**BookVerse.co(Online Book Store)**

👤 **Shivasai Shanigarapu**

📅 **Date: 18-03-2025**

📚 **C3(Wipro .NET React)**

👨‍🏫 **Instructor: Mrs.Jyoti S Patil**

**Table of contents:**

| **Content** | **Description** | **Page** |
| --- | --- | --- |
| **Problem Definition and Objectives** | Project goals and objectives | 2 |
| **Frontend & Backend Architecture** | Overview of frontend and backend technologies used in the Project. | 3 |
| **Component Breakdown & API Design** | Breakdown of major components in frontend,API design, endpoints | 4-16 |
| **Database Design & Storage Optimization** | Entity-Relationship Diagram (ERD) ,Optimization techniques for efficient queries | 17 |
| **Detailed Demonstration of your project needs to be given as per the allocated date** | Front-end part responsiveness, as well as some parts of backend like crud operations . | - |
| **Output Details** | Output Images and some key components. | 9-16 |

**Problem Definition and Objective:**

#### **Problem Statement**

Many people love reading books, but they often face difficulties finding their favorite books in local stores. Traditional shopping methods can be time-consuming, and customers may not always get the best prices or book availability .This platform will provide a **smooth and efficient online book shopping experience**.

#### **Objective**

The goal of this project is to develop an **eCommerce web application** using **.NET Core Web API** for the backend and **React** for the frontend. This platform will allow users to **browse, search, and buy books online** easily. The application will provide:

* A **user-friendly interface** for browsing books by category, author, or title.
* A user can also **filter the books** based on favourite **authors, category/ genre** they love to read and also price range they wanted to buy .
* A user can easily find the books which are trending in the market , he/she can **add their favourite books to wishlist** .
* A **secure payment system** for online purchases.
* Secure payment processing with **credit/debit cards, UPI, and digital wallets**.
* A **smooth and fast shopping experience** with easy checkout and order tracking.
* Customers can view their **past orders, order status, and delivery updates**
* Users can leave **reviews and ratings** to help others choose the best books.
* **User authentication and account management** for personalized recommendations.

**Frontend and Backend Architecture:**

**Frontend:** React (for building an interactive user interface)

**Backend:** ASP.NET Core Web API (for handling business logic and database operations)

**Database:** PostgreSQL / MySQL (for storing books, user data, orders, and transactions)

**Authentication:** JWT-based authentication for secure user login and session management

**Payment Gateway:** Integration with **Stripe, Razorpay, or PayPal** for secure transactions

**Key Features of React in Our Project**

1. **Component-Based Architecture:** The application is broken into reusable UI components such as Navbar, ProductList, Cart, Checkout, and Footer.
2. **Virtual DOM:** React updates only the required parts of the UI, making it **highly efficient**.
3. **React Hooks (for Functional Components):**

* **useState:** Manages component state (e.g., cart items, user input).
* **useEffect:** Handles side effects (e.g., fetching books from the API)

4**. React Router:** Enables navigation between pages (Home, Product Details, Cart.)

5**. Axios:** Used for making API calls to the backend to fetch book details and process orders.

### **Key Features of ASP.NET Core in Our Project**

**1. RESTful API:** Provides structured endpoints like for the crud operations.  
 2. **Entity Framework Core (EF Core):** ORM (Object-Relational Mapping) for interacting with the database.  
 3. **Authentication & Authorization:** Uses **JWT (JSON Web Token)** for secure login and role-based access (e.g., Admin, User).  
 4. **Middleware & Exception Handling:** Improves API security and error handling.

**Component Breakdown & API Design:**

**Frontend Structure:**

bookstore-frontend/

│── node\_modules/

│── public/

│ ├── assets/

│ ├── manifest.json

│ ├── robots.txt

│ ├── index.html # Main HTML template

│── src/

│ ├── components/ # Reusable UI components

│ │ ├── Footer.js

│ │ ├── Navbar.js

│ │ ├── SearchBar.js

| | |---- Auth.js

│ ├── context/

│ │ ├── CartContext.js

│ ├── pages/ # Page components

│ │ ├── AddBook.js

│ │ ├── BookDetails.js

│ │ ├── Books.js

│ │ ├── Cart.js

│ │ ├── CartPage.js

│ │ ├── DeleteBook.js

│ │ ├── EditBook.js

│ │ ├── Home.js

│ │ ├── Login.js

│ │ ├── NewArrivals.js

│ │ ├── Wishlist.js

│ ├── services/

│ ├── App.js # Main app component

│ ├── App.css

│ ├── App.test.js # Test file for App.js

│ ├── index.js # Entry point

│ ├── index.css # Global styles

│── .gitignore # Git ignore file

│── package.json # Dependencies and scripts

│── package-lock.json # Locked dependencies

│── README.md # Project documentation

**Backend Structure:**

bookstore-backend/

│── bookstore-backend.sln # Solution file

│── bookstore-backend/ # Main project directory

│ ├── Controllers/ # API Controllers

│ │ ├── BookController.cs

│ │ ├── CartController.cs

│ │ ├── UserController.cs

│ │ ├── WishlistController.cs

│ │ ├── PaymentController.cs

│ ├── Models/ # Database models/entities

│ │ ├── Book.cs

│ │ ├── Cart.cs

│ │ ├── User.cs

│ │ ├── Wishlist.cs

│ │ ├── Payment.cs

│ ├── Data/ # Database context and migrations

│ │ ├── ApplicationDbContext.cs

│ ├── Migrations/ # EF Core migrations (if applicable)

│ ├── appsettings.json # Configuration settings

│ ├── Program.cs # Entry point for the application

│ ├── Startup.cs # Configure services and middleware

│ ├── bookstore-backend.csproj # Project file

│── .gitignore # Ignore files for version control

│── README.md

### **Frontend:**

### **1. Routing (React Router)**

You have implemented routing using react-router-dom.

* **File:** App.js
* **Package Used:** react-router-dom
* **Routing Implementation:**
  + BrowserRouter (Router) wraps the entire app to enable routing.
  + Routes and Route are used to define paths for different pages.
  + Navigate is used to redirect from / to /home.

### **2. State Management**

You have used **Context API** for managing the cart state.

* **File:** CartContext.js (inside the context folder)
* **Package Used:** React’s built-in useContext and useState.

