

**CSD 3103**

**Project Version B**

**Students May Work in groups of Maximum Two**

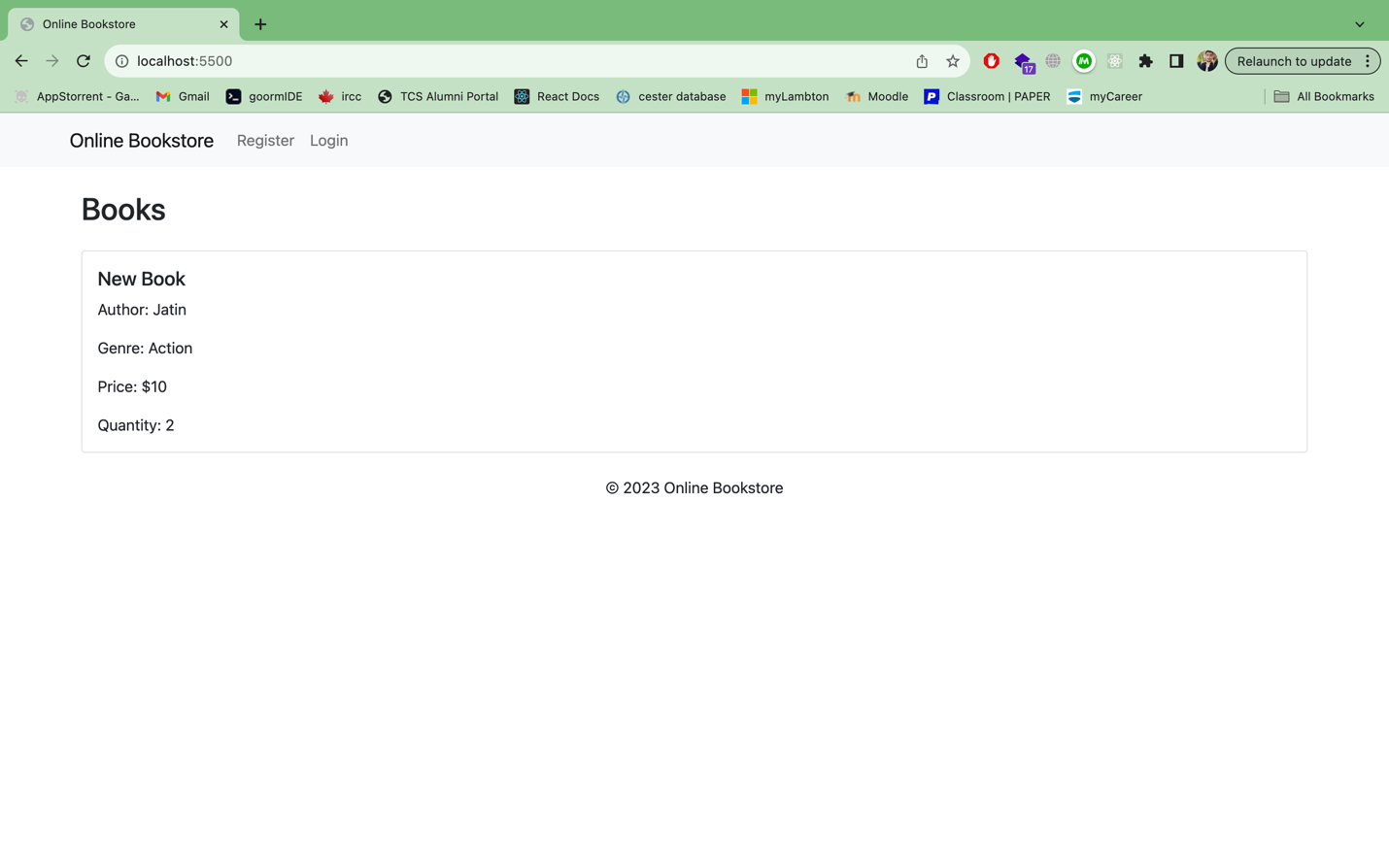
Project Title: Online Bookstore Management System

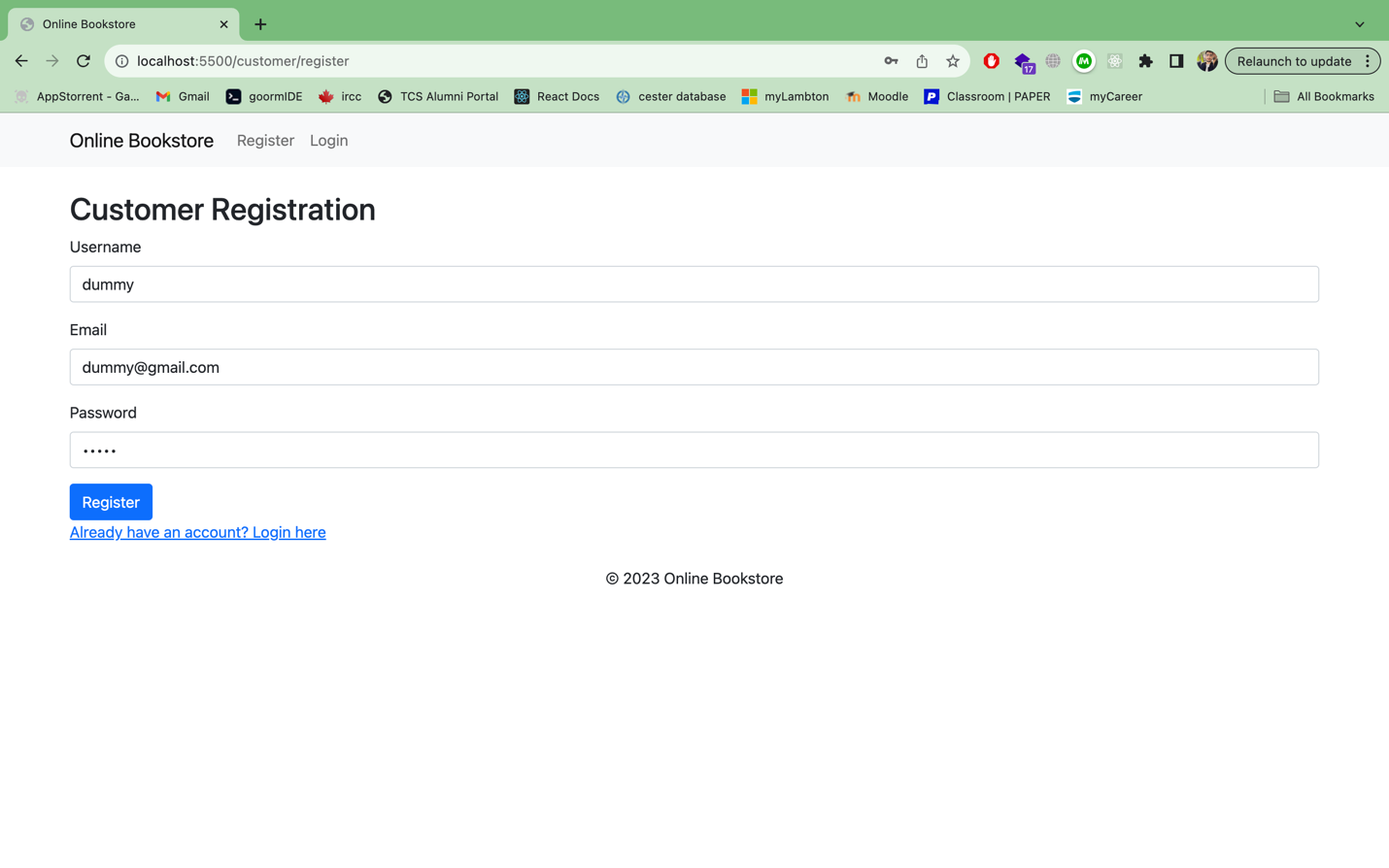
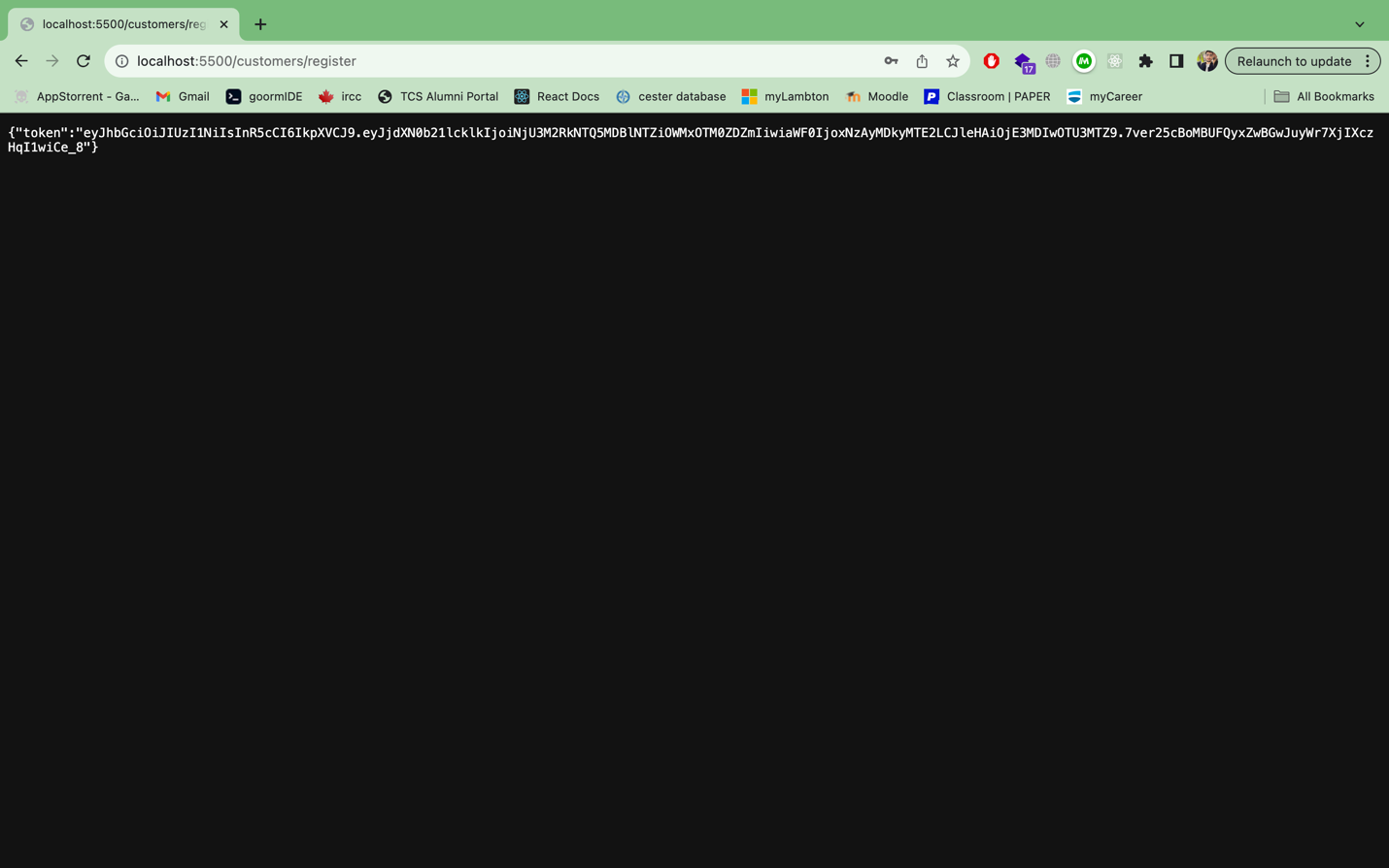
Project Description: The goal of this project is to develop an Online Bookstore Management System using Node.js and server-side programming. The system should provide functionality for managing books, customers, orders, and inventory. Users should be able to browse and search for books, create customer accounts, place orders, and track their order status.

