

NightClub - Projekt Strony Internetowej

Autorzy: [Wpisz Imiona i Nazwiska Autorów]

Spis Treści

- [1. Wstęp](#)
 - [1.1. Cel projektu](#)
 - [1.2. Zakres projektu](#)
 - [1.3. Użytkownicy systemu](#)
- [2. Opis ogólny](#)
 - [2.1. Przeznaczenie witryny](#)
 - [2.2. Główne funkcjonalności](#)
 - [2.3. Wymagania techniczne](#)
 - [2.4. Struktura katalogów i plików projektu](#)
- [3. Struktura strony internetowej](#)
 - [3.1. Strona główna \(index.php\)](#)
 - [3.2. Podstrona Oferta \(oferta.php\)](#)
 - [3.3. Podstrona Kontakt \(kontakt.php\)](#)
 - [3.4. Podstrony Funkcjonalne Klienta \(kup_bilety.php, rezerwacja_sali.php\)](#)
 - [3.5. Logowanie, Rejestracja i Panele Użytkowników](#)
- [4. Elementy statyczne](#)
 - [4.1. HTML – struktura dokumentów](#)
 - [4.2. CSS – stylizacja \(style.css\)](#)
 - [4.3. Biblioteki zewnętrzne](#)
- [5. JavaScript \(script.js\)](#)
 - [5.1. Dynamiczna aktualizacja interfejsu](#)
 - [5.2. Obsługa Modala dla Wiadomości i Walidacja Formularzy](#)
 - [5.3. Obsługa przycisków i efektów interaktywnych](#)
- [6. Elementy dynamiczne \(PHP\)](#)
 - [6.1. Obsługa logowania i rejestracji użytkownika](#)
 - [6.2. Pobieranie i przetwarzanie danych z bazy \(np. oferta.php, panele_użytkowników\)](#)
 - [6.3. Obsługa formularza kontaktowego](#)

- [6.4. Panele administracyjne \(zarządzanie rezerwacjami, użytkownikami, stanem magazynowym\)](#)
- [7. Baza danych](#)
 - [7.1. Projekt struktury bazy danych](#)
 - [7.2. Opis tabel i pól](#)
 - [7.3. Relacje między tabelami](#)
 - [7.4. Przykładowe zapytania SQL](#)
- [8. Bezpieczeństwo](#)
 - [8.1. Weryfikacja danych wejściowych](#)
 - [8.2. Obsługa sesji i uwierzytelnianie](#)
- [9. Możliwości rozbudowy](#)
 - [9.1. Dodanie wyszukiwarki wydarzeń](#)
 - [9.2. Rozbudowany system ocen i komentarzy](#)
 - [9.3. Rozbudowa systemu rezerwacji online](#)
 - [9.4. Zaawansowana responsywność i aplikacja mobilna](#)
- [10. Załączniki](#)

1. Wstęp

1.1. Cel projektu

Celem projektu było stworzenie dynamicznej strony internetowej dla klubu nocnego. [cite: 2] Strona ma umożliwiać użytkownikom przeglądanie oferty klubu, nadchodzących wydarzeń (koncertów), dokonywanie rezerwacji sal prywatnych wraz z dodatkami oraz zakup biletów na koncerty. [cite: 2] Projekt zakłada również implementację panelu klienta, pracownika i właściciela z różnymi poziomami uprawnień i funkcjonalności. [cite: 2]

1.2. Zakres projektu

Zakres projektu obejmuje:

- Stworzenie interfejsu użytkownika (frontend) z wykorzystaniem HTML, CSS i JavaScript.
- Implementację logiki biznesowej i interakcji z bazą danych (backend) przy użyciu PHP i MySQL.
- System rejestracji i logowania dla trzech typów użytkowników: Klient, Pracownik, Właściciel.

- Funkcjonalność przeglądania oferty klubu, w tym opis sal (małej, dużej, koncertowej).
- Możliwość dodawania opinii przez zalogowanych klientów.
- System zakupu biletów na koncerty z wizualną symulacją płatności.
- System rezerwacji sal prywatnych z możliwością wyboru drinków i dodatków, uwzględniający dostępność zasobów i wstępne wyliczenie ceny.
- Panel Klienta: przeglądanie rezerwacji i biletów, możliwość anulowania rezerwacji.
- Panel Pracownika: zatwierdzanie nowych kont klientów, potwierdzanie/anulowanie rezerwacji, zarządzanie stanem magazynowym drinków.
- Panel Właściciela: podgląd sprzedaży biletów i przychodów, przegląd rezerwacji i ich kosztów, zarządzanie rolami użytkowników.
- Formularz kontaktowy dla zalogowanych użytkowników.
- Przechowywanie danych w relacyjnej bazie danych MySQL.