**Backend:**

### **1. API Design Principles**

* **RESTful Structure** → Follows standard HTTP methods (GET, POST, PUT, DELETE).
* **Resource-Oriented** → Operates on the Book resource.
* **JSON Data Exchange** → Requests and responses are structured in JSON format.
* **Stateless** → Each request is independent and contains all necessary information.
* **CRUD Operations** → Implements Create, Read, Update, and Delete functionality.

| **GET** | /api/books | Retrieves all books from the database. |
| --- | --- | --- |

| **GET** | /api/books/{id} | Fetches details of a specific book by its id. |
| --- | --- | --- |

| **POST** | /api/books | Adds a new book to the database. |
| --- | --- | --- |

| **PUT** | /api/books/{id} | Updates an existing book’s details by id. |
| --- | --- | --- |

| **DELETE** | /api/books/{id} | Deletes a book from the database by id. |
| --- | --- | --- |

Json Example Body for **GET**

{

"id": 1,

"title": "Atomic Habits",

"author": "James Clear",

"img": "https://example.com/atomic.jpg",

"price": 20.50,

"description": "A book about building good habits.",

"genre": "Self-help",

"isbn": "9780735211292"

}

Json Example Body for **POST/PUT**

{

"title": "The Power of Now",

"author": "Eckhart Tolle",

"img": "https://example.com/powerofnow.jpg",

"price": 18.99,

"description": "A book on mindfulness and spiritual enlightenment.",

"genre": "Self-help",

"isbn": "9781577314806"

}

## **Authentication & Authorization Flow**

### **User Login**

* A user (either Admin or User) logs in through the React UI.
* The credentials are sent to the ASP.NET Core Web API backend via an API request.

### **Backend Validates Credentials**

* The backend verifies the username and password against the database.
* If valid, a JWT token is generated and sent back along with the user’s role (Admin/User).

### **Storing JWT & Role**

* The frontend stores the token and role in localStorage.

### **Role-Based Routing**

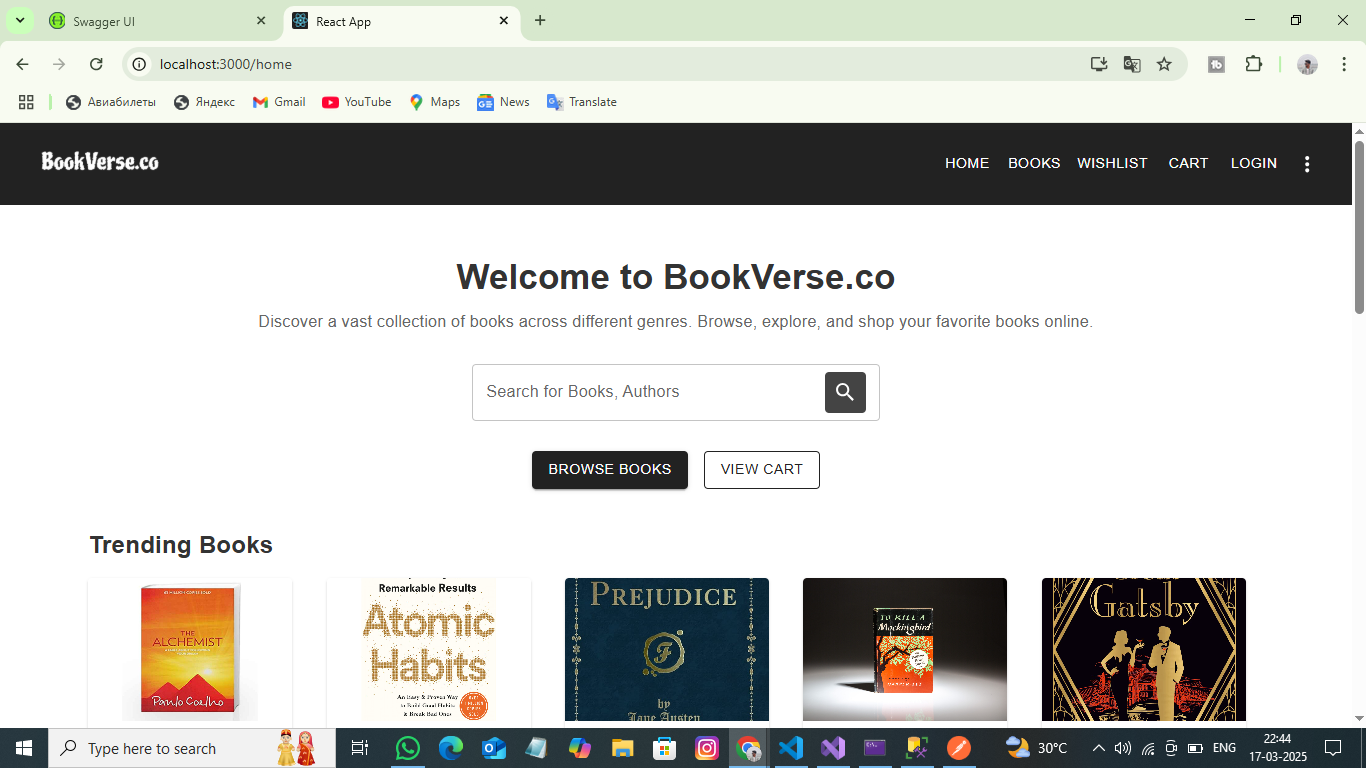
* Users and Admins are redirected to their respective dashboards:
  + Users → / (Home)
  + Admins → /admin-dashboard
* Unauthorized access is restricted using Protected Routes in React.

