelu administratora.

Używane zmienne:

- `User` - model użytkownika.
- `db` - instancja bazy danych.

Odwołania:

- `flash` - funkcja Flask do wyświetlania komunikatów.
- `redirect` - funkcja Flask do przekierowań.
- `url_for` - funkcja Flask do generowania URL.

Opis działania: Funkcja usuwa użytkownika z bazy danych po weryfikacji hasła administratora. W przypadku nieprawidłowego hasła, wyświetla komunikat o błędzie.

2.5.22 deactivate_user()

Funkcja `deactivate_user()` służy do dezaktywacji lub aktywacji konta użytkownika.

Parametry:

- `user_id` - identyfikator użytkownika, którego konto ma zostać dezaktywowane lub aktywowane.

Zwracane wartości: Przekierowanie do widoku szczegółów użytkownika.

Używane zmienne:

- `User` - model użytkownika.
- `db` - instancja bazy danych.

Odwołania:

- `flash` - funkcja Flask do wyświetlania komunikatów.

- `redirect` - funkcja Flask do przekierowań.
- `url_for` - funkcja Flask do generowania URL.

Opis działania: Funkcja zmienia status aktywności konta użytkownika. Jeśli konto jest aktywne, zostaje dezaktywowane, a jeśli jest dezaktywne, zostaje aktywowane. Następnie zapisuje zmiany w bazie danych i wyświetla odpowiedni komunikat.

2.5.23 `update_user_notes()`

Funkcja `update_user_notes()` służy do aktualizacji notatek przypisanych do użytkownika.

Parametry:

- `user_id` - identyfikator użytkownika, którego notatki mają zostać zaktualizowane.

Zwracane wartości: Przekierowanie do widoku szczegółów użytkownika.

Używane zmienne:

- `User` - model użytkownika.
- `UserNotesForm` - formularz do edycji notatek użytkownika.
- `db` - instancja bazy danych.

Odwołania:

- `flash` - funkcja Flask do wyświetlania komunikatów.
- `redirect` - funkcja Flask do przekierowań.
- `url_for` - funkcja Flask do generowania URL.

Opis działania: Funkcja aktualizuje notatki użytkownika na podstawie danych wprowadzonych w formularzu. Po zapisaniu zmian w bazie danych, wyświetla komunikat o pomyślnej aktualizacji.

2.5.24 messages()

Funkcja `messages()` służy do obsługi wysyłania wiadomości między użytkownikami.

Zwracane wartości: Renderowanie szablonu do wysyłania wiadomości.

Używane zmienne:

- `MessageForm` - formularz do wysyłania wiadomości.
- `Message` - model wiadomości.
- `User` - model użytkownika.
- `db` - instancja bazy danych.

Odwołania:

- `flash` - funkcja Flask do wyświetlania komunikatów.
- `redirect` - funkcja Flask do przekierowań.
- `url_for` - funkcja Flask do generowania URL.
- `render_template` - funkcja Flask do renderowania szablonów HTML.

Opis działania: Funkcja umożliwia wysyłanie wiadomości do innych użytkowników. Administrator może wysłać wiadomości do wszystkich użytkowników, a zwykli użytkownicy tylko do siebie. Po wysłaniu wiadomości, zwiększa licznik nieprzeczytanych wiadomości odbiorcy i zapisuje wiadomość w bazie danych.

2.5.25 message()

Funkcja `message()` służy do wyświetlania szczegółów pojedynczej wiadomości.

Parametry:

- `message_id` - identyfikator wiadomości do wyświetlenia.

Zwracane wartości: Renderowanie szablonu z wyświetleniem szczegółów wiadomości.

Używane zmienne:

- Message - model wiadomości.

Odwołania:

- render_template - funkcja Flask do renderowania szablonów HTML.

Opis działania: Funkcja pobiera wiadomość o podanym identyfikatorze z bazy danych i przekazuje ją do szablonu HTML, aby wyświetlić jej szczegóły.

2.5.26 delete_message()

Funkcja delete_message() służy do usuwania wiadomości.

Parametry:

- message_id - identyfikator wiadomości do usunięcia.

Zwracane wartości: Przekierowanie do listy wiadomości.

Używane zmienne:

- Message - model wiadomości.
- db - instancja bazy danych.

Odwołania:

- flash - funkcja Flask do wyświetlania komunikatów.
- redirect - funkcja Flask do przekierowań.
- url_for - funkcja Flask do generowania URL.

Opis działania: Funkcja usuwa wybraną wiadomość z bazy danych i wyświetla komunikat o jej usunięciu.

2.5.27 assign_meters_to_user()

Funkcja assign_meters_to_user() służy do przypisywania liczników do użytkownika.

Parametry:

- user_id - identyfikator użytkownika, do którego mają zostać przypisane liczniki.

Zwracane wartości: Przekierowanie do widoku szczegółów użytkownika.

Używane zmienne:

- User - model użytkownika.
- Meter - model licznika.
- db - instancja bazy danych.

Odwołania:

- flash - funkcja Flask do wyświetlania komunikatów.
- redirect - funkcja Flask do przekierowań.
- url_for - funkcja Flask do generowania URL.

Opis działania: Funkcja przypisuje liczniki do użytkownika na podstawie listy numerów liczników. Jeśli licznik nie jest przypisany do innego użytkownika, zostaje przypisany do wybranego użytkownika. Funkcja zwraca listę pomyślnie przypisanych liczników oraz listę liczników, których przypisanie było niemożliwe.

2.5.28 summary()

Funkcja `summary()` służy do wyświetlania podsumowania odczytów liczników na podstawie wybranego adresu i zakresu dat.

Parametry: Brak parametrów w URL.

Zwracane wartości: Renderowanie szablonu z wyświetleniem odczytów liczników.

Używane zmienne:

- MeterReading - model odczytów liczników.
- Meter - model liczników.

Odwołania:

- render_template - funkcja Flask do renderowania szablonów HTML.

Opis działania: Funkcja filtruje odczyty liczników na podstawie podanego adresu i zakresu dat. Jeśli metoda żądania to POST, pobiera dane z formularza i filtruje odczyty, a następnie przekazuje je do szablonu HTML.

2.5.29 assign_meter_to_superuser()

Funkcja `assign_meter_to_superuser()` służy do przypisywania liczników do superużytkowników przez administratora.

Parametry: Brak parametrów w URL.

Zwracane wartości: Renderowanie szablonu lub przekierowanie.

Używane zmienne:

- Meter - model liczników.
- User - model użytkowników.
- db - instancja bazy danych.

Odwołania:

- flash - funkcja Flask do wyświetlania komunikatów.
- redirect - funkcja Flask do przekierowań.
- url_for - funkcja Flask do generowania URL.

