

Strona tytułowa

- Imię nazwisko autora: Bartłomiej Florek
- Numery albumów: 125115
- Temat projektu: Muzeum czołgów
- Nazwę przedmiotu: Bazy danych
- Grupa laboratoryjna: lab 1

Spis treści

Strona tytułowa	1
Opis programu:.....	2
Wykorzystane technologie:	2
Opis działań	2
Diagram ERD:.....	3
Opis działania aplikacji	5
Funkcje języka proceduralnego	17
Źródła:	24

Opis programu:

Muzeum Czołów to aplikacja desktopowa, która umożliwia zarządzanie kolekcją czołgów. Zalogowani użytkownicy mogą przeglądać czołgi w danych muzeach, a także wyświetlać wszystkie szczegóły czołgów takie jak: nazwa, opis, zdjęcie, amunicja, kraj produkcji itp.

Aby uruchomić program należy:

- Pobrać PostgreSQL
- W pgAdmin4 utworzyć nową bazę, pod nazwą „tanksdb”.
- Zimportować bazę z pliku „BAZA_GIT.sql”, która utworzy wszystkie tabele oraz wprowadzi dane.
- Pobierz kod źródłowy z repozytorium.
- Otwórz projekt w środowisku JavaFX (np. IntelliJ IDEA).
- Skonfiguruj połączenie z bazą danych w pliku konfiguracyjnym „ConnectDB”.

```
1 usage
static String jdbcUrl = "jdbc:postgresql://localhost:5432/tanksdb";
1 usage
static String username = "postgres";
1 usage
static String password = "admin";
```

- Uruchom aplikację.

Wykorzystane technologie:

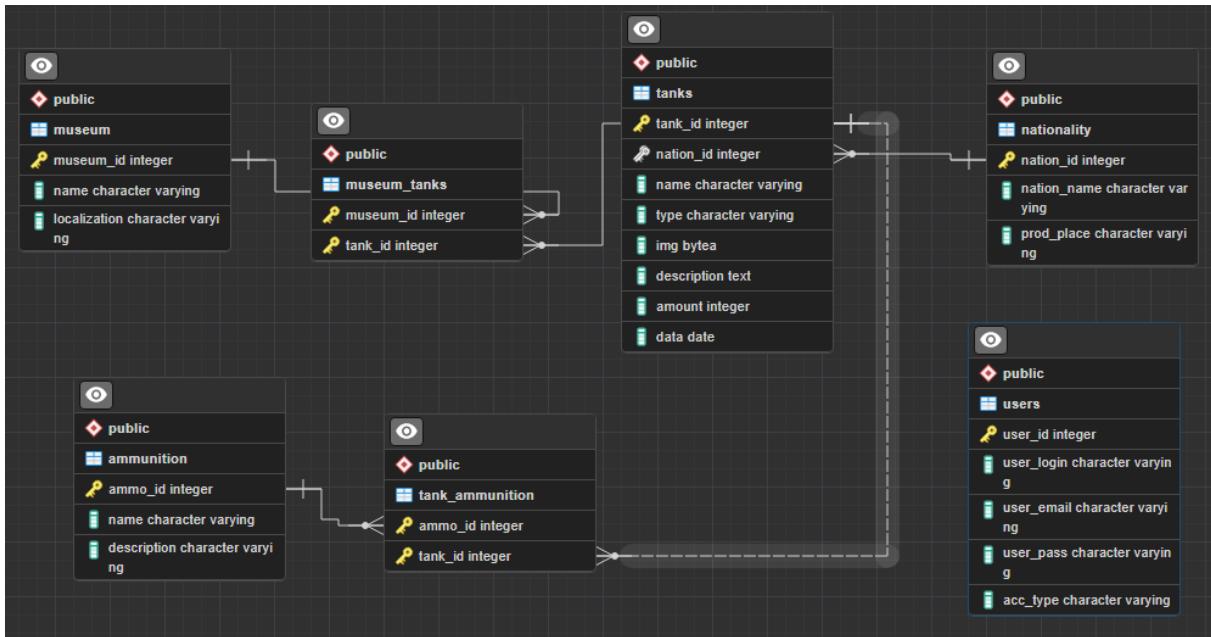
- JavaFX version 21: Do tworzenia interfejsu użytkownika.
- Java SDK version 21.0.2: Do pisania funkcjonalności
- PostgreSQL version 16: Do przechowywania danych o czołgach.
- pgAdmin4 version 7.6: Do zarządzania bazą
- <https://dbdiagram.io/d> - do przygotowania bazy danych

Opis działań

Widoki, funkcje po stronie bazy, wyświetlanie czołgów, dodawanie czołgów, sortowanie, widok dla użytkownika oraz dla admina, baza

Logowanie, tworzenie nowych użytkowników, walidacje, edycje danych, wyszukiwarki

Diagram ERD:



Rysunek 1 - Diagram ERD

Opisy tabel:

Tabela Użytkowników (users):

Kolumny:

- user_id: Liczba całkowita stanowiąca klucz główny identyfikacji użytkownika.
- user_login: Pole typu varchar przechowujące dane logowania użytkownika.
- user_email: Pole typu varchar przechowujące adresy e-mail użytkowników.
- user_pass: Pole typu varchar przechowujące hasła użytkowników.
- acc_type: Pole typu varchar wskazujące typ konta (np. admin, zwykły użytkownik).

Tabela Czołgów (tanks):

Kolumny:

- tank_id: Liczba całkowita stanowiąca klucz główny identyfikacji czołgu.
- nation_id: Liczba całkowita odnosząca się do nation_id z tabeli nationality.
- name: Pole typu varchar zawierające nazwę czołgu.
- type: Pole typu varchar opisujące typ czołgu (np. ciężki, średni, lekki).
- img: Pole typu bytea służące do przechowywania obrazów czołgów.
- description: Pole typu text zawierające opis czołgu.
- amount: Liczba całkowita wskazująca ilość czołgów.

