

Zaawansowane aplikacje internetowe - dokumentacja projektu 1

Łukasz Jezapkowicz - 329121

6 listopada 2023

Spis treści

1	Wstęp	3
2	Konfiguracja	3
3	Hosting	4
4	Baza danych	4
5	Kod źródłowy	5
6	Funkcjonalność	6

1 Wstęp

Celem tego projektu było opracowanie prostego systemu "pamiętnikowego" opartego na języku programowania PHP, technologiach webowych (HTML5, CSS3) oraz dostępnych bibliotekach i frameworkach. System został stworzony, aby umożliwić użytkownikom zapisywanie i śledzenie konkretnych wydarzeń, prezentując je w sposób przejrzysty, estetyczny i łatwo dostępny. Stworzony system jest przeznaczony do śledzenia nadchodzących premier filmowych.

Projekt jest oparty na Laravel, frameworku do tworzenia aplikacji internetowych, napisanym w języku PHP, który stosuje wzorzec architektoniczny Model-View-Controller (MVC). Jako system zarządzania bazami danych wybrano MySQL w wersji 8.0. Jako hosting zastosowano maszynę wirtualną na platformie Google Cloud Platform (GCP).

2 Konfiguracja

Aplikacja została stworzona przy pomocy frameworku Laravel, napisanego w języku PHP. W celu uruchomienia aplikacji lokalnie należy zainstalować:

- język programowania PHP, w wersji $\geqslant 8.2$,
- Composer, menedżer zależności dla języka PHP,
- MySQL, system do zarządzania bazami danych, w wersji $\geqslant 8.0$.

Następnie należyściągnąć kod źródłowy stworzonej aplikacji, dostępny na platformie LeIA. Kod jest również dostępny na platformie [GitHub](#), skąd można go bezpośredniościągnąć. Można go również pobrać przy pomocy komendy (wymagany zainstalowany *git*).

```
1 $ git clone https://github.com/LucasJezap/timeline-project-pw.git
```

Kolejnym krokiem jest stworzenie oraz połączenie aplikacji z bazą danych. Na platformie LeIA (oraz GitHub) znajduje się plik *db_create_query.sql*, który pozwala stworzyć przykładową bazę danych *timeline*, wymaganą do poprawnego działania systemu. Alternatywnie można użyć komendy:

```
1 $ php artisan migrate
```

W przypadku użycia komendy tabeli w bazie danych będą puste, zalecane jest użycie *db_create_query.sql*. By nawiązać połączenie w aplikacji ze stworzoną bazą danych należy odpowiednio uzupełnić plik *.env*. Przykładowa konfiguracja:

```
1 DB_CONNECTION=mysql
2 DB_HOST=127.0.0.1
3 DB_PORT=3308
4 DB_DATABASE=timeline
5 DB_USERNAME=root
6 DB_PASSWORD=root
```

Do poprawnego działania aplikacji należy również stworzyć link symboliczny, pozwalający na poprawne wyświetlanie grafik na stronie. W tym celu należy wykonać komendę:

```
1 $ rm public/storage
2 $ php artisan storage:link
```

Taki link symboliczny został już stworzony w aplikacji, dlatego należy go odświeżyć. Ostatnim krokiem jest uruchomienia serwera PHP, przy pomocy komendy:

```
1 $ php artisan serve --port=8080
```

Jako port można wybrać dowolny dostępny port TCP. Dla portu 8080 aplikacja będzie dostępna pod linkiem:

1 <http://localhost:8080/>

W załączonej bazie danych zostało stworzone konto admina:

1 bossgame32@gmail.com
2 Avatar3232

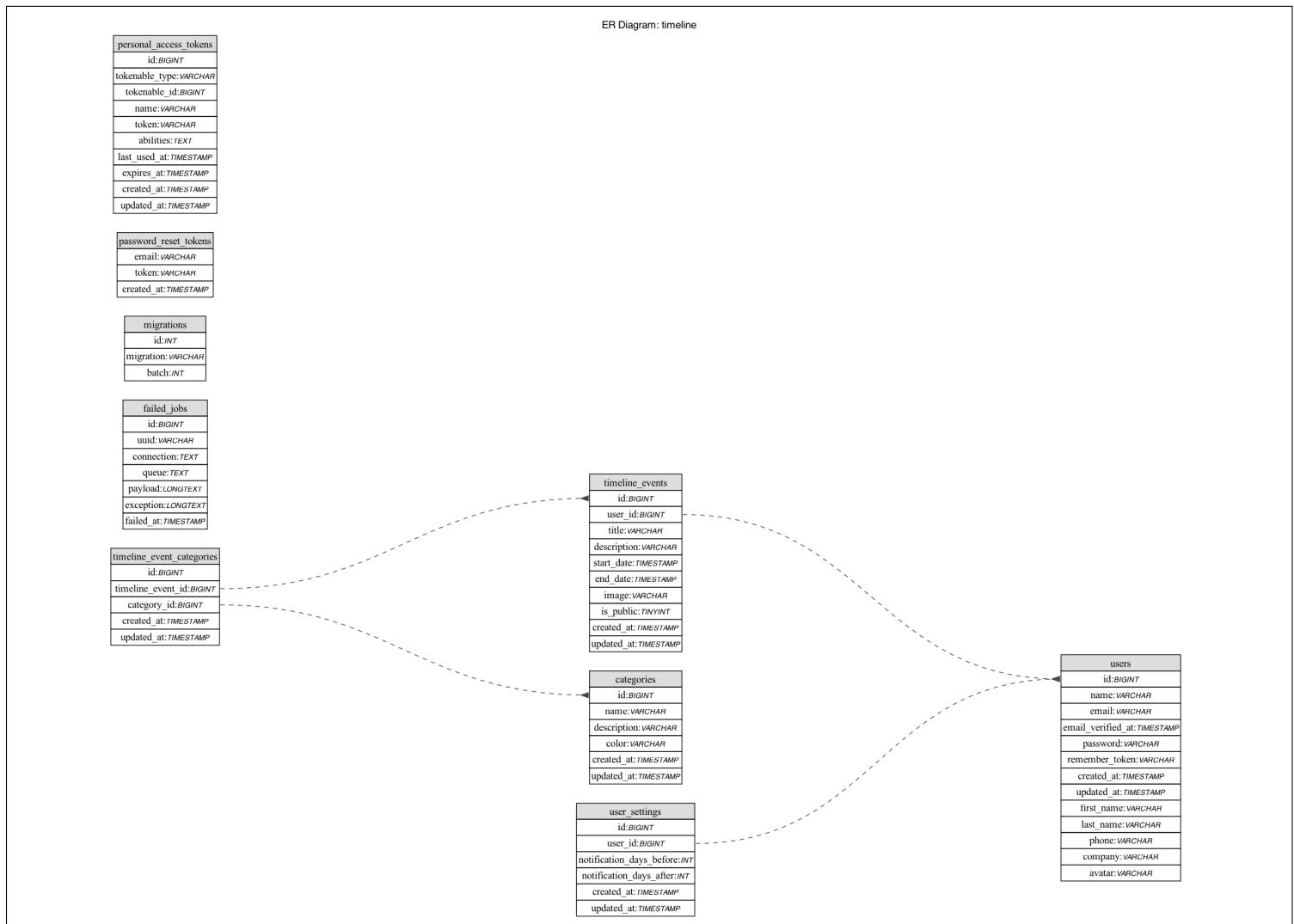
3 Hosting

Strona jest również dostępna publicznie pod linkiem [Timeline Project](#). Hosting zapewniony jest przy pomocy maszyny wirtualnej w Google Cloud Platform. Wybrana maszyna jest jedną ze słabszych dlatego mogą występować małe opóźnienia. W celu zmniejszenia kosztów maszyna nie ma zapewnionego stałego adresu IP, w momencie restartu maszyny może ona otrzymać nowy adres IP (taka sytuacja nie powinna nastąpić, ponieważ maszyny nie重启ują się z byle powodu).

W razie problemów proszę o kontakt.

4 Baza danych

Poniżej zaprezentowany jest diagram ERD bazy danych, stworzony przy pomocy *graphviz*.



