



NAAN MUDHALVAN
MERN STACK POWERED BY MONGODB

PROJECT NAME: BOOK STORE

TEAM MEMBERS:

Nithinsrivatsan S - CA5DD8993A09075BE674974131FF3CC1

Anand A - BC159C7E79E95E9AE1C48BF5E1E2A5BD

Isha R - C713DCFADD023008F57472ED2AF529CC

Shruthi P - 882566B9E2A59ED976A43B9647F69CAA

Sri Manikanda Aaditan M - DB9F62FB1349E5DA55645229D91EA226

Aim:

The aim of the Book Store project is to design and develop an interactive and user-friendly web application using the MERN stack (MongoDB, Express.js, React.js, and Node.js). This project intends to provide users with an efficient platform to browse, search, and purchase books, while administrators can manage book inventory, user data, and order processing seamlessly.

Introduction:

A Book Store Project using the MERN stack (MongoDB, Express.js, React.js, Node.js) is a web application where users can browse, search, and buy books. Users can add books to their cart and place orders. Admins can manage the book list by adding, editing, or removing books. React.js handles the user interface, Node.js and Express.js manage the server and APIs, and MongoDB stores all the data, like books and orders.

Techstacks:

Frontend:

React.js

Builds a dynamic and interactive user interface for browsing and buying books.

Tailwind CSS

Ensures the design is responsive and visually appealing.

Redux and RTK Query Toolkit

Manages the application's state and handles API data efficiently.

Backend:

Node.js and Express.js

Processes user requests, handles server operations, and manages APIs.

Mongoose

Simplifies interactions with the MongoDB database using an intuitive schema-based approach.

JWT (JSON Web Tokens)

Ensures secure user authentication and session management.

Database:

MongoDB

Stores all data, including book details, user accounts, cart items, and orders, providing scalability and reliability for the project.

Features:

User Features:

Cart Management

Users can add or remove products from their carts with ease.

Checkout and Orders

Users can proceed to checkout and place orders, with a convenient cash-on-delivery payment option.

Admin Features:

Secure Admin Dashboard

Admins can access the dashboard securely using a username and password.

Book Management

Admins can upload new books, update existing book information, and delete books as needed.

Inventory Management

A robust system helps admins track and manage book stock efficiently.

Frontend:

Entry Point:

- main.jsx initializes the React app and renders the App component into the DOM.

Routing:

- The routers folder manages app navigation, defining routes for different pages in the application.

Pages:

- The pages directory contains core views like Home, Product Details, Cart, and Authentication.

State Management:

- The redux folder manages global state, including slices for user data, cart functionality, and product listings.

Context:

- Provides shared state or utilities via React Context API for smaller, specific needs.

Components:

- The components folder holds reusable UI elements like headers, footers, and buttons.

Firestore:

- Used for authentication or database integration, configured in the firebase folder.

Utilities:

- Helper functions in the utils folder simplify common operations like formatting or API calls.

Styling:

- Tailwind CSS is configured (tailwind.config.js), and styles are imported in App.css and index.css.

Public Folder:

- Contains static assets (images, icons) and the index.html template.

The app uses **Vite** for faster builds, and deployment is managed via **Vercel** (vercel.json).

Backend:

Order Schema and Model

- Responsible for managing user order details, including name, email, shipping address, phone number, product references (linked to the Book model), and total price.
- Validates all fields to ensure structured order entry.
- Includes a timestamps option for tracking order creation and updates.

Order Routes

- **POST /orders:** Creates a new order by accepting order details from the request body and saves it to the database. The response includes the saved order data.
- **GET /orders/email/:email:** Fetches all orders associated with a user's email address, returning them in descending order by creation date for a chronological view.

Admin Authentication Middleware

- verifyAdminToken middleware validates the JWT in the authorization header.
- Grants access to routes if the token is valid; denies access if it's missing or invalid, ensuring only authenticated admins can access admin-only routes.

