

Dokumentacja Projektu

Autorzy: Szymon Godzwon, Maciek Latawiec

1. Wstęp

1.1. Cel projektu

Celem projektu jest stworzenie dynamicznej witryny internetowej umożliwiającej użytkownikom przeglądanie i możliwość brania udziału w festiwalach poprzez intuicyjny interfejs. Projekt zakłada implementację systemu logowania, panelu administracyjnego, a także integrację z bazą danych MySQL. Witryna ma również zawierać elementy interaktywne stworzone z użyciem JavaScript oraz atrakcyjną i responsywną warstwę wizualną opartą na HTML i CSS.

1.2. Zakres projektu

Projekt obejmuje:

- stworzenie kilku podstron z różnorodną treścią,
- integrację z bazą danych,
- obsługę logowania, rejestracji oraz panelu administratora,
- implementację funkcjonalności dynamicznych w języku PHP i JavaScript,
- przygotowanie dokumentacji technicznej zawierającej opis struktury projektu, kod źródłowy oraz schemat bazy danych.

1.3. Użytkownicy systemu

System przewidziany jest dla dwóch głównych grup użytkowników:

- Użytkownik końcowy – ma możliwość przeglądania treści, korzystania z formularza kontaktowego oraz (po rejestracji) dostępu do dodatkowych funkcji,
- Administrator – posiada dostęp do panelu administracyjnego, gdzie może zarządzać zawartością witryny (np. dodawać, edytować i usuwać wpisy).

2. Opis ogólny

2.1. Przeznaczenie witryny

Witryna przeznaczona jest do możliwości zapoznania się z naszymi usługami, poznania naszych pracowników oraz uczestników, zrzeszanie grupy ludzi uczestniczących w festiwalach w całej Polsce.

2.2. Główne funkcjonalności

Możliwość kupna biletów na dane festiwale, oglądanie zdjęć z poprzednich festiwali, kontakt z naszymi pracownikami, oferujemy również proponowane noclegi oraz transport

2.3. Wymagania techniczne

Xampp, przeglądarka i uruchomiona baza danych MySQL.

2.4. Struktura plików i folderów

Pliki podzielone są w folderach- osobno. JavaScript, php, css oraz baza osobno która znajduje się poza folderem, znajduje się również folder zdjęcia w którym zawarte są wszystkie zdjęcia występujące na stronie.

3.Struktura

3.1. Struktura wszystkich stron

Wszystkie strony zostały zapisane w rozszerzeniu .php, a css lub js były dołączane do nich osobno, panel logowania i rejestracji również zostały zapisane w rozszerzeniu .php, wysyłają one dane użytkowników do bazy co umożliwia utworzenie konta a później zalogowanie się na nie.

4.Elementy statyczne

4.1. HTML

Strona zawiera header, jest podzielona na sekcje oraz posiada stopkę. użyte zostały znaczniki semantyczne takie jak <section> <nav> aby poprawić strukturę i czytelność kodu

4.2. CSS

Css na naszej stronie pozwala nam osiągnąć estetyczny, nowoczesny i responsywny wygląd strony. Kolorystyka nawiązująca do klimatu festiwalu(żywe kolory, gradienty, motywy muzyczne). Dobrano czytelne czcionki. Strona jest responsywna

4.3. Biblioteki zewnętrzne

Zostały zaimportowane różnego rodzaju ikonki oraz zestaw niestandardowych czcionek(google fonts, font awesome).

5.JavaScript

5.1. Dynamika

JavaScript został wykorzystany w projekcie do wprowadzenia elementów interaktywnych, co prowadzi do poprawienia wrażenia użytkownika podczas przewijania strony.

5.2. Obsługa przycisków i efektów interaktywnych

Dodane zostały np efekty hover/click na przyciskach które wywołują animacje, walidacja formularzy na podstronie kontakt, zmiany treści bez przeładowania strony, rozwijane pytania FAQ w podstronie kontakt, możliwość przewijania zdjęć.

6.PHP

6.1. Obsługa logowania i rejestracji użytkownika

Rejestracja- użytkownik może założyć konto podając swoje dane, wprowadzając email lub hasło przez który będzie później mógł się zalogować, dane użytkownika są zapisywane w bazie danych

Logowanie- weryfikacja danych odbywa się poprzez porównanie hasła oraz sesji PHP

Obsługa sesji- po zalogowaniu tworzona jest sesja dzięki której użytkownik pozostaje zalogowany i może uzyskać dostęp do panelu użytkownika lub administrator do panelu admina/ pracownik do panelu pracownika

6.2. Obsługa logowania i rejestracji użytkownika

Strona zawiera dynamiczny katalog, np z artystami, wydarzeniami lub programem festiwalu, dane są pobierane z bazy danych MySQL przy pomocy zapytań SQL, strona może wyświetlić np listę festiwali, ich datę oraz miejsce, artystów którzy będą tam występować, pobliskie noclegi oraz możliwość wspólnego transportu.