- data: Pole typu date przechowujące daty związane z czołgami

Tabela Nationality:

Kolumny:

- nation_id: Liczba całkowita stanowiąca klucz główny identyfikacji narodowości.
- nation_name: Pole varchar przechowujące nazwę narodu (np. Niemcy, USA).
- prod_place: Pole varchar wskazujące miejsce produkcji (np. lokalizacja fabryki).

Tabela Ammunition:

Kolumny:

- ammo_id: Pole varchar stanowiące klucz główny identyfikacji amunicji.
- name: Pole varchar przechowujące nazwę amunicji.
- description: Pole varchar opisujące amunicję.

Tabela Tank-Ammunition:

Kolumny:

- ammo_id: Liczba całkowita odnosząca się do ammo_id z tabeli ammunition.
- tank_id: Liczba całkowita odnosząca się do tank_id z tabeli tanks.

Tabela Museum Tanks:

Kolumny:

- museum_id: Liczba całkowita odnosząca się do museum_id z tabeli museum.
- tank_id: Liczba całkowita odnosząca się do tank_id z tabeli tanks.

Tabela Museum:

Kolumny:

- museum_id: Liczba całkowita stanowiąca klucz główny identyfikacji muzeum.
- name: Pole varchar przechowujące nazwę muzeum.
- localization: Pole varchar wskazujące lokalizację muzeum.

Opis działania aplikacji:

Rejestracja i logowanie:

Widok rejestracji:

Stwórz nowe konto
Podaj swoje dane...

E-mail

Login
 login...

Hasło
 haslo...

Powtór z hasło
 powtórz hasło...

Stwórz

Rysunek 2 - Formularz rejestracji

Program posiada zabezpieczenia przed wprowadzeniem złych wartości i wyświetla odpowiednie komunikaty:

E-mail
 email...
Nieprawidłowy format adresu email!

Login
 login...
Pole login nie może być puste!

Hasło
 hasło...
Pole hasło nie może być puste!

Powtór z hasło
 powtórz hasło...
Pole powtórzone hasło nie może być puste!

Stwórz

Rysunek 3 - Weryfikacja (puste pola)

Stwórz nowe konto
Podaj swoje dane...

E-mail
 123
Nieprawidłowy format adresu email!

Login
 login...
Pole login nie może być puste!

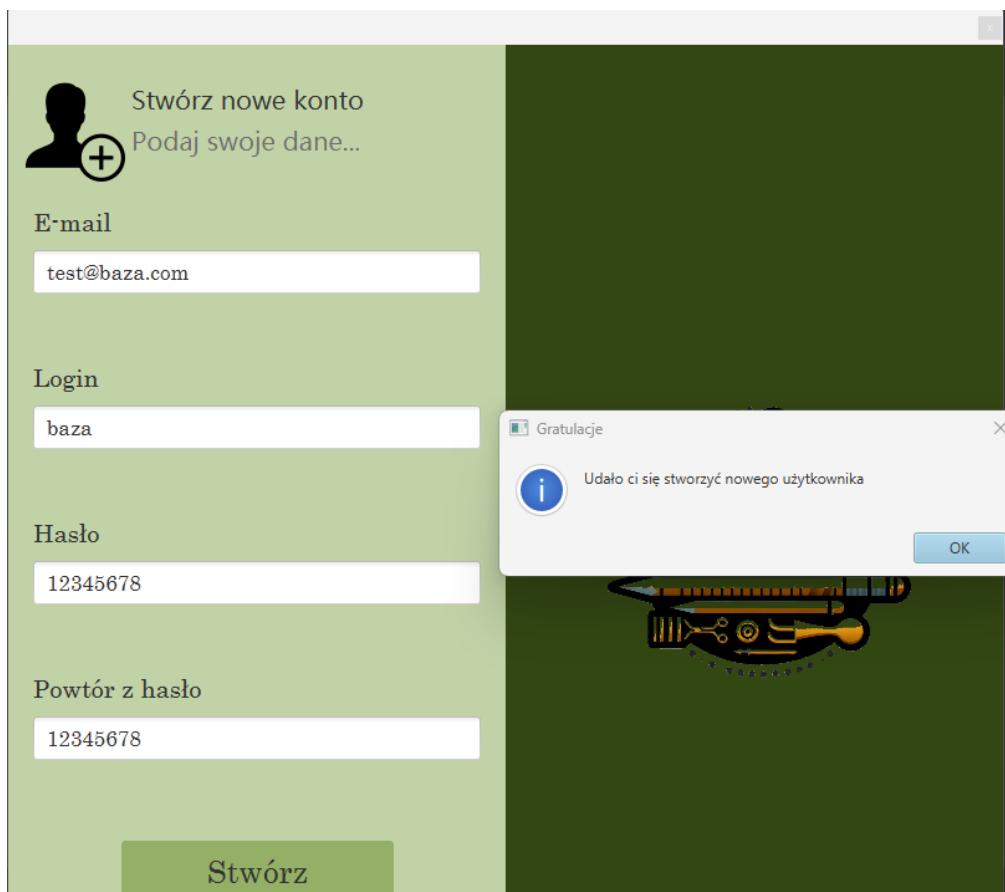
Hasło
 ffffff
Hasła muszą być takie same!

Powtór z hasło
 44444444
Hasła muszą być takie same!