Book Schema and Model

- Defines book data structure, including title, description, category, trending status, cover image URL, price (old and new), and creation date.
- Supports operations like adding, updating, and deleting books.
- timestamps option tracks the creation and update times for each book entry.

Book Routes

- **POST /create-book:** Allows admins to add a new book using the `verifyAdminToken` middleware for authentication.
- **GET /:** Retrieves a list of all books, ordered by creation date.
- **GET /:id:** Fetches detailed information for a specific book by its ID.
- **PUT /edit/:id:** Enables admins to update book information (e.g., price, description).
- **DELETE /:id:** Allows admins to delete a book from the inventory.

User Schema and Model

- Stores user details, including a unique username, hashed password, and role (user or admin).
- Utilizes MongoDB for efficient data storage and retrieval.
- Passwords are hashed using `bcrypt`, and pre-save middleware ensures this hashing is performed only on new or updated passwords.
- Supports role-based access, allowing different features for users and admins.

Admin Authentication Route

- `/admin` route provides secure login for admins.
- Returns a JWT upon valid authentication, which includes the admin's ID, username, and role.
- JWTs are signed with a secret key and have a set expiration (e.g., one hour).
- Validates credentials and returns error messages for invalid inputs.

Admin Dashboard Statistics

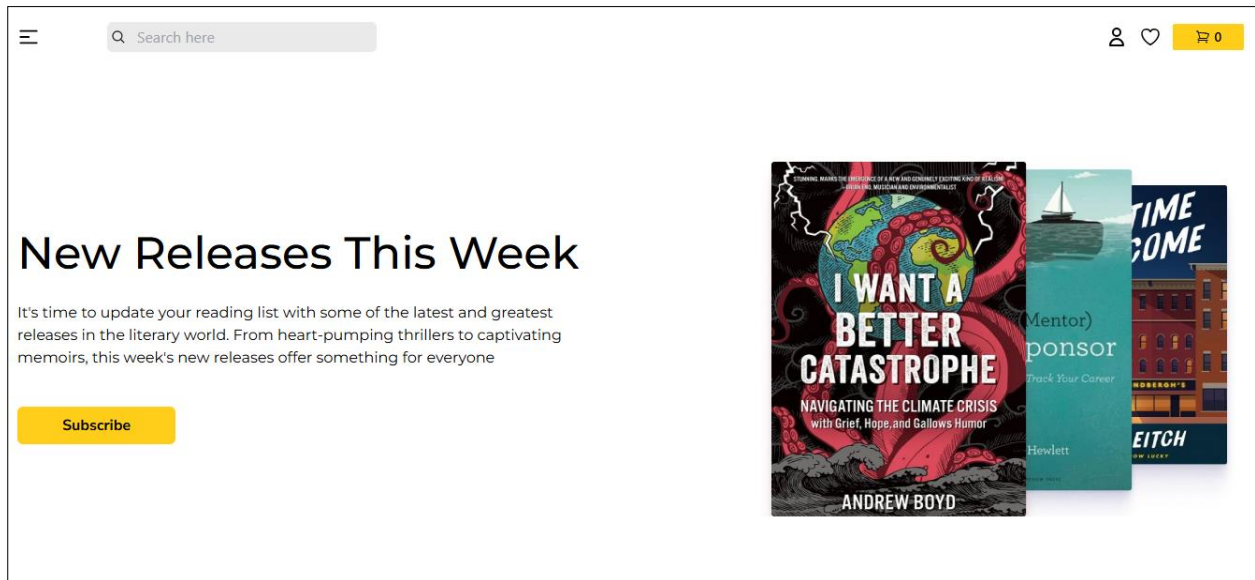
- Provides insights on total orders, overall sales, trending book count, and total book count.
- Uses advanced queries to calculate monthly sales data grouped by year and month, alongside order and sales totals for each period.
- Offers a comprehensive view of platform performance to help admins make data-driven decisions.
- Ensures efficient data processing even with large data sets for scalability.

