

Kraków 08.06.2024

**Sylwia Jaworska
Grzegorz Listwan
Krzysztof Pacura**

Wnioski i Sprawozdanie z Tworzenia Aplikacji Bibliotecznej

Wprowadzenie

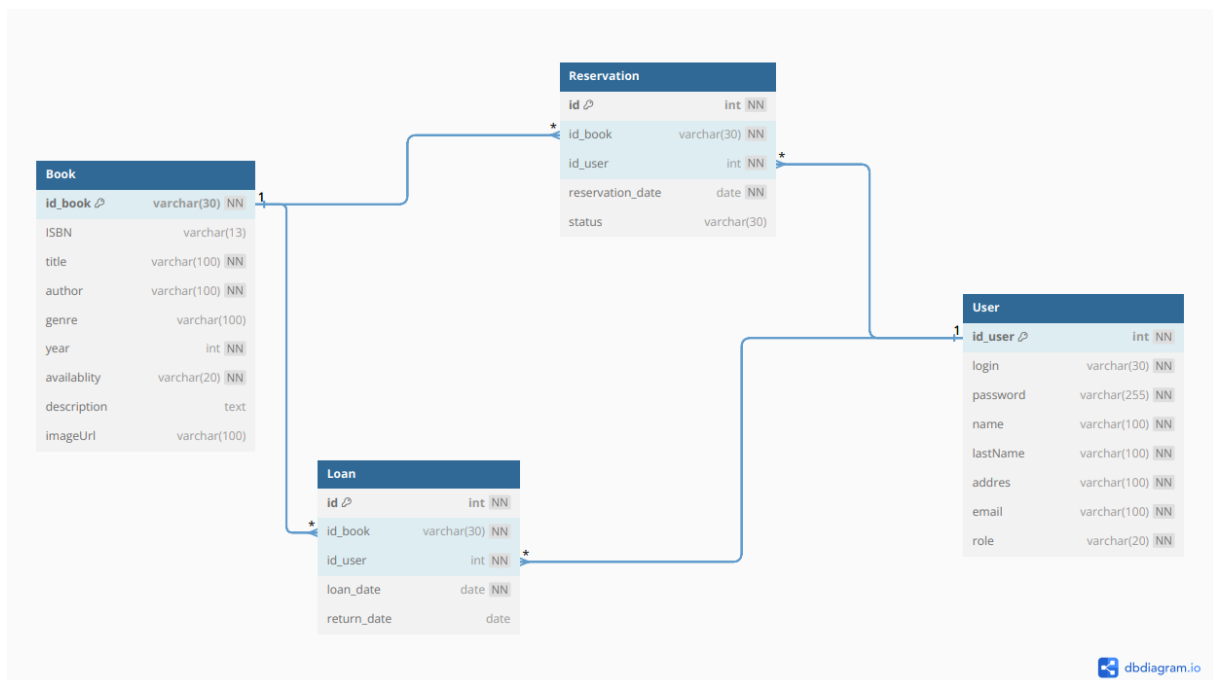
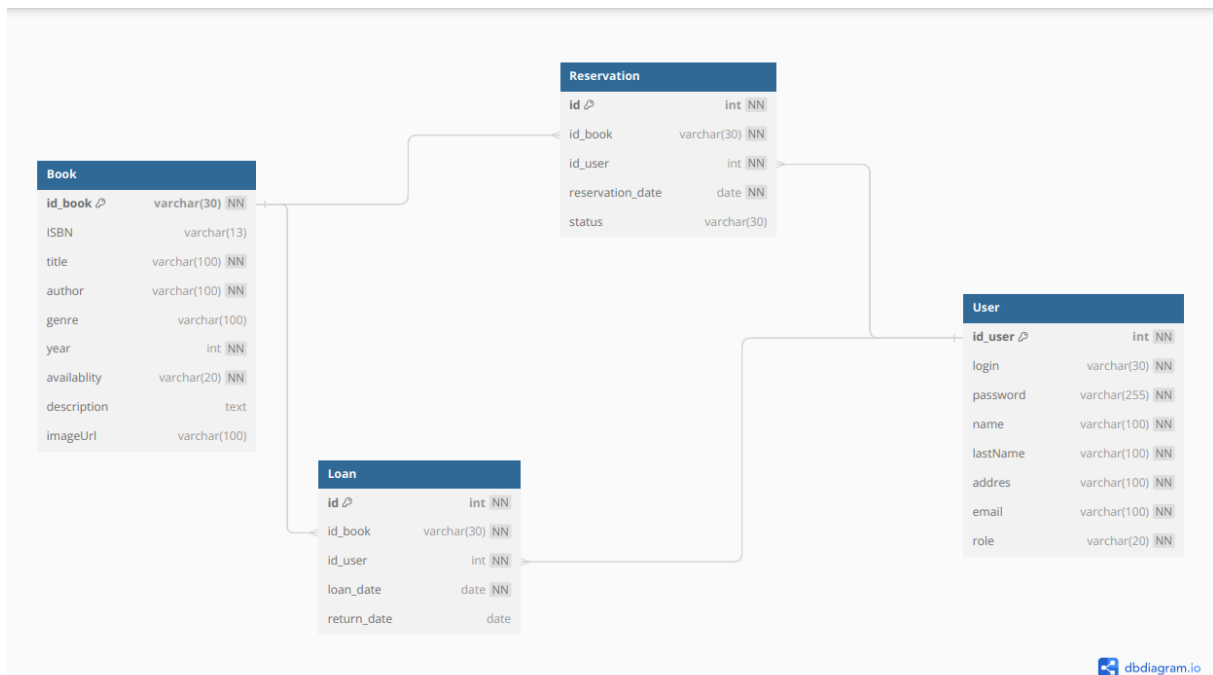
Celem projektu było stworzenie aplikacji bibliotecznej, która umożliwia zarządzanie użytkownikami, książkami oraz operacjami wypożyczeń i rezerwacji. Backend aplikacji został zaimplementowany przy użyciu frameworka Express.js, natomiast frontend w React. Cały proces obejmował konfigurację serwera, połączenie z bazą danych, implementację modeli, tras oraz middleware, a także zapewnienie komunikacji między frontendem a backendem.