### **Logout Implementation**

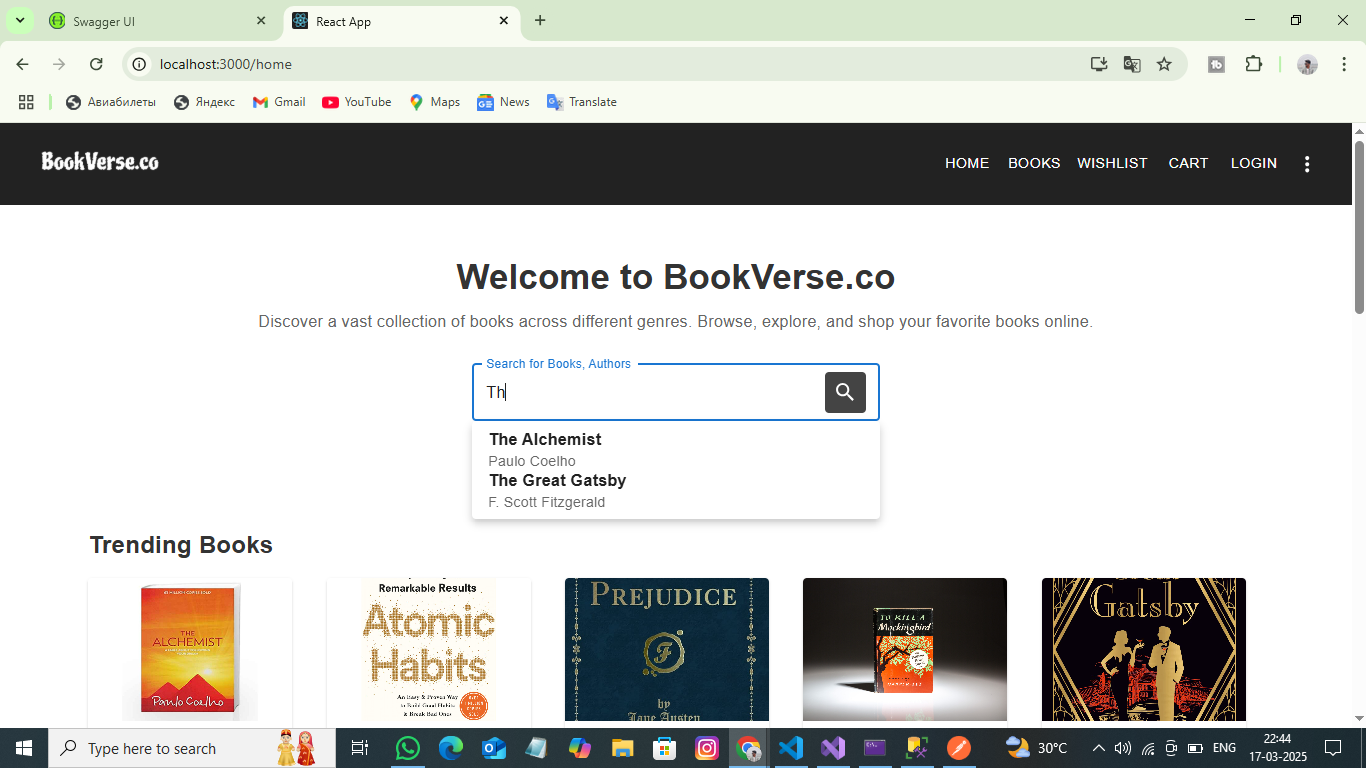
* Clicking Logout clears the localStorage and redirects the user to the login page.

