**Full Stack Development with MERN**

**Final Report format**

**BookNest – Where Stories Nestle**

(MERN Stack Online Book Store)

Introduction

🔹 Project Title: BookNest – Where Stories Nestle

**🔹 Team Members:**

| **Name** | **Role** |
| --- | --- |
| Team Leader | Full Stack Developer (Frontend + Backend) |
| Team Member 2 | UI/UX Designer |
| Team Member 3 | Backend Developer |
| Team Member 4 | Database & Deployment Engineer |

**Project Overview**

**🔹 Purpose:**

BookNest is a full-stack MERN application designed to provide a secure, user-friendly online platform for browsing, searching, and purchasing books. The goal is to create a modern, scalable bookstore with smooth navigation and secure authentication.

**🔹 Key Features:**

* User Registration & Login (JWT Authentication)
* Browse 100+ Books with images
* Search books by title or author
* Add to Cart functionality
* Place Orders
* View Order History
* Cloud Deployment (Render + Vercel)
* Secure MongoDB Atlas database

**Architecture**

**🔹 Frontend (React.js – Vite)**

* Component-based architecture
* Pages: Home, Register, Login, Books, Cart, Orders
* Axios used for API calls
* Protected routes for authentication
* Hosted on Vercel

🔹 Backend (Node.js + Express.js)

* RESTful API structure
* JWT authentication
* Middleware for validation
* CORS configured for production
* Hosted on Render

🔹 Database (MongoDB Atlas)

Collections:

* Users
* Books
* Orders

Database Interactions:

* Mongoose models
* CRUD operations
* Cloud-hosted NoSQL database

**Setup Instructions**

**🔹** Prerequisites

* Node.js (v18+)
* MongoDB Atlas account
* Git
* npm

**🔹 Installation**

**Step 1: Clone Repository**

**git clone** https://github.com/Kushulatha-k/BookNest.git

**cd booknest**

**Step 2: Backend Setup**

cd backend

npm install

Create .env file:

PORT=5000

MONGO\_URI=your\_mongodb\_connection\_string

JWT\_SECRET=your\_secret\_key

Run backend:

npm run dev

**Step 3: Frontend Setup**

cd frontend

npm install

npm run dev

**Folder Structure**

🔹 Client (Frontend)

frontend/

├── src/

│ ├── components/

│ ├── pages/

│ ├── api/

│ ├── App.jsx

│ └── main.jsx

├── package.json

🔹 Server (Backend)

backend/

├── models/

├── routes/

├── config/

├── server.js

├── package.json

└── .env

**Running the Application**

🔹 Frontend

cd frontend

npm run dev

Runs at:

http://localhost:5173

🔹 Backend

cd backend

npm run dev

Runs at:

http://localhost:5000

**API Documentation**

**🔹** User Routes

| Method | Endpoint | Description |
| --- | --- | --- |
| POST | /api/users/register | Register new user |
| POST | /api/users/login | Login user |

Example Request:

{

"email": "user@gmail.com",

"password": "123456"

**}**

**🔹 Book Routes**

| Method | Endpoint | Description |
| --- | --- | --- |
| GET | /api/books | Get all books |
| GET | /api/books/seed | Seed books |

**🔹 Order Routes**

| Method | Endpoint | Description |
| --- | --- | --- |
| POST | /api/orders | Place order |
| GET | /api/orders/myorders | Get user orders |

**Authentication**

* JWT-based authentication
* Token stored in localStorage
* Protected routes
* Password encrypted using bcrypt
* Middleware validates token before accessing protected APIs

**User Interface**

The UI includes:

* Animated backgrounds
* Responsive design
* Card-based book layout
* Navigation bar
* Styled cart and order pages

(Screenshots included in README)

**Testing**

**Testing Strategy:**

* API Testing using Postman
* Manual Integration Testing
* Authentication Testing
* Deployment Testing