6.3. Obsługa formularza kontaktowego

Zastosowano formularz kontaktowy który pozwala odwiedzającym stronę wysłać wiadomość do organizatorów, po przesłaniu PHP sprawdza poprawność danych, a następnie zapisuje dane do bazy

6.4. Panel administracyjny

Dla właściciela jak i dla jego pracowników przygotowaliśmy osobne panele, właściciel może zobaczyć np swoje zarobki, pracownicy mogą edytować dane lub je usuwać, panel działa w oparciu o PHP

7. Baza danych

7.1. Projekt struktury bazy danych

W projekcie wykorzystano relacyjną bazę danych MySQL do przechowywania informacji o użytkownikach, wydarzeniach festiwalowych, wiadomościach kontaktowych, biletach oraz danych administracyjnych itp. Dzięki temu możliwa jest dynamiczna obsługa treści i interakcja użytkownika ze stroną

Baza danych została zaprojektowana tak aby wspierać najważniejsze funkcjonalności czyli: logowanie i rejestracja użytkowników, wyświetlanie programu festiwali i informacji o wydarzeniach, obsługa formularza kontaktowego, zarządzanie treścią przez administratorów

7.2. Opis tabel i pól

7.3 Relacje

Wydarzenia są niezależne ale mogą być powiązane z użytkownikami. użytkownicy, rola decyduje o tym, czy użytkownik ma dostęp do panelu admina lub np panelu właściciela. Wiadomości nie mają relacji z innymi tabelami- jest to zbiór formularzy kontaktowych od anonimowych lub zalogowanych użytkowników

7.4 Przykładowe zapytania SQL

```
INSERT INTO `uzytkownicy` (`uzytkownicy_id`, `nazwa`, `imie`, `nazwisko`,  
`data_urodzenia`, `email`, `telefon`, `haslo`, `rola`, `data_dodania`) VALUES  
(1, 'maciej_latawiec', 'Maciej', 'Latawiec', '2025-05-13',  
'maciej.latawiec29@gmail.com', '668939663',  
'827ccb0eea8a706c4c34a16891f84e7b', 'klient', '2025-05-18 20:08:21')
```

Mamy tutaj przedstawione przykładowe zapytanie które realizuje się podczas rejestracji użytkownika

8. Bezpieczeństwo

Bezpieczeństwo aplikacji webowej jest kluczowe, zwłaszcza gdy przechowuje dane użytkowników, przetwarza formularze, czy umożliwia logowanie.

8.1 Walidacja danych wejściowych

Formularze zawierają atrybuty `required`, `type="email"` i ograniczenia długości, co zapobiega wysyłaniu pustych lub nieprawidłowych danych

przykładowa walidacja adresu e-mail i długości wiadomości została zaimplementowana w JavaScript - użytkownik jest informowany o błędach bez konieczności odświeżania strony

Dane pochodzące od użytkownika są filtrowane i sprawdzane po stronie serwera

8.2 Obsługa sesji i uwierzytelnianie

Po zalogowaniu użytkownika tworzona jest sesja PHP za pomocą `session_start()` i zapisywane są jego dane, dostęp do stron admina lub konra użytkownika jest kontrolowany przez sprawdzanie zmian w sesyjnych np. roli.

Hasła są przechowywane w bazie w postaci zaszyfrowanej (hashowanej) przy pomocy `password_hash()`

Sesje mogą być ustawione na automatyczne wygasanie po określonym czasie braku aktywności

Ograniczony został dostęp do panelu administracyjnego tylko dla zalogowanych administratorów.

9. Możliwość rozbudowy

9.1 Dodanie wyszukiwarki

Możliwość dodania prostego lub zaawansowanego mechanizmu wyszukiwania katalogu wydarzeń lub wykonawców, mogłoby to być wyszukiwanie po tytule, dacie, lokalizacji lub nazwisku artysty, możliwe również byłoby ustawienie filtrowania np. tylko darmowe występy

