

Dokumentacja projektu: Bania u Cygana

Autor:

Krzysztof Tomczyk

Klasa 3AP

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 1: Oferta (offer.php)
 - 3.3. Podstrona 2: Kontakt (contact.php)
 - 3.4. Podstrona 3: Rezerwacja (booking.php)
 - 3.5. Panel logowania i rejestracji
4. Elementy statyczne
 - 4.1. HTML – struktura dokumentów
 - 4.2. CSS – stylizacja (style.css)
 - 4.3. Biblioteki zewnętrzne
5. JavaScript
 - 5.1. Dynamiczne menu rozwijane
 - 5.2. Slider z aktualnościami
 - 5.3. Obsługa przycisków i efektów interaktywnych
6. Elementy dynamiczne (PHP)
 - 6.1. Obsługa logowania i rejestracji użytkownika
 - 6.2. Pobieranie danych z bazy (katalog.php)
 - 6.3. Obsługa formularza kontaktowego
 - 6.4. Panel administracyjny
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. Walidacja danych wejściowych
 - 8.2. Obsługa sesji i uwierzytelnianie
9. Możliwości rozbudowy
 - 9.1. Dodanie wyszukiwarki
 - 9.2. System ocen i komentarzy
 - 9.3. System rezerwacji online
 - 9.4. Responsywność i aplikacja mobilna
10. Załączniki
 - 10.1. Fragmenty kodu źródłowego
 - 10.2. Zrzuty ekranu działania systemu

10.3. Diagram bazy danych

10.4. Lista błędów/uwag i pomysłów na ulepszenia

Celem projektu „Bania u Cygana” jest stworzenie aplikacji internetowej umożliwiającej rezerwację sal imprezowych, zarządzanie zapasami napojów oraz administrację użytkownikami i rezerwacjami. System został zaprojektowany z myślą o łatwości obsługi dla klientów, pracowników i administratorów, wykorzystując technologie PHP, MySQL, HTML, CSS i JavaScript.

1.2. Zakres projektu

Projekt obejmuje:

- Stronę główną z informacjami o firmie, wydarzeniami i opiniami.
- System logowania i rejestracji z weryfikacją przez administratora.
- Formularz rezerwacji sal z wyborem dodatków i napojów.
- Panele użytkownika: klienta (rezerwacje, opinie), pracownika (zapasy, rezerwacje), admina (użytkownicy).
- Formularz kontaktowy i slider opinii.
- Dynamiczne zarządzanie zapasami napojów i anulowaniem rezerwacji.

1.3. Użytkownicy systemu

System obsługuje trzy grupy użytkowników:

- **Klienci:** Rezerwują sale, wybierają dodatki i napoje, dodają opinie, zarządzają rezerwacjami.
- **Pracownicy:** Aktualizują zapasy napojów, potwierdzają lub anulują rezerwacje.
- **Administratorzy:** Zatwierdzają nowych użytkowników, zarządzają rolami, usuwają konta.

2. Opis ogólny

2.1. Przeznaczenie witryny

Witryna „Bania u Cygana” umożliwia organizację imprez poprzez rezerwację sal, zarządzanie zasobami i komunikację z użytkownikami. Zapewnia intuicyjny interfejs i efektywne zarządzanie danymi.

2.2. Główne funkcjonalności

- Logowanie i rejestracja z weryfikacją. (konto: Admin: (email: admin@bania.pl hasło: admin123) Pracownik: (email: pracownik@bania.pl hasło: pracownik123,) Klient:(email: klient@test.pl hasło: klient123))
- Rezerwacja sal (Sala X, Sala Y) z dodatkami (DJ, fotobudka, ochrona) i napojami.
- Wyświetlanie najbliższych wydarzeń (rezerwacje z 7 dni).
- Slider opinii na stronie głównej.
- Panel klienta: przeglądanie/anulowanie rezerwacji, dodawanie opinii.
- Panel pracownika: aktualizacja zapasów, potwierdzanie/anulowanie rezerwacji.
- Panel admina: zatwierdzanie/odrzucając użytkowników, zmiana ról, usuwanie kont.
- Formularz kontaktowy z modalem.
- Zwracanie napojów do zapasów przy anulowaniu rezerwacji.

2.3. Wymagania techniczne

- **Serwer:** XAMPP (Apache 2.4+, PHP 7.4+/8.x, MySQL/MariaDB).
- **Przeglądarka:** Chrome, Firefox, Edge (najnowsze wersje).
- **Biblioteki:** jQuery 3.6.0 (CDN: <https://code.jquery.com/jquery-3.6.0.min.js> (<https://code.jquery.com/jquery-3.6.0.min.js>)).
- **Baza danych:** MySQL/MariaDB z zaimportowanym plikiem database.sql.
- **System:** Windows/Linux/macOS.