Opis działania: Funkcja umożliwia administratorowi przypisanie wybranego licznika do superużytkownika. Po weryfikacji danych z formularza, przypisuje licznik i zapisuje zmiany w bazie danych.

2.5.30 superuser_assign_meter()

Funkcja `superuser_assign_meter()` umożliwia superużytkownikowi przypisanie liczników do użytkowników.

Parametry: Brak parametrów w URL.

Zwracane wartości: Renderowanie szablonu lub przekierowanie.

Używane zmienne:

- User - model użytkowników.
- Meter - model liczników.
- db - instancja bazy danych.

Odwołania:

- flash - funkcja Flask do wyświetlania komunikatów.
- redirect - funkcja Flask do przekierowań.

- `url_for` - funkcja Flask do generowania URL.

Opis działania: Funkcja pozwala superużytkownikowi na przypisanie liczników do użytkowników. Po weryfikacji danych z formularza, przypisuje licznik do użytkownika i zapisuje zmiany w bazie danych.

2.5.31 `superuser_user_overview()`

Funkcja `superuser_user_overview()` służy do wyświetlania szczegółów użytkownika przez superużytkownika.

Parametry:

- `user_id` - identyfikator użytkownika.

Zwracane wartości: Renderowanie szablonu z informacjami o użytkowniku.

Używane zmienne:

- `User` - model użytkowników.
- `Meter` - model liczników.
- `db` - instancja bazy danych.

Odwołania:

- `flash` - funkcja Flask do wyświetlania komunikatów.
- `redirect` - funkcja Flask do przekierowań.
- `url_for` - funkcja Flask do generowania URL.

Opis działania: Funkcja wyświetla szczegółowe informacje o użytkowniku, w tym przypisane mu liczniki. Umożliwia przypisanie nowych liczników do użytkownika oraz aktualizację notatek.

2.5.32 `superuser_panel()`

Funkcja `superuser_panel()` służy do wyświetlania panelu superużytkownika.

Parametry: Brak parametrów w URL.

Zwracane wartości: Renderowanie szablonu panelu superużytkownika.

Używane zmienne:

- User - model użytkowników.
- Meter - model liczników.
- db - instancja bazy danych.

Odwołania:

- flash - funkcja Flask do wyświetlania komunikatów.
- redirect - funkcja Flask do przekierowań.
- url_for - funkcja Flask do generowania URL.

Opis działania: Funkcja umożliwia superużytkownikowi dodawanie nowych użytkowników oraz zarządzanie przypisanymi użytkownikom licznikami.

2.5.33 assign_user_to_superuser()

Funkcja `assign_user_to_superuser()` służy do przypisywania użytkowników do superużytkownika przez administratora.

Parametry:

- `superuser_id` - identyfikator superużytkownika.
- `user_id` - identyfikator użytkownika do przypisania.

Zwracane wartości: Przekierowanie do widoku szczegółów użytkownika.

Używane zmienne:

- User - model użytkowników.
- Meter - model liczników.
- db - instancja bazy danych.

Odwołania:

- flash - funkcja Flask do wyświetlania komunikatów.
- redirect - funkcja Flask do przekierowań.
- url_for - funkcja Flask do generowania URL.

Opis działania: Funkcja umożliwia administratorowi przypisanie wybranego użytkownika do superużytkownika. Po weryfikacji danych, przypisuje użytkownika i zapisuje zmiany w bazie danych.

2.5.34 `remove_assigned_user()`

Funkcja `remove_assigned_user()` służy do usuwania przypisania użytkownika do superużytkownika.

Parametry:

- `user_id` - identyfikator przypisanego użytkownika.

Zwracane wartości: Przekierowanie do widoku szczegółów użytkownika.

Używane zmienne:

- `User` - model użytkowników.
- `db` - instancja bazy danych.

Odwołania:

- `flash` - funkcja Flask do wyświetlania komunikatów.
- `redirect` - funkcja Flask do przekierowań.
- `url_for` - funkcja Flask do generowania URL.

Opis działania: Funkcja umożliwia superużytkownikowi lub administratorowi usunięcie przypisania użytkownika do superużytkownika. Po weryfikacji uprawnień, usuwa przypisanie i zapisuje zmiany w bazie danych.

2.5.35 `generate_report()`

Funkcja `generate_report()` służy do generowania raportu na podstawie wybranych liczników i okresu czasu.

Parametry: Brak parametrów w URL.

Zwracane wartości: Renderowanie szablonu generowania raportu lub przekierowanie do wyświetlenia raportu.

Używane zmienne:

- `User` - model użytkowników.
- `session` - sesja użytkownika do przechowywania danych raportu.

Odwołania:

- `flash` - funkcja Flask do wyświetlania komunikatów.
- `redirect` - funkcja Flask do przekierowań.
- `url_for` - funkcja Flask do generowania URL.
- `create_report_data` - funkcja pomocnicza do tworzenia danych raportu.

Opis działania: Funkcja umożliwia superużytkownikowi lub administratorowi wygenerowanie raportu na podstawie wybranych liczników i okresu czasu. Dane raportu są zapisywane w sesji, a następnie użytkownik jest przekierowywany do strony z wyświetleniem raportu.

2.5.36 `display_report()`

Funkcja `display_report()` służy do wyświetlania wygenerowanego raportu.

Parametry: Brak parametrów w URL.

Zwracane wartości: Renderowanie szablonu wyświetlającego raport.

Używane zmienne:

- `session` - sesja użytkownika do pobierania danych raportu.
- `datetime` - moduł do obsługi dat i czasu.
- `relativedelta` - funkcja do manipulacji datami.

Odwołania:

- `render_template` - funkcja Flask do renderowania szablonów HTML.

Opis działania: Funkcja pobiera dane raportu z sesji, przetwarza je (w tym tłumaczy nazwy miesięcy na polski), a następnie renderuje szablon HTML z wyświetleniem raportu.

2.5.37 `add_multiple_users()`

Funkcja `add_multiple_users()` służy do dodawania wielu użytkowników na podstawie listy adresów email.

Parametry:

- `emails` - lista adresów email pobrana z formularza.

Zwracane wartości: Renderowanie szablonu z podsumowaniem dodanych użytkowników.

Używane zmienne:

- `User` - model użytkowników.
- `db` - instancja bazy danych.

Odwołania:

- `render_template` - funkcja Flask do renderowania szablonów HTML.
- `generate_random_password` - funkcja pomocnicza do generowania losowych haseł.

Opis działania: Funkcja przetwarza listę adresów email, dla każdego adresu tworzy nowego użytkownika z losowo wygenerowanym hasłem, dodaje użytkowników do bazy danych, a następnie renderuje szablon HTML z podsumowaniem dodanych użytkowników.

2.6 Plik `utils.py`

Plik `utils.py` zawiera pomocnicze funkcje i dekoratory używane w aplikacji.