Technical Requirements:

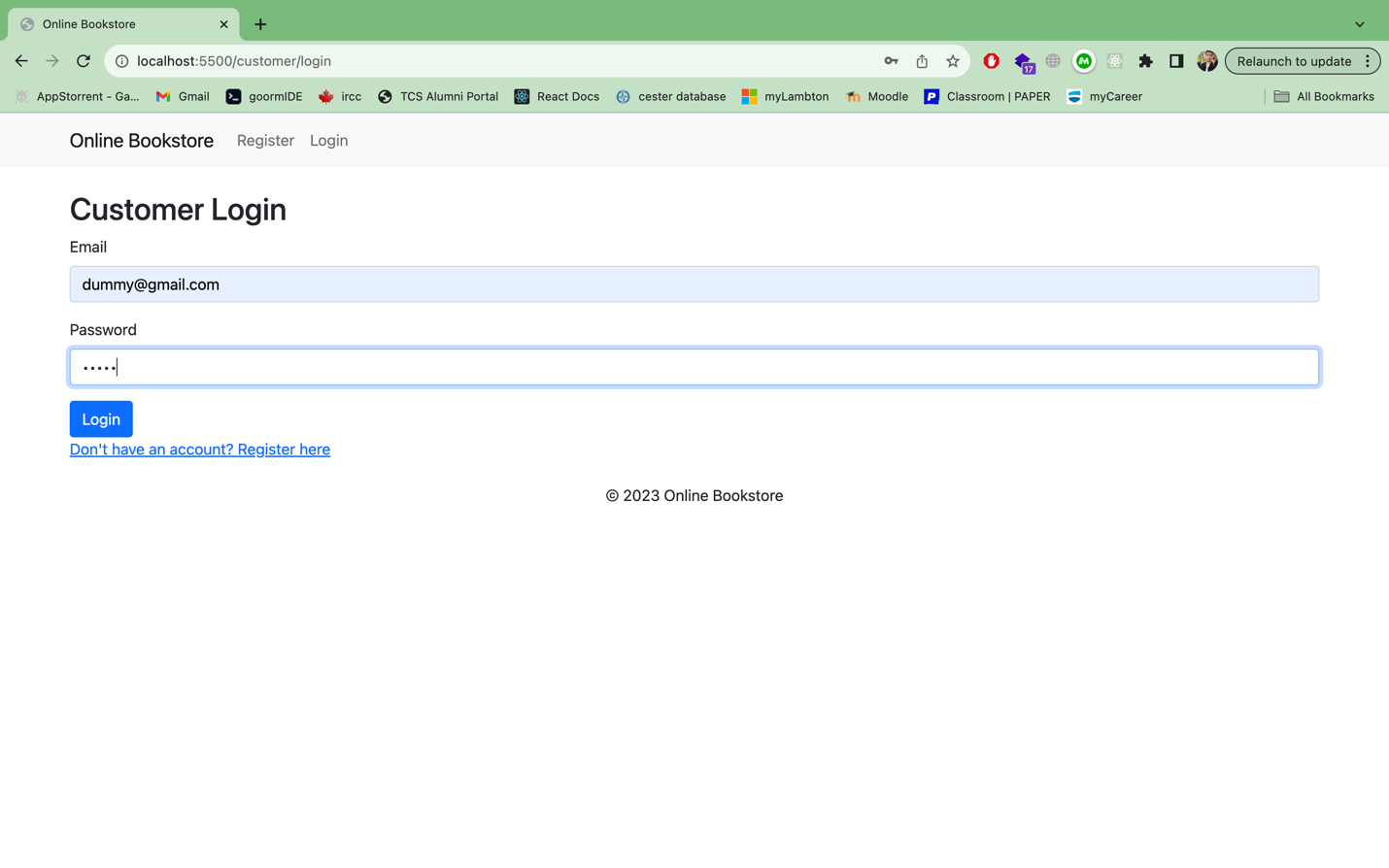
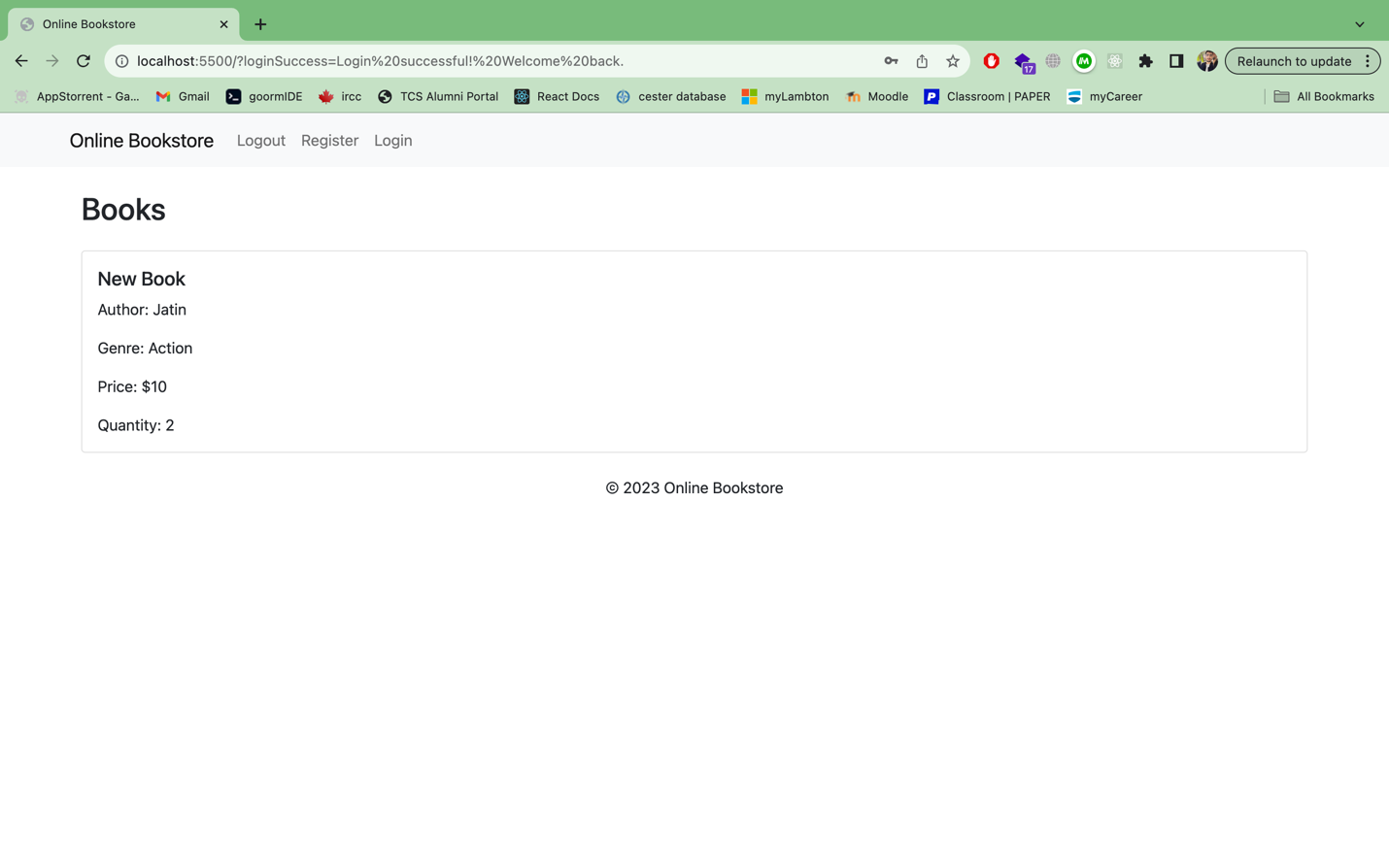
1. Implement a Node.js server to handle HTTP requests and responses.
2. Use a database (e.g., MongoDB, PostgreSQL) to store and manage book, customer, order, and inventory data.
3. Develop a RESTful API to handle CRUD operations for books, customers, and orders.
4. Implement authentication and authorization mechanisms to secure the application.
5. Develop server-side validation and error handling for input data.
6. Implement search functionality to allow users to search for books based on various criteria (e.g., title, author, genre).
7. Implement order tracking functionality for customers to track their order status.
8. Develop an intuitive user interface for customers to browse and interact with the bookstore system.
9. Implement unit tests to ensure the correctness of server-side functionality.
10. Functionality (40%):
    * Book management (CRUD operations, search functionality)
    * Customer management (account creation, authentication, authorization)
    * Order management (placing orders, tracking order status)
    * Inventory management (updating book quantities, managing stock)
11. User Interface (20%):
    * User-friendly and intuitive design
    * Responsive layout for different devices (desktop, mobile)
    * Clear navigation and proper organization of information
12. Code Quality (20%):
    * Well-structured code with proper modularity
    * Consistent coding style and naming conventions
    * Effective use of Node.js best practices
    * Proper error handling and validation
13. Database Design (10%):
    * Appropriate schema design for book, customer, order, and inventory data
    * Efficient indexing and querying for search functionality
14. Testing (10%):
    * Comprehensive unit tests for critical server-side functionality
    * Test coverage and effectiveness of test cases
15. Documentation (5%):
    * Clear and concise project documentation
    * README file with setup instructions and project overview
16. Creativity and Innovation (5%):
    * Implementation of additional features beyond the basic requirements
    * Innovative ideas in design or functionality

