

Aplikacja webowa biblioteki

Marcin Chudy

mchudy@student.agh.edu.pl

Adrian Madej

adrianmadej@student.agh.edu.pl

Opis projektu:

Strona internetowa danej biblioteki. W bazie danych przechowywane są dane książek (wraz z jej autorami i wydawnictwami) oraz użytkowników danej biblioteki.

Użytkownik może przeglądać dostępne książki oraz je wypożyczać, ma również dostęp do swojej historii wypożyczeń.

Administrator może zatwierdzać zwrot danej książki.

Technologie

MongoDB

Angular

Express

Node

Uruchomienie aplikacji

Frontend

```
cd bd_project
npm install
ng serve --proxy-config proxy.conf.json
```

Backend

```
cd bd_project/backend
npm install
npm run dev
```

Schemat bazy danych

Mamy 4 kolekcje: Author

```
{
  "_id": {"$oid": "6460be09478e0704d8a9c19d"},
  "FirstName": "Adam",
  "LastName": "Mickiewicz",
  "BooksID": [{"$oid": "6460d5e70c6c0520811a03ef"}],
```

```
{
  "$oid": "6460d5e70c6c0520811a03ec"},
  {"$oid": "6460d5e70c6c0520811a03ed"},
  {"$oid": "6460d5e70c6c0520811a03ee"}],
  "Description": "Poeta, działacz polityczny, publicysta, filozof.
  Uważany za największego poetę polskiego romantyzmu."
}
```

Book

```
{
  "_id": {"$oid": "6460d5e70c6c0520811a03ef"},
  "Title": "Pan Wołodyjowski",
  "Description": "Trzecia i zarazem ostatnia powieść Trylogii Henryka
  Sienkiewicza.",
  "ReleaseDate": {"$numberInt": "1888"},
  "Genre": "Historyczna",
  "AuthorID": {"$oid": "6460be09478e0704d8a9c19c"},
  "InStock": {"$numberInt": "5"},
  "Copies": [
    {"BookID": {"$numberInt": "16"}, "PublisherID":
    {"$oid": "6460c22109009077c28112c5"},
    "Available": true, "Language": "polski"},
    {"BookID": {"$numberInt": "17"}, "PublisherID":
    {"$oid": "6460c22109009077c28112c5"},
    "Available": true, "Language": "polski"},
    {"BookID": {"$numberInt": "18"}, "PublisherID":
    {"$oid": "6460c22109009077c28112c5"},
    "Available": true, "Language": "polski"},
    {"BookID": {"$numberInt": "19"}, "PublisherID":
    {"$oid": "6460c22109009077c28112c5"},
    "Available": true, "Language": "polski"},
    {"BookID": {"$numberInt": "20"}, "PublisherID":
    {"$oid": "6460c22109009077c28112c5"},
    "Available": true, "Language": "polski"}]
}
```

Publisher

```
{
  "_id": {"$oid": "6460c22109009077c28112c5"},
  "Name": "Fabryka Słów",
  "Email": "fabryka@gmail.com",
  "Phone": "123456789",
  "BooksID": [{"BookID": {"$numberInt": "56"}, "PublisherID":
  {"$oid": "6460c22109009077c28112c5"}, "Available": true,
  "Language": "polski"}, {"BookID": {"$numberInt": "57"},
  "PublisherID": {"$oid": "6460c22109009077c28112c5"},
  "Available": true, "Language": "polski"}, ..., {"BookID":
  {"$numberInt": "85"}, "PublisherID": {"$oid": "6460c22109009077c28112c5"},
  "Available": true, "Language": "polski"}],
  "Address": {
    "Country": "Polska",
    "City": "Rzeszów",
    "PostalCode": "11-111",
    "Street": "Piastowska"
  }
}
```

```

    }
  }
}

User
{
  "_id":{"$_oid":"6460d1e3ac388251224c6732"},
  "FirstName":"Robert",
  "LastName":"Lewandowski",
  "Email":"robert@gmail.com",
  "Phone":"546544542",
  "Pesel":"02112341231",
  "Borrow":[
    {
      "BookID":{"$numberInt":"7"},
      "BorrowDate":{"$date":{"$numberLong":"1678143600000"}},
      "ReturnDate":{"$date":{"$numberLong":"1680904800000"}},
      "ExpectedReturnDate":{"$date":{"$numberLong":"1684101600000"}}
    }
  ]
}

```

Zapytania do bazy danych

Wyciąganie danych o wszystkich książkach

```

const bookPipeline = [
  {
    $lookup: {
      from: "authors",
      localField: "AuthorID",
      foreignField: "_id",
      as: "Author"
    }
  },
  {
    $unwind: "$Author"
  },
];

```

```

const bookData = await
db.collection("book").aggregate(bookPipeline).toArray();

```

Wyciągamy kolekcję book, łącząc ją przy pomocy lookup z kolekcją authors.