2.4. Struktura katalogów i plików projektuStruktura projektu:

```
/bania-u-cygana
├── assets/
│   ├── css/
│   │   └── style.css
│   ├── images/
│   │   ├── club_background.jpg
│   │   ├── party_banner.jpg
│   │   ├── sala_x.jpg
│   │   ├── sala_y.jpg
│   │   ├── header_bg.jpg
│   │   ├── footer_bg.jpg
│   │   └── club_bg.jpg
│   └── js/
│       └── scripts.js
├── includes/
│   ├── config.php |
├── auth.php
│   ├── header.php
│   └── footer.php
├── index.php
├── login.php
├── register.php
├── process_registration.php
├── booking.php
├── process_booking.php
├── panel_klienta.php
├── panel_pracownika.php
├── panel_admin.php
├── logout.php
├── manage_reservation_employee.php
├── cancel_reservation_client.php
├── delete_user.php
├── update_inventory.php
├── approve_user.php
├── reject_user.php
├── change_role_to_employee.php
├── offer.php
├── contact.php
├── wyslij_wiadomosc.php
├── accept_reservation.php
├── cancel_reservation_employee.php
├── add_opinion.php
└── get_drinks.php
```

3. Struktura strony internetowej

3.1. Strona główna (index.php)

Strona główna wyświetla powitanie, informacje o firmie, listę najbliższych wydarzeń (rezerwacje potwierdzone z 7 dni) oraz slider opinii. Zalogowani użytkownicy widzą personalizowaną wiadomość.

3.2. Podstrona 1: Oferta (offer.php)

Prezentuje sale (Sala X, Sala Y) z opisami, zdjęciami, pojemnościami i cenami. Zawiera linki do formularza rezerwacji.

3.3. Podstrona 2: Kontakt (contact.php)

Zawiera przycisk otwierający modal z formularzem kontaktowym, przetwarzanym przez wyslij_wiadomosc.php.

3.4. Podstrona 3: Rezerwacja (booking.php)

Formularz umożliwia wybór sali, daty, dodatków i napojów. Dane są zapisywane przez process_booking.php.

- login.php: Formularz logowania z walidacją.

- register.php: Rejestracja z potwierdzeniem hasła. panel_klienta.php: Rezerwacje i opinie klienta.
- panel_pracownika.php: Zarządzanie zapasami i rezerwacjami. panel_admin.php: Zarządzanie użytkownikami.

4. Elementy statyczne

4.1. HTML – struktura dokumentów

Strony korzystają z szablonów header.php i footer.php. Przykładowa struktura z index.php:

```
<h2>Witamy w Bania u Cygana</h2>
<p>Oferujemy wynajem sal na imprezy...</p>
<div class="events">    <!-- Lista
wydarzeń -->
</div>
<div class="opinions-slider">
    <!-- Slider opinii -->
</div>
```

4.2. CSS – stylizacja (style.css)

Definiuje style dla:

- Responsywnego układu (flexbox, media queries).
- Tła obrazkowego (np. club_background.jpg).
- Animacji (slider, modal).
- Komunikatów (.error-message, .success-message).

Fragment kodu:

```
body {    font-family: 'Arial', sans-serif;    background-image:
url('../images/club_background.jpg'); }
.error-message {    background-color:
#ffe6e6;    border: 1px solid #ff3333;
}
```

4.3. Biblioteki zewnętrzne

- jQuery 3.6.0: Obsługa slidera, modala, AJAX.

5. JavaScript

5.1. Dynamiczne menu rozwijane

Menu w header.php dostosowuje opcje do roli użytkownika (klient, pracownik, admin).

```
var currentIndex = 0; var opinions = $(''.opinion'); function showNextOpinion()
{    opinions.eq(currentIndex).fadeOut(500, function() {        currentIndex =
(currentIndex + 1) % opinions.length;        opinions.eq(currentIndex).fadeIn(500);
    }); } setInterval(showNextOpinion, 5000);
```

5.3. Obsługa przycisków i efektów interaktywnych

- Model kontaktowy (contact.php).
- AJAX do ładowania napojów (get_drinks.php).
- Walidacja formularzy (login.php, register.php).
- Potwierdzenie anulowania rezerwacji.

6. Elementy dynamiczne (PHP)

6.1. Obsługa logowania i rejestracji użytkownika

Logowanie (auth.php) używa prepared statements. Rejestracja (process_registration.php) zapisuje dane do pending_users do zatwierdzenia.

6.2. Pobieranie danych z bazy (katalog.php)

Przykładowe zapytanie z index.php:

```
$stmt = $conn->prepare("SELECT sala, data FROM bookings WHERE status = 'confirmed' AND data BETWEEN CURDATE() AND DATE_ADD(CURDATE(), $stmt->execute());
```

6.3. Obsługa formularza kontaktowego

Formularz w contact.php przetwarza dane w wyslij_wiadomosc.php, zwracając komunikat.

6.4. Panel administracyjny

panel_admin.php umożliwia:

- Zatwierdzanie/odrzucając użytkowników.
- Zmianę roli na pracownika.
- Usuwanie użytkowników.

7. Baza danych

7.1. Projekt struktury bazy danych

Baza bania_u_cygana zawiera 7 tabel: users, pending_users, bookings, drinks, inventory, booking_drinks, opinions.