Steps to Compile and Run the project:

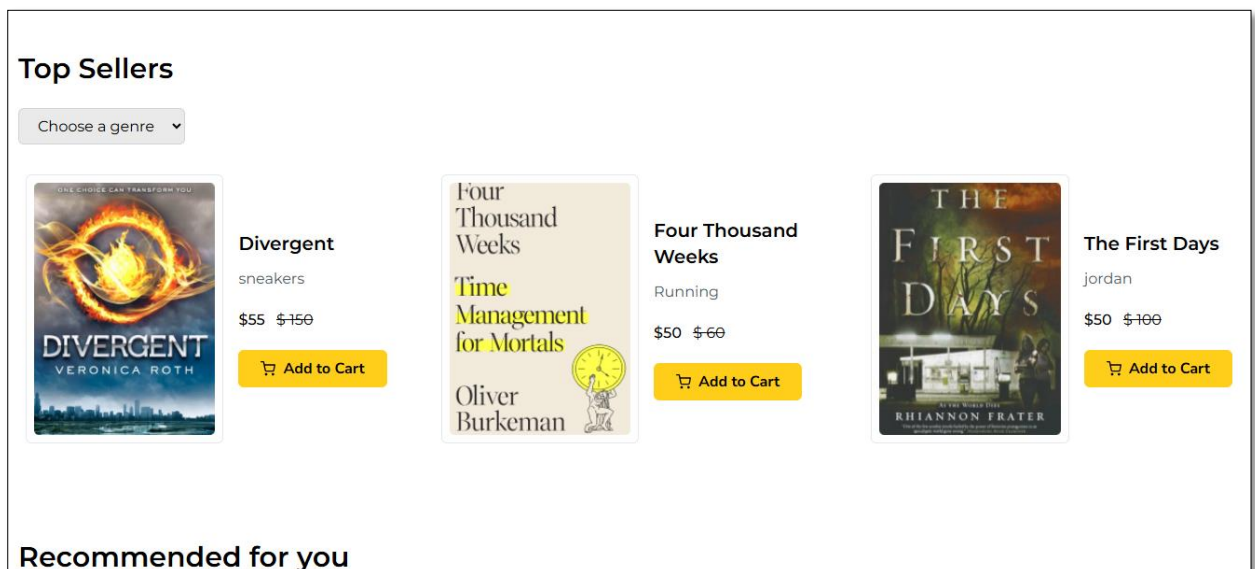
- Create a `.env.local` file in the frontend root directory (same level as `package.json`).
- Add the Firebase environment variables to the `.env.local` file.
- Run `npm install` to install frontend dependencies.
- Run `npm run dev` to start the frontend server.
- Create a `.env` file in the backend root directory (same level as `package.json`).
- Add the database URL and JWT secret key to the `.env` file.
- Run `npm install` to install backend dependencies.
- Run `npm run start:dev` to start the backend server.

Output:


Homepage:






List of Top Sellers:



Login Page:





 0


Please Login

Email

Password

[Login](#)

Haven't an account? Please [Register](#)


 [Sign in with Google](#)



©2025 Book Store. All rights reserved.

Selection of Book:

The image is a screenshot of a web application's interface. At the top, there is a navigation bar with a hamburger menu icon on the left, a search bar with the placeholder text "Search here", and on the right, a user profile icon, a heart icon, and a yellow cart icon with a zero count. The main content area displays a book listing. The book title "Four Thousand Weeks" is in a large, bold, black font. Below it, the subtitle "Time Management for Mortals" is in a smaller, yellow font. The author's name "Oliver Burkeman" is at the bottom of the text. To the right of the text is a book cover illustration featuring a yellow clock face and a small figure of a person standing next to it. Below the book title and subtitle, the following metadata is shown: "Author: admin", "Published: 12/11/2024", "Category: Business", and "Description: Running". At the bottom left of the listing, there is a yellow button with a shopping cart icon and the text "Add to Cart".