Rysunek 1: Diagram ERD bazy danych Timeline

Baza składa się z 9 tabel, ich pola oraz zależności widoczne są na Rysunku 1. Przykładowy opis tabel *timeline_events*, *categories* oraz *timeline_event_categories*:

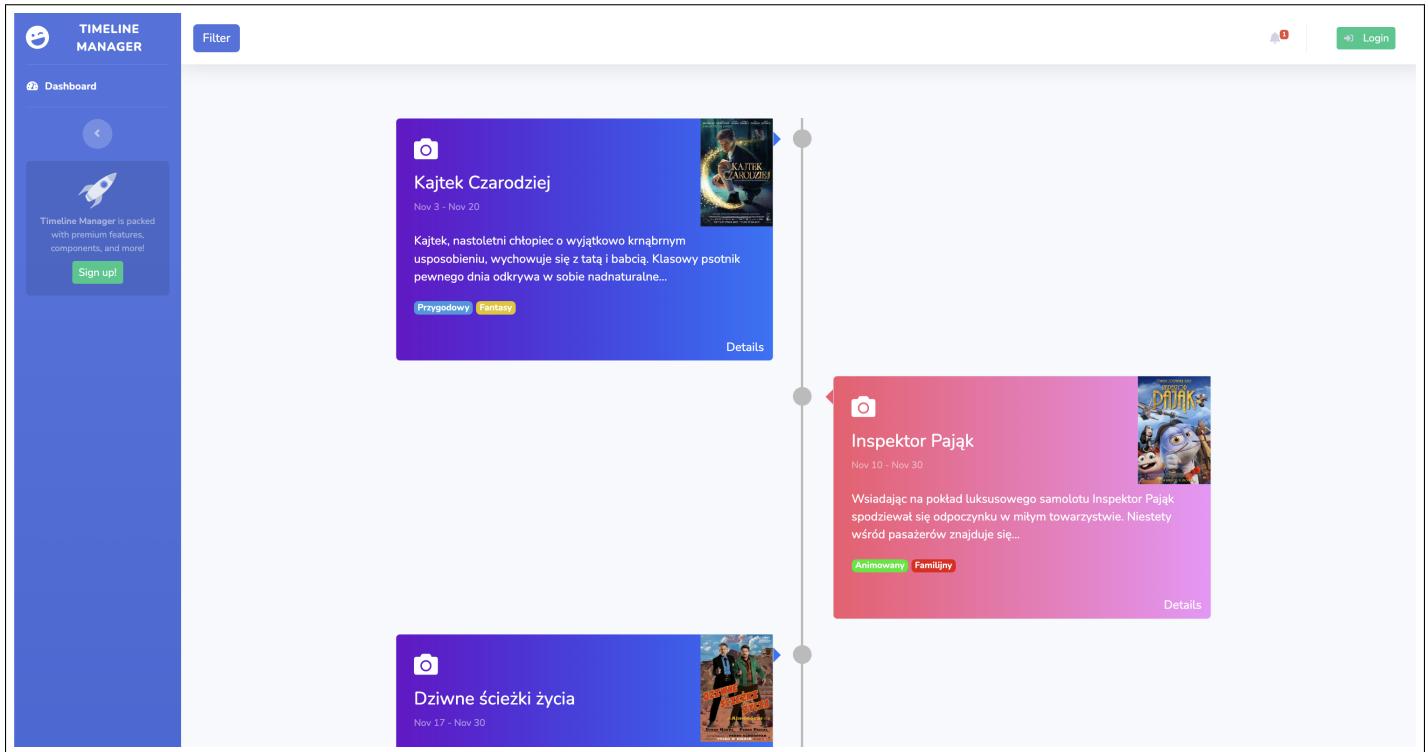
- Tabela *timeline_events* składa się z pola porządkującego *ID*, klucza obcego *user_id* wskazującego użytkownika tworzącego zdarzenie, pól opisujących zdarzenie *title*, *description*, *start_date*, *end_date*, *image*, *is_public* oraz pól śledzących czas zmian *created_at* oraz *updated_at*.
- Tabela *categories* składa się z pola porządkującego *ID*, pól opisujących kategorię *name*, *description*, *color* oraz pól śledzących czas zmian *created_at* oraz *updated_at*.
- Tabela *timeline_event_categories* składa się z pola porządkującego *ID*, kluczy obcych *timeline_event_id*, *category_id*, oraz pól śledzących czas zmian *created_at* oraz *updated_at*.
- Celem tabeli *timeline_event_categories* jest łączenie zdarzeń z konkretnymi kategoriami, przy pomocy dwóch kluczy obcych. Takie rozwiązanie pozwala na proste przypisanie wielu kategorii do jednego zdarzenia.
- Wszystkie opisane klucze obce stworzone są z dodatkową zasadą ON DELETE CASCADE - podczas usunięcia referencji usuwane są również wpisy z kluczami obcymi. Takie rozwiązanie pozwala zachować integralność bazy danych.