Stwórz

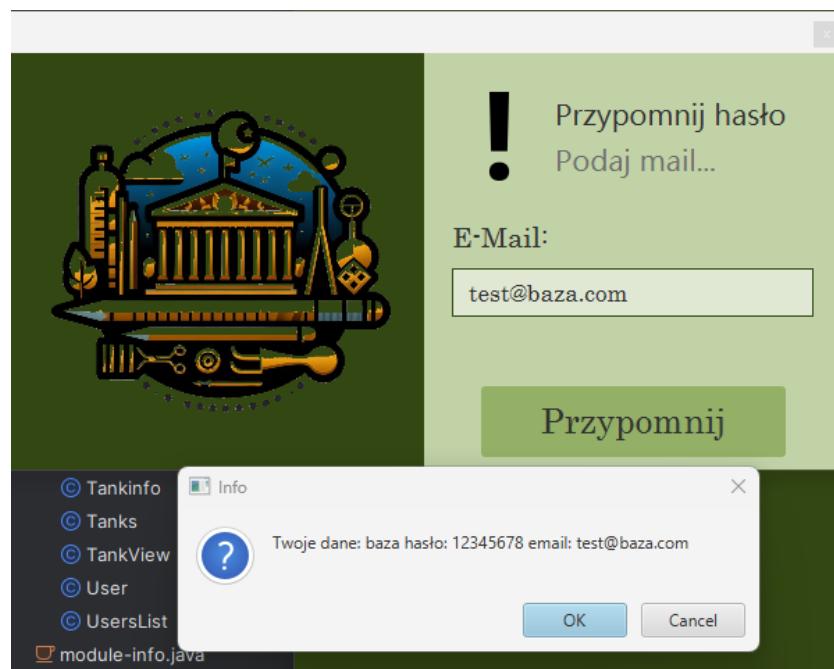
Rysunek 4 - Weryfikacja (niepoprawne wartości)

Po wprowadzeniu poprawnych danych zostanie wyświetlony komunikat:



Rysunek 5 - Poprawne dane rejestracji

W przypadku zapomnienia hasła jest możliwość jego odzyskania przez wpisanie maila:



Rysunek 6 - Formularz odzyskiwania hasła

Widok zwykłego użytkownika po zalogowaniu:

The screenshot shows a user profile on the left with a green header 'Centrum zarządzania'. It displays the user's name 'baza' and ID '16'. Below the profile are two buttons: 'Wyloguj...' (Logout) in red and a placeholder button. On the right, there is a grid of tank cards. Each card contains a thumbnail image, the tank's name, and its origin. There are also 'Wiecej info' (More info) buttons for each tank. The tanks shown are: m1 abrams (USA), t32 (ZSRR), rudy 102 (Poland), bkan1 (Szwecja), and t52.

Rysunek 7 - Widok użytkownika

Widok admina (admin ma dodatkowa opcje widoczne po lewej stronie panelu):

The screenshot shows a user profile on the left with a green header 'Centrum zarządzania'. It displays the user's name 'admin' and ID '2'. Below the profile are several buttons: 'Użytkownicy', 'Nacie', 'Ammunica', 'Muzea', 'Dodaj nowa nacie', 'Dodaj czołg', and 'Wyloguj...'. On the right, the tank cards are identical to those in Figure 7, showing m1 abrams, t32, rudy 102, bkan1, and t52.

Rysunek 8 - Widok admina

Opcje po kliknięciu w wybrany czołg:

SZCZEGOLOWE INFORMACJE

Nazwa: m1 abrams

Typ czołgu cieżki

Kraj pochodzenia USA

Data 2021-05-06

Ilosc 1532

Opis nowoczesna bestia

Amunicja ap-1235

Opis amunicji uzywana to wiel rzeczu ktore sa...

Miejsce produkcji Chigaco

Zdjęcie



X

Edytuj Usun

Rysunek 9 - Szczegóły danego czołgu

Możliwość edycji danych:

Dodawanie nowego czołgu

Nazwa

Typ

Zdjęcie

Opis

Ilosc

Data

Nacja

Amunicja

Muzeum



X

Edytuj

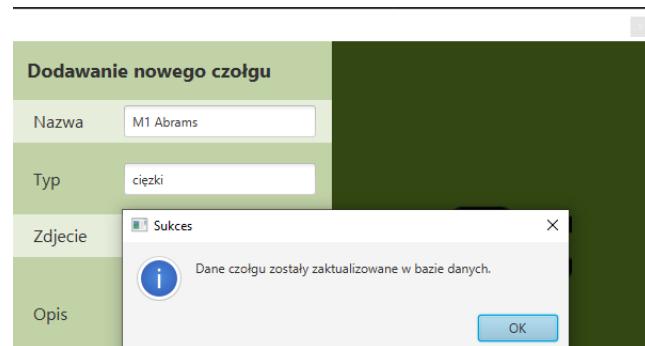
Rysunek 10 - Formularz edycji danych o czołgu

Walidacja danych przy edycji:



Rysunek 11 - Walidacja danych podczas edycji

Poprawne dane przy edycji czołgu:



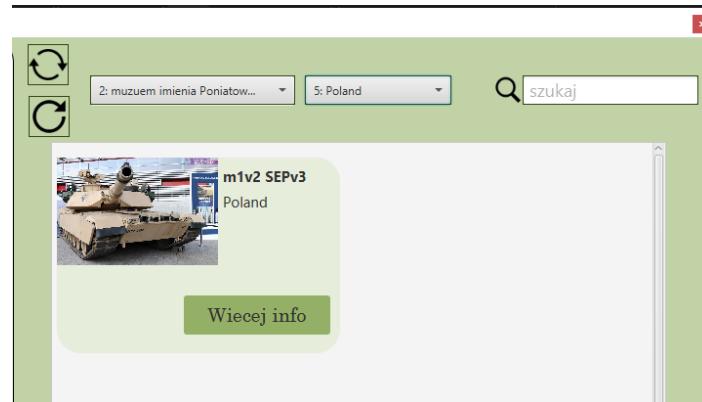
Rysunek 12 - Poprawne dane podczas edycji

Po wybraniu muzeum wyświetlane są tylko czołgi które się w nim znajdują:



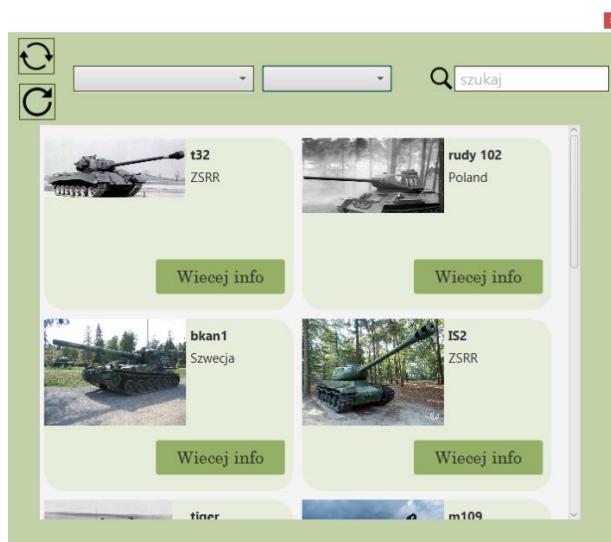
Rysunek 13 - Wybór muzeum

Mogliśmy także filtrować czołgi po nacji:



Rysunek 14 - Filtrowanie czołgów po nacji

Przyciskiem po lewej stronie możemy resetować wcześniej wybrane filtry:



Rysunek 15 - Resetowanie filtrowania

Mogliśmy użyć wyszukiwarki do znajdowania konkretnych czołgów w tym przypadku zawierające literę „l”



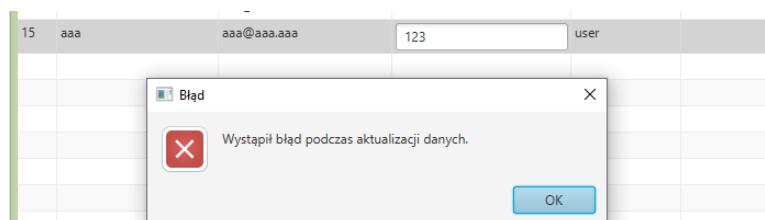
Rysunek 16 - Wyszukiwarka czołgów

Widok wszystkich użytkowników z możliwością ich edycji wraz z walidacją:

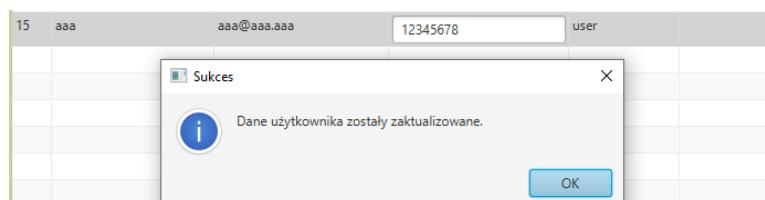
The screenshot shows a window titled "Lista użytkowników" (User List) with a search bar at the top right. Below the title is a user icon consisting of three stylized human figures. The main area contains a table with columns: ID, Login, Email, Password, and Typ konta (Account Type). The table has 6 rows of data. A modal dialog box is overlaid on the table, containing an error message: "Wystąpił błąd podczas aktualizacji danych." (An error occurred during data update).

ID	Login	Email	Password	Typ konta
2	admin	admin@example.com	admin	admin
4	xx	xx	xx	user
10	b	b@b.b	b	user
13	admin2	admin@example.com	adminpassword	admin
14	xxx	xxx@xxx.xxx	xxx	user
15	aaa	aaa@aaa.aaa	aaa	user

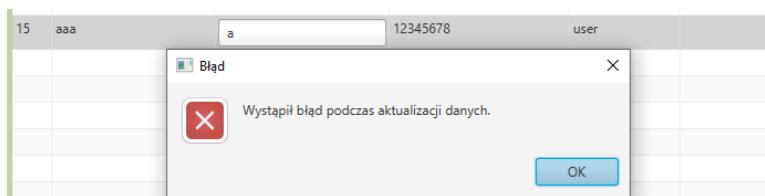
Rysunek 17 - Widok wszystkich użytkowników



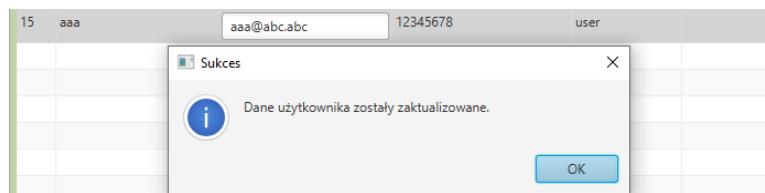
Rysunek 18 - Złe dane podczas próby edycji



Rysunek 19 - Poprawne dane podczas próby edycji



Rysunek 20 - Złe dane podczas próby edycji



Rysunek 21 - Poprawne dane podczas próby edycji

Mogliśmy wyszukiwać użytkowników:

The screenshot shows a Windows application window titled "Lista użytkowników". At the top right, there is a search bar containing the letter "a". Below the title, there is a small icon of two user profiles. The main area contains a table with the following data:

ID	Login	Email	Password	Typ konta
2	admin	admin@example.com	admin	admin
13	admin2	admin@example.com	adminpassword	admin
15	aaa	aaa@abc.abc	12345678	user

Rysunek 22 - Wyszukiwanie użytkowników

Widok wszystkichacji, który działa na tej samej zasadzie co lista użytkowników, tzn. wyszukiwarka oraz edycja wraz walidacją:

The screenshot shows a Windows application window titled "Lista nacji". At the top right, there is a search bar. Below the title, there is a small icon of two flags. The main area contains a table with the following data:

ID	Nazwa nacji	Miejsce produkcji
1	ZSRR	Stalingrad
3	USA	Chicago
4	England	London
5	Poland	Warsaw
6	Szwecja	Sztokholm
2	Germany	Berlin
7	Francja	Paryż

Rysunek 23 - Lista nacji i wyszukiwanie

Dodawanie nowego czołgu (walidacja):

Dodawanie nowego czołgu

Nazwa	nazwa
Pole nazwa czołgu nie może być puste!	
Typ	typ
Pole typ czołgu nie może być pustym!	
Zdjecie	Dodaj fotę
Opis	opis
Pole opis czołgu nie może być pustym!	
Ilosc	ilosz
Pole ilość czołgów nie może być pustym!	
Data	11-11-1999
Pole data czołgu nie może być pustym!	
Nacja	
Pole nacja czołgu nie może być pustym!	
Amunicja	
Pole amunicja czołgu nie może być pustym!	
Muzuem	
Pole muzuem czołgu nie może być pustym!	