Adding the Selected Book to the cart:






1

Shopping cart

Clear Cart



Four Thousand Weeks

Category: Business

Qty: 1

\$50

Remove

Subtotal

Shipping and taxes calculated at checkout.

\$50.00

Checkout

or Continue Shopping →

Filling Up Personal Details:

Total Price: \$50.00

Items: 1

Personal Details

Please fill out all the fields.

Full Name

Ram Kumar

Email Address

nithicknazareth1@gmail.com

Phone Number

445767588765

Address / Street

No 15 Gandhi street

City

Chennai

Country / region

India

State / province

Tamil Nadu

Zipcode

600543

☒ I am agree to the [Terms & Conditions](#) and [Shoping Policy](#).

Place an Order

Confirmation of Order:


Your Orders

1

Order ID: 67596f708a94a0d53b820031
Name: Ram Kumar
Email: nithicknazareth1@gmail.com
Phone: 445767588765
Total Price: \$50

Address:
Chennai, Tamil Nadu, India, 600543

Products Id:
66f04d2b8568a4e7c7938c1a



Confirmed Order




Your order placed successfully!

Yes, It's Okay!Cancel

Order Details:

☰

🔍 Search here



Your Orders






1

Order ID: 67596f708a94a0d53b820031
Name: Ram Kumar
Email: nithicknazareth1@gmail.com
Phone: 445767588765
Total Price: \$50

Address:
Chennai, Tamil Nadu, India, 600543


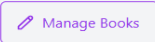
Products Id:
66f04d2b8568a4e7c7938c1a


Dashboard of Admin:





Dashboard


Book Store Inventory




**3**
Products

**\$87879**
Total Sales

**2(13%)**
Trending Books
in This Month

**115**
Total Orders

The number of orders per month



Monthly Revenue

Monthly Revenue

Revenue (USD)

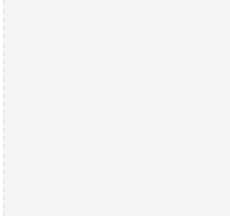
Month	Revenue (USD)
Jan	800
Feb	850
Mar	900
Apr	950
May	1000
Jun	1050
Jul	1100
Aug	1150
Sep	1200

Users by average order

Descending ▾

User	Average Order
Annette Watson	9.3
Calvin Steward	8.9
Ralph Richards	8.7
Bernard Murphy	8.2

Students by type of studying



Managing Books:

The screenshot shows a dashboard for a book store inventory. At the top, there's a search bar and a user profile for Grace Simmons, Lecturer. The dashboard title is 'Dashboard' with the subtitle 'Book Store Inventory'. On the right, there are two buttons: 'Manage Books' and 'Add New Book'. The main content area displays a table titled 'All Books' with a 'SEE ALL' button. The table lists three books with their details and actions.