2.6.1 `admin_required`

Dekorator `admin_required` służy do ograniczenia dostępu do określonych funkcji tylko dla użytkowników z uprawnieniami administratora.

Parametry: Funkcja przyjmuje jako argument inną funkcję.

Zwracane wartości: Funkcja dekorowana lub przekierowanie do strony głównej w przypadku braku uprawnień.

Opis działania: Sprawdza, czy bieżący użytkownik posiada uprawnienia administratora. Jeśli nie, wyświetla komunikat o braku uprawnień i przekierowuje na stronę główną.

2.6.2 `superuser_required`

Dekorator `superuser_required` służy do ograniczenia dostępu do określonych funkcji tylko dla superużytkowników lub administratorów.

Parametry: Funkcja przyjmuje jako argument inną funkcję.

Zwracane wartości: Funkcja dekorowana lub przekierowanie do strony głównej w przypadku braku uprawnień.

Opis działania: Sprawdza, czy bieżący użytkownik jest superużytkownikiem lub administratorem. Jeśli nie, wyświetla komunikat o braku uprawnień i przekierowuje na stronę główną.

2.6.3 `allowed_file`

Funkcja `allowed_file` sprawdza, czy przesłany plik ma dozwolone rozszerzenie.

Parametry:

- `filename` - nazwa pliku do sprawdzenia.

Zwracane wartości: True jeśli plik ma dozwolone rozszerzenie, w przeciwnym razie False.

Opis działania: Sprawdza, czy nazwa pliku zawiera kropkę i czy rozszerzenie pliku to 'csv'.

2.6.4 `is_valid_link`

Funkcja `is_valid_link` sprawdza, czy link aktywacyjny jest ważny.

Parametry:

- `user_link` - link aktywacyjny do sprawdzenia.

Zwracane wartości: True jeśli link jest ważny, w przeciwnym razie False.

Opis działania: Sprawdza, czy link aktywacyjny istnieje w bazie danych i czy nie został już użyty.

2.6.5 `process_csv_water`

Funkcja `process_csv_water` przetwarza plik CSV z odczytami wodomierzy.

Parametry:

- `file_path` - ścieżka do pliku CSV.

Zwracane wartości: Komunikat o wyniku przetwarzania pliku.

Opis działania: Funkcja odczytuje plik CSV, przetwarza dane, tworzy lub aktualizuje obiekty w bazie danych. Obsługuje różne formaty plików i kodowania.

2.6.6 `process_csv_heat`

Funkcja `process_csv_heat` przetwarza plik CSV z odczytami ciepłomierzy.

Parametry:

- `file_path` - ścieżka do pliku CSV.

Zwracane wartości: Komunikat o wyniku przetwarzania pliku.

Opis działania: Podobnie jak `process_csv_water`, funkcja odczytuje i przetwarza plik CSV, tworząc lub aktualizując obiekty w bazie danych.

2.6.7 `process_csv_events`

Funkcja `process_csv_events` przetwarza plik CSV z zdarzeniami dla wodomierzy lub ciepłomierzy.

Parametry:

- `file_path` - ścieżka do pliku CSV.
- `type` - typ urządzenia ('water' lub 'heat').

Zwracane wartości: Brak. Funkcja zapisuje dane bezpośrednio do bazy danych.

Opis działania: Funkcja odczytuje plik CSV i na jego podstawie tworzy zdarzenia w bazie danych, przypisując je do odpowiednich liczników.

2.6.8 `create_report_data`

Funkcja `create_report_data` tworzy dane do raportu na podstawie wybranych liczników i okresu.

Parametry:

- `selected_meters` - lista wybranych liczników.
- `report_period` - okres czasu dla raportu.

Zwracane wartości: Lista danych do wykorzystania w raporcie.

Opis działania: Funkcja generuje dane do raportu, łącząc informacje o licznikach, ich odczytach i adresach w określonym okresie czasu.

Rozdział 3

Szablony

- `base.html` - Szablon główny aplikacji.
- `admin_panel.html` - główny panel administratora.
- `display_report.html` - wyświetlenie raportów.
- `edit_account.html` - zmiana hasła do konta.
- `generate_report.html` - generowanie raportów.
- `home.html` - strona domowa.
- `login.html` - strona logowania.
- `messages.html` - skrzynka wiadomości.
- `meter_details.html` - szczegóły licznika.
- `send_message.html` - wysyłanie wiadomości przez administratora.
- `superuser_panel.html` - Panel administratora zasobów.
- `superuser_user_overview.html` - podgląd użytkownika przez superusera
- `upload_csv.html` - przesyłanie plików wsadowych.
- `user_overview.html` - Przegląd użytkownika przez administratora.

3.1 base.html

Plik `base.html` jest podstawowym szablonem HTML używanym w aplikacji *Zdalny Odczyt*. Ten szablon definiuje strukturę podstawową, wspólną dla wszystkich innych szablonów w aplikacji. Zawiera definicje nagłówka, stopki, nawigacji oraz głównego kontenera treści.

3.1.1 Nagłówek

Nagłówek zawiera podstawowe informacje o stronie, takie jak tytuł, metadane, linki do arkuszy stylów CSS oraz skryptów JavaScript. Używa Bootstrapa dla stylizacji oraz responsywności strony.

3.1.2 Nawigacja

Sekcja nawigacji definiuje pasek nawigacyjny, który jest stały (przyklejony do górnej części ekranu). Zawiera linki do różnych sekcji aplikacji, które są dostępne w zależności od statusu autoryzacji użytkownika.

3.1.3 Główny Kontener Treści

Główny kontener (`<div class="center-box">`) jest miejscem, w którym wyświetlane są różne treści, w zależności od konkretnej strony. Zawartość tego kontenera jest dynamicznie zmieniana przez inne szablony, które rozszerzają `base.html`.

3.1.4 Stopka

Stopka zawiera informacje kontaktowe, linki do polityki prywatności, warunków użytkowania, regulaminu oraz podpis autora aplikacji.

3.1.5 Skrypty JavaScript

- **Bootstrap JS:** Używany do obsługi elementów interaktywnych w Bootstrapie, takich jak rozwijane menu.
- **jQuery:** Biblioteka JavaScript używana przez Bootstrap.
- **Popper.js:** Biblioteka używana do pozycjonowania elementów (np. dymków) w Bootstrapie.

- **Plotly.js:** Biblioteka do tworzenia interaktywnych wykresów.
- **Własny skrypt:** Skrypt do dostosowywania marginesu górnego głównego kontenera treści, tak aby nie był zasłaniany przez pasek nawigacyjny.

3.1.6 Stylizacja

Stylizacja jest realizowana głównie przez Bootstrap oraz dodatkowy arkusz stylów `style.css`, który definiuje niestandardowe style specyficzne dla aplikacji.