Dodaj

Rysunek 24 - Walidacja formularza przy dodawaniu czołgu

W przypadku podania prawidłowych danych: Kiedy nie wybierzemy zdjęcia, aplikacja ustawi zdjęcie domyślne:

Dodawanie nowego czołgu

Nazwa	nazwa
Typ	typ
Zdjecie	Dodaj fotę
Opis	Sukces
Nowy czołg został dodany do bazy danych.	
Ilosc	
Data	11-11-1999
Nacja	1: ZSRR
Amunicja	2: apcr
Muzuem	3: muzuem 3 brygady

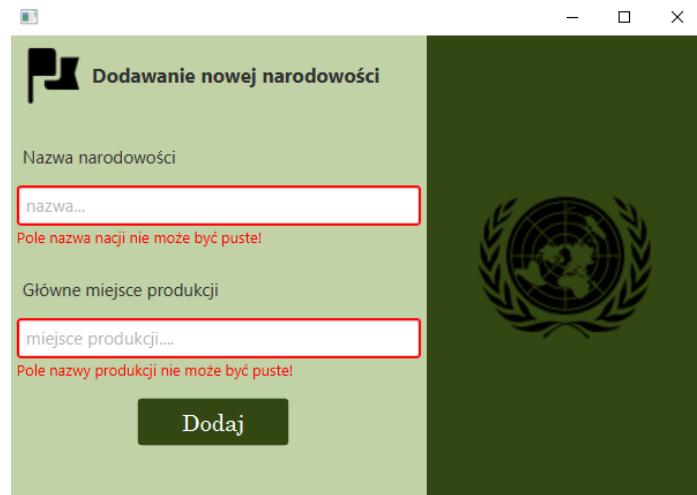
Dodaj

Rysunek 25 - Dodanie czołgu



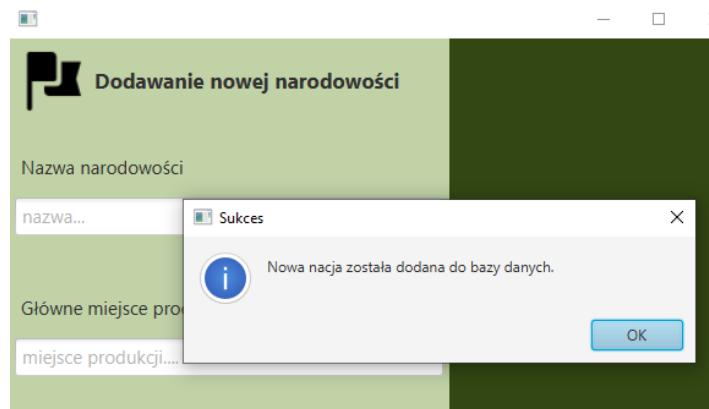
Rysunek 26 - Domyślne zdjęcie czołgu

Dodawanie nowej narodziny wraz z walidacją danych:



Rysunek 27 - Walidacja formularza przy dodawaniu nowej narodziny

W przypadku wprowadzenie poprawnych danych:

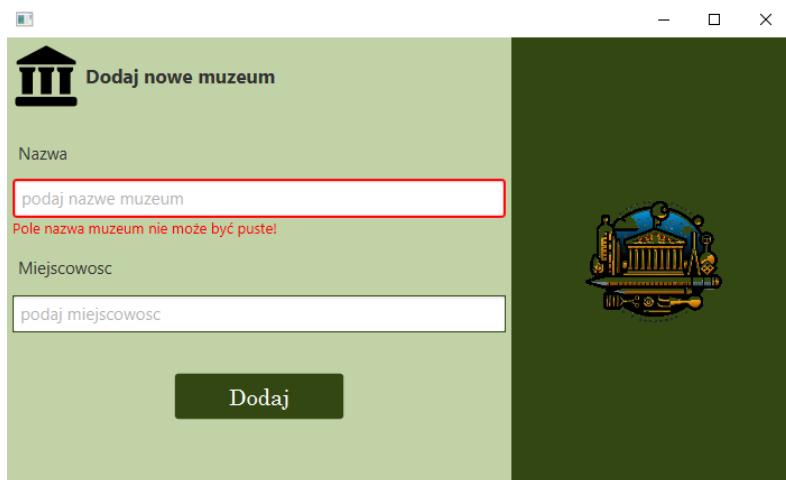


Rysunek 28 - Dodanie nowej narodziny

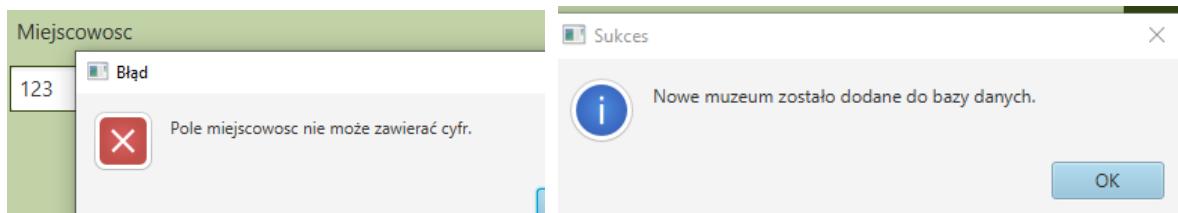
8	Indie	Taj Majal
---	-------	-----------

Rysunek 29 - Dodana wcześniej narodzina w liście

Dodawanie nowego muzeum wraz z walidacją: Miejscowość może być pusta wtedy aplikacja ustawi domyślną wartość na 'Warszawę':