5 Kod źródłowy

Poniżej znajduje się krótki opis najważniejszych elementów kodu źródłowego:

- folder *app* - zawiera logikę programu (serwer/backend),
 - folder *app/Http* - zawiera logikę requestów Http
 - * *app/Http/Controllers* - zawiera logikę kontrolerów, sterujących aplikacją,
 - * *app/Http/Livewire* - zawiera logikę wyświetlania tabelek,
 - * *app/Http/Requests* - zawiera logikę walidacji requestów Http,
 - folder *app/Models* - zawiera opis encji bazy danych,
 - folder *app/Traits* - zawiera logikę wrzucania plików/zdjęć,
- folder *database/migrations* - zawiera opis migracji (tworzenia/edykcji) bazy danych,
- folder *public* - zawiera wszystkie zasoby strony, w tym pliki *CSS*, pliki *JavaScript*, grafiki/zdjęcia,
- folder *resources/views* - zawiera wszystkie pliki *blade.php*, szablony/widoki zawierające faktyczny kod strony,
- plik *routes/web.php* - zawiera całą logikę kierowania (routowania) requestów w aplikacji, łączy szablony/widoki z kontrolerami.

6 Funkcjonalność



Rysunek 2: Dashboard, dla niezalogowanego użytkownika

The screenshot shows a form for creating a new event. On the left is a movie poster for 'Kajtek Czarodziej'. To the right are input fields: 'Title' (filled with 'Kajtek Czarodziej'), 'Description' (containing a brief plot summary), 'Start Date' (set to '03/11/2023, 00:00'), 'End Date' (set to '20/11/2023, 00:00'), and 'Categories' (listing 'Przygodowy' and 'Fantasy').

Rysunek 3: Widok zdarzenia, dla niezalogowanego użytkownika

Name

Description

Gatunek filmowy obejmujący obrazy filmowe, w fabule, których występują motywy związane z magią, nadprzyrodzonością, mitologią, wymyślonym folklorem czy też istnieniem fantastycznych światów

Color

Rysunek 4: Widok kategorii, dla niezalogowanego użytkownika

TIMELINE

Dashboard

Timeline M... with some components

Filter Data

Title

Category

- Dramat
- Western
- Przygodowy
- Fantasy

Description

Start Date

End Date

Close

Rysunek 5: Widok filtrowania zdarzeń

Sign In

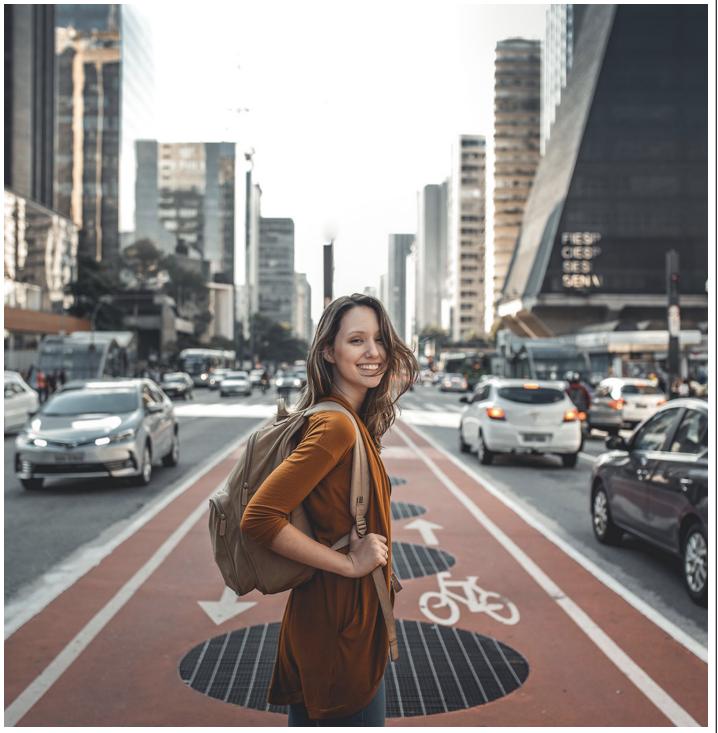
Email

Password

Remember me [Forgot Password](#)

Log In

[Not on Timeline yet? Sign up](#)