**Output Details:**

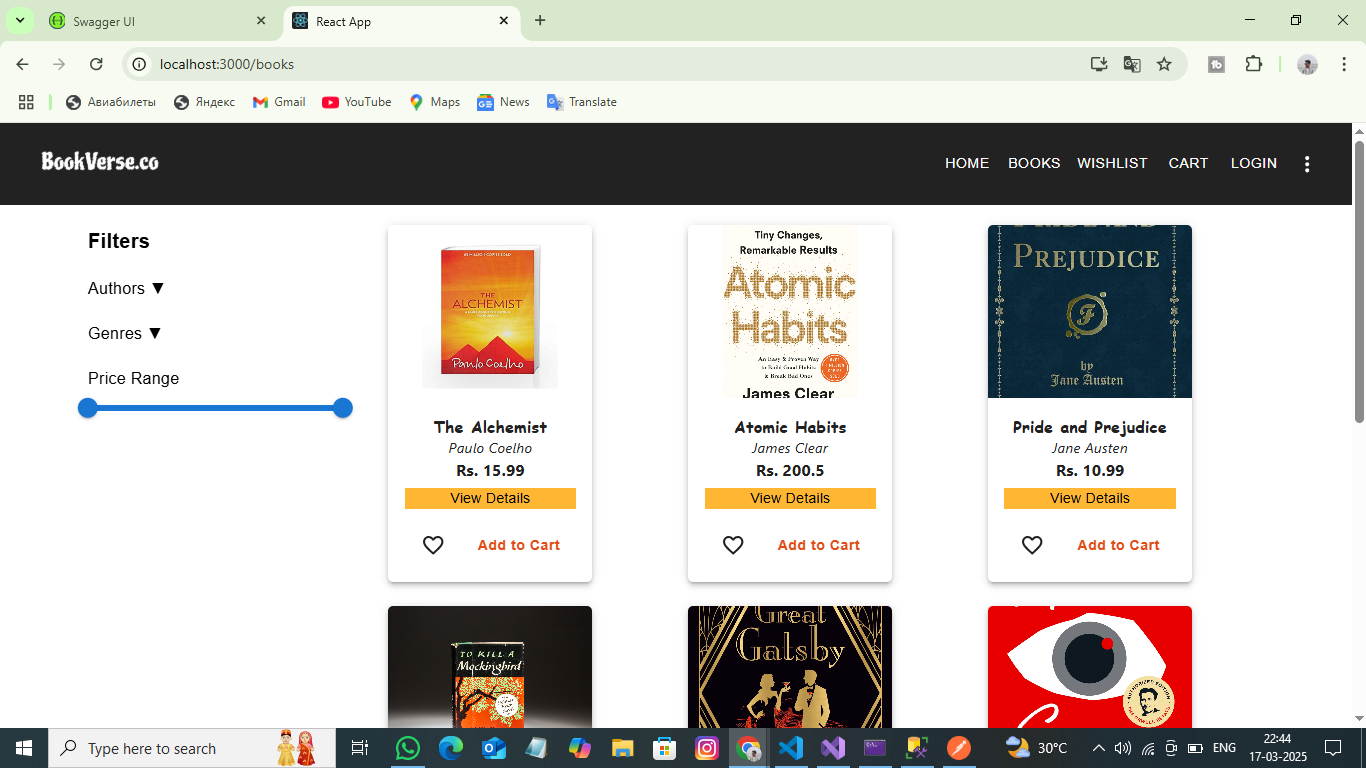
**HomePage:**



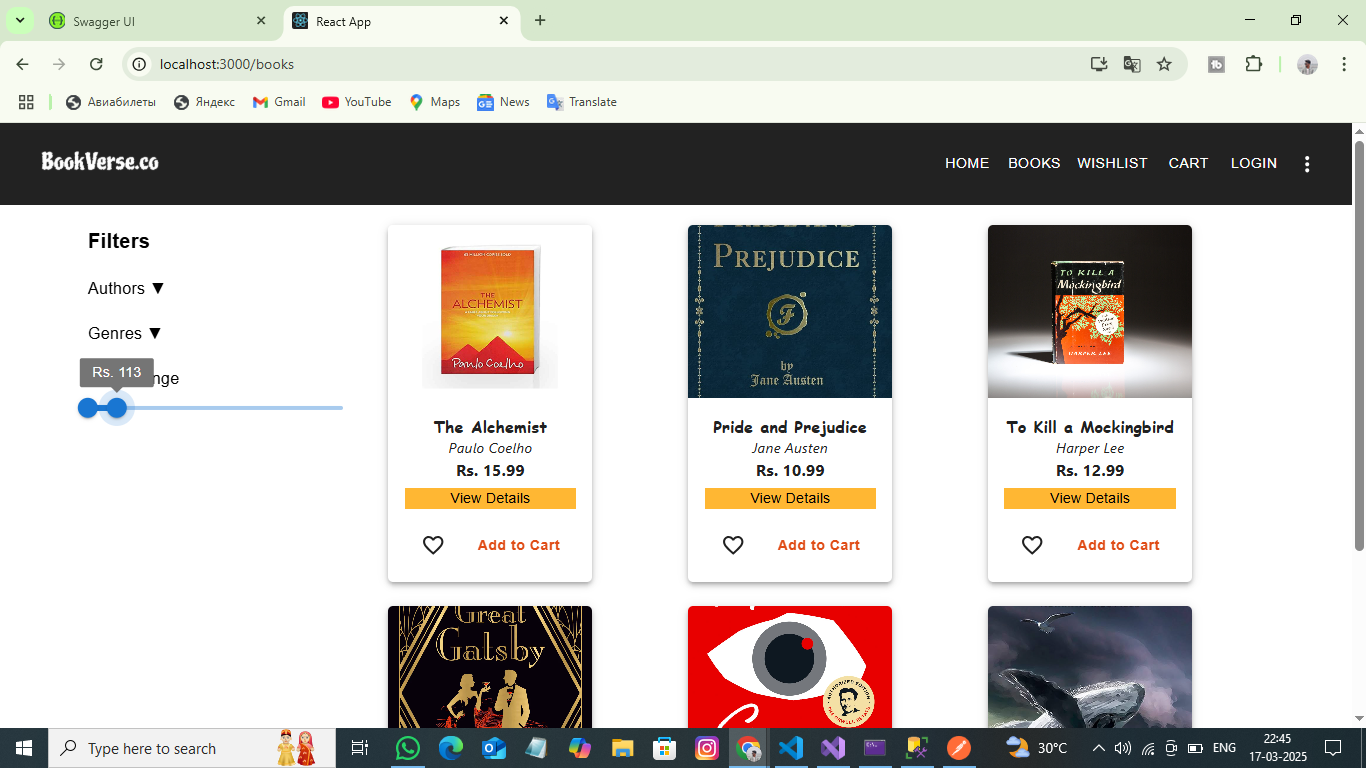
**Search Books:**



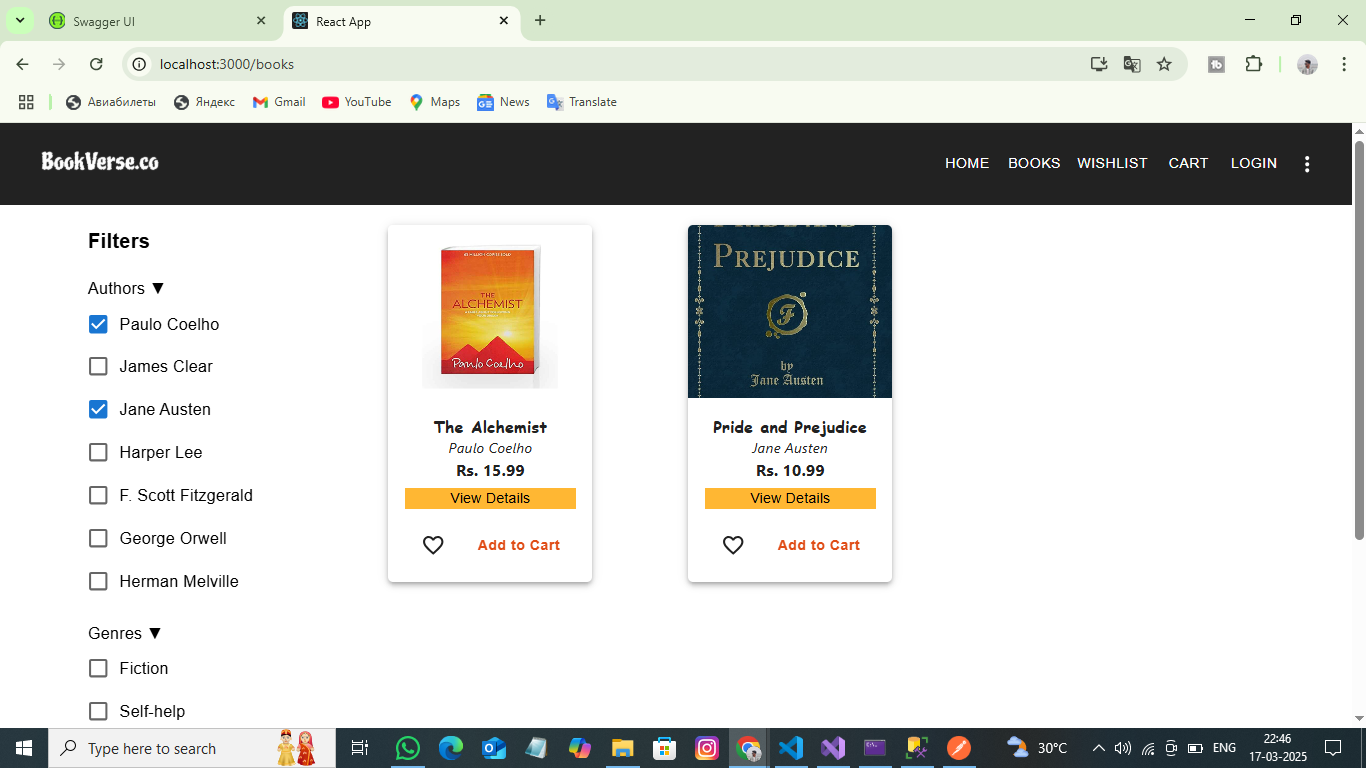
**BooksPage:**



**Filter Books:**



**Filter Based on Authors and Genres:**



**Authentication & Authorization**

Authentication and Authorization are fundamental security concepts that control user access to a system.

**Authentication** is the process of verifying a user’s identity before granting access to a system.

**Types of Authentication:**

* **Password-Based Authentication** – Using a username & password.
* **Token-Based Authentication (JWT, OAuth)** – Using a generated token for session validation.
* **Multi-Factor Authentication (MFA)** – Combining two or more verification methods (e.g., password + OTP).

Authorization is the process of determining what resources and actions a user can access after authentication.