1.3. Użytkownicy systemu

System przewiduje trzy główne role użytkowników:

- **Klient:** Może przeglądać ofertę, wydarzenia, dodawać opinie, rezerwować sale, kupować bilety oraz zarządzać swoimi rezerwacjami i biletami w panelu klienta.
- **Pracownik:** Odpowiedzialny za obsługę operacyjną – zatwierdzanie nowych kont klientów, zarządzanie rezerwacjami (potwierdzanie, anulowanie), aktualizowanie stanu drinków w systemie.
- **Właściciel:** Ma dostęp do danych zarządczych – podgląd sprzedaży biletów, przychodów z rezerwacji, zarządzanie kontami użytkowników (np. zmiana roli klienta na pracownika).

2. Opis ogólny

2.1. Przeznaczenie witryny

Witryna internetowa klubu nocnego ma na celu promocję oferty klubu, ułatwienie klientom dostępu do informacji o wydarzeniach oraz umożliwienie interakcji z klubem poprzez systemy rezerwacji, zakupu biletów i pozostawiania opinii. Strona służy również jako narzędzie administracyjne dla personelu i właściciela klubu.

2.2. Główne funkcjonalności

- Prezentacja oferty klubu (sale, wydarzenia).
- Rejestracja i logowanie użytkowników z podziałem na role.
- System rezerwacji sal z wyborem drinków i dodatków oraz sprawdzaniem dostępności.
- System sprzedaży biletów na koncerty.
- Dodawanie i wyświetlanie opinii klientów.
- Formularz kontaktowy.
- Panele administracyjne dla klienta, pracownika i właściciela dostosowane do ich ról.

2.3. Wymagania techniczne

Do poprawnego działania i rozwoju strony wymagane są:

- Serwer WWW obsługujący PHP (np. XAMPP, WAMP, MAMP).
- PHP w wersji 7.4 lub nowszej.
- Serwer bazy danych MySQL lub MariaDB.
- Nowoczesna przeglądarka internetowa (np. Google Chrome, Mozilla Firefox, Microsoft Edge).

2.4. Struktura katalogów i plików projektu [cite: 3]

```
nightclub/
|
├── index.php      - Strona główna
├── oferta.php    - Strona z ofertą sal
├── kontakt.php   - Strona kontaktowa
├── zaloguj.php   - Strona logowania
├── zarejestruj.php - Strona rejestracji
├── wyloguj.php   - Skrypt wylogowania
|
└── client/        - Pliki dostępne dla zalogowanego klienta
    └── kup_bilety.php - Strona zakupu biletów
```


└── database.sql - Plik eksportu bazy danych

Struktura strony internetowej

3.1. Strona główna (*index.php*) Strona główna (*index.php*) pełni rolę wizytówki klubu.

Zawiera:

- Sekcję powitalną
- Dynamicznie ładowaną listę najbliższych wydarzeń (koncertów) pobieranych z bazy danych
- Sekcję z ostatnimi opiniami klientów, również pobieranymi z bazy
- Formularz dodawania opinii dla zalogowanych klientów
- Komunikat zachęcający do zalogowania się w celu skorzystania z pełnej funkcjonalności

3.2. Podstrona Oferta (*oferta.php*)

- Opis Sali Małej (do rezerwacji prywatnych)
- Opis Sali Dużej (do rezerwacji prywatnych)
- Opis Sali Koncertowej (miejsce odbywania się koncertów)
- Informacje o pojemności i podstawowych cenach
- Przyciski umożliwiające przejście do rezerwacji dla zalogowanych klientów

3.3. Podstrona Kontakt (*kontakt.php*)

- Formularz kontaktowy (widoczny tylko dla zalogowanych)
- Zapisywanie zapytań w bazie danych
- Podstawowe dane kontaktowe klubu (adres, telefon, email)

3.4. Podstrony Funkcjonalne Klienta *kup_bilety.php*:

- Przeglądanie listy nadchodzących koncertów
- Formularz wyboru ilości biletów
- Przekierowanie do symulacji płatności

rezerwacja_sali.php:

- Rezerwacja Sali Małej lub Dużej
- Wybór terminu i liczby drinków
- Dynamiczne sprawdzanie dostępności
- Wyświetlanie wstępnej ceny

3.5. Logowanie, Rejestracja i Panele Użytkowników *zaloguj.php*:

- Formularz logowania z wyborem roli

zarejestruj.php:

- Formularz rejestracji dla nowych klientów
- Wymagane zatwierdzenie przez pracownika

Panele Użytkowników:

- *panel_klienta.php*: Zarządzanie rezerwacjami i biletami
- *panel_pracownika.php*: Zarządzanie kontami i rezerwacjami
- *panel_właściciela.php*: Podgląd finansów i zarządzanie

Elementy statyczne

4.1. HTML – struktura dokumentów

- Semantyczne znaczniki HTML5
- Wspólne elementy (header.php, footer.php) dołączane dynamicznie
- Główna treść w

4.2. CSS – stylizacja (style.css)

- Jeden główny arkusz stylów
- Ciemna paleta barw z akcentami
- Layout oparty na Flexbox i Grid
- Pełna responsywność

JavaScript (script.js)

5.1. Dynamiczna aktualizacja interfejsu

- Obliczanie wstępnej ceny rezerwacji
- Aktualizacja dostępności drinków

5.2. Obsługa Modala i Walidacja

- Modal do wyświetlania wiadomości
- Podstawowa walidacja formularzy
- Potwierdzenia przed akcjami

Elementy dynamiczne (PHP)

6.1. Obsługa logowania i rejestracji

- Sprawdzanie unikalności emaila
- Ustawianie zmiennych sesyjnych
- Weryfikacja statusu konta

6.2. Pobieranie danych z bazy

- Listy wydarzeń, opinii, sal
- Dane specyficzne dla roli użytkownika

6.3. Formularz kontaktowy

- Walidacja i zapis do bazy
- Tabela contact_messages

6.4. Panele administracyjne

- Zatwierdzanie kont klientów
- Zarządzanie rezerwacjami
- Aktualizacja stanu magazynowego
- Raporty finansowe

Baza danych

7.1. Projekt struktury

- Relacyjna baza danych
- Tabele: users, halls, events, tickets, drinks, extras, reservations

7.2. Główne tabele

- users: ID, email, hasło, rola, status
- halls: ID, nazwa, pojemność, cena
- events: ID, nazwa, data, ID sali
- tickets: ID, ID wydarzenia, ID użytkownika
- drinks: ID, nazwa, ilość, cena

7.3. Relacje

- 1:1, 1:many, many:many
- Klucze obce łączące tabele

Bezpieczeństwo

8.1. Walidacja danych

- Po stronie klienta (JS) i serwera (PHP)
- Ochrona przed XSS i SQL Injection

8.2. Obsługa sesji

- Zmienne \$_SESSION
- Funkcje kontroli dostępu
- Niszczenie sesji przy wylogowaniu

9. Możliwości rozbudowy

9.1. Wyszukiwarka wydarzeń

9.2. Rozbudowany system ocen

9.3. Płatności online

9.4. Możliwość dodawania koncertów przez pracownika

Załączniki

10.1. Fragmenty kodu

Skrypt symulacja płatności:

```
<?php
require_once __DIR__ . '/../includes/header.php';
requireRole('klient');
requireApprovedUser();

if (!isset($_SESSION['ticket_order'])) {
    $_SESSION['message'] = "Brak danych zamówienia do przetworzenia.";
    $_SESSION['message_type'] = "error";
    header("Location: /nightclub/client/kup_bilety.php");
    exit();
}

$order = $_SESSION['ticket_order'];
?>

<h2>Potwierdzenie Zakupu i Symulacja Płatności</h2>

<div class="payment-summary styled-form">
    <h3>Podsumowanie Zamówienia Biletów</h3>
    <p><strong>Wydarzenie:</strong> <?php echo sanitize_output($order['event_name']); ?></p>
    <p><strong>Ilość biletów:</strong> <?php echo $order['quantity']; ?></p>
    <p><strong>Cena za bilet:</strong> <?php echo number_format($order['ticket_price'], 2); ?> PLN</p>
    <p><strong>Łączna kwota do zapłaty:</strong> <?php echo number_format($order['total_price'], 2); ?>
        PLN</p>

    <hr>
    <h4>Symulacja Płatności</h4>
    <p>To jest jedynie wizualna strona symulacji płatności dla projektu szkolnego. W rzeczywistej
        aplikacji tutaj znajdowałby się formularz płatności lub integracja z bramką płatniczą.</p>

    <form action="/nightclub/actions/process_payment_action.php" method="POST">
        <p>Proszę kliknąć "Zapłać", aby sfinalizować zakup.</p>
        <button type="submit" name="finalize_payment" class="btn btn-large">Zapłać i Kup
            Bilety</button>
    </form>
    <a href="/nightclub/client/kup_bilety.php?event_id=<?php echo $order['event_id']; ?>" class="btn
        btn-secondary" style="margin-top: 10px;">Anuluj i wróć</a>
</div>

<?php require_once __DIR__ . '/../includes/footer.php'; ?>
```