Rysunek 6: Widok logowania

Register

Name

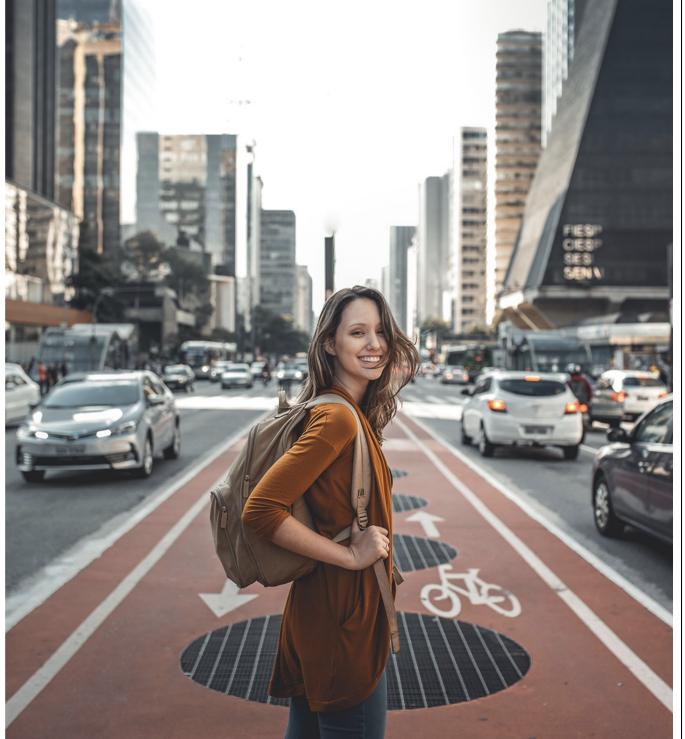
Email

Password

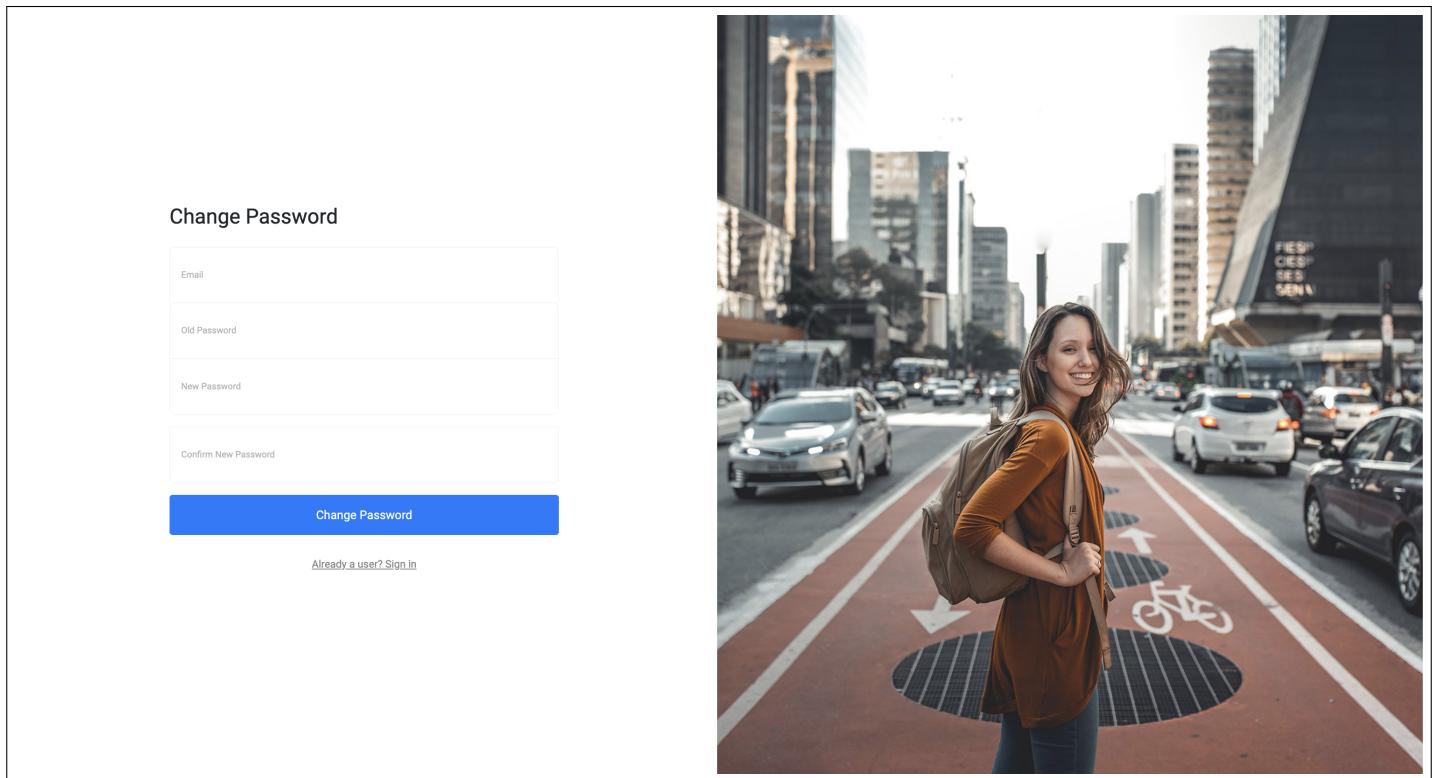
Confirm Password

Register

[Already a user? Sign in](#)



Rysunek 7: Widok rejestracji



Rysunek 8: Widok zmiany hasła (zapomniane hasło)

Rysunek 9: Dashboard, dla zalogowanego użytkownika



Poster
Accepted file type .png.

Title

Description

Kajtek, nastoletni chłopiec o wyjątkowo krynickim usposobieniu, wychowuje się z tatą i babcią. Klasowy psotnik pewnego dnia odkrywa w sobie nadnaturalne umiejętności. Czarodziejskie moce przejmują kontrolę nad życiem Kajtka, który od teraz rozprawia się

Start Date

End Date

Categories

Dramat
Western
Przygodowy
Fantasy

Is Public

✓

Rysunek 10: Widok zdarzenia, dla zalogowanego użytkownika

Name

Description

Film o akcji skonstruowanej z ciągu przygód protagonisty, charakteryzujący się szybkim tempem akcji, brawurowymi rozwiązaniami fabularnymi oraz licznymi przeszkodami pokonywanymi przez bohaterów. Schemat filmów przygodowych opiera się na licznych scenach pościgów, ucieczek, eksplozji oraz niebezpiecznych dla protagonyty wydarzeniach.

Color

Rysunek 11: Widok kategorii, dla zalogowanego użytkownika



The screenshot shows a user interface for creating a new event. On the left, there is a placeholder image for a movie poster with the text "NO POSTER FOUND" below it. To the right, there are several input fields and buttons:

- Title:** An input field for entering the event title.
- Description:** A text area for entering a detailed description of the event.
- Start Date:** An input field for selecting the start date, with a date picker icon to the right.