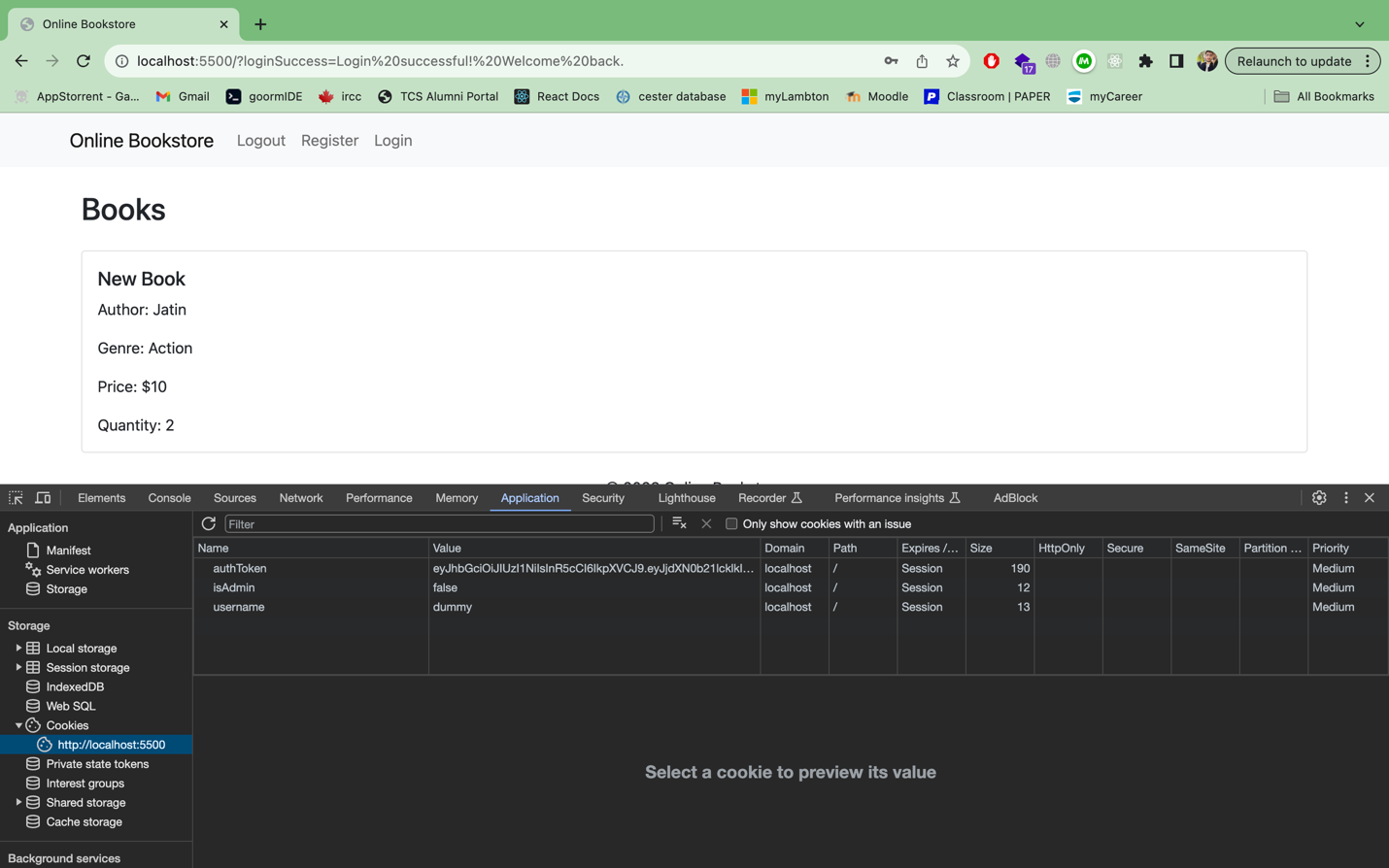
Note: The weights assigned to each category are suggestive and can be adjusted according to the specific requirements of the project.

Project Demonstration:  
  
HomePage:  