3.2 admin_panel.html

Plik `admin_panel.html` jest szablonem HTML używanym w panelu administratora aplikacji *Zdalny Odczyt*. Szablon ten rozszerza `base.html` i zawiera specyficzne dla panelu administratora elementy interfejsu użytkownika oraz logikę.

3.2.1 Sekcja Dodawania Użytkowników

Sekcja ta zawiera przyciski do pokazywania i ukrywania formularzy do dodawania pojedynczych użytkowników oraz dodawania wielu użytkowników naraz. Umożliwia administratorowi zarządzanie użytkownikami w systemie.

3.2.2 Lista Użytkowników i Liczników

Sekcja ta zawiera rozwijane listy użytkowników i liczników. Użytkownik może wyszukiwać użytkowników i liczniki, a także przeglądać szczegóły każdego z nich. Lista jest dynamicznie filtrowana na podstawie wprowadzonego tekstu.

3.2.3 Formularze

Szablon zawiera dwa formularze:

- **Formularz Dodawania Użytkownika:** Pozwala na dodanie nowego użytkownika do systemu.

- **Formularz Dodawania Wielu Użytkowników:** Umożliwia dodanie wielu użytkowników jednocześnie poprzez wprowadzenie adresów e-mail.

3.2.4 Skrypty JavaScript

Szablon zawiera skrypty JavaScript do obsługi interakcji użytkownika, takich jak pokazywanie/ukrywanie formularzy, wyszukiwanie użytkowników/liczników oraz zmiana ikon w przyciskach rozwijanych.

3.2.5 Stylizacja

Stylizacja jest zgodna z `base.html`, z dodatkowymi stylami specyficznymi dla panelu administratora, takimi jak szerokość i układ elementów.

3.2.6 Zastosowanie

Szablon `admin_panel.html` jest używany jako interfejs dla administratorów systemu, umożliwiając im zarządzanie użytkownikami i licznikami, a także dostęp do funkcji administracyjnych, takich jak generowanie raportów i przesyłanie plików CSV.

3.3 `display_report.html`

Plik `display_report.html` jest szablonem HTML używanym do wyświetlania raportów w aplikacji *Zdalny Odczyt*. Szablon ten rozszerza `base.html` i zawiera elementy interfejsu użytkownika oraz logikę specyficzną dla wyświetlania raportów.

3.3.1 Nagłówek Raportu

Nagłówek raportu zawiera logo aplikacji, tytuł raportu oraz informacje o okresie, którego raport dotyczy. Dodatkowo, wyświetla listę unikalnych adresów e-mail użytkowników uwzględnionych w raporcie.

3.3.2 Przyciski Akcji

Sekcja zawiera przyciski umożliwiające pobranie raportu w formacie CSV lub JSON oraz drukowanie raportu w formacie PDF. Każdy przycisk jest powiązany ze skrypcem JavaScript, który realizuje odpowiednią funkcjonalność.

3.3.3 Tabela Raportu

Tabela zawiera szczegółowe dane raportu, w tym adresy e-mail użytkowników, adresy liczników, numery radiowe, typy liczników oraz odczyty dla poszczególnych miesięcy. Dane są dynamicznie generowane na podstawie danych przekazanych do szablonu.

3.3.4 Skrypty JavaScript

- **Eksport do CSV:** Skrypt umożliwia eksport danych tabeli do pliku CSV. Użytkownik ma możliwość wyboru między eksportem z polskimi znakami a trybem zgodności.
- **Drukowanie PDF:** Skrypt umożliwia wydrukowanie raportu. Użytkownik jest informowany o sugerowanej nazwie pliku PDF.
- **Eksport do JSON:** Skrypt umożliwia eksport danych tabeli do pliku JSON.

3.3.5 Stylizacja

Stylizacja jest zgodna z `base.html`, z dodatkowymi stylami specyficznymi dla raportu, takimi jak formatowanie tabeli i ukrywanie niektórych elementów podczas drukowania.

3.3.6 Zastosowanie

Szablon `display_report.html` jest używany do prezentacji wygenerowanych raportów, umożliwiając użytkownikom przeglądanie, eksportowanie i drukowanie szczegółowych danych dotyczących odczytów liczników.

3.4 edit_account.html

Plik `edit_account.html` jest szablonem HTML używanym do umożliwienia użytkownikom edycji ich profilu w aplikacji *Zdalny Odczyt*. Szablon ten rozszerza `base.html` i zawiera formularz do zmiany hasła użytkownika.

3.4.1 Przycisk Powrotu

Na górze strony znajduje się przycisk "Powrót", który umożliwia użytkownikowi powrót do strony głównej aplikacji.

3.4.2 Formularz Edycji Profilu

Formularz umożliwia użytkownikowi zmianę swojego hasła. Zawiera następujące pola:

- **Obecne hasło:** Pole do wprowadzenia obecnego hasła użytkownika.
- **Nowe hasło:** Pole do wprowadzenia nowego hasła.
- **Potwierdź nowe hasło:** Pole do potwierdzenia nowego hasła.
- **Przycisk Zmień hasło:** Przycisk do zatwierdzenia zmian.

3.4.3 Walidacja Formularza

Formularz zawiera walidację po stronie serwera, która zapewnia, że nowe hasło jest poprawnie wprowadzone i potwierdzone.

3.4.4 Stylizacja

Stylizacja formularza jest zgodna z ogólnym stylem aplikacji, z wykorzystaniem klas Bootstrapa do formatowania pól formularza i przycisków.

3.4.5 Zastosowanie

Szablon `edit_account.html` jest używany do zapewnienia użytkownikom możliwości bezpiecznej zmiany ich hasła, co jest kluczowym aspektem zarządzania kontem użytkownika w aplikacji.

3.5 `generate_report.html`

Plik `generate_report.html` jest szablonem HTML używanym do generowania raportów w aplikacji *Zdalny Odczyt*. Szablon ten rozszerza `base.html` i zawiera interfejs do wyboru użytkowników i liczników, dla których ma być wygenerowany raport.

3.5.1 Przycisk Powrotu

Na górze strony znajduje się przycisk "Powrót", który umożliwia użytkownikowi powrót do strony głównej aplikacji.

3.5.2 Formularz Generowania Raportu

Formularz umożliwia użytkownikowi wybór konkretnych użytkowników i ich liczników, dla których ma być wygenerowany raport. Zawiera następujące elementy:

- **Pole wyszukiwania:** Umożliwia filtrowanie listy użytkowników i liczników.
- **Lista użytkowników i liczników:** Wyświetla użytkowników i przypisane do nich liczniki, które można wybrać do raportu.
- **Wybór okresu raportu:** Pozwala na wybór okresu, za który ma być wygenerowany raport.
- **Przycisk generowania raportu:** Przycisk do zatwierdzenia wyboru i generowania raportu.