Wyciąganie danych o książkach danego autora

```

const firstName, lastName
const bookPipeline = [
  {
    $lookup: {
      from: "authors",
      localField: "AuthorID",

```

```

        foreignField: "_id",
        as: "Author"
      }
    },
    {
      $unwind: "$Author"
    },
    {
      $match: {
        "Author.FirstName": firstName,
        "Author.LastName": lastName
      }
    }
  ],
];

```

```

const bookData = await
db.collection("book").aggregate(bookPipeline).toArray();

```

Wyciągamy informacje o książkach danego autora znając jego imię oraz nazwisko.

Wyciąganie wszystkich danych o danym użytkowniku

```

const userId
const userPipeline = [
  {
    $match: {
      _id: new mongoose.Types.ObjectId(userId)
    }
  },
  {
    $unwind: "$Borrow"
  },
  {
    $lookup: {
      from: "book",
      localField: "Borrow.BookID",
      foreignField: "Copies.BookID",
      as: "Borrow.bookDetails"
    }
  },
  {
    $unwind: "$Borrow.bookDetails"
  },
  {
    $group: {
      _id: "$_id",
      username: { $first: "$username" },
      borrowedBooks: { $push: "$Borrow" }
    }
  }
]

```

```
    const userData = await
db.collection("user").aggregate(userPipeline).toArray();
```

Wyciągamy wszystkie dane o danym użytkowniku przy znajomości jego `_id`. Dodatkowo tworzymy tablicę obiektów Book wypożyczonych przez niego książek.

Wyciąganie danych o autorach

```
const authorsData = await db.collection('authors').aggregate([
  {
    $lookup: {
      from: 'book',
      localField: 'BooksID',
      foreignField: '_id',
      as: 'books'
    }
  }
]).toArray();
```

Wyciągamy wszystkie dane o autorach i przy pomocy BookID tworzymy listę książek napisanych przez danego autora.

Funkcja wypożyczenia danej książki

```
const Copy = await db.collection("book").aggregate([
  { $match: { _id: new mongoose.Types.ObjectId(_id) } },
  { $unwind: '$Copies' },
  { $match: { 'Copies.Available': true } },
  { $sample: { size: 1 } }
]).toArray();

if (Copy.length === 0) {
  res.status(404).send('Brak dostępnych egzemplarzy');
  return;
}
const result = await db.collection("book").updateOne(
  {
    _id: new mongoose.Types.ObjectId(_id),
    'Copies.BookID': Copy[0].Copies.BookID
  },
  {
    $inc: { InStock: -1 },
    $set: { 'Copies.$.Available': false }
  }
);

if (result.modifiedCount === 0) {
  res.status(404).send('Egzemplarz książki nie jest dostępny.');
```

```
  } else {
    await db.collection("user").updateOne(
      { _id: new mongoose.Types.ObjectId(userId) },
      { $push: { Borrow: {
```

```

        BookID: Copy[0].Copies.BookID,
        BorrowDate: new Date(),
        ReturnDate: null,
        ExpectedReturnDate: new Date(new Date().getTime() + (14 * 24
* 60 * 60 * 1000))
    })))
    res.send('Książka została wypożyczona.');
```

Na początku sprawdzamy czy istnieje egzemplarz danej książki, który nie jest wypożyczony. Następnie zaznaczamy w bazie, że dany egzemplarz zostaje wypożyczony (jeśli któraś z czynności się nie powiedzie użytkownik zostaje o tym poinformowany). Na końcu dodajemy dany egzemplarz do książek wypożyczonych przez danego użytkownika.

Funkcja zwracająca egzemplarz do biblioteki

```

const BookID = parseInt(req.params.id);
const user = await db.collection('user').findOne({ 'Borrow.BookID':
BookID, 'Borrow.ReturnDate': null })
if (user) {
    const borrowedBook = user.Borrow.find((book) => book.BookID ===
BookID && book.ReturnDate == null);
    if (borrowedBook == null) {
        console.log(borrowedBook)
        res.status(404).send('Nikt nie wypożyczył książki o takim
ID.');
```

Na początku sprawdzamy czy istnieje użytkownik, który wypożyczył książkę o danym numerze id i czy posiada on jakąś wypożyczoną książkę, jeśli tak to sprawdzamy czy to poszukiwany przez nas egzemplarz jest wypożyczony. Jeśli wszystko się zgadza to

wpisujemy datę zwrocenia książki na aktualną datę oraz zapisujemy w bazie że egzemplarz jest gotowy to ponownego wypożyczenia. W przeciwnym przypadku wyrzucamy błąd.

Wyciąganie danych o konkretnym egzemplarzu

```
const BookID = parseInt(req.params.id);
try {
  let user = await db.collection('user').findOne({ 'Borrow.BookID':
BookID, 'Borrow.ReturnDate': null });
  let borrowedBook = user;
  if (borrowedBook) {
    borrowedBook = borrowedBook.Borrow.find((book) => book.BookID
=== BookID && book.ReturnDate == null);
    borrowedBook.user = user._id
  }
}
```

Przeszukujemy użytkowników w celu znalezienia, czy któryś z nich aktualnie ma wypożyczony dany egzemplarz.