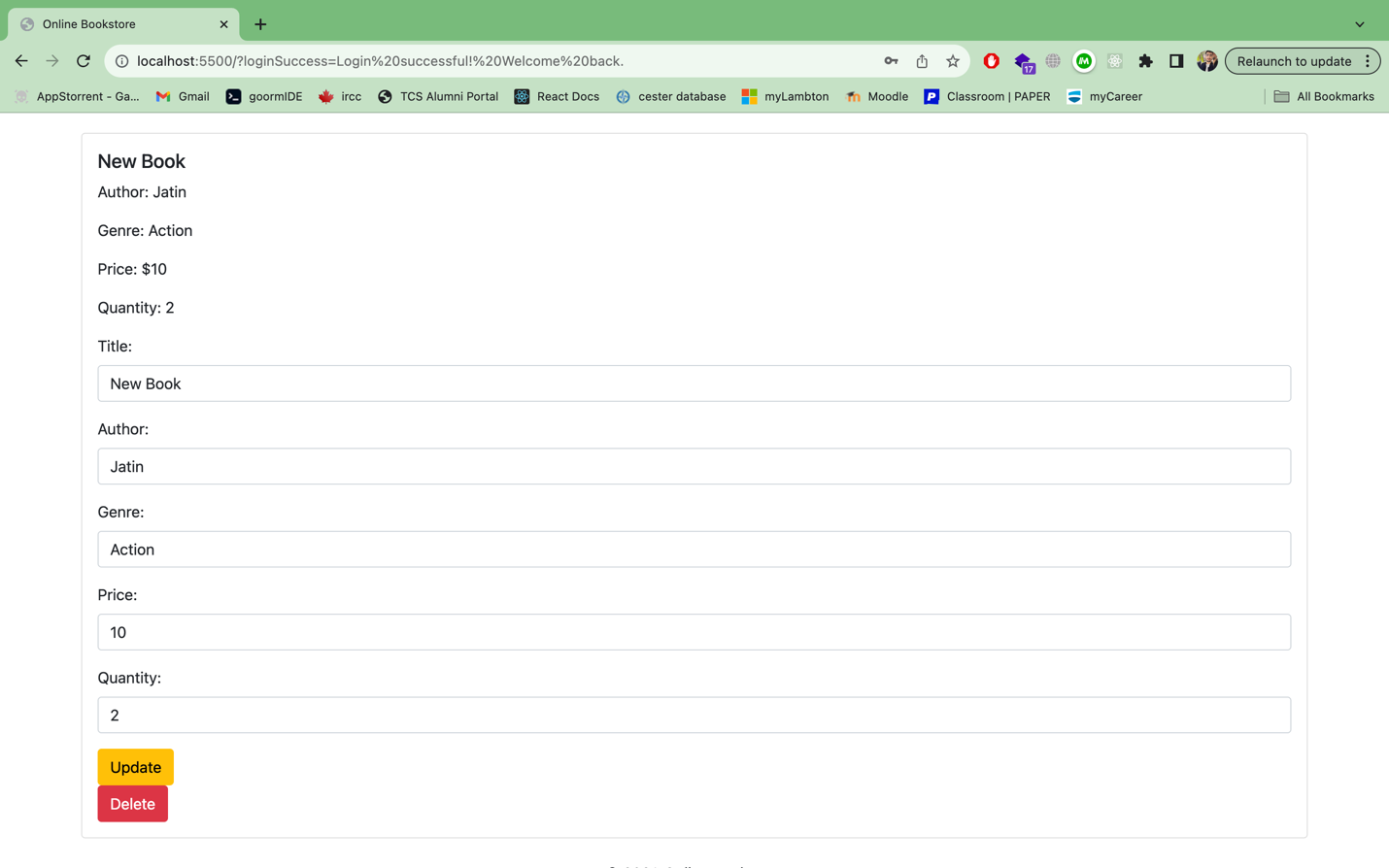
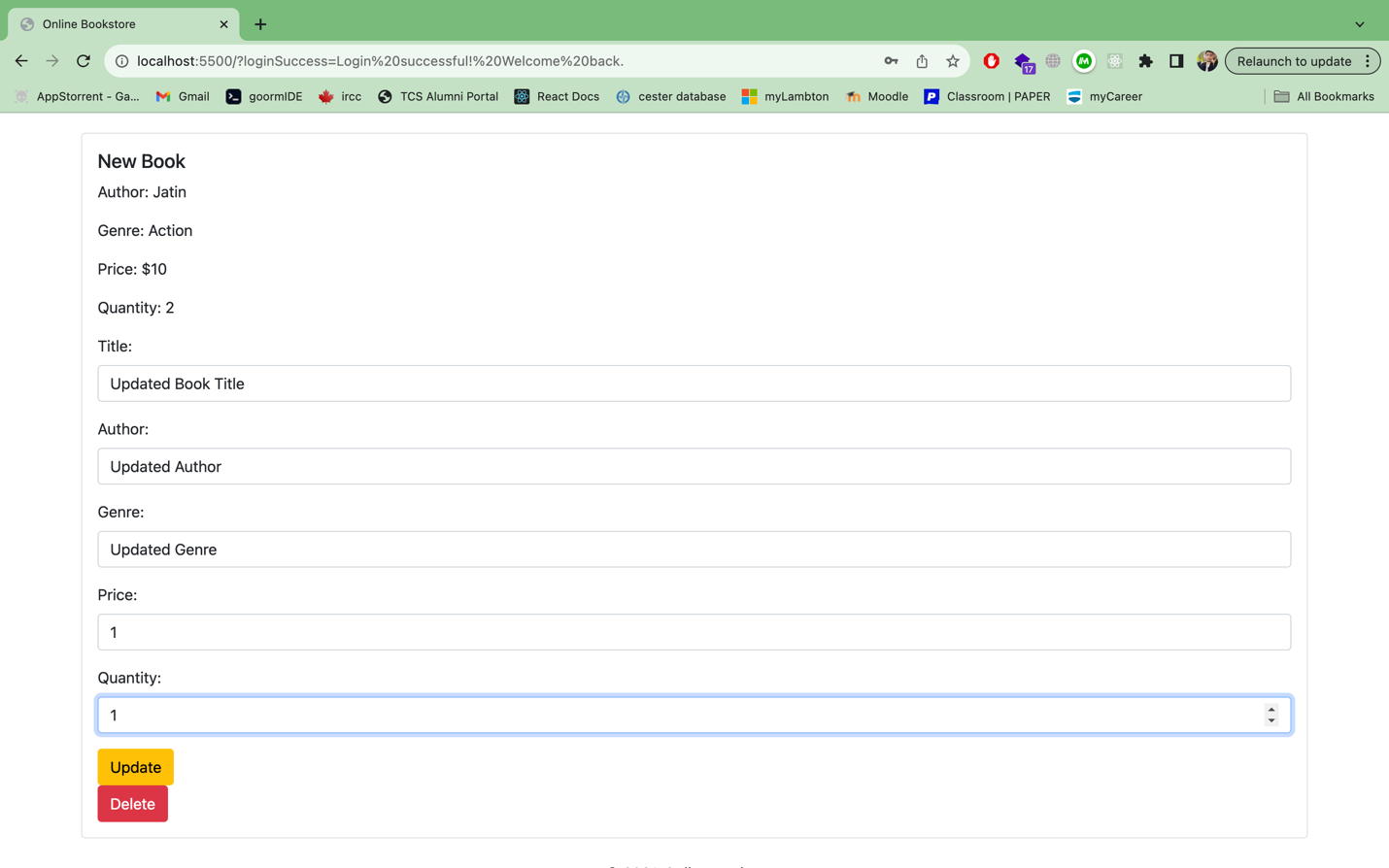
Registration Page: (Users can register)  
  
  
We can put in values and register.  
After registering we can simply login.  
  


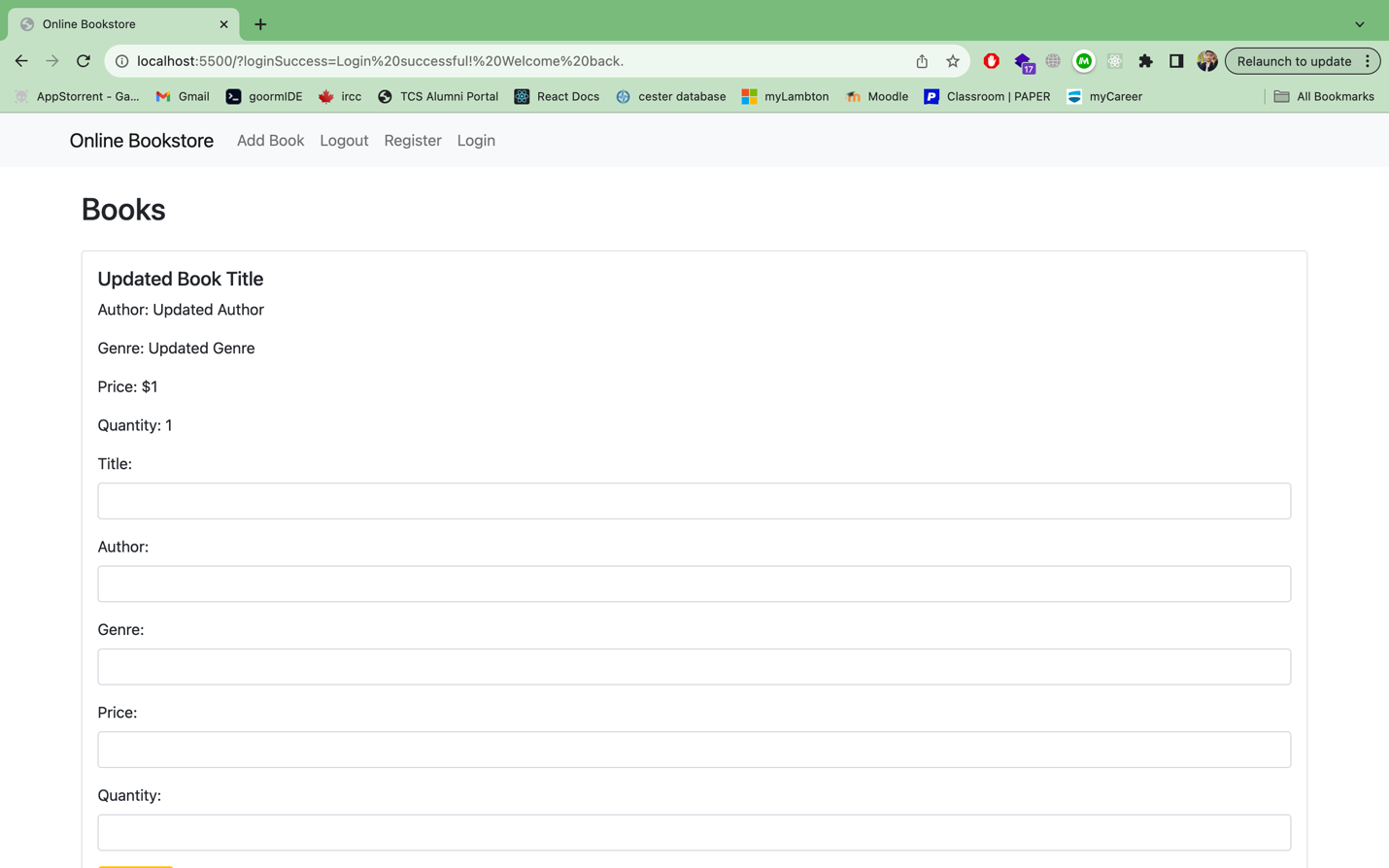
Login Page: ( we can login with the new user )