3.5.3 Interaktywność

Szablon zawiera skrypty JavaScript, które zapewniają interaktywność formularza:

- **Rozwijanie listy liczników:** Użytkownik może rozwijać i związać listę liczników dla każdego użytkownika.
- **Zaznaczanie liczników:** Możliwość zaznaczenia wszystkich liczników dla wybranego użytkownika.
- **Filtrowanie listy:** Możliwość wyszukiwania użytkowników i liczników na podstawie wprowadzonego tekstu.

3.5.4 Stylizacja

Stylizacja formularza jest zgodna z ogólnym stylem aplikacji, z wykorzystaniem klas Bootstrapa do formatowania elementów formularza.

3.5.5 Zastosowanie

Szablon `generate_report.html` jest używany do umożliwienia administratorom i superużytkownikom generowania szczegółowych raportów dotyczących odczytów liczników dla wybranych użytkowników i okresów, co jest kluczowym aspektem zarządzania danymi w aplikacji.

3.6 home.html

Plik `home.html` jest szablonem HTML używanym jako strona główna aplikacji *Zdalny Odczyt* dla zalogowanych użytkowników. Szablon ten rozszerza `base.html` i zawiera interfejs użytkownika prezentujący listę przypisanych liczników oraz inne informacje.

3.6.1 Powitanie Użytkownika

Na górze strony znajduje się powitanie użytkownika, które wykorzystuje adres e-mail zalogowanego użytkownika do wyświetlenia jego nazwy (część adresu e-mail przed znakiem @).

3.6.2 Lista Przypisanych Liczników

Sekcja ta zawiera listę liczników przypisanych do zalogowanego użytkownika. Każdy element listy jest linkiem do szczegółów danego licznika. Lista ta może być filtrowana za pomocą pola wyszukiwania.

- **Pole wyszukiwania:** Umożliwia filtrowanie listy liczników na podstawie wprowadzonego tekstu.
- **Lista liczników:** Wyświetla liczniki przypisane do użytkownika. Liczniki mogą być wyświetlane z nazwą (jeśli jest dostępna) i numerem radiowym.

3.6.3 Interaktywność

Szablon zawiera skrypt JavaScript, który umożliwia dynamiczne filtrowanie listy liczników na podstawie wprowadzonego tekstu w polu wyszukiwania.

3.6.4 Stylizacja

Stylizacja strony jest zgodna z ogólnym stylem aplikacji, z wykorzystaniem klas Bootstrapa oraz niestandardowych stylów CSS.

3.6.5 Zastosowanie

Szablon `home.html` jest używany jako strona startowa dla zalogowanych użytkowników, zapewniając szybki dostęp do informacji o ich licznikach oraz umożliwiając łatwe zarządzanie i monitorowanie ich danych. Jest to kluczowy element interfejsu użytkownika, który ułatwia nawigację i dostęp do funkcji aplikacji.

3.7 login.html

Plik `login.html` jest szablonem HTML używanym do prezentowania formularza logowania w aplikacji *Zdalny Odczyt*. Ten szablon rozszerza `base.html` i zawiera interfejs użytkownika umożliwiający zalogowanie się do systemu.

3.7.1 Formularz Logowania

Formularz logowania jest centralnym elementem tego szablonu. Umożliwia użytkownikom wprowadzenie swoich danych logowania, takich jak adres e-mail i hasło.

- **Pole e-mail:** Użytkownik wprowadza swój adres e-mail.
- **Pole hasła:** Użytkownik wprowadza swoje hasło.
- **Pole zapamiętaj mnie:** Opcja umożliwiająca użytkownikowi pozostanie zalogowanym nawet po zamknięciu przeglądarki.
- **Przycisk logowania:** Przycisk, który użytkownik naciska, aby zalogować się do systemu.

3.7.2 Link do Rejestracji

Pod formularzem znajduje się link do strony firmy *Smartbits*, gdzie nowi użytkownicy mogą skontaktować się w celu utworzenia konta. Jest to przydatne dla osób, które jeszcze nie mają konta w systemie.

3.7.3 Stylizacja

Formularz logowania jest stylizowany za pomocą klas Bootstrapa oraz niestandardowych stylów CSS, aby zapewnić spójność z resztą aplikacji. Formularz jest umieszczony w kontenerze o ograniczonej szerokości, co ułatwia czytelność i koncentrację na formularzu.

3.7.4 Zastosowanie

Szablon `login.html` jest kluczowym elementem interfejsu użytkownika, umożliwiającym dostęp do zabezpieczonych części aplikacji. Jest to pierwszy punkt interakcji dla użytkowników, którzy muszą się zalogować, aby korzystać z funkcji aplikacji. Zapewnia bezpieczny i wygodny sposób na weryfikację tożsamości użytkownika.

3.8 message.html

Plik `message.html` jest szablonem HTML używanym do wyświetlania szczegółów pojedynczej wiadomości w aplikacji *Zdalny Odczyt*. Ten szablon rozszerza `base.html` i prezentuje treść, nadawcę, temat oraz datę wysłania wiadomości.

3.8.1 Szczegóły Wiadomości

Szablon prezentuje następujące informacje o wiadomości:

- **Temat:** Wyświetla temat wiadomości.
- **Nadawca:** Pokazuje adres e-mail nadawcy wiadomości.
- **Data:** Data i czas wysłania wiadomości.
- **Treść:** Zawartość wiadomości.

3.8.2 Przycisk Usunięcia Wiadomości

Użytkownik ma możliwość usunięcia wiadomości za pomocą przycisku "Usuń wiadomość". Po kliknięciu, użytkownik jest przekierowywany do akcji usuwania wiadomości w systemie.

3.8.3 Przycisk Powrotu

Dodatkowo, szablon zawiera przycisk "Powrót", który umożliwia użytkownikowi powrót do poprzedniej strony. Jest to realizowane przez funkcję JavaScript `goBack()`, która wywołuje `window.history.back()`.

3.8.4 Stylizacja i Układ

Szablon jest stylizowany za pomocą Bootstrapa i niestandardowych stylów CSS, aby zapewnić spójność z resztą aplikacji. Informacje o wiadomości są prezentowane w czytelny i uporządkowany sposób, co ułatwia użytkownikowi zrozumienie treści.

3.8.5 Zastosowanie

Szablon `message.html` jest używany do wyświetlania szczegółów konkretnych wiadomości w aplikacji. Jest to ważny element interfejsu użytkownika, umożliwiający użytkownikom przeglądanie i zarządzanie swoimi wiadomościami. Zapewnia łatwy dostęp do ważnych informacji zawartych w komunikatach.