Skrypt logowania:

```
<?php
require_once __DIR__ . '/../../includes/db_connect.php';

if ($_SERVER['REQUEST_METHOD'] == "POST") {
    $email = trim($_POST['email']);
    $password = $_POST['password'];
    $role_choice = $_POST['role_choice'];

    $_SESSION['form_data'] = ['email' => $email, 'role_choice' => $role_choice]; // Zapamiętaj dane na wypadek błędu

    if (empty($email) || empty($password) || empty($role_choice)) {
        $_SESSION['message'] = "Wszystkie pola są wymagane.";
        $_SESSION['message_type'] = "error";
        header("Location: /nightclub/zaloguj.php");
        exit();
    }

    $stmt = $conn->prepare("SELECT user_id, email, password, role, is_approved FROM users WHERE email = ? AND role = ?");
    $stmt->bind_param("ss", $email, $role_choice);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows == 1) {
        $user = $result->fetch_assoc();
        if ($password === $user['password']) { // Porównanie haseł (bez szyfrowania)
            // Sprawdzenie zatwierdzenia tylko dla Klienta
            if ($user['role'] === 'Klient' && !$user['is_approved']) {
                $_SESSION['message'] = "Twoje konto oczekuje na zatwierdzenie przez pracownika. Nie możesz się jeszcze zalogować!";
                $_SESSION['message_type'] = "warning";
                header("Location: /nightclub/zaloguj.php");
                exit();
            }

            $_SESSION['user_id'] = $user['user_id'];
            $_SESSION['email'] = $user['email'];
            $_SESSION['role'] = $user['role'];
            // $_SESSION['is_approved'] nie jest już potrzebne w sesji, bo mamy funkcję isUserApproved()

            unset($_SESSION['form_data']); // Usuń zapamiętane dane formularza
            $_SESSION['message'] = "Logowanie pomyślne. Witaj " . sanitize_output($user['email']) .
        "!";
            $_SESSION['message_type'] = "success";
        }

        $redirect_url = $_SESSION['redirect_url'] ?? '/nightclub/index.php';
        unset($_SESSION['redirect_url']);
        header("Location: " . $redirect_url); // Przekieruj na stronę główną lub zapamiętaną
        exit();
    } else {
        $_SESSION['message'] = "Nie znaleziono użytkownika z podanym adresem e-mail i rolą.";
        $_SESSION['message_type'] = "error";
        header("Location: /nightclub/zaloguj.php");
    }
    $stmt->close();
} else {
    header("Location: /nightclub/zaloguj.php");
}
exit();
?>
```

10.2. Zrzuty ekranu

Strona Główna Oferta Kontakt Panel Pracownika Wyloguj (pracownik@example.com)

Nasza Oferta

Sale Prywatne do Wynajęcia



Sala Mała
Przytulna sala idealna na mniejsze imprezy, urodziny czy spotkania w gronie przyjaciół. Wypożyczana w wygodne loże i dostęp do baru.
Pojemność: do 30 osób
Cena bazowa za wynajem: 250.00 PLN

