|  |  |  |
| --- | --- | --- |
| **DOKUMENTACJA PROJEKTU** | | **BD2** |
| **Autor** | Dawid Łukasz |  |
| **Kierunek, rok** | Informatyka, II rok, st. stacjonarne (3,5-l) |  |
| **Temat projektu** | *System rezerwacji biletów* |  |

Spis treści

[1. Wstęp 2](#_Toc168274420)

[3. Narzędzia i technologie 2](#_Toc168274421)

[4. Baza danych 3](#_Toc168274422)

[4.1. Diagram ERD 3](#_Toc168274423)

[4.2. Opis bazy danych 3](#_Toc168274424)

# Opis

System Rezerwacji Biletów ma na celu umożliwienie użytkownikom przeglądania dostępnych wydarzeń (np. filmy, koncerty, przedstawienia teatralne), rezerwacji miejsc oraz zakupu biletów przez internet. Użytkownicy mogą przeglądać wydarzenia według kategorii, daty i miejsca. Po wybraniu interesującego wydarzenia, mogą wybrać miejsca na sali i dokonać płatności online. System będzie również obsługiwał zarządzanie wydarzeniami przez administratorów, w tym dodawanie nowych wydarzeń, konfigurację sali (miejsca, sektory) i monitorowanie sprzedaży biletów.

# Funkcjonalności:

Przeglądanie wydarzeń:

Użytkownicy mogą przeglądać dostępne wydarzenia według różnych kryteriów, takich jak kategoria (film, koncert, teatr), data, miejsce, czy popularność. Opcje filtrowania i sortowania pomogą znaleźć wydarzenia odpowiadające ich zainteresowaniom.

Szczegóły wydarzeń:

Po wybraniu wydarzenia, użytkownicy mogą zobaczyć szczegółowe informacje, takie jak opis wydarzenia, lokalizacja, czas trwania, dostępne daty oraz ceny biletów.

Rezerwacja i zakup:

Użytkownicy mogą wybrać konkretne miejsca na planie sali (jeśli to koncert lub kino) i dokonać rezerwacji biletów. Proces zakupu powinien umożliwiać płatności online poprzez różne metody, takie jak karty kredytowe, PayPal, czy inne lokalne systemy płatności.

Zarządzanie użytkownikami:

System powinien umożliwiać rejestrację i logowanie użytkowników, zarządzanie ich profilami, przeglądanie historii zakupów i zarządzanie rezerwacjami.

Zarządzanie wydarzeniami dla administratorów:

Administratorzy systemu mogą dodawać nowe wydarzenia, edytować istniejące, zarządzać harmonogramami, cenami biletów, konfiguracją miejsc, a także monitorować sprzedaż biletów i analizować dane sprzedaży.

# Narzędzia i technologie

* **HTML 5**  
  HTML (HyperText Markup Language) to podstawowy język znaczników używany do tworzenia i strukturyzowania treści na stronach internetowych. HTML pozwala na definiowanie elementów, które tworzą strukturę strony, takich jak nagłówki, paragrafy, linki, obrazy, listy i inne. Każdy element w HTML jest reprezentowany przez tagi, które otaczają zawartość elementu.
* **CSS 3**  
  CSS (Cascading Style Sheets) to język stylizacji używany do opisania wyglądu i formatu dokumentu napisanego w języku znaczników, takim jak HTML. CSS pozwala na oddzielenie struktury strony (HTML) od jej wyglądu (stylu). Dzięki CSS możemy kontrolować wygląd różnych elementów na stronie, takich jak kolory, czcionki, odstępy, układ i wiele innych.
* **JavaScript ES6**  
  JavaScript to wszechstronny język programowania, który jest szeroko stosowany w tworzeniu interaktywnych i dynamicznych stron internetowych. Jest to język skryptowy, który działa po stronie klienta.
* **Node.js**  
  Otwartoźródłowa platforma programistyczna, która pozwala na uruchamianie kodu JavaScript poza przeglądarką internetową. Jest to środowisko wykonawcze oparte na silniku V8 firmy Google.
* **Oracle Database**  
  Oracle Database to relacyjny system zarządzania bazami danych (RDBMS) opracowany przez Oracle Corporation. Jest to jedno z najbardziej zaawansowanych i szeroko stosowanych narzędzi do zarządzania danymi, szczególnie w dużych przedsiębiorstwach i instytucjach finansowych.
* **Oracle SQL/PL SQL**  
  SQL to standardowy język zapytań używany do komunikacji z relacyjnymi bazami danych. Umożliwia wykonywanie różnych operacji na danych, takich jak pobieranie, wstawianie, aktualizowanie i usuwanie danych. PL/SQL to proceduralne rozszerzenie języka SQL stworzone przez Oracle Corporation. PL/SQL wprowadza programistyczne funkcje do SQL, umożliwiając pisanie pełnowartościowych programów i skryptów z użyciem struktur kontrolnych, takich jak pętle, instrukcje warunkowe, procedury, funkcje i pakiety.
* **Visual Studio Code**  
  Wieloplatformowy edytor kodu źródłowego.