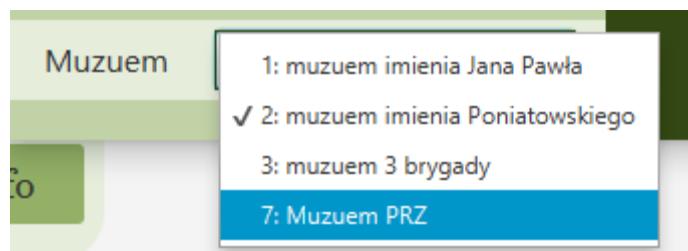


Rysunek 30 - Walidacja nazwy



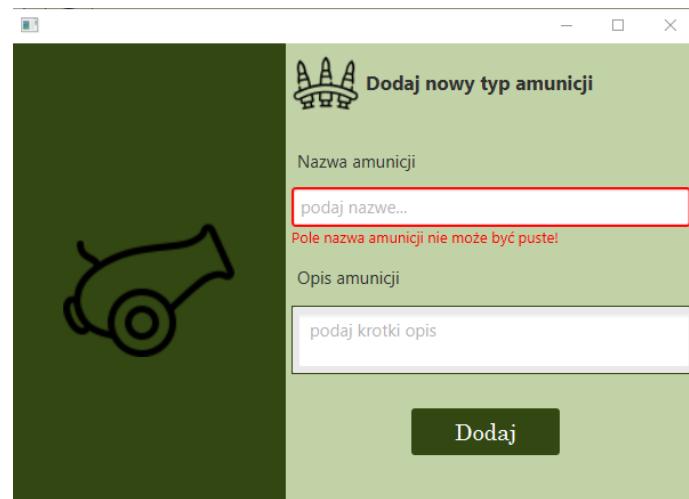
Rysunek 31 – Komunikat niepoprawnych danych

Rysunek 32 - Komunikat dodania muzeum

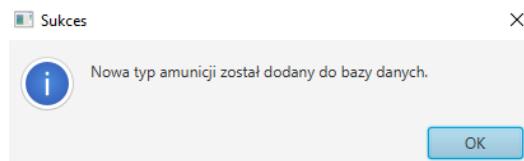


Rysunek 33 - Dowód dodania muzeum

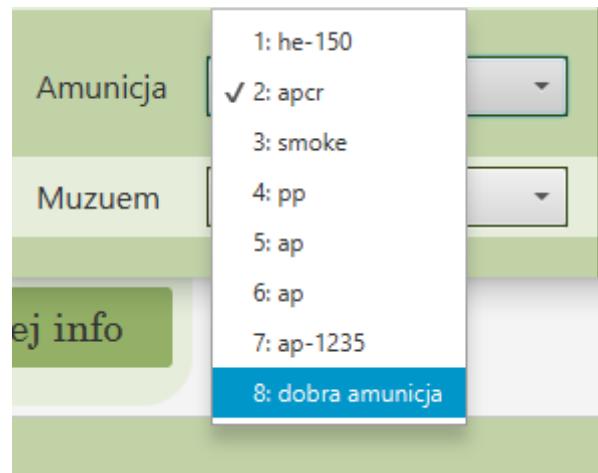
Dodawanie nowego rodzaju amunicji wraz z walidacją: Pole opis może być puste wtedy aplikacja ustawi domyślny opis



Rysunek 34 - Weryfikacja przy dodawaniu amunicji



Rysunek 35 - Komunikat przy poprawnym dodaniu amunicji



Rysunek 36 – Dowód

Edycja zdjęcia czołgu:

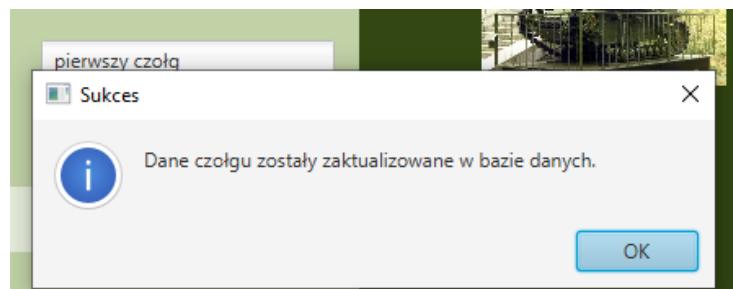
Dodawanie nowego czołgu

Nazwa	<input type="text" value="MS1"/>
Typ	<input type="text" value="traktor"/>
Zdjęcie	<input type="button" value="Dodaj foto"/>
Opis	<input type="text" value="pierwszy czołg"/>
Ilosc	<input type="text" value="0"/>
Data	<input type="text" value="26.05.2003"/> <input type="button" value=""/>
Nacja	<input type="text" value="1: ZSRR"/>

Edytuj



Rysunek 37 - Formularz edycji zdjęcia



Rysunek 38 - Komunikat



Rysunek 39- Nowe zdjęcie czołgu

Funkcje języka proceduralnego:

Funkcje:

- check_no_digits:

```
CREATE OR REPLACE FUNCTION check_no_digits(input_text TEXT)
RETURNS BOOLEAN AS $$

BEGIN RETURN NOT (regexp_match(input_text, '[0-9]') IS NOT NULL);

END;

$$ LANGUAGE plpgsql;
```

Opis: Funkcja sprawdzająca, czy w podanym tekście nie znajdują się żadne cyfry. Argumenty: input_text TEXT - tekst do sprawdzenia. Zwraca: BOOLEAN - wartość TRUE, jeśli w tekście nie ma cyfr, w przeciwnym razie FALSE.

- is_valid_email

```
CREATE OR REPLACE FUNCTION is_valid_email(email TEXT)
RETURNS BOOLEAN AS $$

BEGIN RETURN email ~* '^[A-Za-z0-9.%+-]+@[A-Za-z0-9.-]+.[A-Z|a-z]{2,}$';

END; $$ LANGUAGE plpgsql;
```

Opis: Funkcja sprawdzająca poprawność adresu email. Argumenty: email TEXT - adres email do sprawdzenia. Zwraca: BOOLEAN - wartość TRUE, jeśli email jest poprawny, w przeciwnym razie FALSE

- is_valid_password

```
CREATE OR REPLACE FUNCTION is_valid_password(password TEXT)
RETURNS BOOLEAN AS $$ BEGIN RETURN LENGTH(password) >= 8;

END;

$$ LANGUAGE plpgsql;
```