Konfiguracja Backend

Na początku skonfigurowaliśmy serwer Express.js, który nasłuchuje na odpowiednim porcie. Wykorzystaliśmy middleware takie jak `'body-parser'`, `'cookie-parser'` oraz `'express-session'`, aby obsługiwać żądania, zarządzać ciasteczkami i sesjami użytkowników. Ponadto, skonfigurowaliśmy CORS (Cross-Origin Resource Sharing) za pomocą pakietu `'cors'`, aby umożliwić komunikację między frontendem a backendem działającym na różnych portach.

Połączenie z Bazą Danych

Do zarządzania bazą danych użyliśmy Sequelize, ORM dla Node.js. Zdefiniowaliśmy modele dla użytkowników (User), książek (Book), rezerwacji (Reservation), wypożyczeń (Loan) oraz logów (Logi). Modele te zostały powiązane odpowiednimi relacjami, co pozwoliło na kompleksowe zarządzanie danymi w bazie.



Implementacja Modeli

Modele zostały zaimplementowane w plikach umieszczonych w katalogu `models`. Każdy model zawiera definicję tabeli oraz jej kolumn, a także relacje między tabelami. Na przykład, model `User` posiada relacje `hasMany` z modelami `Reservation`, `Loan` i `Logi`, natomiast model `Book` ma relacje `hasMany` z modelami `Reservation` i `Loan`.

Implementacja Tras

Trasy aplikacji zostały zaimplementowane w katalogu ``routes``. Każdy plik w tym katalogu odpowiada za zestaw operacji związanych z konkretnym modelem. Przykładowo, plik ``user.js`` zawiera trasy do logowania, wylogowania oraz sprawdzania sesji użytkownika. W pliku ``book.js`` znajdują się trasy do dodawania nowych książek, pobierania wszystkich książek oraz wyszukiwania książek na podstawie różnych kryteriów. Pliki ``loan.js`` i ``reservation.js`` obsługują operacje związane z wypożyczeniami i rezerwacjami książek.

Zarządzanie Sesjami i Ciasteczkami

Sesje i ciasteczka zostały skonfigurowane za pomocą ``express-session`` i ``cookie-parser``. Sesje są używane do przechowywania danych użytkownika po zalogowaniu, co pozwala na zachowanie stanu użytkownika między różnymi żądaniami HTTP. Ciasteczka umożliwiają przesyłanie informacji o sesji między klientem a serwerem.