7.2. Opis tabel i pól

- users: id (PK), email, password, role (client/employee/admin), created_at, status (unconfirmed/confirmed).
- pending_users: id (PK), email, password, created_at.
- bookings: id (PK), user_id (FK), sala, data, dodatki, status (pending/confirmed/cancelled).
- drinks: id (PK), name. inventory: id (PK), sala, drink_id (FK), quantity, ostatnia_aktualizacja.
- booking_drinks: id (PK), booking_id (FK), drink_id (FK), quantity.
- opinions: id (PK), user_id (FK), opinion, created_at.

7.3. Relacje między tabelami

- users.id → bookings.user_id (1:N).
- users.id → opinions.user_id (1:N).
- drinks.id → inventory.drink_id (1:N).
- bookings.id → booking_drinks.booking_id (1:N).
- drinks.id → booking_drinks.drink_id (1:N).

7.4. Przykładowe zapytania SQL

```
-- Pobieranie rezerwacji klienta
SELECT b.*, GROUP_CONCAT(CONCAT(d.name, ' ', bd.quantity)) AS drinks
FROM bookings b
LEFT JOIN booking_drinks bd ON b.id = bd.booking_id
LEFT JOIN drinks d ON bd.drink_id = d.id
WHERE b.user_id = ?
GROUP BY b.id;

-- Aktualizacja zapasów
UPDATE inventory SET quantity = quantity - ?
WHERE sala = ? AND drink_id = ?;
```

8. Bezpieczeństwo

8.1. Walidacja danych wejściowych

- Czyszczenie danych: `trim()`, `htmlspecialchars()`.
- Prepared statements w PHP (np. `process_booking.php`).
- Walidacja formularzy w JavaScript (`scripts.js`).

8.2. Obsługa sesji i uwierzytelnianie

- Sesje PHP w `header.php`.
- Funkcja `isLoggedIn()` w `auth.php`.
- Przekierowania dla nieautoryzowanego dostępu.

9. Możliwości rozbudowy

9.1. Dodanie wyszukiwarki

Wyszukiwarka rezerwacji po sali lub dacie z zapytaniem SQL WHERE.

9.2. System ocen i komentarzy

Dodanie pola rating (1–5) do tabeli opinions.

9.3. System rezerwacji online

Integracja z płatnościami (np. Stripe) lub kalendarzem Google.

9.4. Responsywność i aplikacja mobilna

Optymalizacja CSS dla urządzeń mobilnych i aplikacja w React Native.

10. Załączniki

10.1. Fragmenty kodu źródłowego

1. HTML – Struktura strony głównej (`index.php`, sekcja 4.1)

Opis: Fragment kodu HTML z `index.php`, który definiuje strukturę strony głównej, zawierającą powitanie, listę wydarzeń i slider opinii. Wykorzystuje szablony `header.php` i `footer.php` dla spójnego układu.

```

<!-- index.php -->
<div class="main-content">
  <h2>Witamy w Bania u Cygana</h2>
  <p>Oferujemy wynajem sal na imprezy okolicznościowe. Sprawdź naszą ofertę!</p>
  <div class="events">
    <!-- Lista najbliższych wydarzeń generowana dynamicznie przez PHP -->
    <?php
      $stmt = $conn->prepare("SELECT sala, data FROM bookings WHERE status = 'confirmed' AND data
    BETWEEN CURDATE() AND DATE_ADD(CURDATE(), INTERVAL 7 DAY) ORDER BY data");
      $stmt->execute();
      $result = $stmt->get_result();
      while ($row = $result->fetch_assoc()) {
        echo "<div class='event'>Sala: {$row['sala']}, Data: {$row['data']}</div>";
      }
    ?>
  </div>
  <div class="opinions-slider">
    <!-- Slider opinii ładowany przez JavaScript -->
  </div>
</div>

```

2. CSS – Stylizacja strony (style.css, sekcja 4.2)

Opis: Fragment kodu CSS z assets/css/style.css, definiujący style dla strony głównej, w tym responsywność dla mniejszych ekranów. Uwzględnia poprawkę responsywności po usunięciu folderu assets/css/assets, który mógł powodować konflikty stylów.

```
/* assets/css/style.css */
body {
  font-family: 'Arial', sans-serif;
  background-image: url('../images/club_background.jpg');
  background-size: cover;
}
.main-content {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
.error-message {
  background-color: #ffe6e6;
  border: 1px solid #ff3333;
  padding: 10px;
}
@media (max-width: 768px) {
  .main-content {
    flex-direction: column;
    align-items: center;
  }
  .events, .opinions-slider {
    width: 100%;
    max-width: 100%;
    margin: 0 auto;
    box-sizing: border-box;
  }
}
```