# Baza danych

## Diagram ERD

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Strona internetowa

Opis wygenerowany automatycznie

## Opis bazy danych

Tabela użytkownicy przechowuje podstawowe informacje o każdym użytkowniku systemu. Każdy użytkownik ma unikalny identyfikator, nazwę użytkownika, adres email, hasło oraz rolę (np. klient, administrator), co umożliwia zarządzanie dostępem i uprawnieniami w systemie.

Tabela wydarzenia zawiera informacje o wszystkich wydarzeniach dostępnych w systemie, takich jak filmy, koncerty czy przedstawienia teatralne. Każde wydarzenie ma swój tytuł, opis, datę, lokalizację i czas utworzenia.

Tabela bilety przechowuje dane o biletach na poszczególne wydarzenia, w tym numer miejsca, cenę, status (dostępny, zarezerwowany, sprzedany) i czas utworzenia. Każdy bilet jest powiązany z określonym wydarzeniem.

Tabela miejsca opisuje miejsca, w których odbywają się wydarzenia, takie jak sale kinowe, teatry czy areny koncertowe. Każde miejsce ma swoją nazwę, pojemność, adres i czas utworzenia.

Tabela wydarzenia\_miejsca to tabela pośrednicząca umożliwiająca przypisanie wielu miejsc do jednego wydarzenia i na odwrót. Dzięki tej tabeli system może obsługiwać wydarzenia odbywające się w różnych lokalizacjach.

Tabela rezerwacje zarządza rezerwacjami biletów dokonywanymi przez użytkowników. Każda rezerwacja jest powiązana z użytkownikiem, który ją dokonał, i zawiera informacje o dacie rezerwacji, statusie (zarezerwowana, opłacona, anulowana), kwocie oraz czasie utworzenia.

## Opis relacji

Użytkownicy **i** Rezerwacje

* **Relacja**: Jeden do wielu
* **Opis**: Każdy użytkownik może dokonać wielu rezerwacji, ale każda rezerwacja jest przypisana do jednego użytkownika.
* **Implementacja**: Klucz obcy ID\_Użytkownika w tabeli Rezerwacje odnosi się do klucza głównego ID\_Użytkownika w tabeli Użytkownicy.

Wydarzenia **i** Bilety

* **Relacja**: Jeden do wielu
* **Opis**: Każde wydarzenie może mieć wiele biletów, ale każdy bilet jest przypisany do jednego wydarzenia.
* **Implementacja**: Klucz obcy ID\_Wydarzenia w tabeli Bilety odnosi się do klucza głównego ID\_Wydarzenia w tabeli Wydarzenia.

Wydarzenia **i** Wydarzenia\_Miejsca

* **Relacja**: Jeden do wielu
* **Opis**: Każde wydarzenie może być zorganizowane w wielu miejscach, ale każda pojedyncza lokalizacja w tabeli Wydarzenia\_Miejsca jest przypisana do jednego wydarzenia.
* **Implementacja**: Klucz obcy ID\_Wydarzenia w tabeli Wydarzenia\_Miejsca odnosi się do klucza głównego ID\_Wydarzenia w tabeli Wydarzenia.

Miejsca **i** Wydarzenia\_Miejsca

* **Relacja**: Jeden do wielu
* **Opis**: Każde miejsce może być miejscem przeprowadzenia wielu wydarzeń. Każda pozycja w tabeli Wydarzenia\_Miejsca określa, że konkretne miejsce jest wykorzystywane dla danego wydarzenia.
* **Implementacja**: Klucz obcy ID\_Miejsca w tabeli Wydarzenia\_Miejsca odnosi się do klucza głównego ID\_Miejsca w tabeli Miejsca.

Wydarzenia **i** Miejsca **przez** Wydarzenia\_Miejsca

* **Relacja**: Wielu do wielu
* **Opis**: Jedno wydarzenie może odbywać się w wielu miejscach, a jedno miejsce może być używane dla wielu wydarzeń.
* **Implementacja**: Relacja wielu do wielu jest realizowana przez tabelę pośredniczącą Wydarzenia\_Miejsca, która zawiera klucze obce ID\_Wydarzenia i ID\_Miejsca, tworząc połączenia między tabelami Wydarzenia i Miejsca.