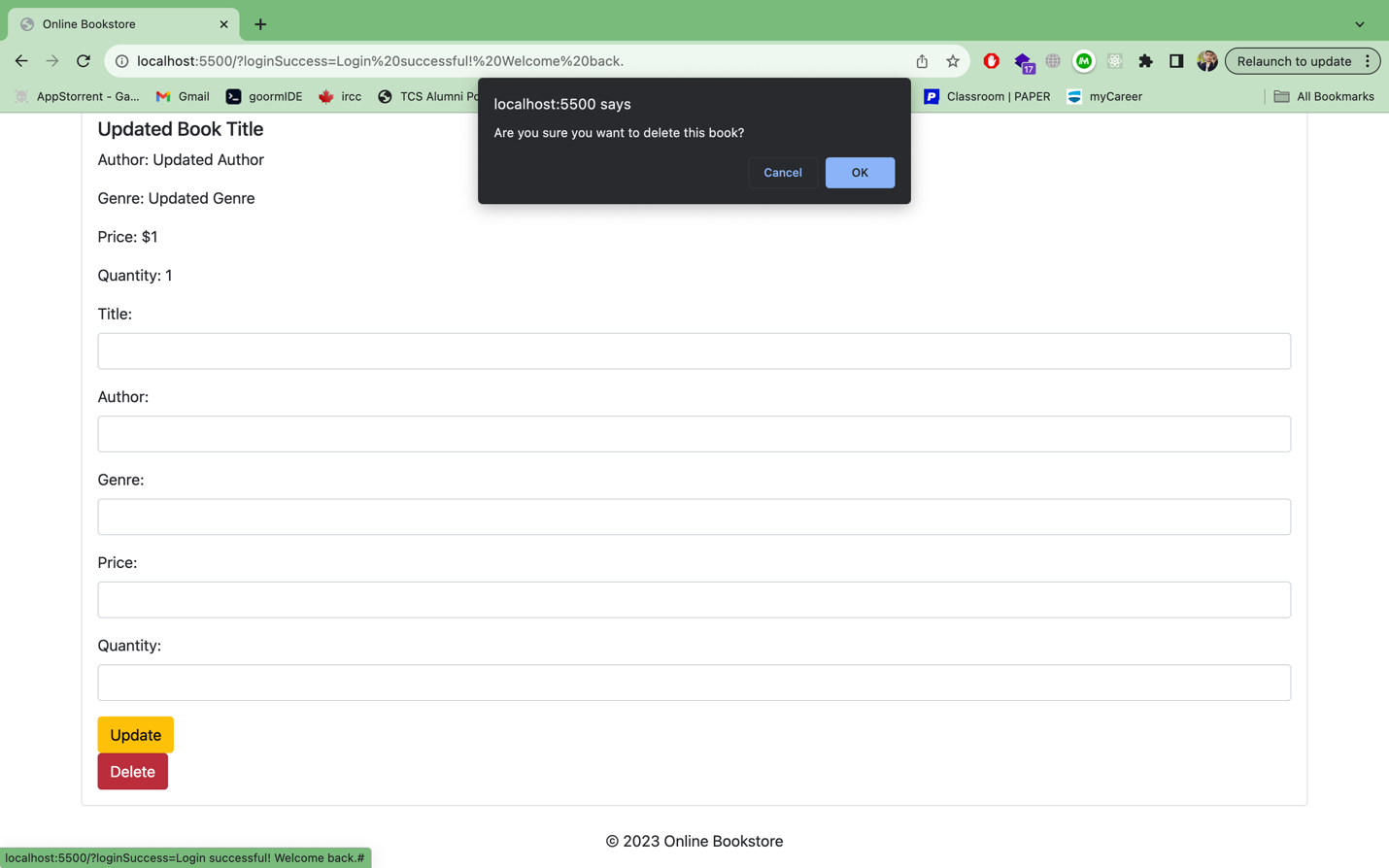
  
  
After logging in, we can see the books:  


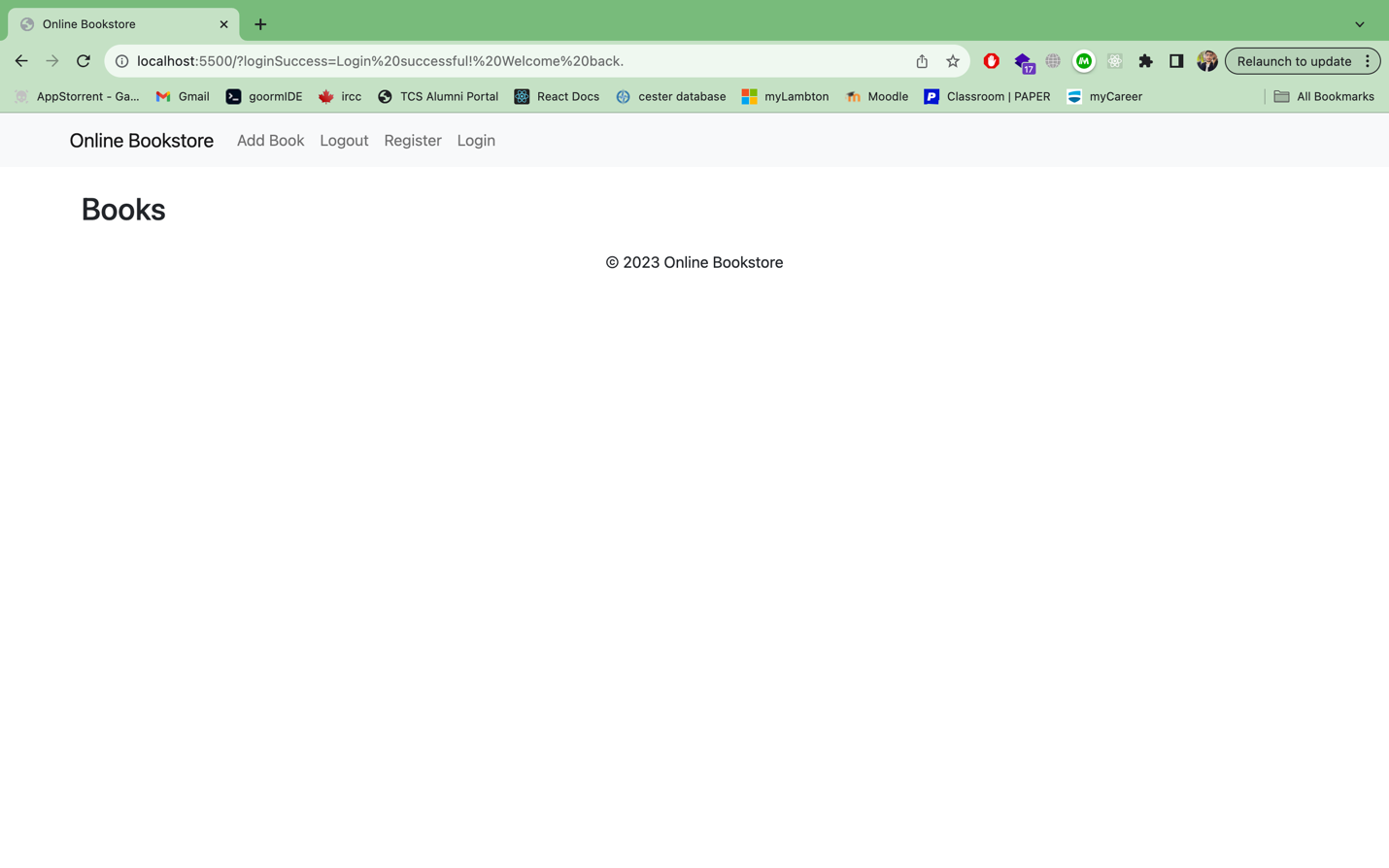
After logging in, we can see a logout button appearing.  
The app uses cookies to handle everything:  


