BOOK STORE APPLICATION USING MERN

PROJECT DOCUMENTATION

1. Introduction

- **Project Title:** BOOK STORE APPLICATION
- **Team Members:**
  - **Priyadarshini-2021503036**
  - **Madhumitha-2021503520**
  - **Shalini-2021503554**
  - **Shruthi-2021503556**

2. Project Overview

- **Purpose:**
  The Book Store Application is designed to facilitate the discovery of books in an organized and user-friendly environment. Users have access to a variety of features that enhance their browsing experience.
- **Features:**
  - **Show Books:** Users can browse a collection of books presented in visually appealing cards and tables, making it easy to find and explore various titles.
  - **Create New Books:** Admin users have the ability to add new books to the collection, ensuring a continuous and diverse selection for users.
  - **Edit Existing Books:** Authorized users can modify details of existing books, such as title, author, genre, ensuring accurate and up-to-date information.
  - **Delete Books:** Users with appropriate permissions can remove books from the collection, helping maintain the relevance of the library.
  - **Show Description:** Each book displays a detailed description, including plot summaries, reviews, and other pertinent information to assist users in their decisions.

3. Architecture

- **Frontend:**
  Built with React, the frontend architecture utilizes a component-based structure, enabling reusable components for book cards, forms, and navigation. The application state is managed with hooks and context, while Axios handles API requests to the backend.

- **Backend:**
  The backend is developed using Node.js and Express.js, structured to provide RESTful API endpoints that facilitate CRUD operations for books and user management. Middleware is implemented for handling authentication and validation of requests.

- **Database:**
  The application uses MongoDB as the database, with a schema that includes collections for users and books. Mongoose is used to model the data and simplify interactions with the database, providing a clear structure for data retrieval and manipulation.

4. Setup Instructions

- **Prerequisites:**
  - Node.js (v16 or higher)
  - MongoDB (local installation or MongoDB Atlas)
  - npm (Node package manager)

- **Installation:**

i. Navigate to the project directory:

```
bash

cd book-store
```

ii. Install dependencies for both the client and server:

```
bash

cd client

npm install

cd ../server

npm install
```

iii. Set up environment variables:
Create a .env file in the server directory and add necessary variables such as:

```
MONGO_URI=your_mongo_uri

JWT_SECRET=your_jwt_secret
```

5. Folder Structure

- **Client:**

java

```
/client
├── /public
├── /src
|   ├── /components      // Reusable UI components (BookCard, Header, etc.)
|   ├── /pages           // Main application pages (Home, About, etc.)
|   ├── /context         // Context and hooks for state management
|   ├── App.js
|   └── index.js
└── package.json
```

- **Server:**

arduino

```
/server
├── /models            // Mongoose models for users and books
├── /routes            // Routes for API endpoints
├── /config            // Configuration files, e.g., database connection
├── /controllers       // Controllers for handling business logic
├── server.js          // Main entry point for the server
└── package.json
```

6. Running the Application

- **Frontend:**
  Navigate to the client directory and run:

  bash

  ```
  npm start
  ```

- **Backend:**
  Navigate to the server directory and run:

  bash

  ```
  npm start
  ```

7. API Documentation

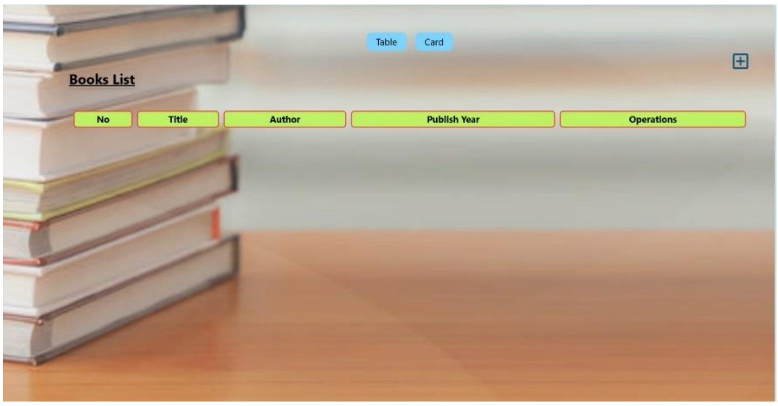
- **Endpoints:**
  - **GET /api/books**
    - **Description:** Retrieve all books
    - **Response:** JSON array of book objects
  - **POST /api/books**
    - **Description:** Add a new book to the collection
    - **Request Body:** { title, author, publish year }
    - **Response:** Confirmation message and added book object
  - **PUT /api/books/:id**
    - **Description:** Update an existing book's details
    - **Request Body:** { title, author, genre, publish year }
    - **Response:** Confirmation message and updated book object
  - **DELETE /api/books/:id**
    - **Description:** Remove a book from the collection
    - **Response:** Confirmation message

8. Authentication

- Authentication is managed through JWT (JSON Web Tokens). After successful login, a token is created and stored in local storage. This token is sent with subsequent requests for protected routes to verify user identity and permissions, allowing access to CRUD operations on books.

