

Opis tabel bazy danych

1. Books - Tabela przechowująca informacje o książkach w bibliotece

- **book_id** (PK, Integer, Auto-increment): Unikalny identyfikator książki.
- **title** (Varchar, 255): Tytuł książki.
- **author** (Varchar, 255): Autor książki.
- **isbn** (Varchar, 13): Numer ISBN, unikalny dla każdej książki.
- **year** (Integer): Rok wydania książki.
- **available** (Boolean): Status dostępności książki, gdzie true oznacza, że książka jest dostępna, a false, że jest aktualnie wypożyczona.

2. Users - Tabela przechowująca dane użytkowników biblioteki

- **user_id** (PK, Integer, Auto-increment): Unikalny identyfikator użytkownika.
- **first_name** (Varchar, 100): Imię użytkownika.
- **last_name** (Varchar, 100): Nazwisko użytkownika.
- **email** (Varchar, 255): Adres e-mail użytkownika, musi być unikalny.
- **phone_number** (Varchar, 15): Numer telefonu użytkownika, musi być unikalny.

3. Loans - Tabela rejestrująca wypożyczenia książek

- **loan_id** (PK, Integer, Auto-increment): Unikalny identyfikator wypożyczenia.
- **user_id** (FK, Integer, References Users(user_id)): Identyfikator użytkownika wypożyczającego książkę.
- **book_id** (FK, Integer, References Books(book_id)): Identyfikator książki, która jest wypożyczana.
- **loan_date** (Date): Data wypożyczenia książki.
- **return_date** (Date): Planowana data zwrotu książki.

4. Categories - Tabela przechowująca kategorie książek

- **category_id** (PK, Integer, Auto-increment): Unikalny identyfikator kategorii.
- **category_name** (Varchar, 100): Nazwa kategorii, np. „literatura piękna”, „naukowa”.

5. BookCategories - Tabela pośrednicząca, która mapuje książki na ich kategorie

- **book_id** (FK, Integer, References Books(book_id)): Identyfikator książki.

- **category_id** (FK, Integer, References Categories(category_id)): Identyfikator kategorii, do której przypisana jest książka.

Architektura projektu

Projekt zbudowany jest na architekturze klient-serwer, gdzie serwer Node.js obsługuje zarówno statyczne zasoby, jak i endpointy REST API. Aplikacja składa się z trzech głównych warstw: warstwy dostępu do danych, warstwy serwera aplikacji oraz warstwy klienta.

1. Warstwa dostępu do danych (PostgreSQL)

Warstwa bazy danych PostgreSQL przechowuje wszystkie informacje związane z książkami, użytkownikami, wypożyczeniami oraz kategoriami.

2. Warstwa serwera aplikacji (Node.js, Express, Sequelize)

Warstwa serwera napisana z użyciem technologii node.js, używająca Express do obsługi żądań HTTP. Zawiera kontroler, oraz model danych zarządzany przy użyciu Sequelize.

3. Warstwa klienta (React)

Warstwa frontendu (klienta) będzie dostarczać użytkownikowi interfejs do interakcji z systemem bibliotecznym. Prosta aplikacja HTML/JavaScript, napisana z użyciem technologii React komunikująca się z backendem przy użyciu Fetch API.