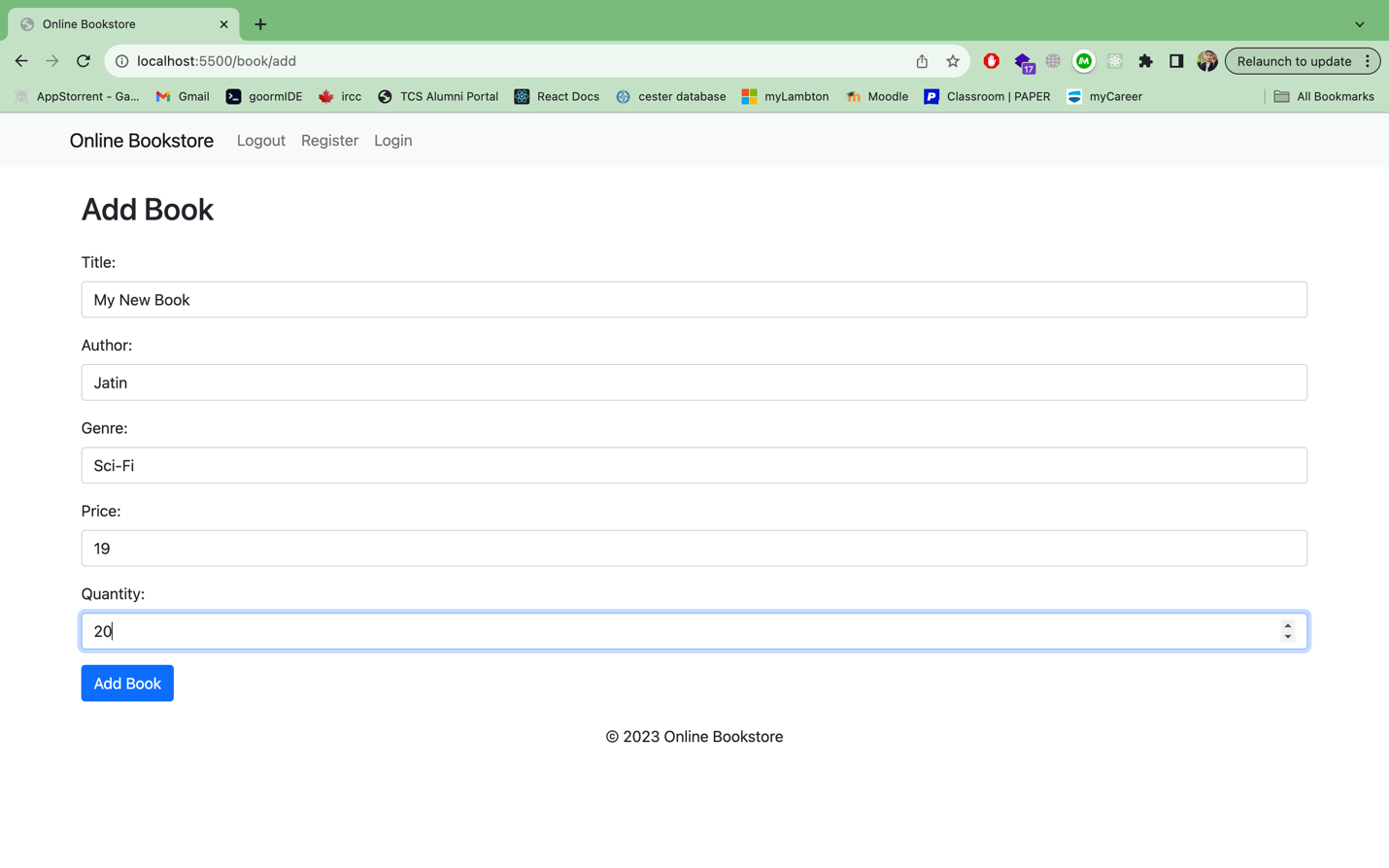
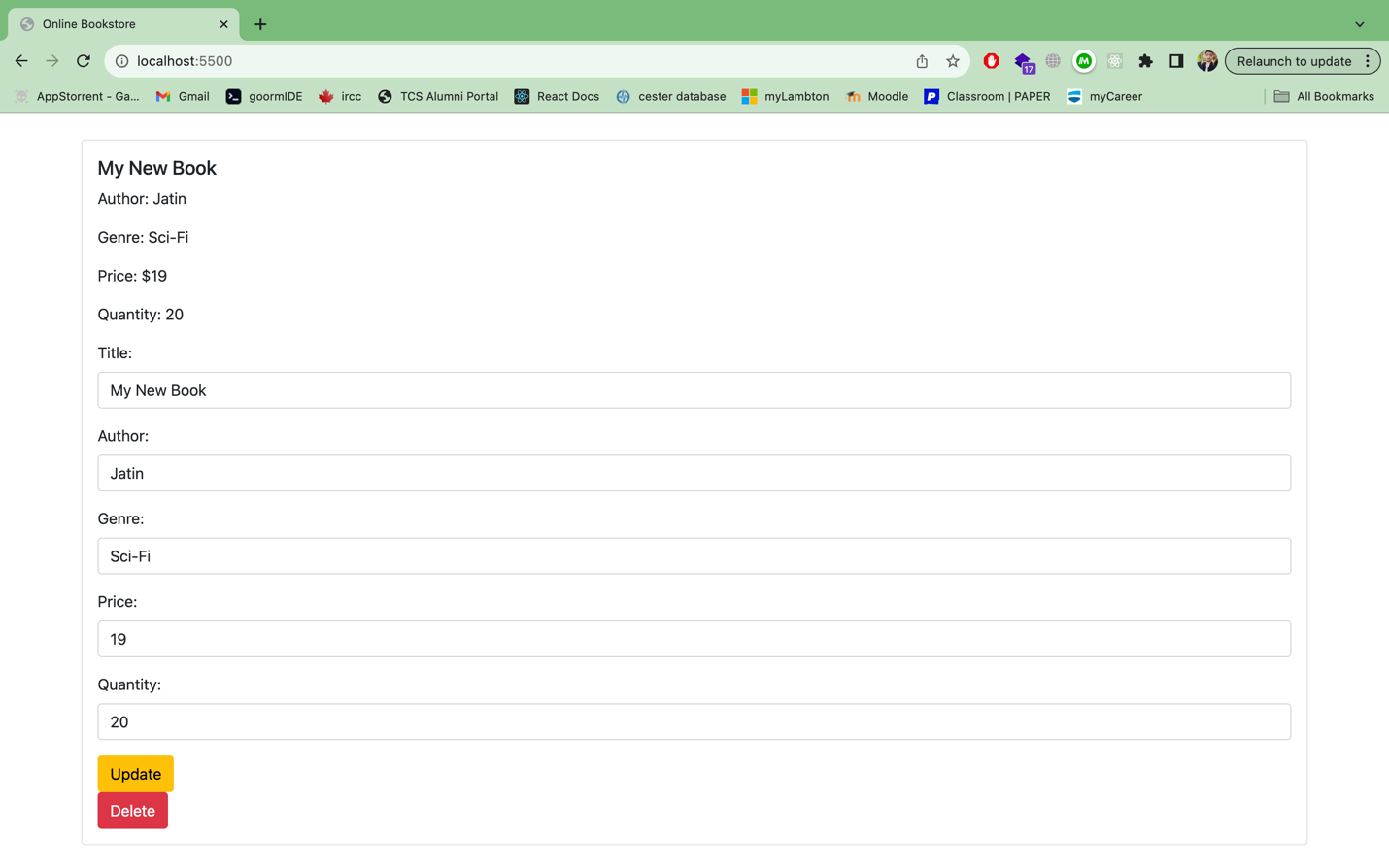
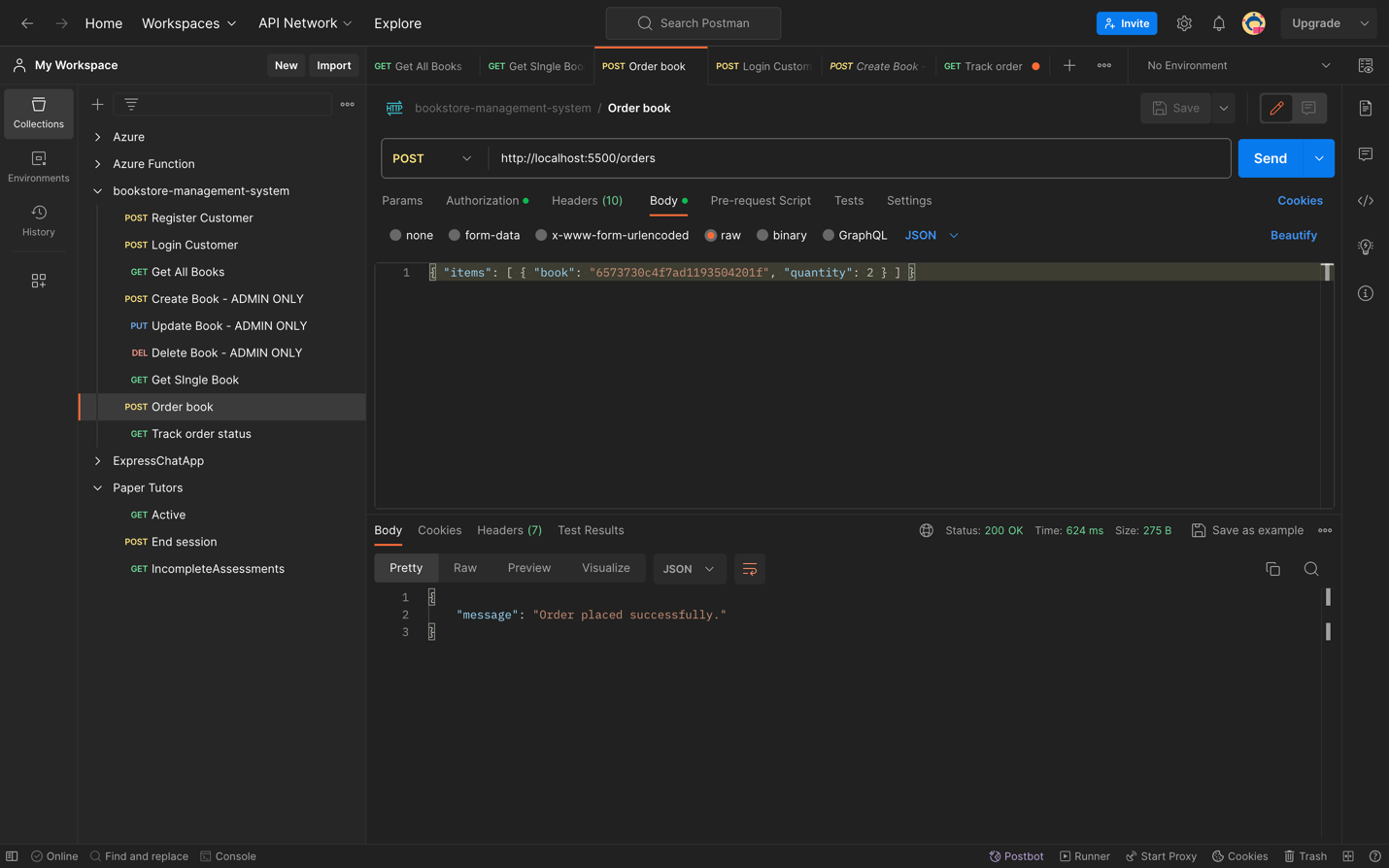
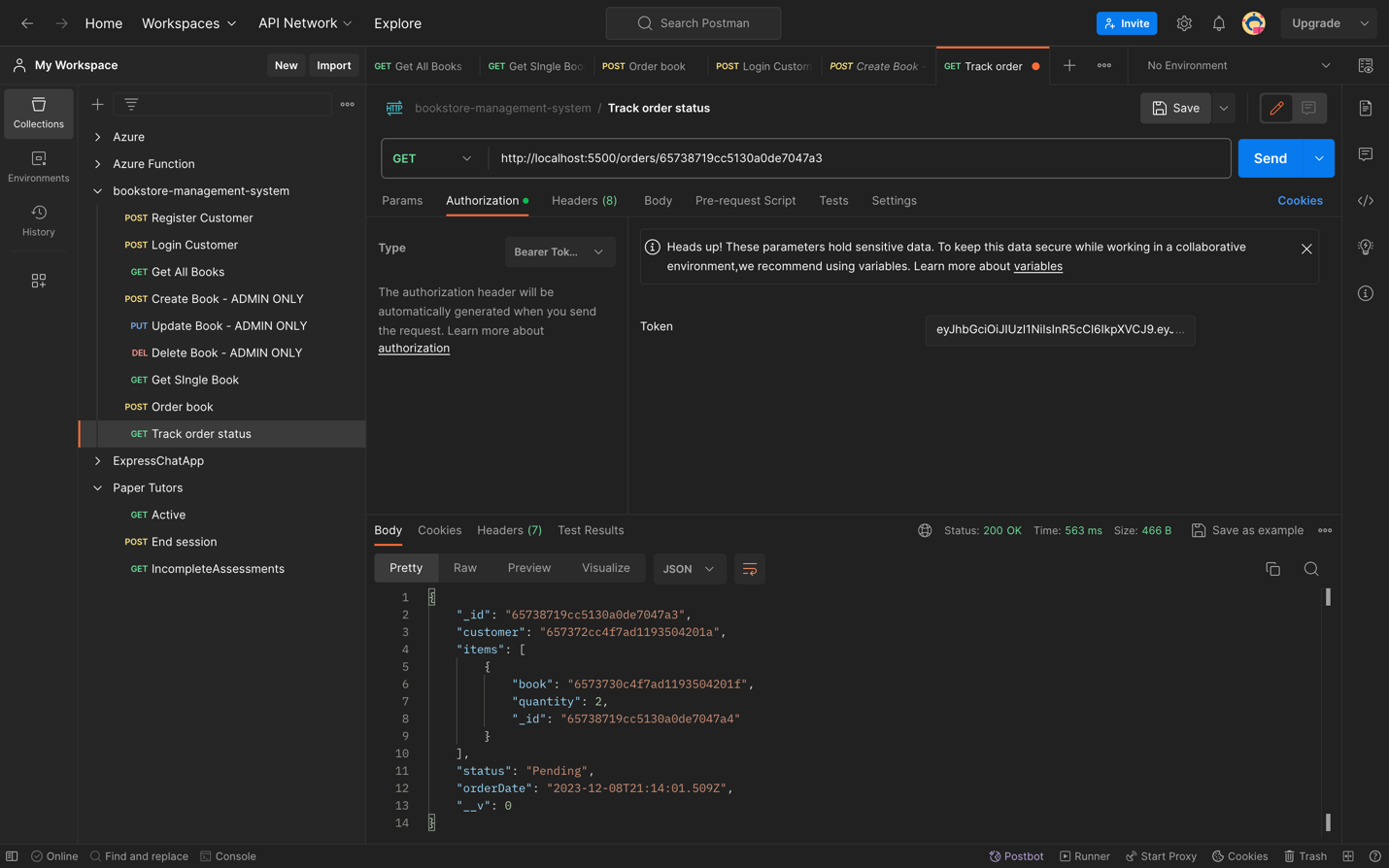
Since the user is not an Admin, he wont be able to add or remove books.