3.9 messages.html

Plik `messages.html` jest szablonem HTML wykorzystywanym do zarządzania wiadomościami w aplikacji *Zdalny Odczyt*. Ten szablon rozszerza `base.html` i umożliwia użytkownikom wysyłanie nowych wiadomości oraz przeglądanie skrzynki odbiorczej i wysłanej.

3.9.1 Formularz Wysyłania Wiadomości

Dla użytkowników z uprawnieniami administratora, szablon zawiera formularz do wysyłania wiadomości. Formularz umożliwia wybór odbiorców, wpisanie tematu oraz treści wiadomości. Użytkownik może również wybrać opcję wysłania wiadomości do wszystkich użytkowników.

3.9.2 Lista Wiadomości

Szablon prezentuje listę wysłanych i otrzymanych wiadomości. Dla każdej wiadomości wyświetlane są informacje o nadawcy, temacie oraz dacie wysłania. Użytkownik może kliknąć na wiadomość, aby zobaczyć jej szczegóły lub usunąć ją.

3.9.3 Filtrowanie Wiadomości

Szablon zawiera funkcjonalność filtrowania wiadomości według nadawcy i tematu. Użytkownik może wpisać tekst w pola filtrów, a lista wiadomości zostanie automatycznie zaktualizowana, aby wyświetlić tylko te, które pasują do kryteriów wyszukiwania.

3.9.4 Skrypty JavaScript

Szablon wykorzystuje JavaScript do obsługi filtrowania wiadomości oraz interakcji z formularzem. Skrypty pozwalają na dynamiczne ukrywanie i wyświetlanie wiadomości oraz elementów interfejsu użytkownika w zależności od akcji użytkownika.

3.9.5 Stylizacja i Układ

Podobnie jak inne szablony, `messages.html` korzysta z Bootstrapa i niestandardowych stylów CSS. Układ jest zaprojektowany tak, aby zapewnić łatwą nawigację i czytelność, a także spójność z resztą aplikacji.

3.9.6 Zastosowanie

Szablon `messages.html` jest kluczowym elementem interfejsu użytkownika, umożliwiającym zarządzanie komunikacją w aplikacji. Zapewnia użytkownikom platformę do wysyłania i odbierania wiadomości, co jest ważnym aspektem współpracy i komunikacji w ramach systemu.

3.10 `meter_details.html`

Plik `mete_details.html` jest szablonem HTML wykorzystywanym do wyświetlania szczegółowych informacji o liczniku w aplikacji *Zdalny Odczyt*. Ten szablon rozszerza `base.html` i umożliwia użytkownikom przeglądanie danych o konkretnym liczniku, w tym odczytów, adresu, a także zarządzanie nazwą i adresem licznika.

3.10.1 Informacje o Liczniku

Szablon prezentuje podstawowe informacje o liczniku, takie jak numer radiowy, typ licznika (wodomierz/ciepłomierz), oraz przypisany adres. Użytkownik może edytować adres licznika oraz, w przypadku bycia właścicielem, zmieniać jego nazwę.

3.10.2 Formularz Edycji Adresu

Jeśli licznik nie ma przypisanego adresu, użytkownik może dodać nowy adres za pomocą formularza. Jeśli adres jest już przypisany, użytkownik może go edytować. Formularz umożliwia wprowadzenie miasta, ulicy, numeru budynku, mieszkania oraz kodu pocztowego.

3.10.3 Wykresy i Tabele Odczytów

Szablon zawiera funkcjonalność do wyświetlania odczytów licznika w formie wykresów (liniowych lub słupkowych) oraz tabel. Użytkownik

może wybrać zakres dat dla wyświetlanych danych. Dodatkowo, istnieje możliwość eksportu danych do formatu CSV oraz drukowania wykresów.

3.10.4 Skrypty JavaScript

Szablon wykorzystuje JavaScript do obsługi interakcji z formularzami, wykresami i tabelami. Skrypty pozwalają na dynamiczne aktualizowanie wykresów i tabel w zależności od wybranego zakresu dat, a także na eksport danych do CSV i drukowanie.

3.10.5 Stylizacja i Układ

Stylizacja szablonu jest zgodna z resztą aplikacji, wykorzystując Bootstrap i niestandardowe style CSS. Układ jest zaprojektowany tak, aby zapewnić intuicyjną nawigację i łatwość obsługi.

3.10.6 Zastosowanie

Szablon `meter_details.html` jest kluczowym elementem interfejsu użytkownika, umożliwiającym szczegółowy wgląd w dane licznika. Jest to ważne dla zarządzania i monitorowania zużycia w ramach systemu *Zdalny Odczyt*.

3.11 `send_message.html`

Plik `send_message.html` jest szablonem HTML używanym do wysyłania wiadomości w aplikacji *Zdalny Odczyt*. Ten szablon rozszerza `base.html` i umożliwia użytkownikom tworzenie oraz wysyłanie wiadomości do innych użytkowników systemu.

3.11.1 Formularz Wysyłania Wiadomości

Szablon zawiera formularz do wysyłania wiadomości, który składa się z następujących elementów:

- **Odbiorca:** Pole wyboru, które pozwala użytkownikowi wybrać odbiorcę wiadomości z listy dostępnych użytkowników.

- **Temat:** Pole tekstowe do wprowadzenia tematu wiadomości.
- **Treść:** Pole tekstowe do wprowadzenia treści wiadomości.
- **Przycisk Wysyłania:** Przycisk do wysłania wiadomości.

3.11.2 Stylizacja i Układ

Stylizacja formularza jest spójna z resztą aplikacji, wykorzystując Bootstrap oraz niestandardowe style CSS. Układ formularza jest przejrzysty i intuicyjny, co ułatwia użytkownikowi proces tworzenia i wysyłania wiadomości.

3.11.3 Zastosowanie

Szablon `send_message.html` odgrywa ważną rolę w komunikacji między użytkownikami systemu *Zdalny Odczyt*. Umożliwia on wysyłanie informacji, zapytań czy powiadomień, co jest kluczowe dla efektywnej współpracy i zarządzania w ramach systemu.

3.12 superuser_panel.html

Plik `superuser_panel.html` jest szablonem HTML używanym w aplikacji *Zdalny Odczyt* do zarządzania zasobami przez superusera. Szablon ten rozszerza `base.html` i zapewnia interfejs do przeglądania i zarządzania zasobami użytkowników.

3.12.1 Panel Administratora Zasobów

Szablon zawiera panel superusera, który umożliwia przeglądanie listy użytkowników i zarządzanie nimi. Panel składa się z dwóch głównych sekcji:

- **Lewy Panel:** Zawiera listę użytkowników. Użytkownik może wyszukać konkretnego użytkownika za pomocą paska wyszukiwania.
- **Prawy Panel:** Zawiera szczegółowe informacje o wybranym użytkowniku. Ta sekcja jest dynamicznie wypełniana w zależności od wybranego użytkownika.