3. JavaScript – Slider opinii (scripts.js, sekcja 5.2)

Opis: Fragment kodu JavaScript z assets/js/scripts.js, odpowiedzialny za dynamiczny slider opinii na stronie głównej. Używa jQuery do animacji przełączania opinii co 5 sekund.



```
// assets/js/scripts.js
var currentIndex = 0;
var opinions = $('.opinion');
function showNextOpinion() {
    opinions.eq(currentIndex).fadeOut(500, function() {
        currentIndex = (currentIndex + 1) % opinions.length;
        opinions.eq(currentIndex).fadeIn(500);
    });
}
setInterval(showNextOpinion, 5000);
```

4. PHP – Pobieranie danych z bazy (index.php, sekcja 6.2)

Opis: Fragment kodu PHP z index.php, który pobiera potwierdzone rezerwacje z bazy danych na najbliższe 7 dni. Używa prepared statements dla bezpieczeństwa.

```
// index.php
<?php
require_once 'includes/config.php';
require_once 'includes/auth.php';

$stmt = $conn->prepare("SELECT sala, data FROM bookings WHERE status = 'confirmed' AND data BETWEEN
CURDATE() AND DATE_ADD(CURDATE(), INTERVAL 7 DAY) ORDER BY data");
$stmt->execute();
$result = $stmt->get_result();
while ($row = $result->fetch_assoc()) {
    echo "<div class='event'>Sala: " . htmlspecialchars($row['sala']) . ", Data: " .
htmlspecialchars($row['data']) . "</div>";
}
$stmt->close();
?>
```

5. PHP – Przetwarzanie rezerwacji (process_booking.php, sekcja 6.2)

Opis: Fragment kodu PHP z poprawionego pliku process_booking.php, który przetwarza formularz rezerwacji, zapisuje dane do tabel bookings i booking_drinks, oraz aktualizuje zapasy w inventory. Błąd mysqli_stmt object is already closed został rozwiązany przez poprawne zamykanie statementów.

```

// process_booking.php
<?php
session_start();
require_once 'includes/config.php';
require_once 'includes/auth.php';

if (!isLoggedIn()) {
    header("Location: login.php");
    exit;
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $user_id = $_SESSION['user_id'];
    $sala = $_POST['sala'];
    $data = $_POST['data'];
    $dodatkil = isset($_POST['dodatkil']) ? implode(" ", $_POST['dodatkil']) : '';
    $drinks = isset($_POST['drinks']) ? $_POST['drinks'] : [];

    // Wstaw rezerwacje
    $stmt = $conn->prepare("INSERT INTO bookings (user_id, sala, data, dodatki, status) VALUES (?, ?, ?, ?, 'pending')");
    $stmt->bind_param("iss", $user_id, $sala, $data, $dodatkil);
    $stmt->execute();
    $booking_id = $stmt->insert_id;
    $stmt->close();

    // Wstaw napoje i aktualizuj zapasy
    foreach ($drinks as $drink_id => $quantity) {
        if ($quantity > 0) {
            $stmt = $conn->prepare("INSERT INTO booking_drinks (booking_id, drink_id, quantity) VALUES (?, ?, ?)");
            $stmt->bind_param("iii", $booking_id, $drink_id, $quantity);
            $stmt->execute();
            $stmt->close();

            $stmt = $conn->prepare("UPDATE inventory SET quantity = quantity - ? WHERE sala = ? AND drink_id = ?");
            $stmt->bind_param("isi", $quantity, $sala, $drink_id);
            $stmt->execute();
            $stmt->close();
        }
    }

    header("Location: panel_klienta.php?success=Rezerwacja została złożona");
    exit;
}
?>