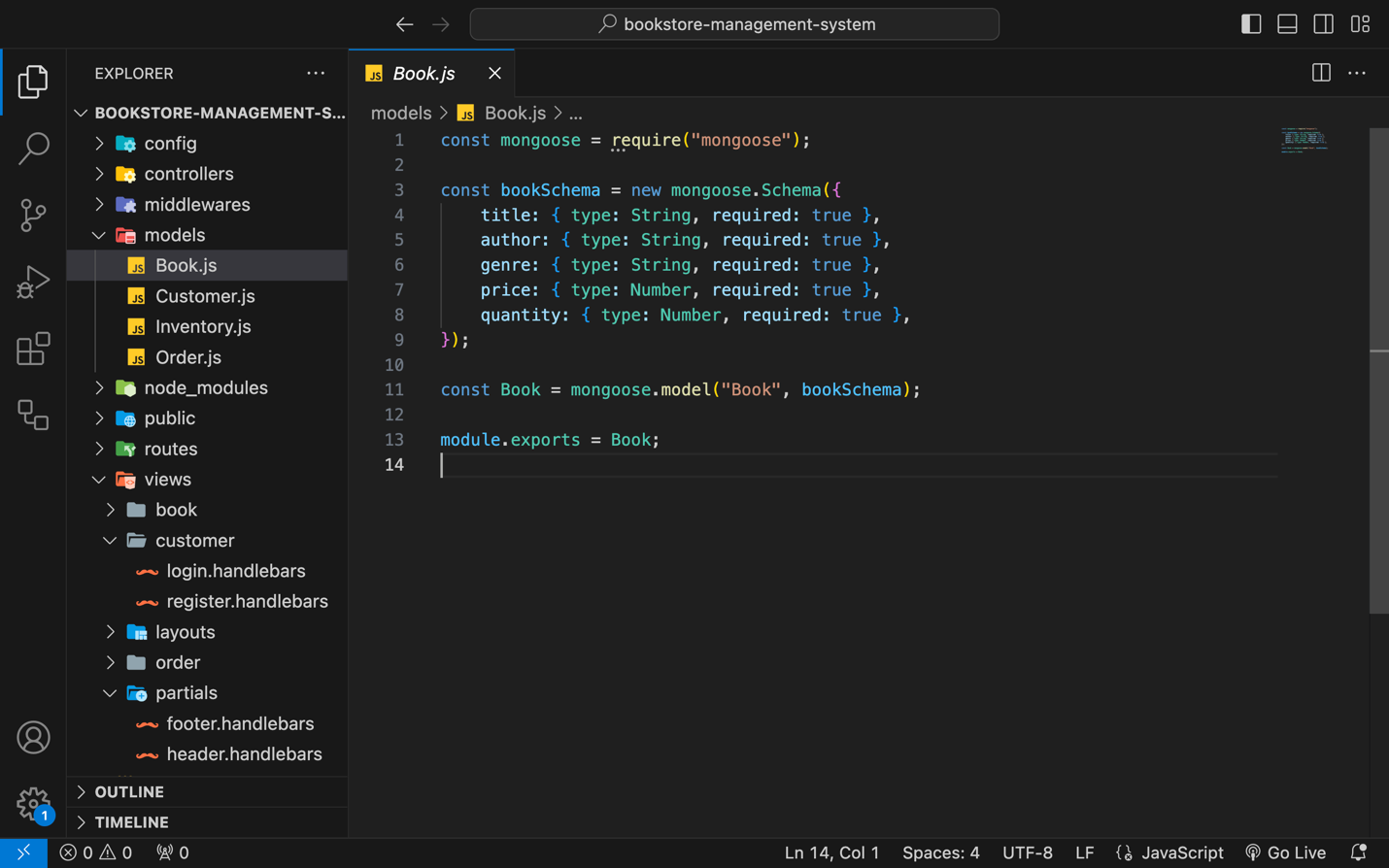
Lets login using an Admin user.  
  
  
For each book, the Admin user will be able to update the values, and make changes to any book.  
  