9. User Interface

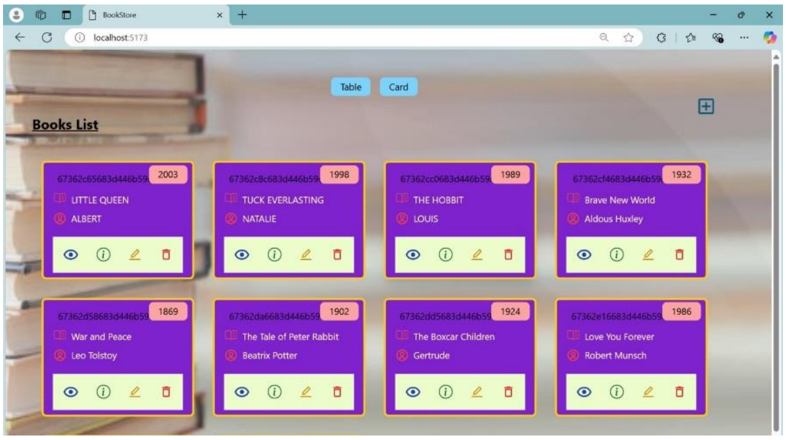
Home Page:



Books List

No	Title	Author	Publish Year	Operations

Book details:



Books List

67362c65683d446b59 2003
LITTLE QUEEN
ALBERT

67362c8c683d446b59 1998
TUCK EVERLASTING
NATALIE

67362cc0683d446b59 1989
THE HOBBIT
LOUIS

67362cf4683d446b59 1932
Brave New World
Aldous Huxley

67362d58683d446b59 1869
War and Peace
Leo Tolstoy

67362da6683d446b59 1902
The Tale of Peter Rabbit
Beatrix Potter

67362dd5683d446b59 1924
The Boxcar Children
Gertrude

67362e16683d446b59 1986
Love You Forever
Robert Munsch

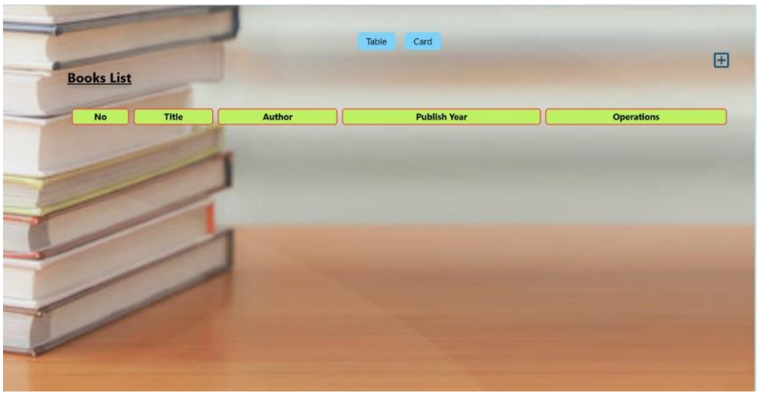
10. Testing

- **Testing Strategy:**
  The project employs Jest and React Testing Library for frontend testing and Mocha/Chai for the backend. Unit tests ensure that all critical functionalities, including CRUD operations for books, are functioning correctly.

11. Screenshots or Demo

- **Demo Link:** [MERN-DEMO-BOOK STORE - Google Drive]
- **Screenshots:**

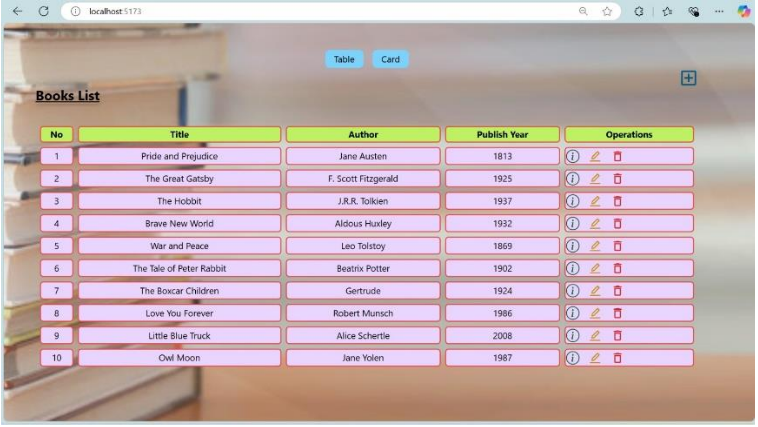
  HOME PAGE:



Books List

No	Title	Author	Publish Year	Operations

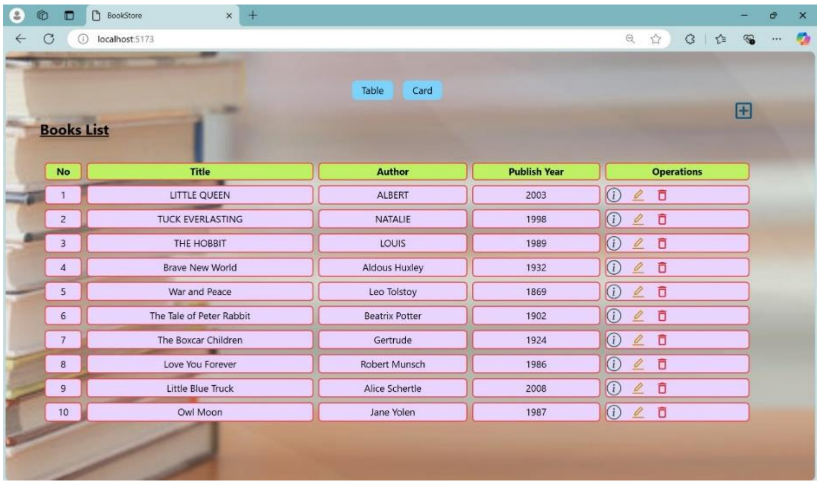
AFTER BOOK CREATION:



Books List

No	Title	Author	Publish Year	Operations
1	Pride and Prejudice	Jane Austen	1813	
2	The Great Gatsby	F. Scott Fitzgerald	1925	
3	The Hobbit	J.R.R. Tolkien	1937	
4	Brave New World	Aldous Huxley	1932	
5	War and Peace	Leo Tolstoy	1869	
6	The Tale of Peter Rabbit	Beatrix Potter	1902	
7	The Boxcar Children	Gertrude	1924	
8	Love You Forever	Robert Munsch	1986	
9	Little Blue Truck	Alice Schertle	2008	
10	Owl Moon	Jane Yolen	1987	

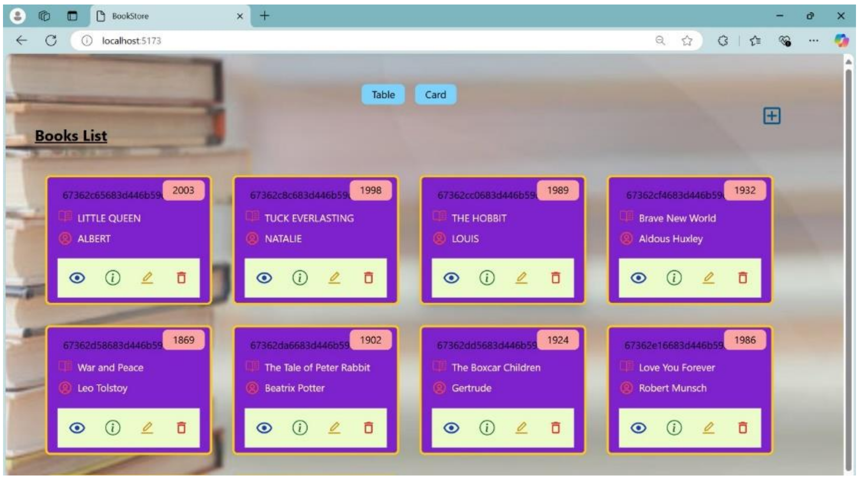
EDITING THE BOOKS:



Books List

No	Title	Author	Publish Year	Operations
1	LITTLE QUEEN	ALBERT	2003	
2	TUCK EVERLASTING	NATALIE	1998	
3	THE HOBBIT	LOUIS	1989	
4	Brave New World	Aldous Huxley	1932	
5	War and Peace	Leo Tolstoy	1869	
6	The Tale of Peter Rabbit	Beatrix Potter	1902	
7	The Boxcar Children	Gertrude	1924	
8	Love You Forever	Robert Munsch	1986	
9	Little Blue Truck	Alice Schertle	2008	
10	Owl Moon	Jane Yolen	1987	