```

6. SQL – Przykładowe zapytania (database.sql, sekcja 7.4)

Opis: Fragment kodu SQL z pliku database.sql, zawierający zapytania do pobierania rezerwacji klienta oraz aktualizacji zapasów napojów. Ilustruje relacje między tabelami bookings, booking_drinks i drinks.

```

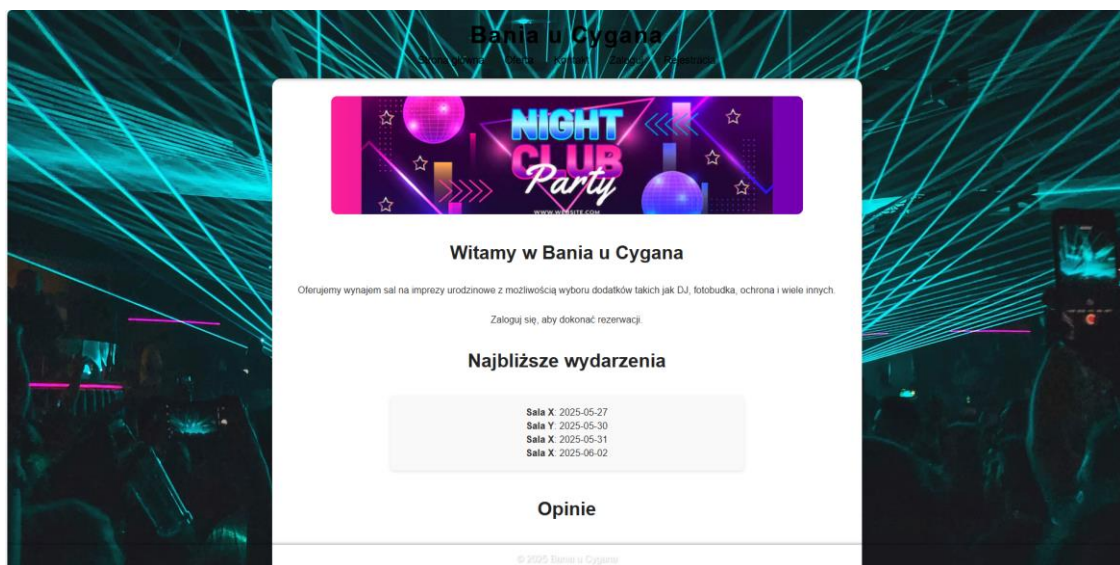
-- Pobieranie rezerwacji klienta z napojami
SELECT b.*, GROUP_CONCAT(CONCAT(d.name, ': ', bd.quantity)) AS drinks
FROM bookings b
LEFT JOIN booking_drinks bd ON b.id = bd.booking_id
LEFT JOIN drinks d ON bd.drink_id = d.id
WHERE b.user_id = ?
GROUP BY b.id;

-- Aktualizacja zapasów przy rezerwacji
UPDATE inventory SET quantity = quantity - ?
WHERE sala = ? AND drink_id = ?;

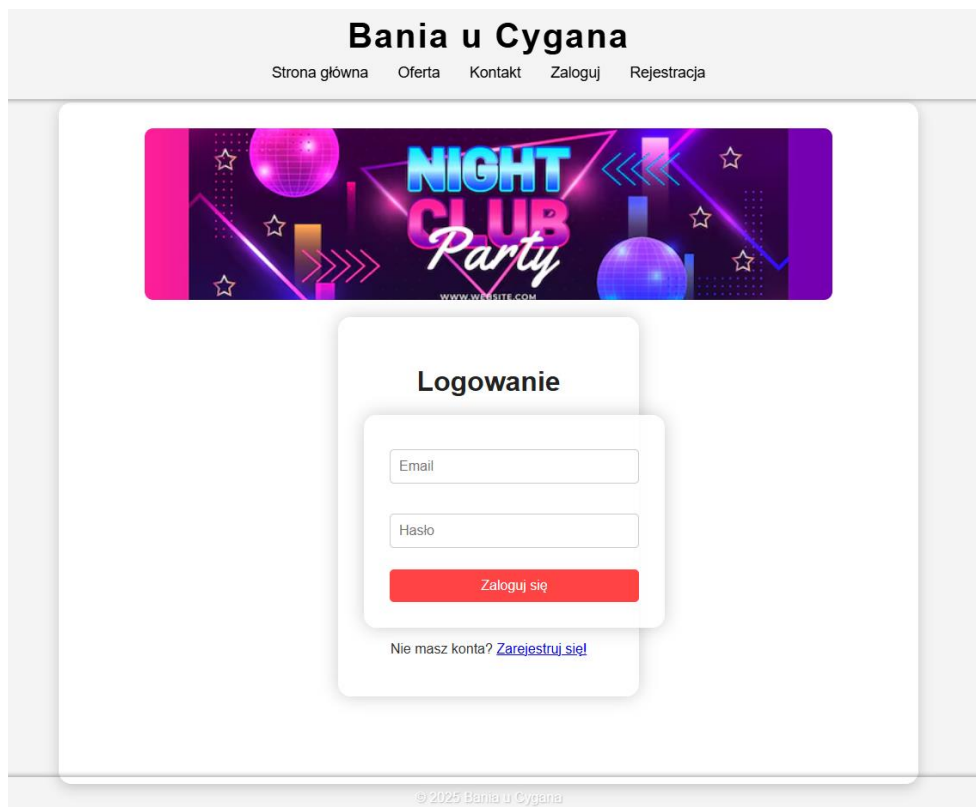
```