**Types of Authorization:**

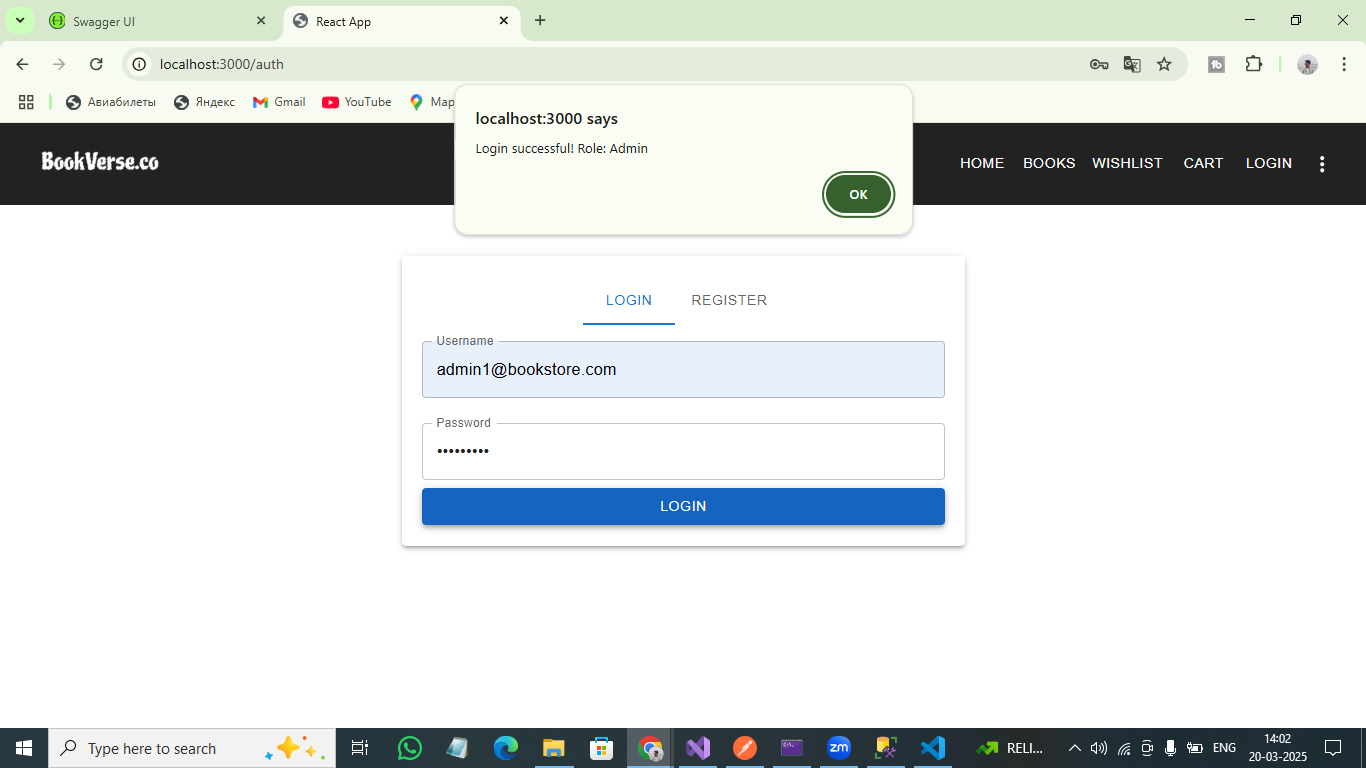
* **Role-Based Access Control (RBAC)** – Assigns users predefined roles (e.g., Admin, User).
* **Permission-Based Access Control** – Grants users specific actions (e.g., Read, Write, Delete).

**Implementation in Project**

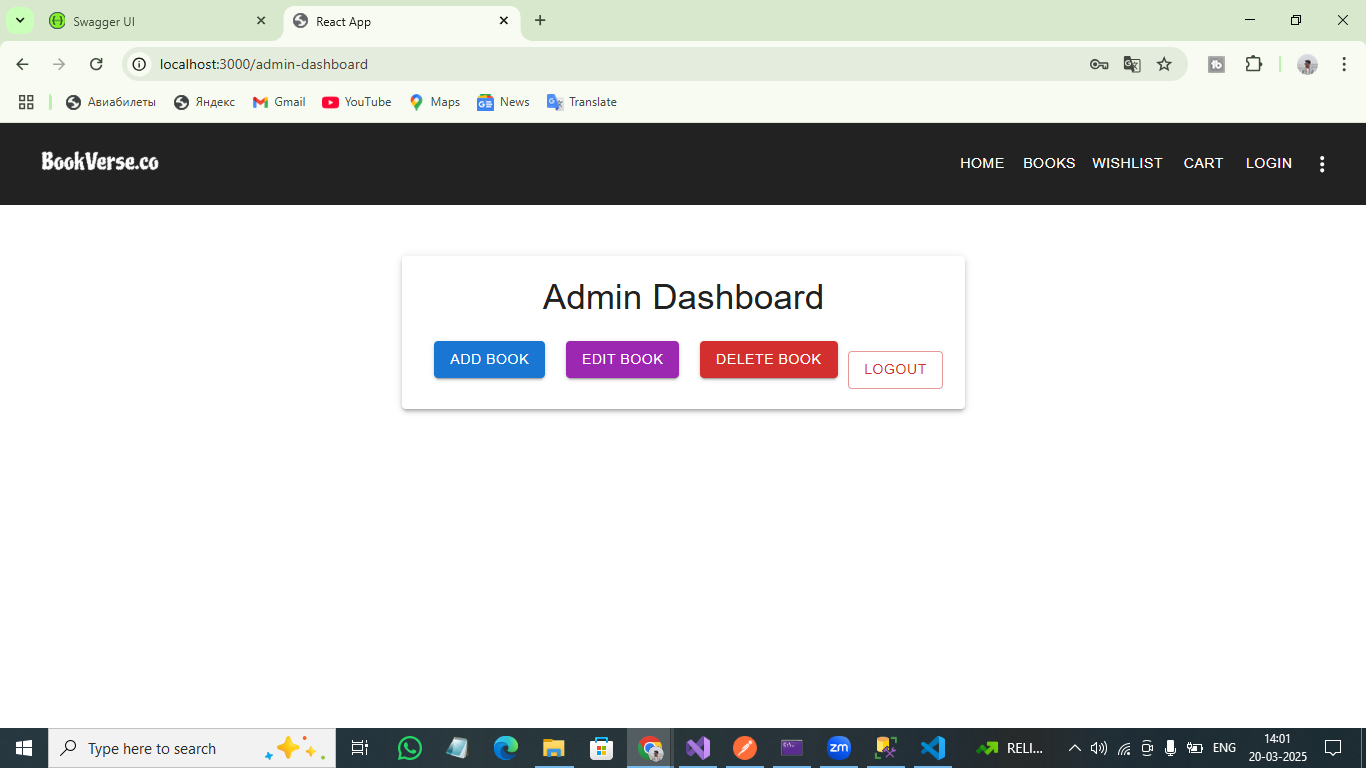
**Implemented role-based authentication for User , Admin**

* User can login and view books , search , filter , order , add to cart , wishlist etc .
* An Admin can Add new books , edit books , Delete books as well .