DISPLAYING THE BOOKS DETAILS IN CARD FORMAT:



Books List

67362c65683d446b59 2003
LITTLE QUEEN
ALBERT

67362c8c683d446b59 1998
TUCK EVERLASTING
NATALIE

67362cc0683d446b59 1989
THE HOBBIT
LOUIS

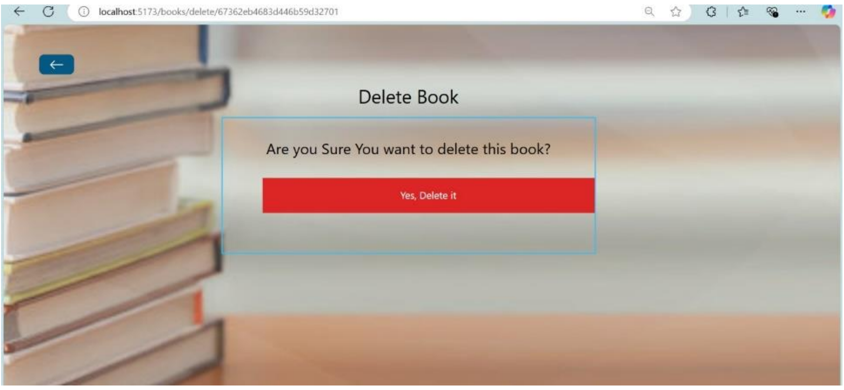
67362cf4683d446b59 1932
Brave New World
Aldous Huxley

67362d58683d446b59 1869
War and Peace
Leo Tolstoy

67362da6683d446b59 1902
The Tale of Peter Rabbit
Beatrix Potter

67362dd5683d446b59 1924
The Boxcar Children
Gertrude

67362e16683d446b59 1986
Love You Forever
Robert Munsch

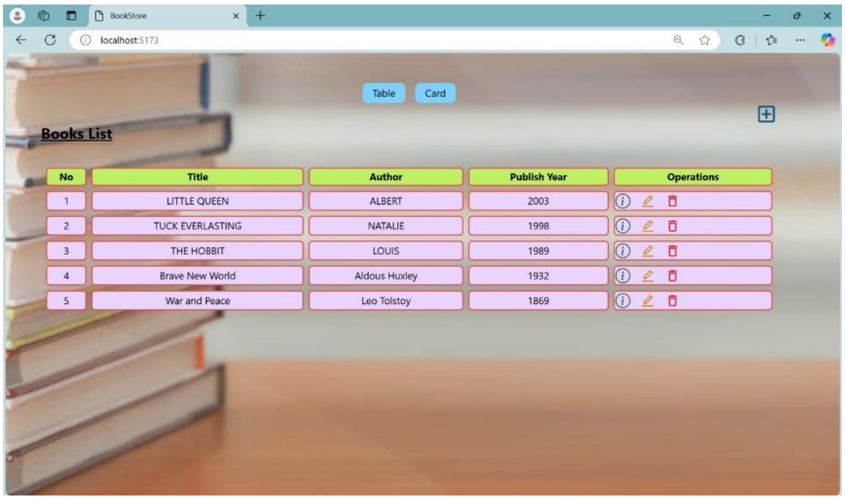

localhost:5173/books/delete/67362eb4683d446b59d32701

Delete Book

Are you Sure You want to delete this book?

Yes, Delete it

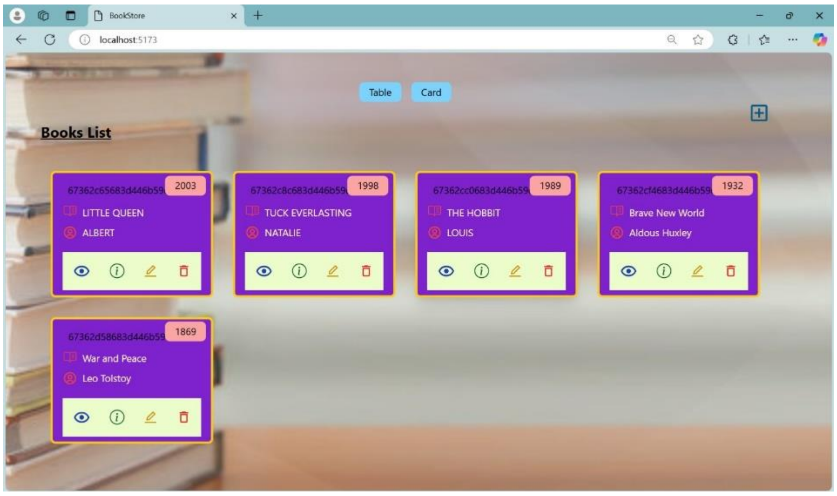
AFTER DELETING THE BOOKS:



Books List

No	Title	Author	Publish Year	Operations
1	LITTLE QUEEN	ALBERT	2003	
2	TUCK EVERLASTING	NATALIE	1998	
3	THE HOBBIT	LOUIS	1989	
4	Brave New World	Aldous Huxley	1932	
5	War and Peace	Leo Tolstoy	1869	

CARD FORMAT:



Books List

67362c65683d446b59 2003
LITTLE QUEEN
ALBERT

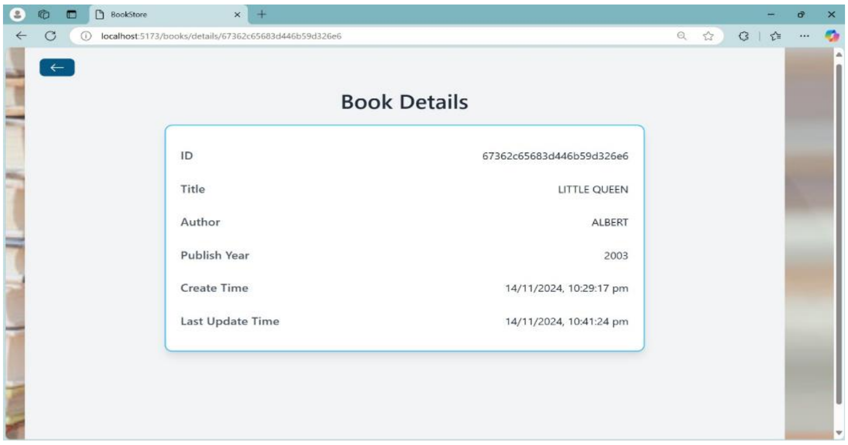
67362c8c683d446b59 1998
TUCK EVERLASTING
NATALIE

67362cc0683d446b59 1989
THE HOBBIT
LOUIS

67362cf4683d446b59 1932
Brave New World
Aldous Huxley

67362d58683d446b59 1869
War and Peace
Leo Tolstoy

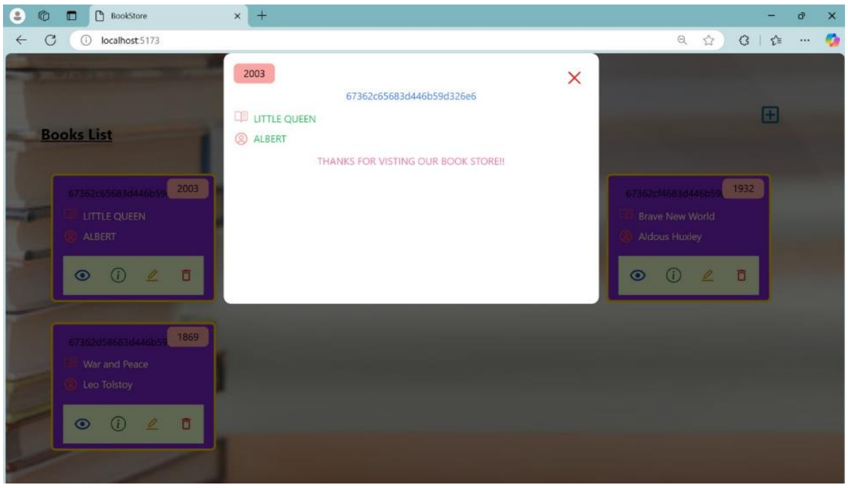
BOOK DETAILS:



Book Details

ID	67362c65683d446b59d326e6
Title	LITTLE QUEEN
Author	ALBERT
Publish Year	2003
Create Time	14/11/2024, 10:29:17 pm
Last Update Time	14/11/2024, 10:41:24 pm

VIEWING THE BOOK DETAILS:



2003

67362c65683d446b59d326e6

LITTLE QUEEN

ALBERT

THANKS FOR VISTING OUR BOOK STORE!!

12. Known Issues

- Users may occasionally encounter errors during the deletion of books, which will be monitored for resolution in future updates.

13. Future Enhancements

- Potential future features could include:
  - Implementing user roles and permissions for advanced access control
  - Integration of social media login options
  - Adding advanced search and filtering capabilities
  - Recommendations based on user preferences.