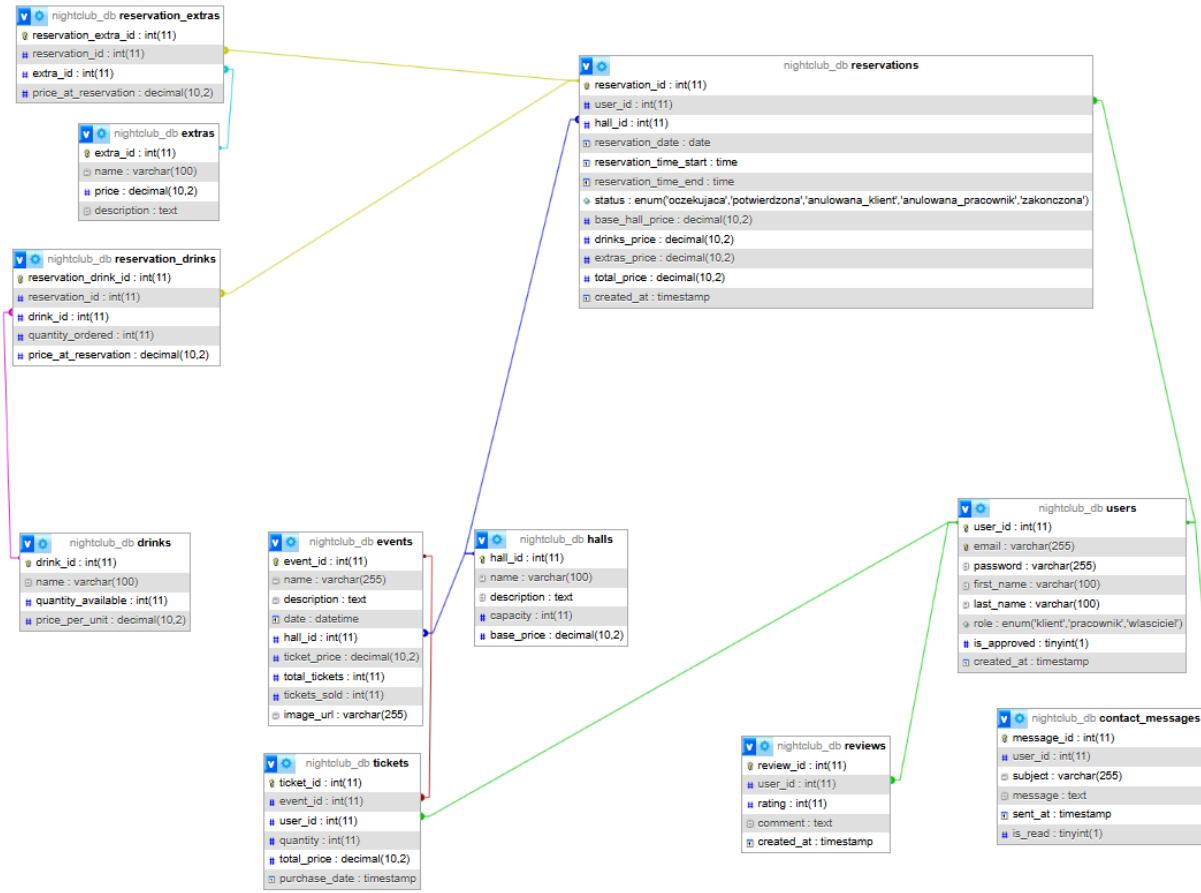


Sala Duża
Przestronna sala z największym parkietem, idealna na większe przyjęcia, imprezy firmowe. Możliwość aranżacji według potrzeb.
Pojemność: do 80 osób
Cena bazowa za wynajem: 600.00 PLN

Sala Koncertowa



10.3. Diagram bazy



10.4. Lista uwag

Bezpieczeństwo Haseł, Walidacja i Obsługa Błędów w Aplikacjach Webowych

Bezpieczeństwo Haseł

Najważniejszą uwagę jest przechowywanie haseł w bazie danych jako zwykły tekst. W środowisku produkcyjnym jest to **niedopuszczalne**.

- ◆ **Rozwiążanie:** Należy zaimplementować haszowanie haseł (np. za pomocą `password_hash()`) oraz ich weryfikację przez `password_verify()` w PHP.

Walidacja po stronie serwera

Chociaż zaimplementowano podstawową walidację, warto ją rozszerzyć i upewnić się, że wszystkie dane wejściowe są dokładnie sprawdzane pod kątem: **Typów danych, Zakresów wartości, Potencjalnie szkodliwych danych**

Dzięki temu zapytania SQL oraz wyświetlane treści będą bezpieczne.

Obsługa błędów

System wyświetla komunikaty o błędach, ale warto rozważyć:

- Bardziej szczegółowe **logowanie błędów** po stronie serwera (np. do pliku logów), co ułatwi debugowanie.
- Prezentowanie użytkownikowi **ogólnych komunikatów**, które nie zdradzają szczegółów technicznych.