10.2. Zrzuty ekranu działania systemu

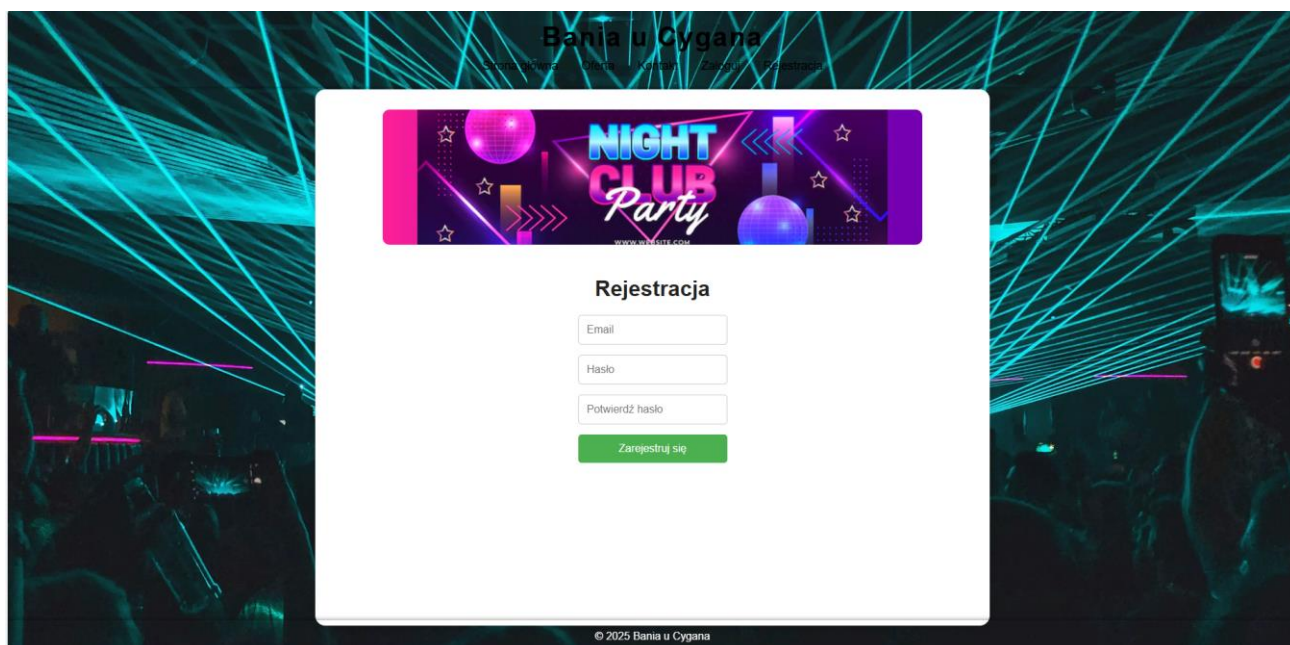
Index.php



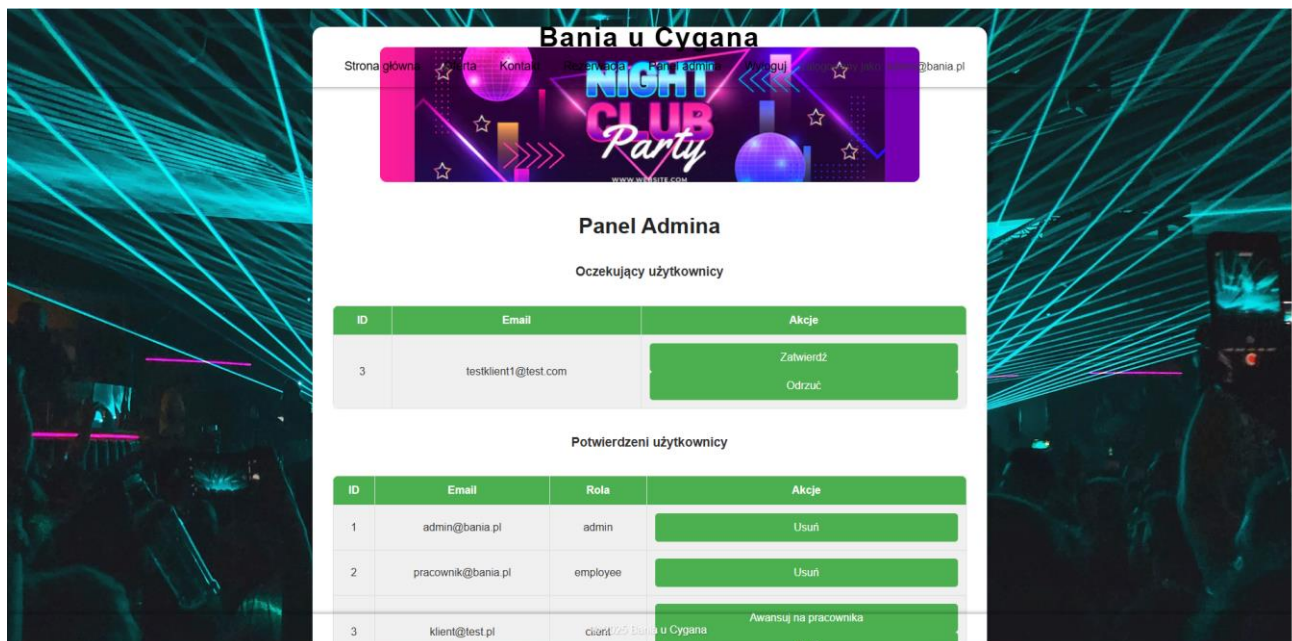
Logowanie



Rejestracja



Panel admina:



Bania u Cygana
NIGHT CLUB Party

Strona główna | Oferta | Kontakt | Rezerwacja | Panel admina | Wyloguj | Zalogowany jako: admin@bania.pl

