**Book Reviews App**

**Overview**

The Book Reviews App is a web application designed to manage a collection of books and their reviews. This app allows users to view, add, update, and delete books, as well as add, update, and delete reviews for each book. It uses Express.js for the backend, MongoDB for data storage, and React.js for the frontend. Data is fetched from the Open Library API with pagination support.

**Table of Contents**

1. [Project Description](#project-description)
2. [Installation](#installation)
3. [Usage](#usage)
4. [API Endpoints](#api-endpoints)
5. [Frontend](#frontend)
6. [Seeding the Database](#seeding-the-database)
7. [Contributing](#contributing)
8. [License](#license)

**Project Description**

The Book Reviews App allows users to:

* **View all books** with pagination.

A white background with black lines

Description automatically generated

* **View details** of a specific book.

A screenshot of a computer

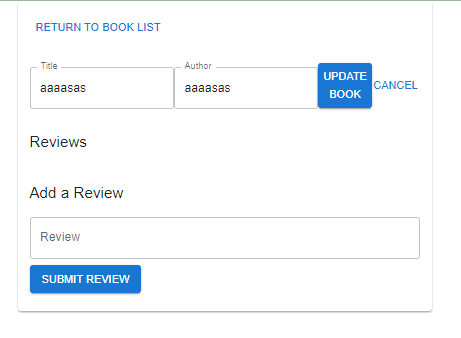
Description automatically generated

* **Add new books** to the collection. (new book is added)

A white background with black lines

Description automatically generated

* **Update existing books**.



A screenshot of a computer

Description automatically generated

A white rectangular object with black lines

Description automatically generated

* **Delete books** from the collection.

UpdatedBook is deleted by clicking the deletedBook Button

A white background with black lines

Description automatically generated

* **Add reviews** to books.

A screenshot of a review

Description automatically generated

* **Update reviews**.

A screenshot of a review

Description automatically generated

A screenshot of a computer

Description automatically generated

* **Delete reviews**.

A screenshot of a book list

Description automatically generated

The application is structured into three main components:

1. **Backend API**: Built with Express.js and MongoDB.
2. **Frontend**: Built with React.js and Material-UI.
3. **Database Seeding Script**: Populates the database with initial data from the Open Library API.

**Installation**

**Backend**

1. Clone the repository:

bash

Copy code

git clone <repository-url>

cd <repository-folder>

1. Navigate to the backend directory:

bash

Copy code

cd backend

1. Install dependencies:

bash

Copy code

npm install

1. Start the MongoDB server (ensure MongoDB is installed and running):

bash

Copy code

mongod

1. Start the backend server:

bash

Copy code

npm start

**Frontend**

1. Navigate to the frontend directory:

bash

Copy code

cd frontend

1. Install dependencies:

bash

Copy code

npm install

1. Start the React development server:

bash

Copy code

npm start

**Usage**

1. Open your browser and go to http://localhost:3000 to view the React app.
2. Use the interface to interact with the books and reviews.
3. For backend API testing, you can use tools like Postman or curl to interact with endpoints.

**API Endpoints**

**Books**

* **GET /books**: Retrieve all books with pagination.
  + **Query Parameters**: page (number), limit (number)
  + **Example**: /books?page=1&limit=10
* **GET /books/**

: Retrieve a specific book by ID.

* **POST /books**: Create a new book.
* **PUT /books/**

: Update a book by ID.

* **DELETE /books/**

: Delete a book by ID.

**Reviews**

* **POST /books/**

**/reviews**: Add a review to a specific book.

* **PUT /books/**

**/reviews/**

: Update a specific review for a book.

* **DELETE /books/**

**/reviews/**

: Delete a specific review for a book.

**Frontend**

The frontend application uses React.js and Material-UI for UI components. It allows users to:

* **View a list of books** with pagination.
* **Select a book** to view its details and manage reviews.
* **Add, update, and delete books** and reviews.

**File Structure**

* **src/App.js**: Main application component.
* **src/BookList.js**: Component for listing books with pagination.
* **src/BookDetails.js**: Component for displaying book details and managing reviews.

**Seeding the Database**

To populate the database with initial data from the Open Library API:

1. Navigate to the seeding script directory:

bash

Copy code

cd seed

1. Run the seeding script:

bash

Copy code

node seedDB.js

**Data Fetching and Pagination**

* **Data Source**: Open Library API.
* **Pagination**: Implemented on both the backend and frontend to handle large datasets. The backend supports pagination via query parameters (page and limit).

The seeding script fetches data from the Open Library API, handles pagination, and inserts it into the MongoDB database.

**Contributing**

Contributions are welcome! Please follow these steps:

1. Fork the repository.
2. Create a new branch (git checkout -b feature/your-feature).
3. Make your changes.
4. Commit your changes (git commit -am 'Add new feature').
5. Push to the branch (git push origin feature/your-feature).
6. Create a new Pull Request.