3.12.2 Wyszukiwanie Użytkowników

Szablon zawiera funkcjonalność wyszukiwania, która pozwala filtrować listę użytkowników w czasie rzeczywistym. Użytkownik może wpisać nazwę użytkownika w pasku wyszukiwania, a lista użytkowników zostanie automatycznie zaktualizowana, aby wyświetlić tylko te, które pasują do kryteriów wyszukiwania.

3.12.3 Interakcje i Skrypty JavaScript

Szablon wykorzystuje JavaScript do obsługi interakcji z użytkownikiem, takich jak wyszukiwanie użytkowników. Skrypt nasłuchuje zmian w polu wyszukiwania i aktualizuje wyświetlaną listę użytkowników w zależności od wprowadzonego zapytania.

3.12.4 Zastosowanie

Szablon `admin_resources.html` jest kluczowym elementem interfejsu administratora w systemie *Zdalny Odczyt*. Umożliwia on administratorom efektywne zarządzanie zasobami użytkowników, co jest niezbędne do utrzymania porządku i efektywności w systemie.

3.13 superuser_user_overview.html

Plik `superuser_user_overview.html` jest szablonem HTML używanym w aplikacji *Zdalny Odczyt* do przeglądu informacji o użytkownikach przez superużytkownika. Szablon ten rozszerza `superuser_panel.html` i zapewnia interfejs do przeglądania szczegółów dotyczących konkretnego użytkownika oraz zarządzania przypisanymi do niego licznikami.

3.13.1 Przegląd Użytkownika

Szablon zawiera sekcję z podstawowymi informacjami o użytkowniku, takimi jak email. Jest to punkt wyjścia do dalszego zarządzania zasobami przypisanymi do tego użytkownika.

3.13.2 Przypisane Liczniki

W tej sekcji wyświetlana jest lista liczników przypisanych do użytkownika. Każdy element listy zawiera numer radiowy licznika oraz linki do szczegółów licznika i opcji usunięcia licznika.

3.13.3 Interakcje i Skrypty JavaScript

- **Filtrowanie Liczników:** Użytkownik może filtrować listę przypisanych liczników, wpisując numer radiowy w pole wyszukiwania. Lista jest aktualizowana w czasie rzeczywistym, aby wyświetlać tylko te liczniki, które pasują do kryteriów wyszukiwania.
- **Usuwanie Liczników:** Szablon zawiera funkcję JavaScript `confirmDelete`, która wyświetla okno dialogowe potwierdzenia przed usunięciem licznika. Jeśli użytkownik potwierdzi, przekierowuje do odpowiedniego URL, aby usunąć licznik.

3.13.4 Zastosowanie

Szablon `superuser_user_overview.html` jest kluczowym elementem interfejsu superużytkownika w systemie *Zdalny Odczyt*. Umożliwia on superużytkownikom efektywne zarządzanie zasobami przypisanymi do konkretnych użytkowników, co jest niezbędne do utrzymania porządku i efektywności w systemie.

3.14 upload_csv.html

Plik `upload_csv.html` jest szablonem HTML używanym w aplikacji *Zdalny Odczyt* do przesyłania plików CSV przez administratora. Szablon ten rozszerza `base.html` i zapewnia interfejs do przesyłania plików CSV zawierających dane odczytów liczników.

3.14.1 Formularz Przesyłania Pliku

Szablon zawiera formularz umożliwiający użytkownikowi przesłanie pliku CSV. Formularz składa się z następujących elementów:

- **Typ Urządzenia:** Pole wyboru (`form.device_type`), które pozwala użytkownikowi wybrać typ urządzenia, dla którego plik jest przesyłany (np. wodomierz, ciepłomierz).
- **Plik:** Pole do przesyłania pliku (`form.file`), które umożliwia użytkownikowi wybranie pliku CSV z lokalnego systemu plików.
- **Przycisk Prześlij:** Przycisk (`form.submit`), który po kliknięciu przesyła formularz i załączony plik do serwera.

3.14.2 Zastosowanie

Szablon `upload_csv.html` jest kluczowym elementem interfejsu administratora w systemie *Zdalny Odczyt*. Umożliwia on administratorom łatwe przesyłanie danych odczytów liczników w formacie CSV, co jest niezbędne do aktualizacji i utrzymania bazy danych odczytów. Jest to ważne dla zapewnienia ciągłości i dokładności danych w systemie.

3.14.3 Stylizacja i Układ

Formularz jest stylizowany za pomocą Bootstrapa, co zapewnia spójność i estetykę interfejsu użytkownika. Układ formularza jest prosty i intuicyjny, co ułatwia użytkownikom przesyłanie plików

3.15 `user_meters.html`

Plik `user_meters.html` jest szablonem HTML w aplikacji *Zdalny Odczyt*, który wyświetla listę liczników przypisanych do zalogowanego użytkownika. Szablon ten rozszerza `base.html` i zapewnia spersonalizowany widok dla użytkownika.

3.15.1 Lista Liczników

Szablon zawiera listę (`<ul class="user-list">`) elementów (`<li class="user-item">`), gdzie każdy element reprezentuje jeden licznik przypisany do użytkownika. Dla każdego licznika wyświetlane są następujące informacje:

- **Numer Licznika:** Numer identyfikacyjny licznika (`meter.radio_number`).

- **Nazwa:** Nazwa przypisana licznikowi (`meter.name`), jeśli istnieje.
- **Typ:** Typ licznika, np. wodomierz lub ciepłomierz (`meter.type`).

3.15.2 Linki do Szczegółów Licznika

Każdy element listy zawiera link (``), który przekierowuje użytkownika do strony ze szczegółami danego licznika. Umożliwia to użytkownikowi szybki dostęp do szczegółowych informacji o każdym z jego liczników.

3.15.3 Zastosowanie

Szablon `user_meters.html` jest używany do prezentowania użytkownikom listy ich liczników w łatwy do przeglądania sposób. Jest to szczególnie przydatne w systemach monitorowania zużycia, gdzie użytkownicy mogą mieć przypisanych wiele liczników.

3.15.4 Stylizacja i Układ

Lista liczników jest stylizowana za pomocą klas CSS zdefiniowanych w Bootstrapie oraz niestandardowych arkuszy stylów, co zapewnia spójny i estetyczny wygląd. Układ jest prosty i przejrzysty, co ułatwia użytkownikom nawigację i dostęp do informacji o swoich licznikach.

3.16 user_overview.html

Plik `user_overview.html` jest szablonem HTML w aplikacji *Zdalny Odczyt*, który wyświetla szczegółowe informacje o użytkowniku oraz zarządza przypisanymi do niego licznikami i innymi użytkownikami. Szablon ten rozszerza `admin_panel.html` i jest przeznaczony głównie dla administratorów i superużytkowników.