9.2 System ocen i komentarzy

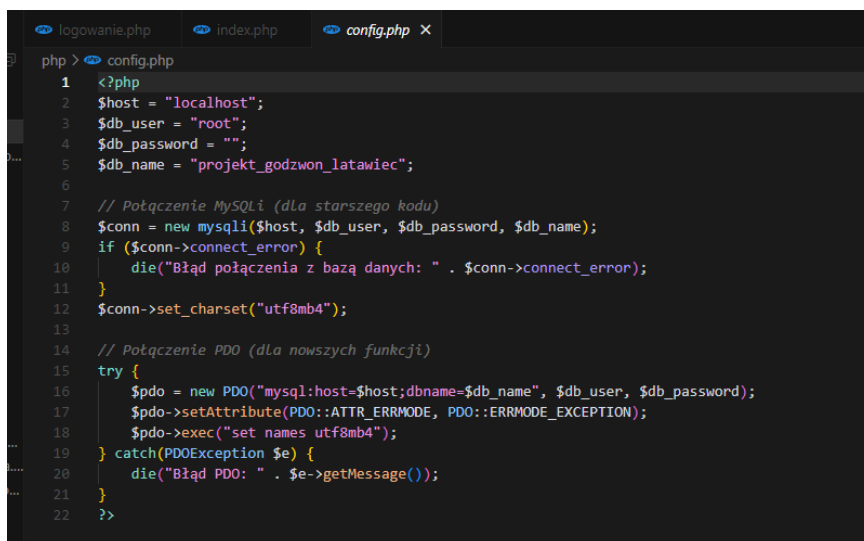
Po zalogowaniu użytkownik mógłby dodać komentarz i ocenić wydarzenie np. w skali od 1 do 5 gwiazdek, na temat festiwalu w którym brał udział lub na temat naszej strony, byłoby to zapisywane w bazie danych, a administrator mógłby mieć kontrolę nad tymi komentarzami

9.3 Aplikacja mobilna

Możliwość stworzenia natywnej aplikacji mobilnej z funkcjami programu festiwalu, mapka, bilety, przypomnienia

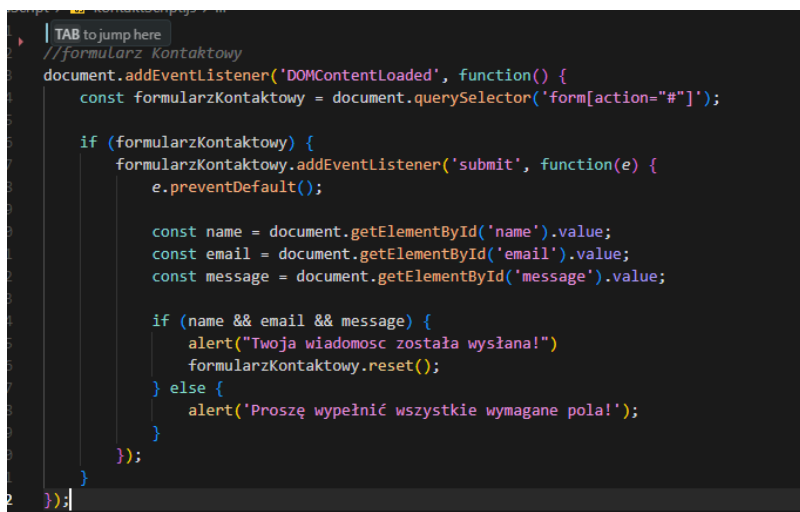
10. Załączniki

10.1 Fragmenty kodu źródłowego



```
1 <?php
2 $host = "localhost";
3 $db_user = "root";
4 $db_password = "";
5 $db_name = "projekt_godzwon_latawiec";
6
7 // Połączenie MySQLi (dla starszego kodu)
8 $conn = new mysqli($host, $db_user, $db_password, $db_name);
9 if ($conn->connect_error) {
10     die("Błąd połączenia z bazą danych: " . $conn->connect_error);
11 }
12 $conn->set_charset("utf8mb4");
13
14 // Połączenie PDO (dla nowszych funkcji)
15 try {
16     $pdo = new PDO("mysql:host=$host;dbname=$db_name", $db_user, $db_password);
17     $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
18     $pdo->exec("set names utf8mb4");
19 } catch(PDOException $e) {
20     die("Błąd PDO: " . $e->getMessage());
21 }
22 ?>
```