Opis: Funkcja sprawdzająca, czy hasło spełnia określone wymagania (długość, obecność cyfr, małych i wielkich liter oraz znaków specjalnych). Argumenty: password TEXT - hasło do sprawdzenia. Zwraca: BOOLEAN - wartość TRUE jeśli hasło spełnia wymagania, w przeciwnym razie FALSE

-is_valid_acc_type

```
CREATE OR REPLACE FUNCTION is_valid_acc_type(acc_type VARCHAR)
RETURNS BOOLEAN AS $$

BEGIN RETURN acc_type = 'admin'
OR acc_type = 'user';

END; $$

LANGUAGE plpgsql;
```

Opis: Ta funkcja is_valid_acc_type sprawdza, czy podany typ konta użytkownika jest prawidłowy. Przyjmuje jeden argument acc_type typu VARCHAR, który reprezentuje typ konta. Funkcja zwraca wartość logiczną TRUE, jeśli typ konta jest równy "admin" lub "user", a w przeciwnym razie zwraca FALSE.

Procedury:

- add_ammunition

```
CREATE OR REPLACE PROCEDURE add_ammunition( p_name VARCHAR, p_description VARCHAR )
LANGUAGE plpgsql AS $$

BEGIN INSERT INTO ammunition(name, description) VALUES (p_name, p_description);

END; $$;
```

Opis: Procedura add_ammunition służy do dodawania nowej amunicji do bazy danych. Procedura przyjmuje dwa argumenty: nazwę amunicji (p_name) oraz jej opis (p_description). Następnie wstawia te dane do tabeli ammunition. Argumenty: p_name (VARCHAR): Nazwa amunicji, która ma być dodana do bazy danych. p_description (VARCHAR): Opis amunicji, który ma być dodany do bazy danych. Działanie: Procedura wykonuje operację INSERT, aby dodać nowy rekord do tabeli ammunition z podanymi wartościami nazwy i opisu.

- add_museum

```

CREATE OR REPLACE PROCEDURE add_museum( p_name VARCHAR, p_localization VARCHAR )
LANGUAGE plpgsql AS $$

BEGIN INSERT INTO museum(name, localization) VALUES (p_name, p_localization);

END; $$;

```

Opis: Procedura add_museum dodaje nowe muzeum do bazy danych. Przyjmuje dwa argumenty: nazwę muzeum i jego lokalizację, które są następnie wstawiane do tabeli museum. Argumenty: p_name VARCHAR - nazwa muzeum. p_localization VARCHAR - lokalizacja muzeum. Działanie: Procedura wstawia nowy rekord do tabeli museum z podanymi wartościami dla kolumn name i localization.

- **add_nation**

```

CREATE OR REPLACE PROCEDURE add_nation( p_nation_name VARCHAR, p_prod_place VARCHAR )
LANGUAGE plpgsql AS $$ BEGIN INSERT INTO nationality(nation_name, prod_place) VALUES
(p_nation_name, p_prod_place); END; $$;

```

Opis: Procedura add_nation służy do dodawania nowej nacji do bazy danych. Procedura przyjmuje dwa argumenty: nazwę nacji (p_nation_name) oraz miejsce produkcji (p_prod_place). Następnie wstawi te dane do tabeli nationality. Argumenty: p_nation_name (VARCHAR): Nazwa nacji, która ma być dodana do bazy danych. p_prod_place (VARCHAR): Miejsce produkcji związane z nacją, które ma być dodane do bazy danych. Działanie: Procedura wykonuje operację INSERT, aby dodać nowy rekord do tabeli nationality z podanymi wartościami nazwy nacji i miejsca produkcji.

Triggery:

- **check_input_trigger_function**

```
CREATE OR REPLACE FUNCTION check_input_trigger_function()
```

```
RETURNS TRIGGER AS $$
```

```

BEGIN IF NEW.nation_name IS NOT NULL AND NOT check_no_digits(NEW.nation_name::TEXT) THEN
RAISE EXCEPTION 'Nazwa nacji nie może zawierać cyfr.'; END IF; IF NEW.prod_place IS NOT NULL AND
NOT check_no_digits(NEW.prod_place::TEXT) THEN RAISE EXCEPTION 'Miejsce produkcji nie może
zawierać cyfr.';

END IF; RETURN NEW;

END; $$ LANGUAGE plpgsql;
```

Opis: Funkcja check_input_trigger_function jest funkcją wyzwalającą (trigger function), która służy do walidacji danych wprowadzanych do tabeli nationality. Jej głównym celem jest zapewnienie, że pola nation_name i prod_place nie zawierają cyfr. Jeśli którakolwiek z tych kolumn zawiera cyfry, funkcja zgłasza wyjątek i zapobiega wstawieniu lub aktualizacji danych. Działanie: Funkcja sprawdza, czy kolumna nation_name w nowym rekordzie (NEW) jest różna od NULL i czy nie zawiera cyfr, korzystając z funkcji check_no_digits. Jeśli nation_name zawiera cyfry, funkcja zgłasza wyjątek z komunikatem "Nazwa nacji nie może zawierać cyfr." Podobnie, funkcja sprawdza, czy kolumna prod_place w nowym rekordzie (NEW) jest różna od NULL i czy nie zawiera cyfr, korzystając z funkcji check_no_digits. Jeśli prod_place zawiera cyfry, funkcja zgłasza wyjątek z komunikatem "Miejsce produkcji nie może zawierać cyfr." Jeśli oba pola są prawidłowe, funkcja zwraca nowy rekord (NEW), pozwalając na wstawienie lub aktualizację danych w tabeli nationality.

- set_default_acc_type

```
CREATE OR REPLACE FUNCTION set_default_acc_type()

RETURNS TRIGGER AS $$

BEGIN NEW.acc_type := 'user';

RETURN NEW;

END;

$$ LANGUAGE plpgsql;
```

Opis: Funkcja set_default_acc_type jest funkcją wyzwalającą (trigger function), której celem jest automatyczne ustawienie domyślnej wartości kolumny acc_type na 'user' dla nowych rekordów wstawianych do tabeli. Jest używana w kontekście wyzwalacza, który uruchamia się przed operacją INSERT na określonej tabeli. Działanie: Funkcja przypisuje wartość 'user' do kolumny acc_type nowego rekordu (NEW). Funkcja zwraca zmodyfikowany nowy rekord (NEW), co pozwala na dalsze przetwarzanie rekordu przez operację INSERT.