Frontend

Aplikacja frontendowa jest zbudowana przy użyciu Reacta i składa się z wielokrotnego użytku komponentów. Główne funkcjonalności obejmują zarządzanie użytkownikami (logowanie i rejestracja), przeglądanie katalogu książek, rezerwacje oraz śledzenie historii wypożyczeń. Funkcjonalność:

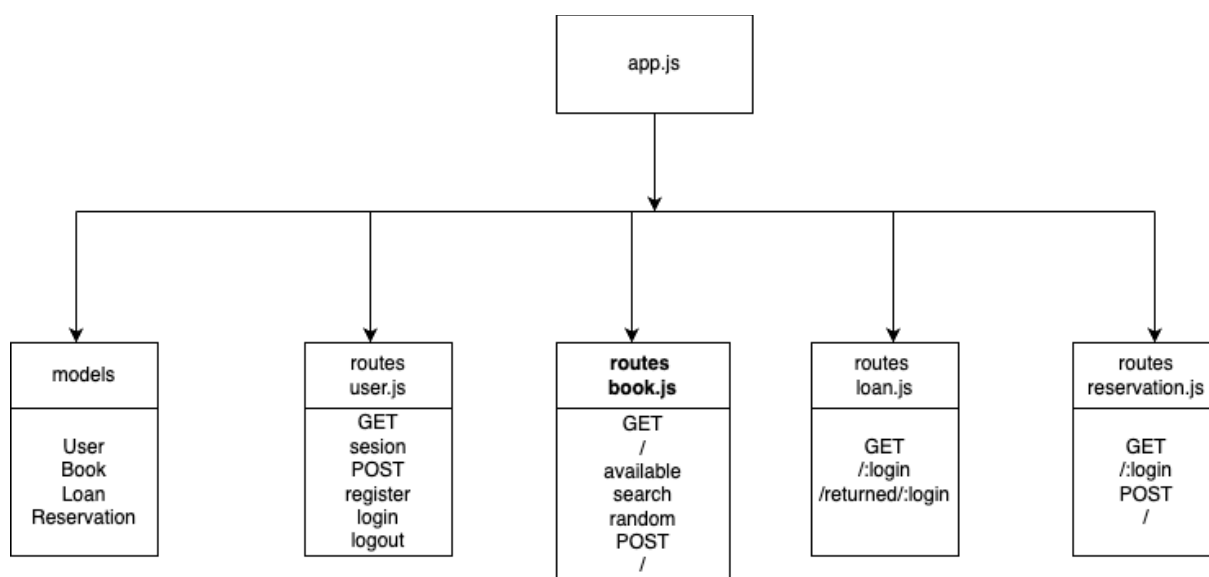
- Aplikacja jest przyjazna dla użytkownika i zapewnia intuicyjny interfejs.
- Zawiera funkcjonalności do korzystania z biblioteki online, takie jak logowanie, rejestracja, przeglądanie książek, rezerwacja oraz śledzenie historii wypożyczeń.
- Struktura aplikacji jest modularna, co ułatwia dodawanie nowych funkcjonalności i utrzymanie kodu.
- Każdy komponent ma jasno określoną odpowiedzialność, co sprawia, że aplikacja jest łatwa w zarządzaniu.
- Aplikacja jest estetycznie zaprojektowana, z naciskiem na czytelność i spójność wizualną.
- Implementacja mechanizmu zapamiętywania danych logowania przy użyciu ciasteczek zwiększa wygodę użytkownika.

Testowanie

Testowanie aplikacji odbywało się przy użyciu Postmana, co pozwoliło na weryfikację poprawności działania endpointów backendu. Przeprowadziliśmy testy logowania, wylogowania, zarządzania książkami, rezerwacjami oraz wypożyczeniami. Każdy endpoint został przetestowany pod kątem różnych scenariuszy, takich jak poprawne dane, brakujące dane, czy błędne dane.

Diagram Struktury Aplikacji

Diagram struktury aplikacji został stworzony za pomocą narzędzia Draw.io. Diagram ten przedstawia relacje między poszczególnymi komponentami aplikacji, w tym serwerem, modelami, trasami oraz konfiguracją. Diagram pomógł w wizualizacji architektury aplikacji, co ułatwiło jej implementację i utrzymanie.



Wnioski

Projekt aplikacji bibliotecznej był ambitnym przedsięwzięciem, które wymagało integracji wielu technologii. Kluczowymi wyzwaniami było zarządzanie sesjami i ciasteczkami, konfiguracja CORS oraz implementacja kompleksowych relacji między modelami w bazie danych. Udało się jednak zrealizować wszystkie założone cele, a końcowy produkt jest funkcjonalną aplikacją, która umożliwia zarządzanie użytkownikami, książkami oraz operacjami bibliotecznymi.

Stworzenie aplikacji od podstaw pozwoliło na zdobycie cennych doświadczeń w zakresie programowania backendu w Node.js z użyciem Express, zarządzania bazą danych z wykorzystaniem Sequelize oraz tworzenia interaktywnych interfejsów użytkownika w React.