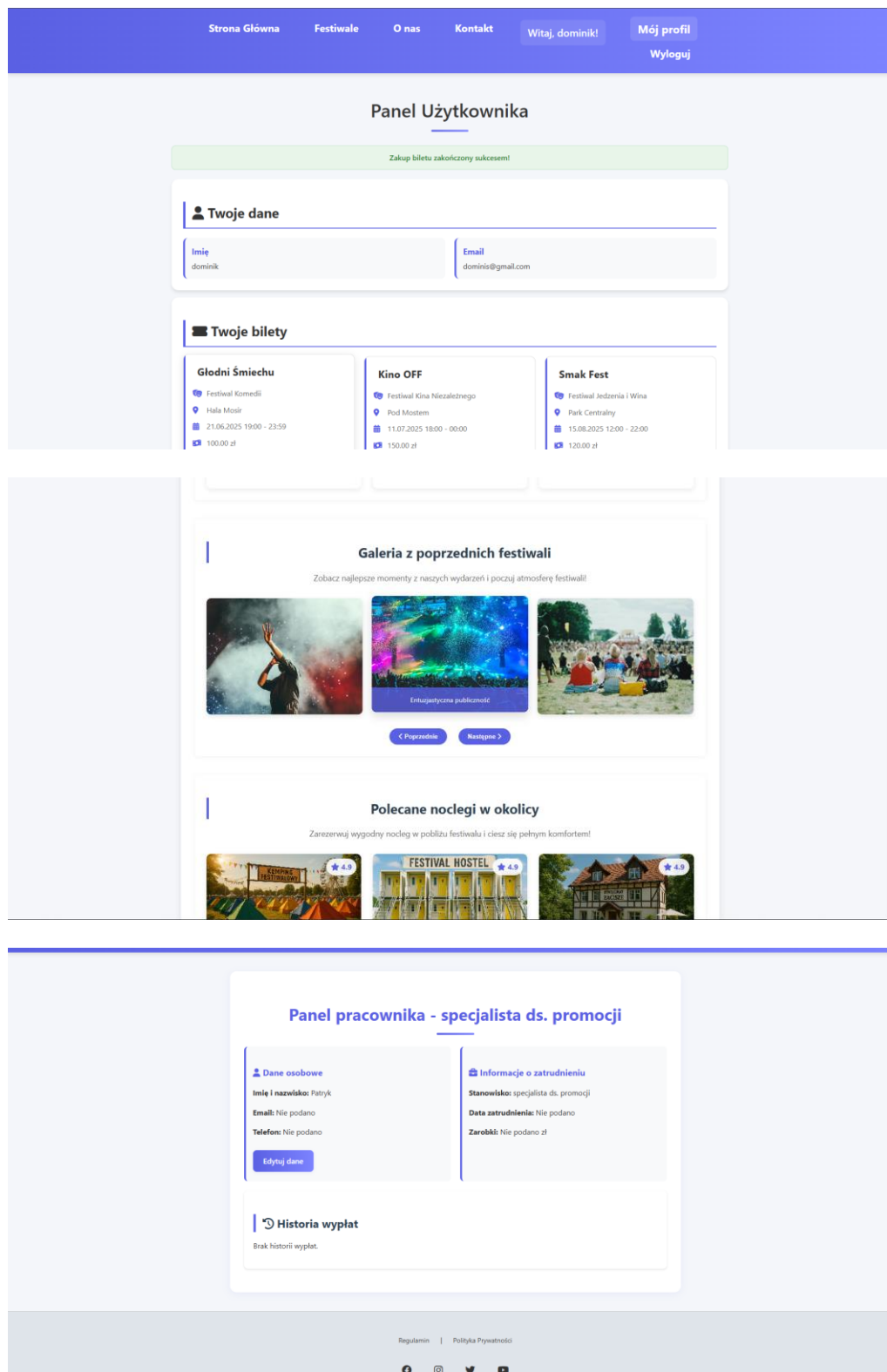
//przykładowe łączenie się z bazą



```
1 //formularz Kontaktowy
2 document.addEventListener('DOMContentLoaded', function() {
3     const formularzKontaktowy = document.querySelector('form[action="#"]');
4
5     if (formularzKontaktowy) {
6         formularzKontaktowy.addEventListener('submit', function(e) {
7             e.preventDefault();
8
9             const name = document.getElementById('name').value;
10            const email = document.getElementById('email').value;
11            const message = document.getElementById('message').value;
12
13            if (name && email && message) {
14                alert("Twoja wiadomość została wysłana!");
15                formularzKontaktowy.reset();
16            } else {
17                alert('Proszę wypełnić wszystkie wymagane pola!');
18            }
19        });
20    }
21 });
```

//Wysyłanie formularza

10.2. Zrzuty działania systemu



10.3 Diagram Bazy Danych



10.4 Lista błędów/uwag i pomysłów na ulepszenia

- wszystkie dane wyciągane z bazy oraz zdjęcia
- Rozbudowa panelów
- zrobienie koszyka do kupowania biletów
- formularz aplikowanie do pracy
- dokładniejsze informacje co do transportu