- set_default_description

```
CREATE OR REPLACE FUNCTION set_default_description()

RETURNS TRIGGER AS $$

BEGIN IF NEW.description IS NULL OR NEW.description = "" THEN NEW.description := 'opis zostanie
jeszcze dodany';

END IF; RETURN NEW;

END; $$ LANGUAGE plpgsql;
```

Opis: Funkcja set_default_description ma na celu automatyczne ustawianie wartości domyślnej dla kolumny description w przypadku, gdy nowo wstawiany rekord nie zawiera wartości dla tej kolumny lub gdy wartość ta jest pusta.

- **set_default_localization**

```
CREATE OR REPLACE FUNCTION set_default_localization()
RETURNS TRIGGER AS $$ BEGIN IF NEW.localization IS NULL OR NEW.localization = "" THEN
NEW.localization := 'Warszawa';
END IF;
RETURN NEW;
END; $$ LANGUAGE plpgsql;
```

Opis: Funkcja set_default_localization ma na celu automatyczne ustawianie wartości domyślnej dla kolumny localization w przypadku, gdy nowo wstawiany rekord nie zawiera wartości dla tej kolumny lub gdy wartość ta jest pusta.

- **set_negative_amount_to_zero**

```
CREATE OR REPLACE FUNCTION set_negative_amount_to_zero()
RETURNS TRIGGER AS $$ BEGIN IF NEW.amount < 0 THEN NEW.amount := 0;
END IF;
RETURN NEW; END; $$ LANGUAGE plpgsql;
```

Do tej funkcji utworzone są trigger'y: przed dodaniem nowego czołgu oraz podczas edycji:

```
CREATE TRIGGER set_negative_amount_to_zero_before_insert BEFORE INSERT ON tanks FOR EACH
ROW EXECUTE FUNCTION set_negative_amount_to_zero();
CREATE TRIGGER set_negative_amount_to_zero_before_update BEFORE UPDATE ON tanks FOR EACH
ROW EXECUTE FUNCTION set_negative_amount_to_zero();
```

Opis: Funkcja set_negative_amount_to_zero ma na celu zapobieganie ustawieniu ujemnej wartości dla kolumny amount w tabeli tanks. Jeśli nowa wartość dla tej kolumny jest ujemna, zostanie automatycznie zmieniona na zero.

- **validate_user_update**

```
CREATE OR REPLACE FUNCTION validate_user_update()
RETURNS TRIGGER AS $$

BEGIN IF NEW.user_email IS NOT NULL AND NOT is_valid_email(NEW.user_email) THEN RAISE
EXCEPTION 'Niepoprawny adres e-mail: %', NEW.user_email;

END IF; IF NEW.user_pass IS NOT NULL AND NOT is_valid_password(NEW.user_pass) THEN RAISE
EXCEPTION 'Hasło musi mieć co najmniej 8 znaków';

END IF; RETURN NEW; END; $$ LANGUAGE plpgsql;
```

Do tej funkcji utworzony jest trigger, który wywoła funkcje przed dodaniem nowego użytkownika albo przed jego edycją:

```
CREATE TRIGGER check_input_trigger BEFORE INSERT OR UPDATE ON nationality FOR EACH ROW
EXECUTE FUNCTION check_input_trigger_function();
```

Opis: Funkcja validate_user_update ma na celu sprawdzenie poprawności aktualizacji danych użytkownika w tabeli users. Sprawdza, czy nowy adres e-mail jest w prawidłowym formacie oraz czy nowe hasło spełnia minimalne wymagania dotyczące długości.

- validate_and_set_acc_type

```
CREATE OR REPLACE FUNCTION validate_and_set_acc_type()
RETURNS TRIGGER AS $$ BEGIN IF NEW.acc_type IS NULL
OR NOT is_valid_acc_type(NEW.acc_type)
THEN NEW.acc_type := 'user';
END IF;
RETURN NEW;
END; $$ LANGUAGE plpgsql;
```

Do tej funkcji utworzony jest trigger, który wywoła te funkcje przed każdą operacją update

```
CREATE TRIGGER validate_and_set_acc_type_before_update
BEFORE UPDATE ON users FOR EACH ROW
EXECUTE FUNCTION validate_and_set_acc_type()
```

Opis: Ta funkcja validate_and_set_acc_type służy do walidacji i ustawienia domyślnego typu konta użytkownika w przypadku, gdy podany typ jest nieprawidłowy lub nullowy. Funkcja ta jest często

używana w kontekście tworzenia nowych rekordów użytkowników lub aktualizacji istniejących rekordów, aby zapewnić spójność danych w bazie danych.

Sekwencje są użytkowane przy inkrementowaniu kluczy głównych.

```
CREATE SEQUENCE IF NOT EXISTS public.ammunition_ammo_id_seq
    INCREMENT 1
    START 1
    MINVALUE 1
    MAXVALUE 2147483647
    CACHE 1
    OWNED BY ammunition.ammo_id;
```

Źródła:

Ikony pobrane ze strony: <https://icons8.com/>

Grafika muzeum utworzona przez: <https://copilot.microsoft.com/>

Dynamiczna siatka z czołgami: <https://www.youtube.com/watch?v=L3PLDAZWU9s>

Drugi materiał dotyczący siatki: <https://www.youtube.com/watch?v=XIAzQ170kzM>

Wzór: <https://www.youtube.com/watch?v=Aliw3lNRzfc>

Dodawanie zdjęć do bazy: <https://www.tutorialspoint.com/how-to-insert-an-image-in-to-mysql-database-using-java-program>

Również dodawanie: <https://stackoverflow.com/questions/30447539/uploading-image-to-a-mysql-database-using-java>