- End Date:** An input field for selecting the end date, with a date picker icon to the right.
- Categories:** A dropdown menu containing the following categories:
 - Dramat
 - Western
 - Przygodowy
 - Fantasy
- Is Public:** A checkbox for indicating if the event is public.
- Add Event:** A teal-colored button at the bottom right for saving the event.

Rysunek 12: Widok tworzenia nowego zdarzenia, dla zalogowanego użytkownika

The screenshot shows a user interface for creating a new category. It includes the following fields:

- Name:** An input field for the category name.
- Description:** A text area for a detailed description of the category.
- Color:** A color picker or selection bar showing a solid black color.
- Add Category:** A teal-colored button at the bottom right for saving the category.

Rysunek 13: Widok tworzenia nowej kategorii, dla zalogowanego użytkownika

A screenshot of a web-based application interface. At the top, there is a search bar with a magnifying glass icon. Below the search bar is a table header with columns: ID, Name, Email, Password (Hash), Created, and Updated. The table contains one row of data:

ID	Name	Email	Password (Hash)	Created	Updated
1	Łukasz	bossgame32@gmail.com	\$2y\$10\$ZfFa7.4SPHuPEXZgBQSmHOO4r52i7sqV0qcnif9UZwpSgX1S33nyC	2023-11-05 19:24:44	2023-11-05 20:45:32

Rysunek 14: Widok tabel, dla zalogowanego użytkownika

A screenshot of a user profile edit form titled "Profile". The form includes fields for profile photo, username, email address, first name, last name, phone number, and company, each with a corresponding input field. There are also "Upload Image" and "Save Avatar" buttons. A "Save Changes" button is located at the bottom of the form.