**Admin login:**

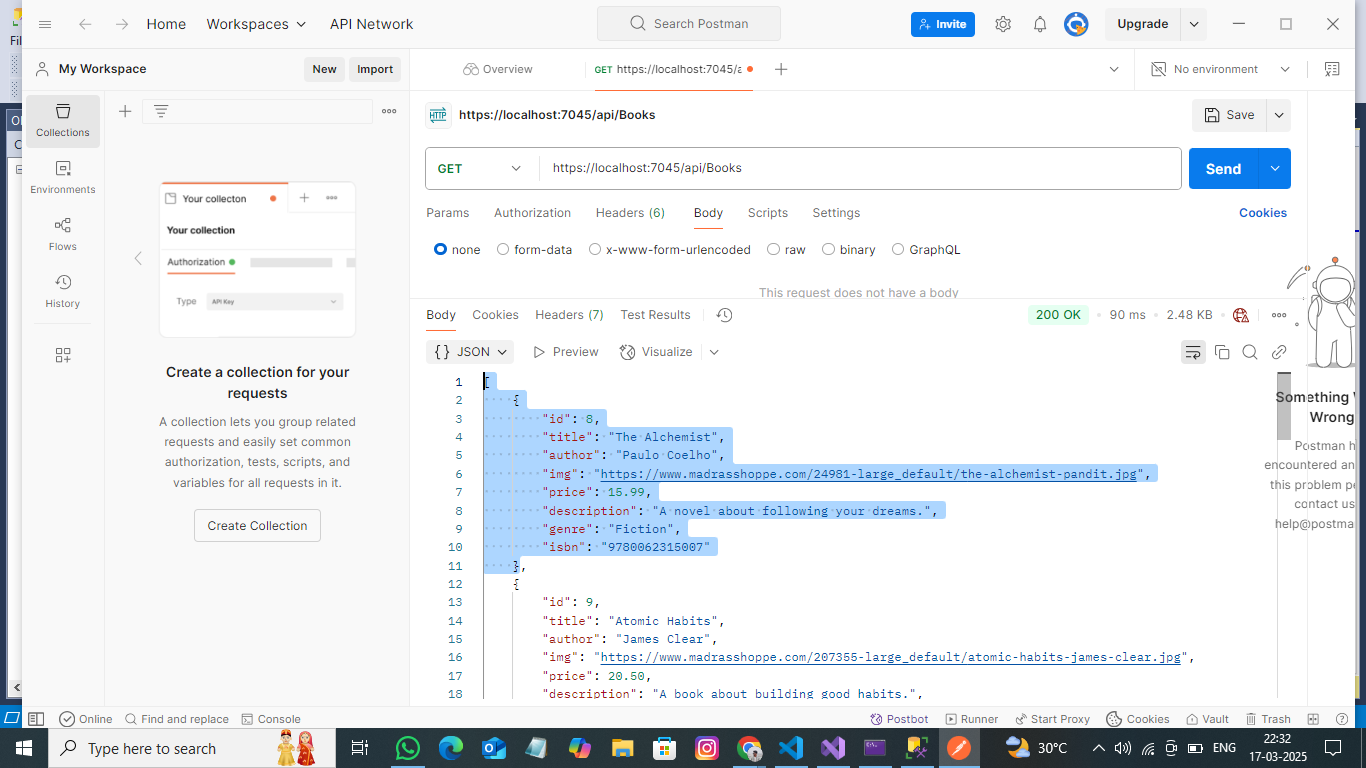
****

**Admin DashBoard:**

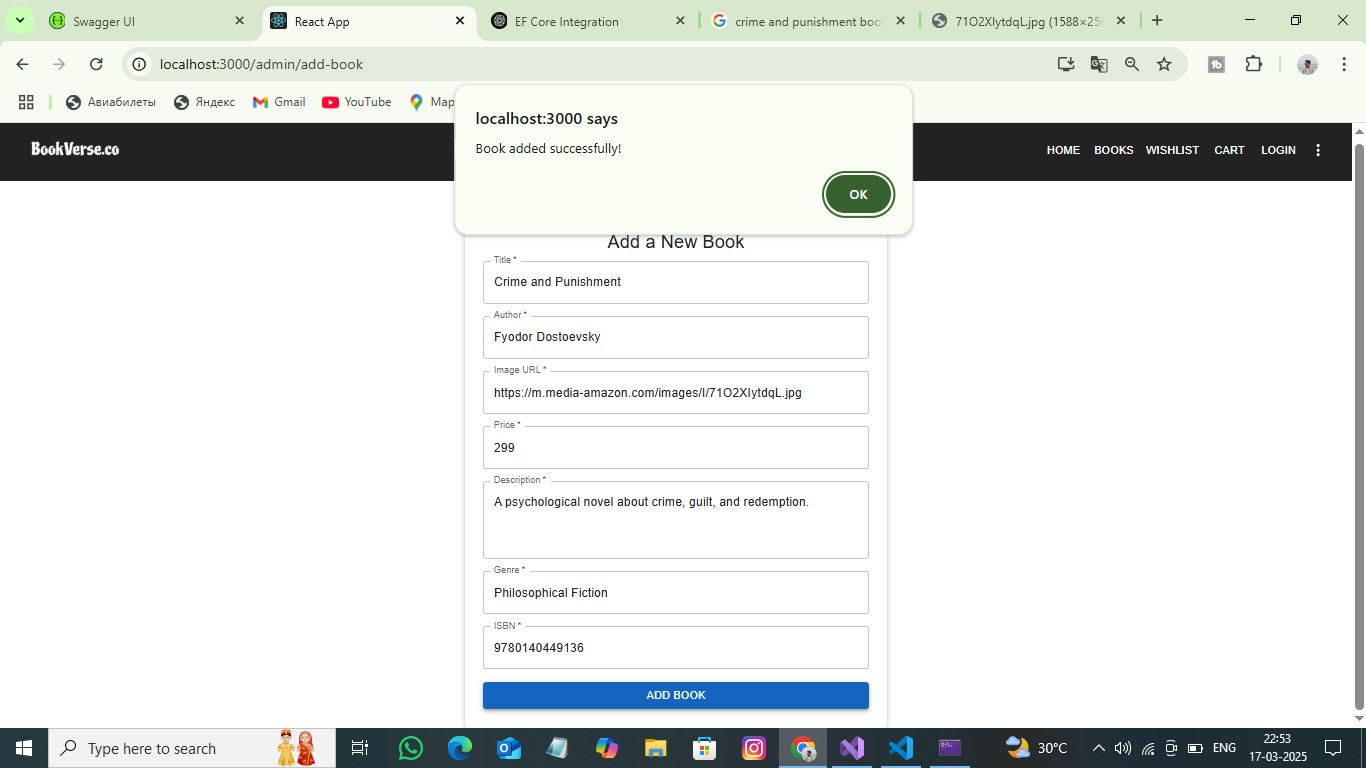