3.16.1 Informacje o Użytkowniku

Sekcja ta zawiera podstawowe informacje o użytkowniku, takie jak email, przypisanie do superużytkownika (jeśli istnieje), oraz rolę użytkownika (`superuser`, `administrator`).

3.16.2 Notatki o Użytkowniku

W tej sekcji wyświetlane są notatki dotyczące użytkownika. Umożliwia to administratorom i superużytkownikom dodawanie i edycję notatek, co może być przydatne do zarządzania informacjami o użytkownikach.

3.16.3 Przypisane Liczniki

Lista przypisanych liczników jest wyświetlana wraz z opcjami przeglądania szczegółów każdego licznika oraz możliwością ich usunięcia. Dostępne są również funkcje do przypisywania nowych liczników do użytkownika.

3.16.4 Wolne Liczniki

Sekcja ta zawiera listę wolnych liczników, które mogą być przypisane do użytkownika. Użytkownik może przypisać liczniki, wpisując ich numery radiowe.

3.16.5 Przypisani Użytkownicy (dla Superużytkowników)

Jeśli zalogowany użytkownik jest superużytkownikiem, wyświetlana jest lista użytkowników przypisanych do niego. Umożliwia to zarządzanie przypisaniami użytkowników.

3.16.6 Usuwanie i Dezaktywacja Użytkownika

Administratorzy mają możliwość usunięcia lub dezaktywacji konta użytkownika. Wymaga to potwierdzenia hasłem administratora.

3.16.7 Skrypty JavaScript

Szablon zawiera skrypty JavaScript do obsługi dynamicznych elementów interfejsu, takich jak filtrowanie listy liczników i użytkowników, edycja notatek, oraz potwierdzenie akcji usunięcia.

3.16.8 Stylizacja i Układ

Stylizacja i układ szablonu są zgodne z ogólnym motywem aplikacji, zapewniając spójność i czytelność interfejsu użytkownika.

Rozdział 4

Style CSS

4.1 GeneralStyles.css

4.1.1 Opis

Plik `GeneralStyles.css` zawiera ogólne style, które są wspólne dla całej aplikacji. Definiuje podstawowe ustawienia dla elementów HTML, takie jak kolor tekstu, tło, czcionki i inne podstawowe style.

4.1.2 Zawartość

- **Kolory tekstu:** Ustawia domyślny kolor tekstu dla nagłówków (h1-h6).
- **Tło:** Definiuje tło dla całej strony.
- **Box-shadow:** Stosuje cień do elementów z klasą `.box-element`.

4.2 Layout.css

4.2.1 Opis

Plik `Layout.css` odpowiada za układ i rozmieszczenie elementów na stronie. Zawiera style dotyczące układu strony, takie jak marginesy, paddingi, flexbox i inne właściwości CSS służące do tworzenia układu strony.

4.2.2 Zawartość

- **Grid i Flexbox:** Style dla układu strony wykorzystującego grid i flexbox.
- **Responsywność:** Style dla responsywnego zachowania elementów na różnych rozmiarach ekranu.
- **Nawigacja:** Style dla paska nawigacyjnego i jego elementów.

4.3 MediaQueries.css

4.3.1 Opis

Plik `MediaQueries.css` zawiera zapytania medialne, które są używane do tworzenia responsywnego designu. Dzięki temu strona będzie poprawnie wyświetlana na różnych urządzeniach, takich jak telefony komórkowe, tablety i monitory o różnych rozdzielczościach.

4.3.2 Zawartość

- **Zapytania medialne:** Zawiera różne zapytania medialne dla różnych rozmiarów ekranu.
- **Responsywne style:** Dostosowuje układ, rozmiar czcionek i inne elementy, aby strona była czytelna i funkcjonalna na urządzeniach mobilnych.

4.4 ButtonsAndForms.css

4.4.1 Opis

Plik `ButtonsAndForms.css` zawiera style specyficzne dla przycisków i formularzy. Definiuje wygląd przycisków, pól formularzy, etykiet i innych elementów formularzy.

4.4.2 Zawartość

- **Przyciski:** Style dla różnych typów przycisków, w tym przycisków akcji, przycisków anulowania itp.

- **Formularze:** Style dla pól formularzy, takich jak pola tekstowe, pola wyboru, przyciski radiowe i inne.

4.5 UserList.css

4.5.1 Opis

Plik `UserList.css` zawiera style specyficzne dla listy użytkowników. Zawiera style dla elementów listy, takich jak elementy listy użytkowników, stylowanie tekstu, marginesy i inne właściwości związane z wyświetlaniem listy użytkowników.

4.5.2 Zawartość

- **Lista użytkowników:** Style dla listy użytkowników, w tym marginesy, paddingi, tło, obramowanie i inne właściwości CSS.
- **Elementy listy:** Style dla poszczególnych elementów listy, w tym stylowanie linków, przycisków i innych elementów interfejsu użytkownika związanych z listą użytkowników.

Rozdział 5

Linki i kontakt

5.1 serwer SQL

```
1 {
2   "kind": "v12.0",
3   "properties": {
4     "administratorLogin": "develop",
5     "version": "12.0",
6     "state": "Ready",
7     "fullyQualifiedDomainName": "wodomierze-online.
      database.windows.net",
8     "privateEndpointConnections": [],
9     "minimalTlsVersion": "1.2",
10    "publicNetworkAccess": "Enabled",
11    "administrators": {
12      "administratorType": "ActiveDirectory",
13      "principalType": "User",
14      "login": "wwadamski_gmail.com#EXT#@365smartbits.
        onmicrosoft.com",
15      "sid": "985b0fc8-f283-40a9-b18d-d0c436cc7fdd",
16      "tenantId": "e671a479-5d4c-4f94-993f-7a6cf6722c6
        e",
17      "azureADOnlyAuthentication": false
18    },
19    "restrictOutboundNetworkAccess": "Disabled"
20  },
21  "location": "polandcentral",
```

```
22     "tags": {},  
23     "id": "/subscriptions/f17ad7e8-2a8d-481a-b86d-b775b0be5a  
24         26/resourceGroups/Dev-Smartbits/providers/Microsoft.  
25         Sql/servers/wodomierze-online",  
26     "name": "wodomierze-online",  
27     "type": "Microsoft.Sql/servers"  
28 }
```

5.2 Serwer aplikacji

https://app.azure.com/e671a479-5d4c-4f94-993f-7a6cf6722c6e/subscriptions/f17ad7e8-2a8d-481a-b86d-b775b0be5a26/resourceGroups/Dev-Smartbits/providers/Microsoft.Web/sites/webapp01-smbts?utm_source=Portal

5.3 Github

<https://github.com/SmartbitsSp/wodomierze-online.git>

5.4 Kontakt

email: wwadamski@gmail.com
tel.: 728 326 669