Panel Admina

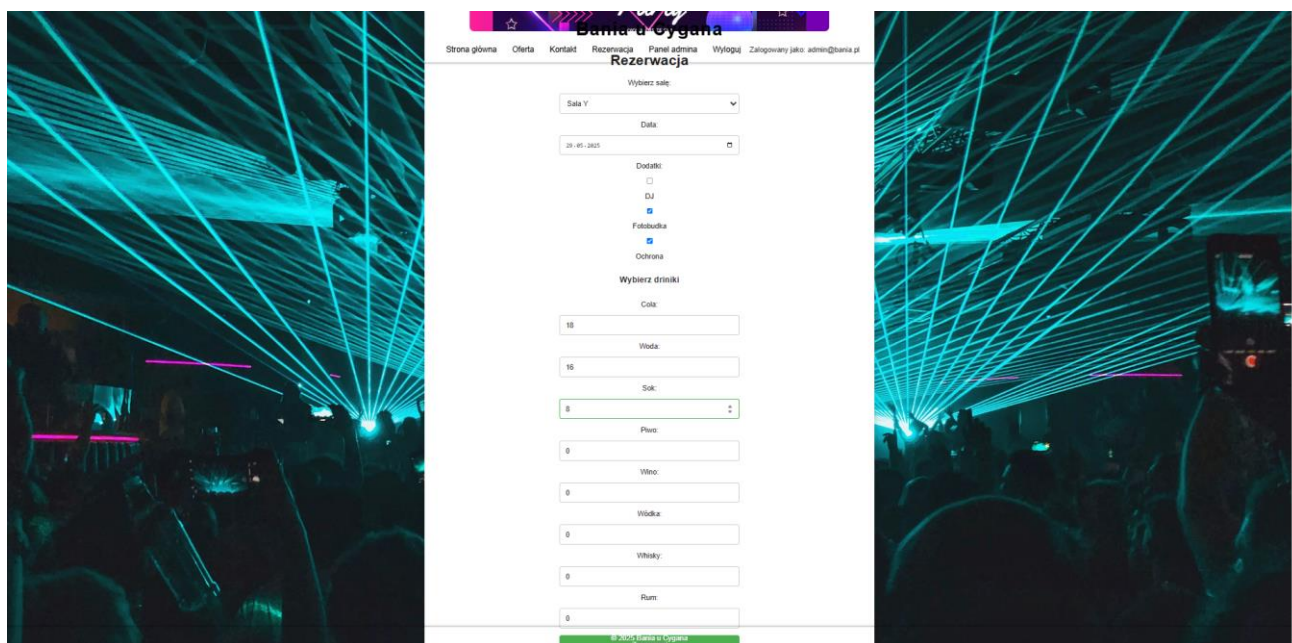
Oczekujący użytkownicy

ID	Email	Akcje
3	testklient1@test.com	Zatwierdź Odrzuć

Potwierdzeni użytkownicy

ID	Email	Rola	Akcje
1	admin@bania.pl	admin	Usuń
2	pracownik@bania.pl	employee	Usuń
3	klient@test.pl	ciężki	Awansuj na pracownika

Panel rezerwacji (pomniejszony)



Bania u Cygana
NIGHT CLUB Party

Strona główna | Oferta | Kontakt | Rezerwacja | Panel admina | Wyloguj | Zalogowany jako: admin@bania.pl

Rezerwacja

Wybierz salę:

Sala Y

Data:

20-01-2025

Dodatk:

☐ DJ

☒ Fotobudka

☒ Ochrona

Wybierz drinki:

Cola:

15

Woda:

15

Sok:

8

Plan:

0

Wino:

0

Wódka:

0

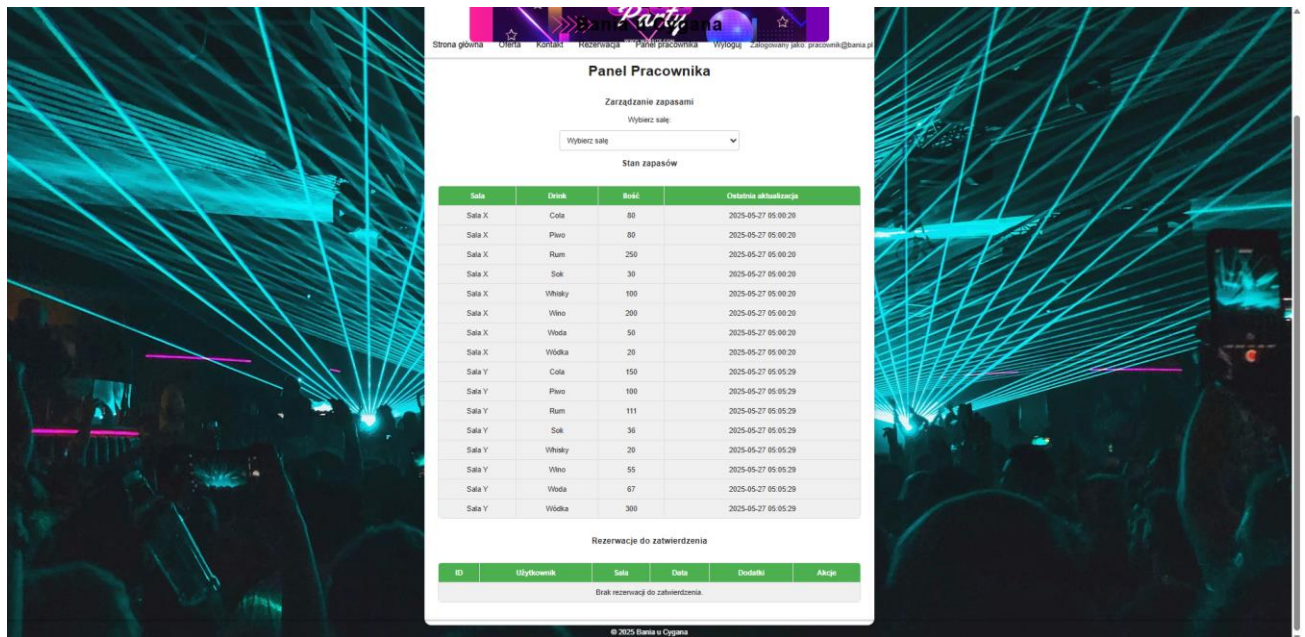
Whisky:

0

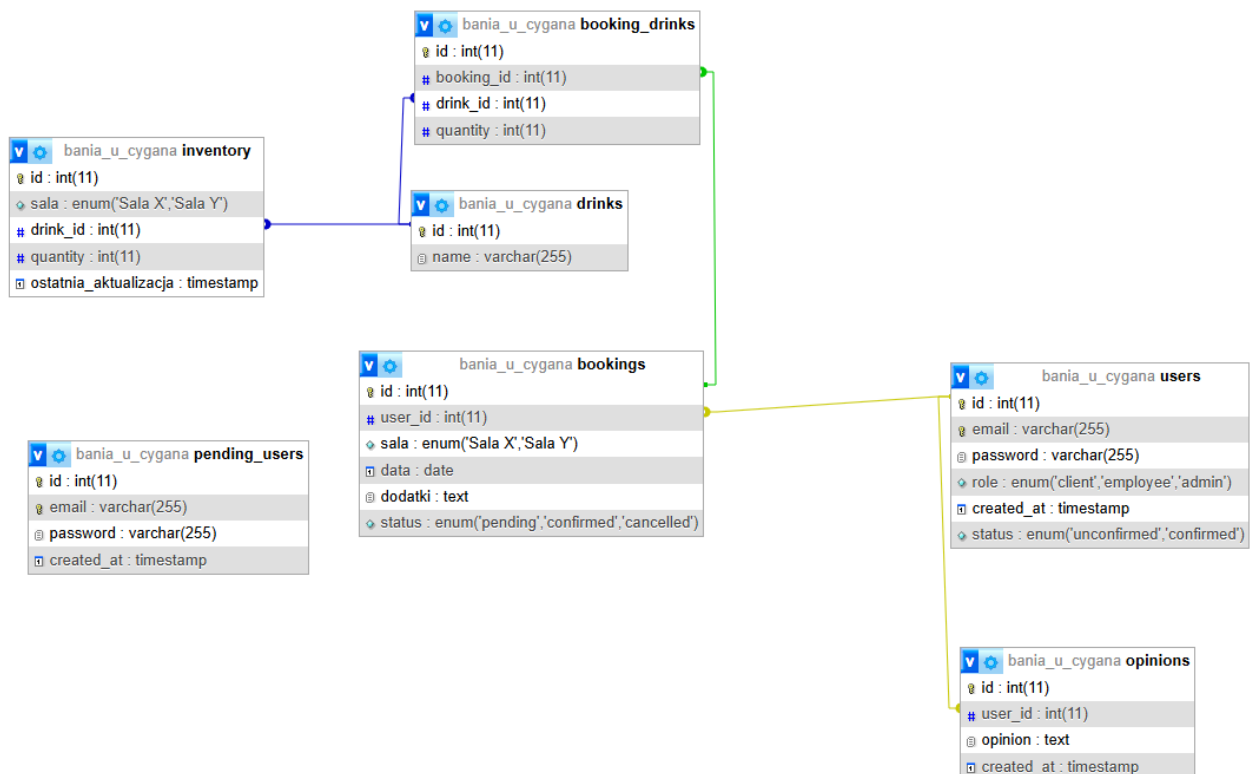
Rum:

0

Panel pracownika(pomniejszony)



10.3. Diagram bazy danych



10.4. Lista błędów/uwag i pomysłów na ulepszenia

Błędy/uwagi:

- • CSS nie zawsze działa prawidłowo, np. elementy mogą się przesuwać na mniejszych ekranach lub w różnych przeglądarkach, strona logowania nie wyświetlana jest prawidłowo.
- Niedokonczone i błędnie działające opinie

Pomysły na ulepszenia:

- Wdrożenie hashowania haseł za pomocą funkcji `password_hash()` w PHP, aby zwiększyć bezpieczeństwo przechowywania danych użytkowników.
- Dodanie powiadomień e-mail o nowych rezerwacjach, potwierdzeniach lub anulowaniach, np. za pomocą biblioteki PHPMailer.
- Poprawa responsywności