****

**Admin CRUD operations:**

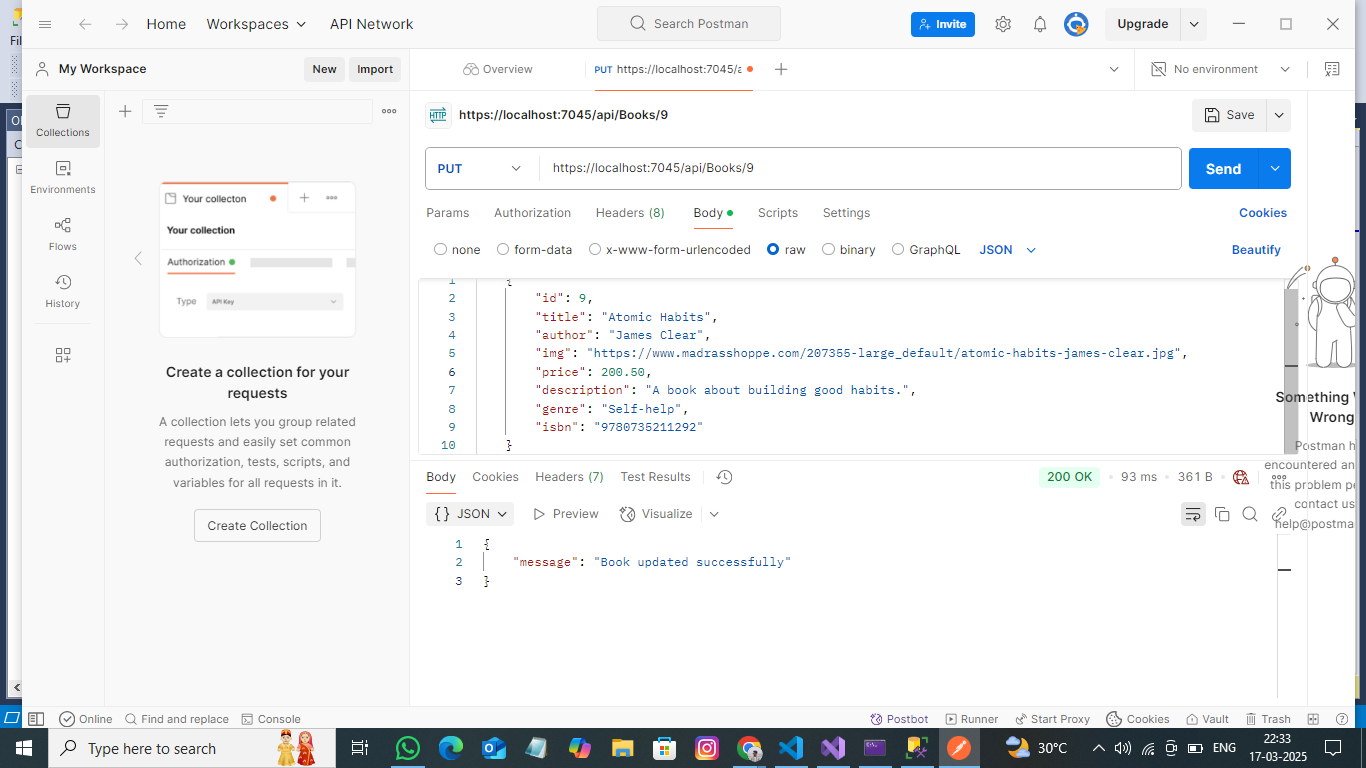
**GetBooks:**

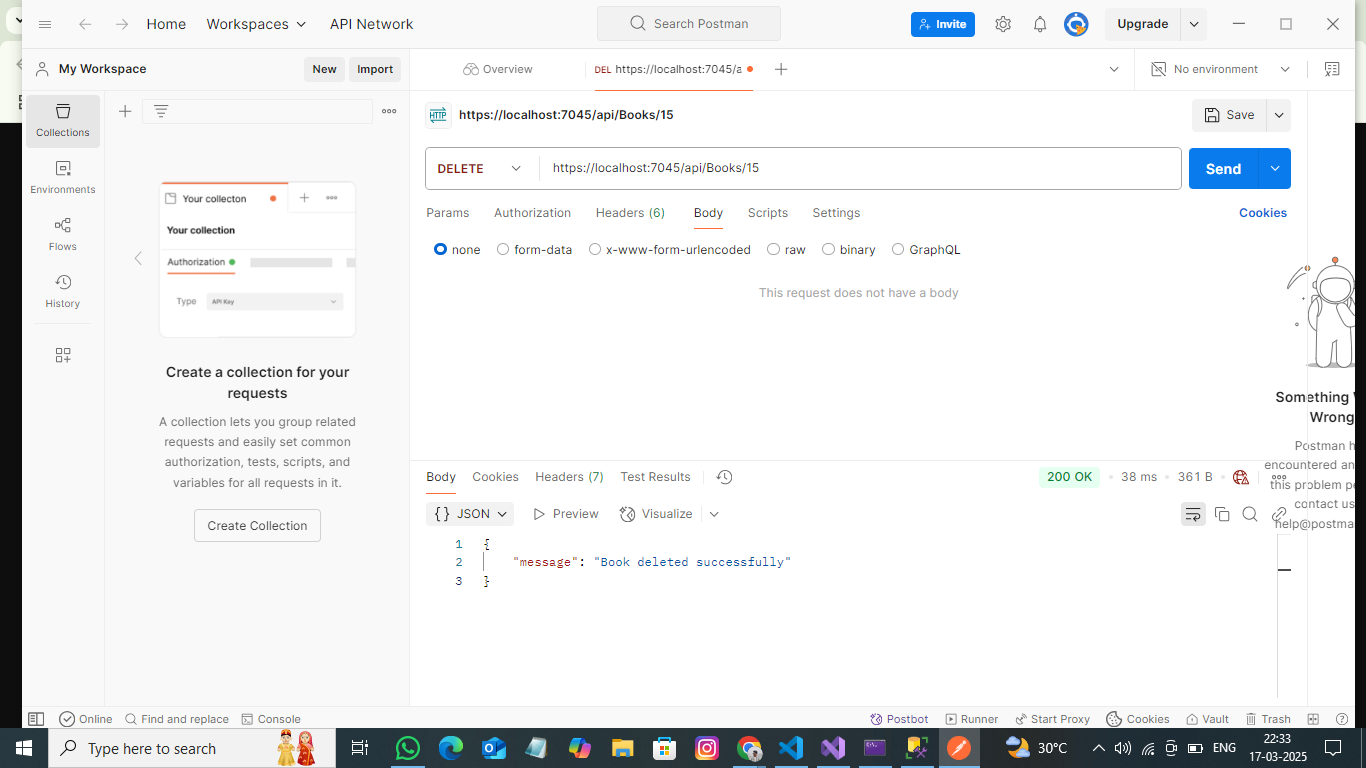


Add Book:

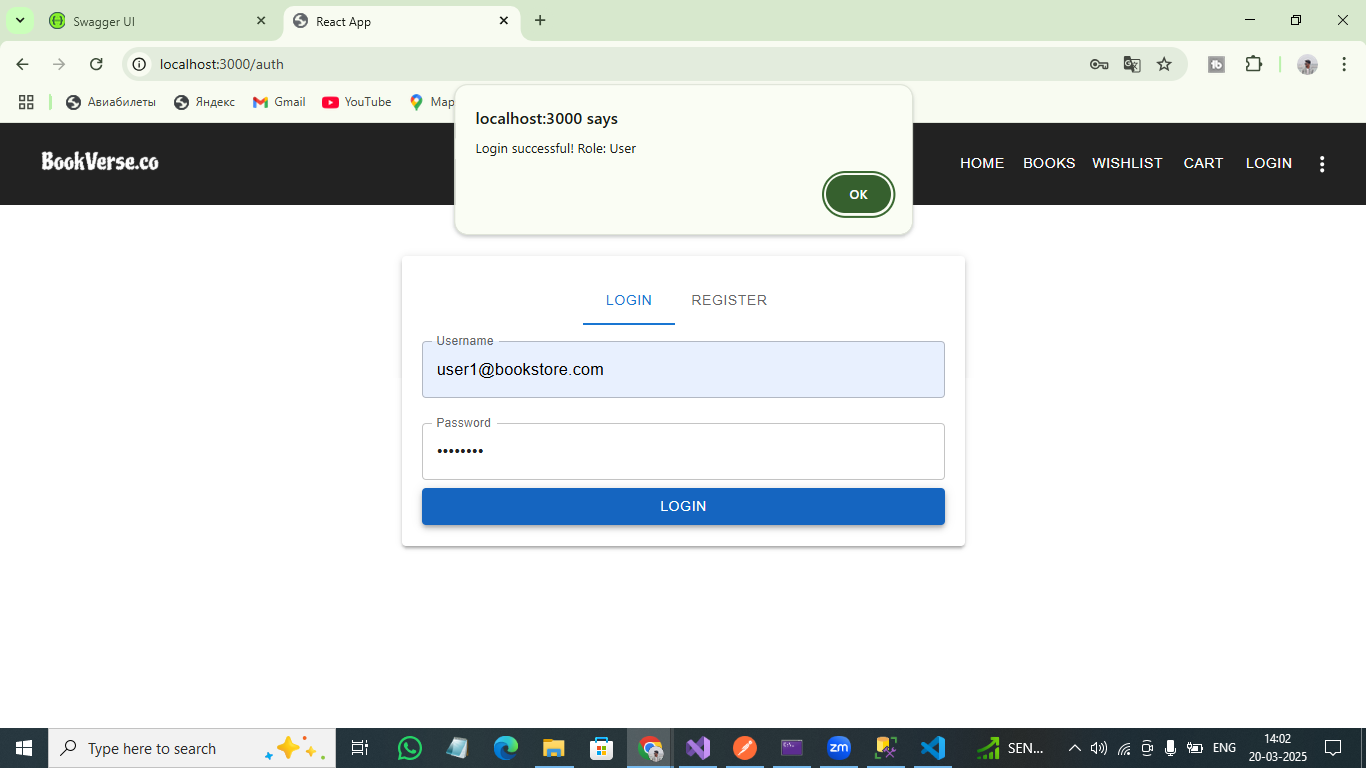
****

Edit Book:



DeleteBook:  


**UserLogin:**

****

# **SQL Server & Its Implementation in Our Project**

## **1. Overview of SQL Server**

Microsoft SQL Server (MSSQL) is a **relational database management system (RDBMS)** used to store and manage structured data. It supports **T-SQL (Transact-SQL)** for querying, managing transactions, and implementing stored procedures.

### **Key Features of SQL Server:**

✅ **ACID Compliance** → Ensures reliable transactions.  
 ✅ **Stored Procedures** → Optimized query execution.  
 ✅ **User Authentication** → Secure login via SQL or Windows Authentication.  
 ✅ **Scalability & Performance** → Supports indexing, partitioning, and caching.  
 ✅ **Security** → Role-based access control, encryption, and audit logs.

## **Database Design & Storage Optimization**

## **Overview of SQL Server**

Microsoft SQL Server (MSSQL) is a **relational database management system (RDBMS)** used to store and manage structured data. It supports **T-SQL (Transact-SQL)** for querying, managing transactions, and implementing stored procedures.

**Key Features:**

**Stored Procedures** → Optimized query execution.

**User Authentication** → Secure login via SQL or Windows Authentication.

**Scalability & Performance** → Supports indexing, partitioning, and caching.

**Security** → Role-based access control, encryption, and audit logs.

### **Database Setup & Configuration**

### **Create Database & Tables**

CREATE DATABASE BookStoreDB;

USE BookStoreDB;

CREATE TABLE Users (

id INT PRIMARY KEY IDENTITY(1,1),

userName NVARCHAR(100) UNIQUE NOT NULL,

passwordHash NVARCHAR(255) NOT NULL,

role NVARCHAR(50) CHECK (role IN ('Admin', 'User'))

);

CREATE TABLE Books (

id INT PRIMARY KEY IDENTITY(1,1),

title NVARCHAR(255) NOT NULL,

author NVARCHAR(100),

price DECIMAL(10,2) NOT NULL,

stock INT NOT NULL

);

**Thank You**

**Shivasai Shanigarapu**

**Batch:C3(Wipro .NET React).**

**Attachment files:**

**client.zip**

**server.zip**

**Database.zip**

**Github link :**[**github**](https://github.com/ShanigarapuShivasai/BookStoreApp_)