Profile Photo	Accepted file type .png.
<input type="button" value="Upload Image"/>	<input type="button" value="Save Avatar"/>
Username <input type="text" value="Łukasz"/>	Email Address <input type="text" value="bossgame32@gmail.com"/>
First Name <input type="text"/>	Last Name <input type="text"/>
Phone Number <input type="text"/>	Company <input type="text"/>
<input type="button" value="Save Changes"/>	

Rysunek 15: Widok profilu, dla zalogowanego użytkownika

Account settings

Control the amount of alerts you see

Alert Days Before
Alert Days After

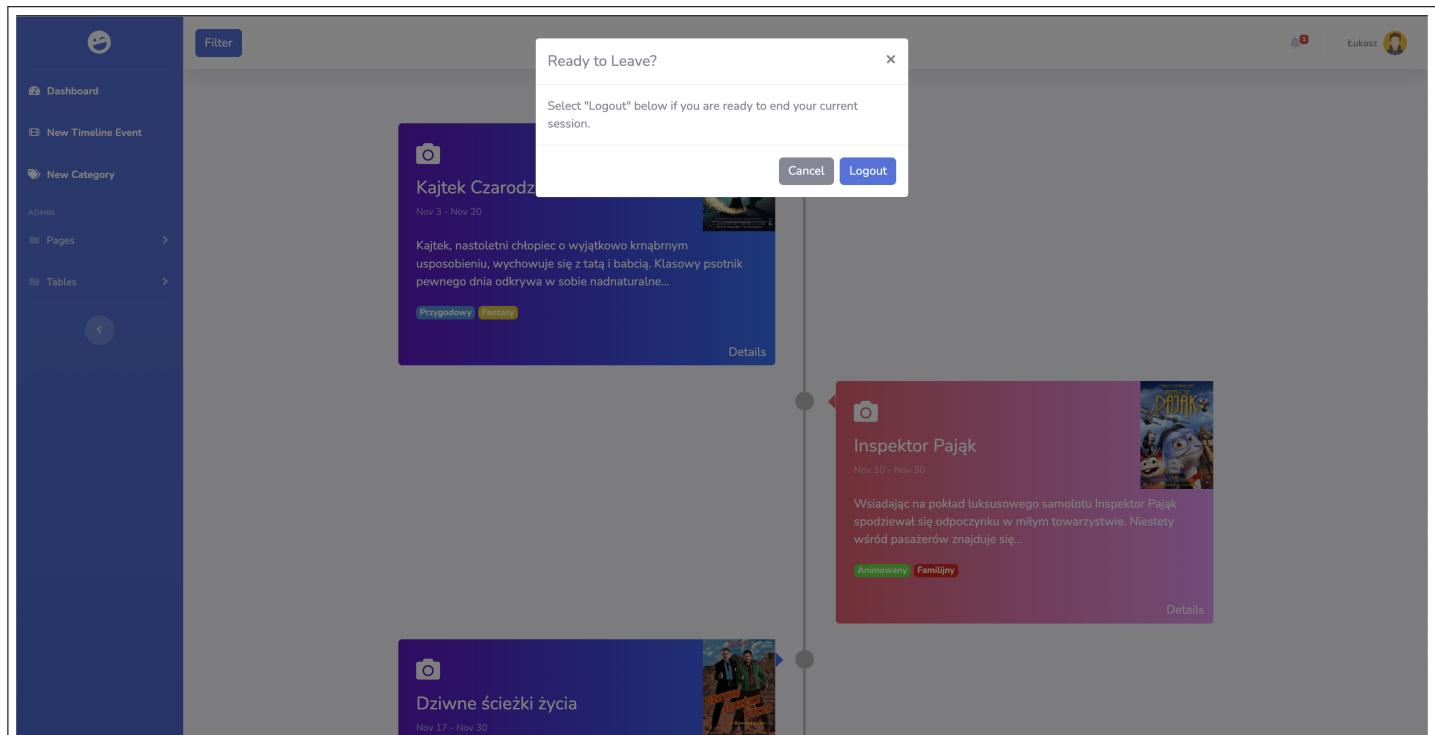
Change password

Old Password

New Password

New Password Confirmation

Rysunek 16: Widok ustawień, dla zalogowanego użytkownika



Rysunek 17: Możliwość wylogowania, dla zalogowanego użytkownika