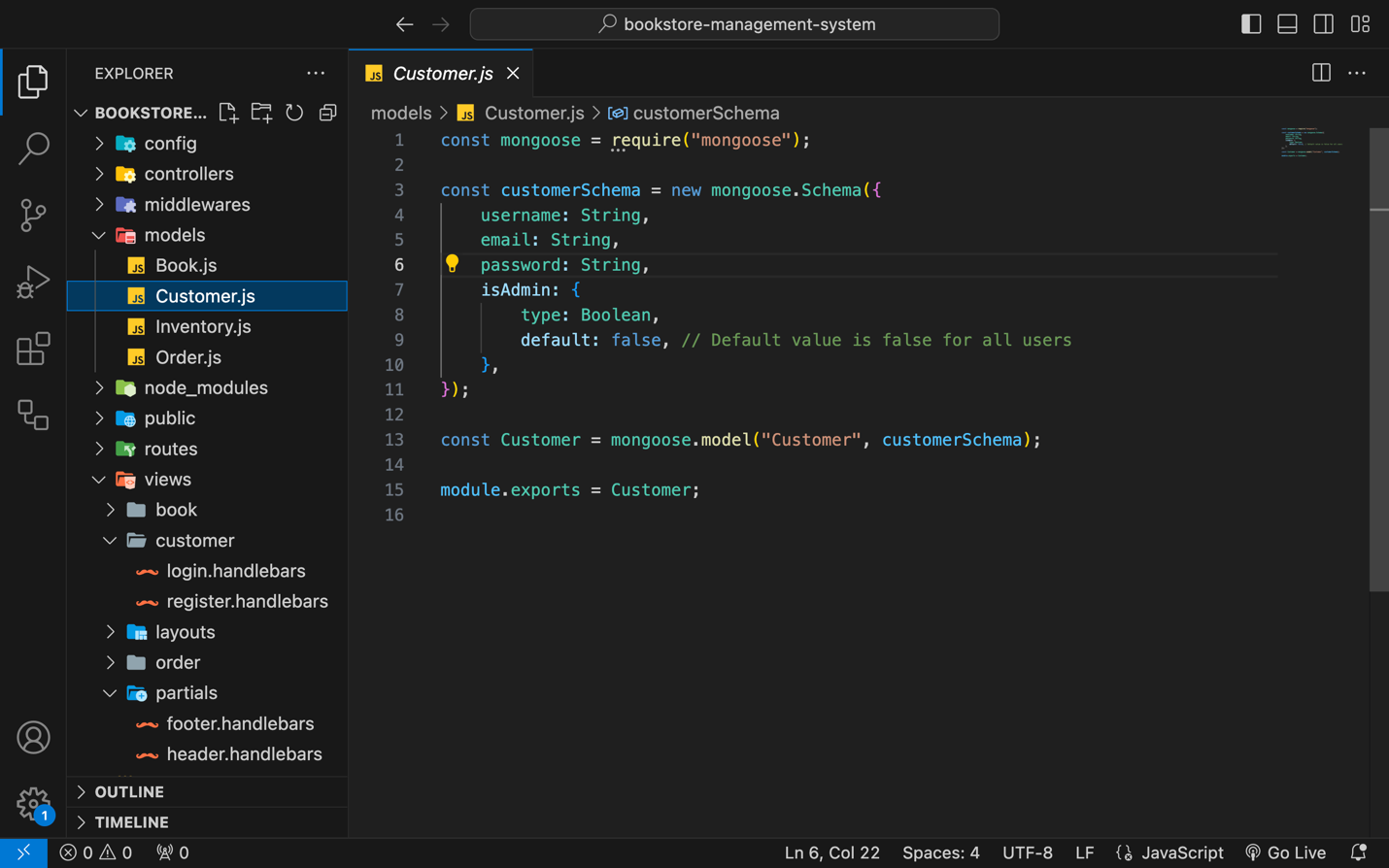


We can also delete book. (after clicking delete button):  
  


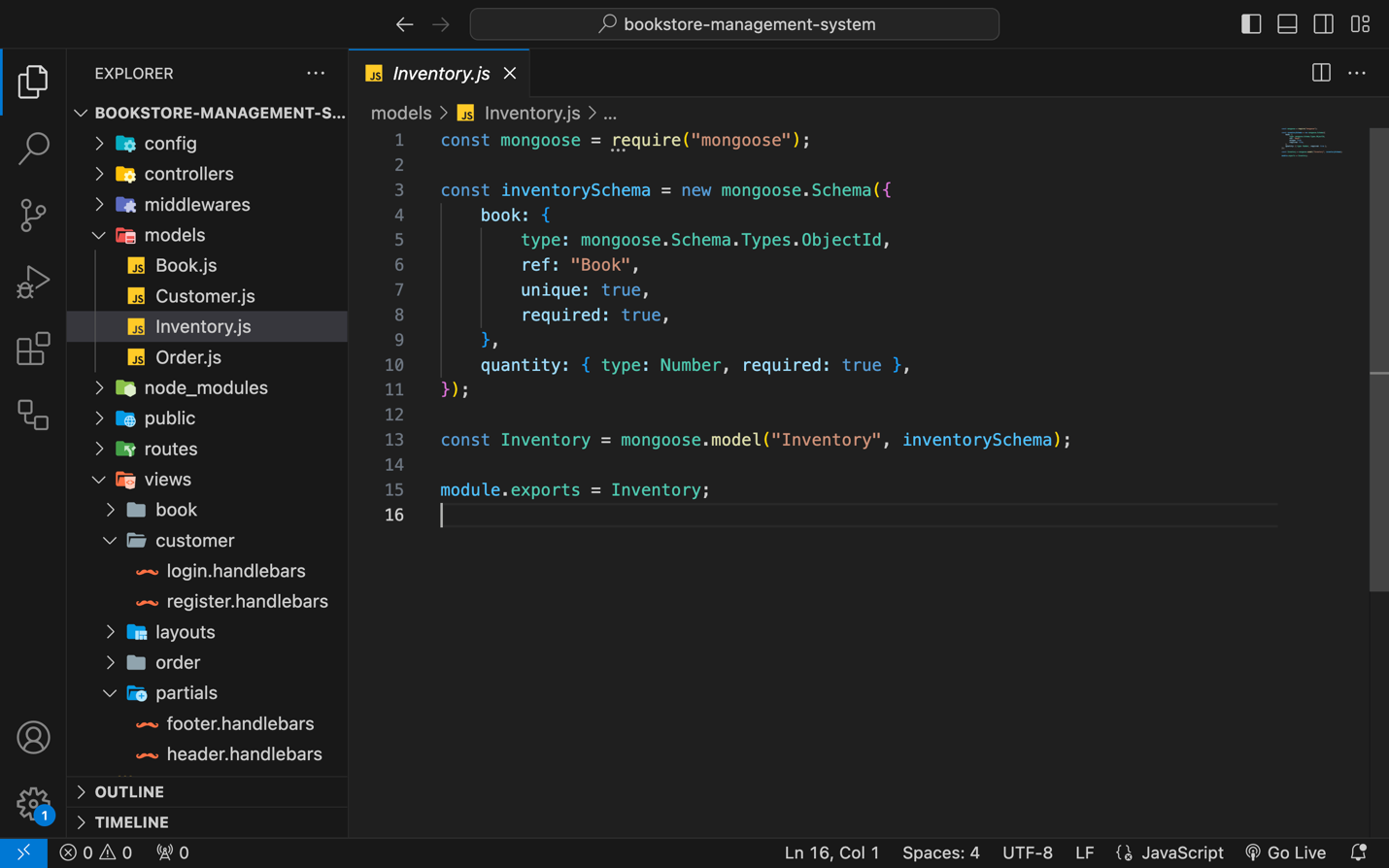


Adding a book:  
  
  
  
After clicking on add book.  
  
  
We have more apis that further needs to be integrated on the front end.  
  
  
Order Book:  
  
  
Tracking Order Status:  


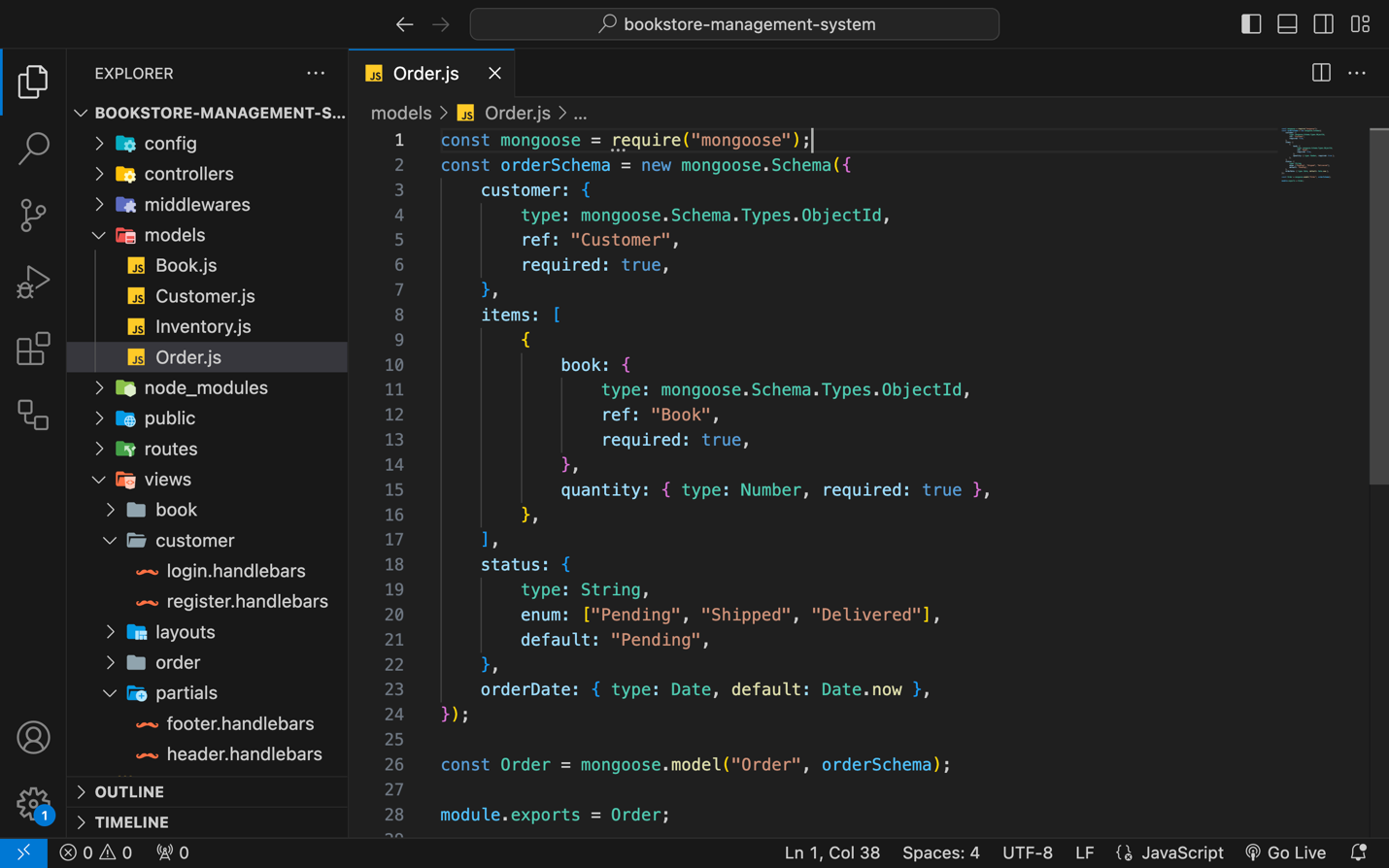
Database Model:  
  
Book.js  


Customer.js  


Inventory.js



Order.js



MongoDB Database:  
User’s password are registered by using encryption