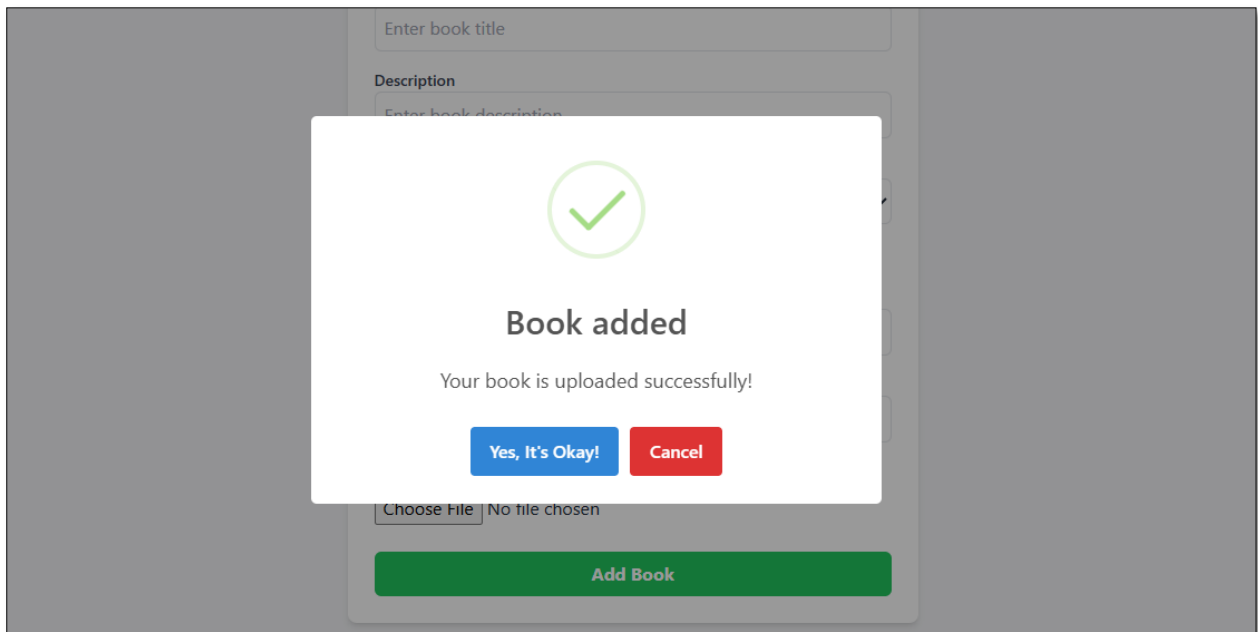
#	BOOK TITLE	CATEGORY	PRICE	ACTIONS
1	Divergent	shoes	\$55	Edit Delete
2	Four Thousand Weeks	business	\$50	Edit Delete
3	The First Days	horror	\$50	Edit Delete

Adding a New Book:

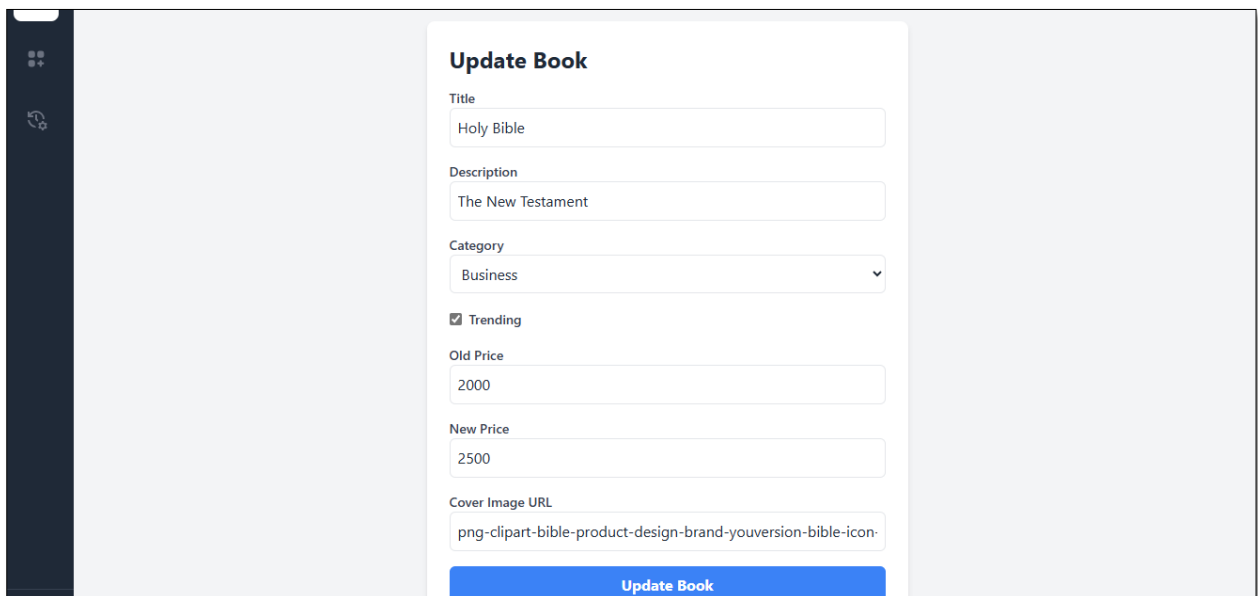
The screenshot shows the 'Add New Book' form on the dashboard. The form fields are as follows:

- Title: Holy Bible
- Description: The New Testament
- Category: Technology (dropdown menu)
- ☒ Trending
- Old Price: 2000
- New Price: 2500

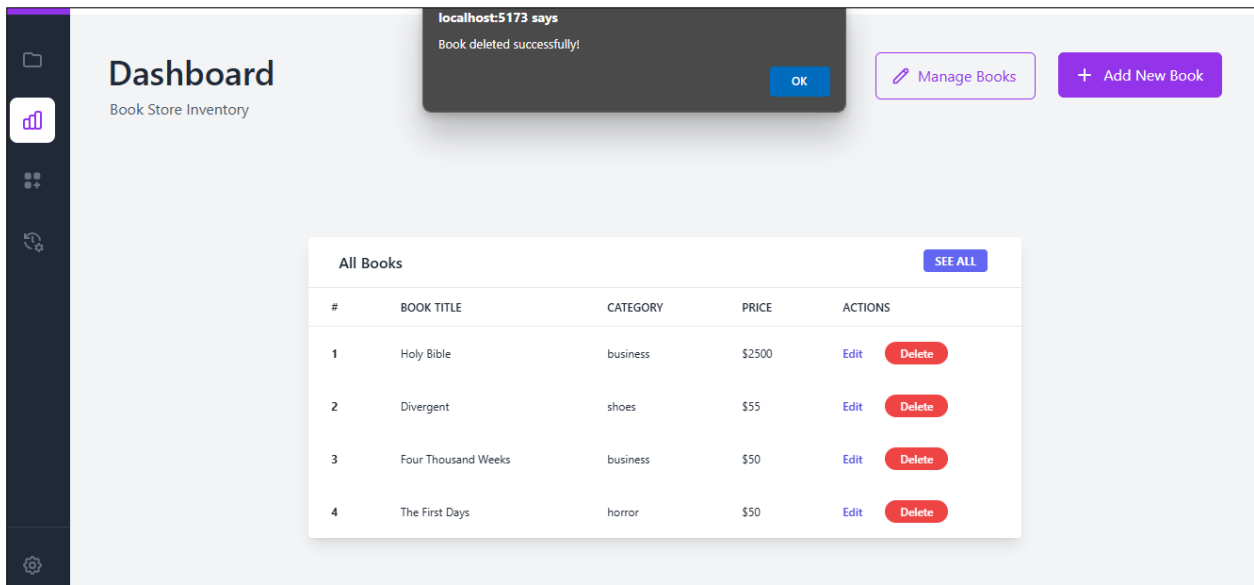
Successful Adding of Book:



Updation of Book:

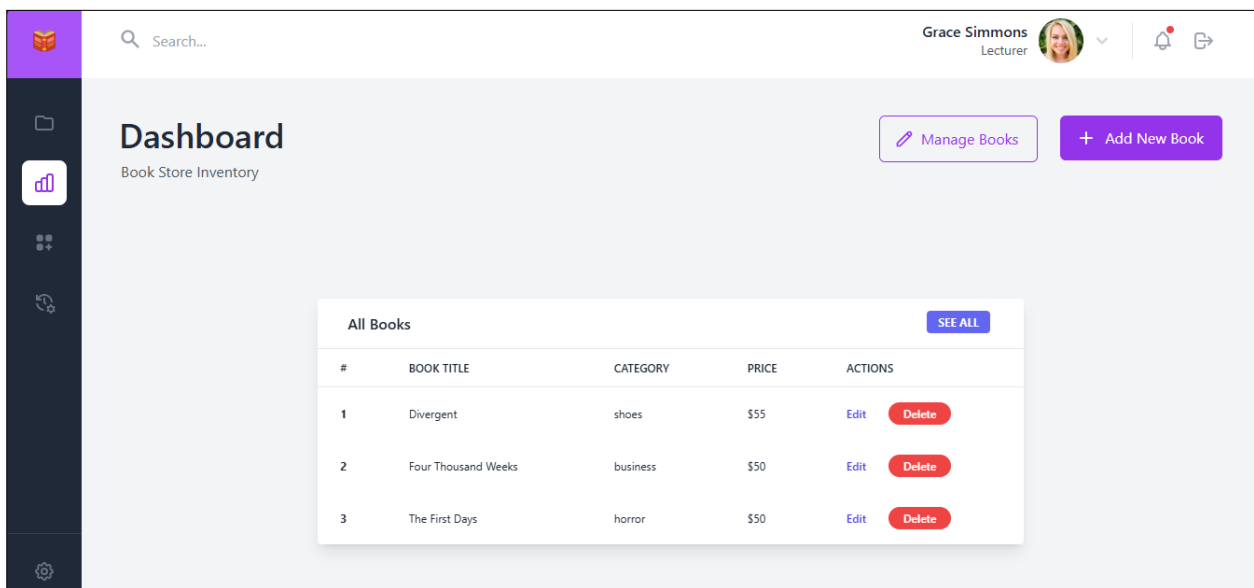


Deletion of Book:



The screenshot shows the 'Dashboard' for 'Book Store Inventory'. A dark notification box at the top center displays the message 'localhost:5173 says Book deleted successfully!' with an 'OK' button. To the right of the notification are two buttons: 'Manage Books' (with a pencil icon) and 'Add New Book' (with a plus icon). Below these, a table titled 'All Books' lists four books. The table has columns for '#', 'BOOK TITLE', 'CATEGORY', 'PRICE', and 'ACTIONS'. The books listed are 'Holy Bible' (business, \$2500), 'Divergent' (shoes, \$55), 'Four Thousand Weeks' (business, \$50), and 'The First Days' (horror, \$50). Each book has 'Edit' and 'Delete' buttons in the 'ACTIONS' column. A 'SEE ALL' button is located at the top right of the table.

#	BOOK TITLE	CATEGORY	PRICE	ACTIONS
1	Holy Bible	business	\$2500	Edit Delete
2	Divergent	shoes	\$55	Edit Delete
3	Four Thousand Weeks	business	\$50	Edit Delete
4	The First Days	horror	\$50	Edit Delete



The screenshot shows the 'Dashboard' for 'Book Store Inventory' after a book has been deleted. The notification box is no longer present. The 'All Books' table now lists only three books: 'Divergent' (shoes, \$55), 'Four Thousand Weeks' (business, \$50), and 'The First Days' (horror, \$50). The 'ACTIONS' column for each book still contains 'Edit' and 'Delete' buttons. The 'SEE ALL' button remains at the top right of the table. The top right of the dashboard shows the user 'Grace Simmons' with the role 'Lecturer' and a profile picture.

#	BOOK TITLE	CATEGORY	PRICE	ACTIONS
1	Divergent	shoes	\$55	Edit Delete
2	Four Thousand Weeks	business	\$50	Edit Delete
3	The First Days	horror	\$50	Edit Delete

Conclusion:

The Book Store project successfully demonstrates the development of a full-stack web application that leverages the MERN stack to deliver a responsive and feature-rich online book retailing experience. The project highlights the integration of front-end and back-end technologies to create an engaging and interactive user interface, secure user authentication, and a robust backend for managing data efficiently. This project emphasizes the importance of modern web development practices and provides a comprehensive platform that could be scaled and enhanced with further features.