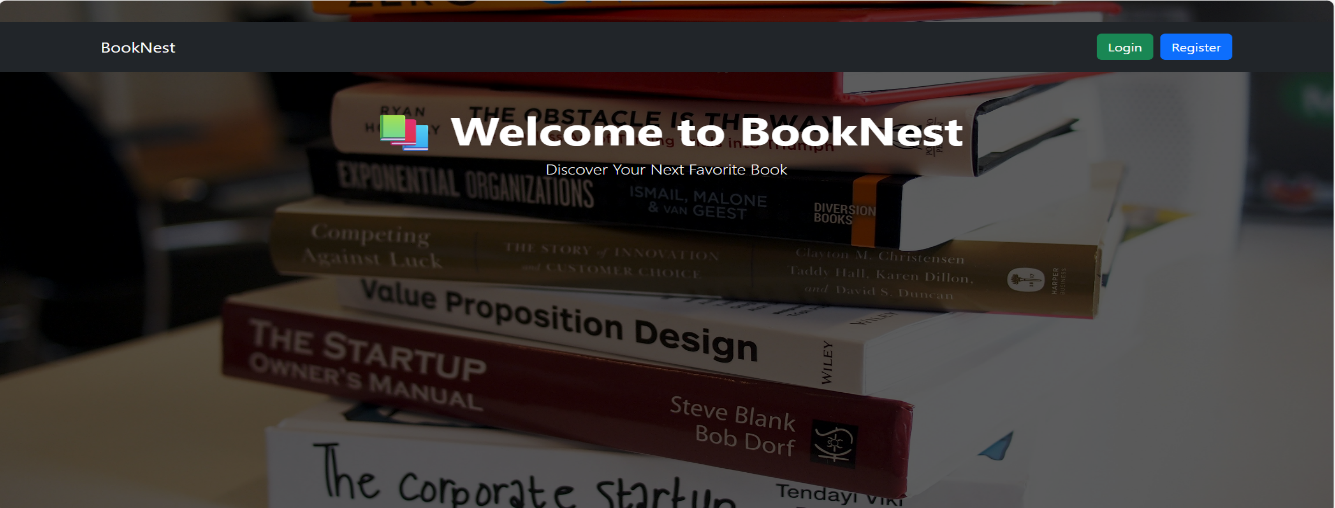
All modules tested before production deployment.

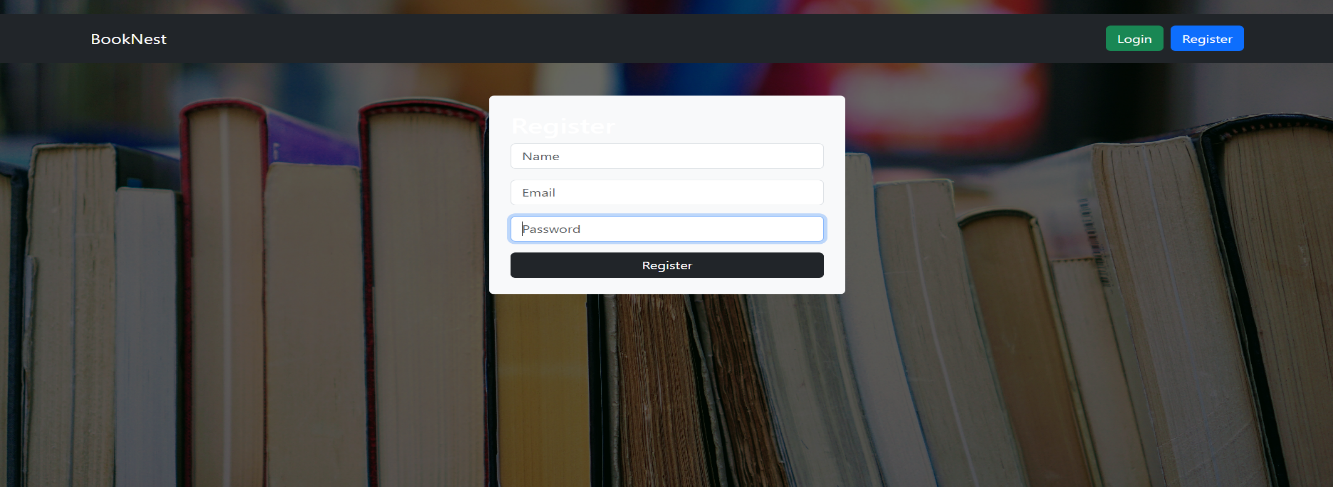
**Screenshots / Demo**

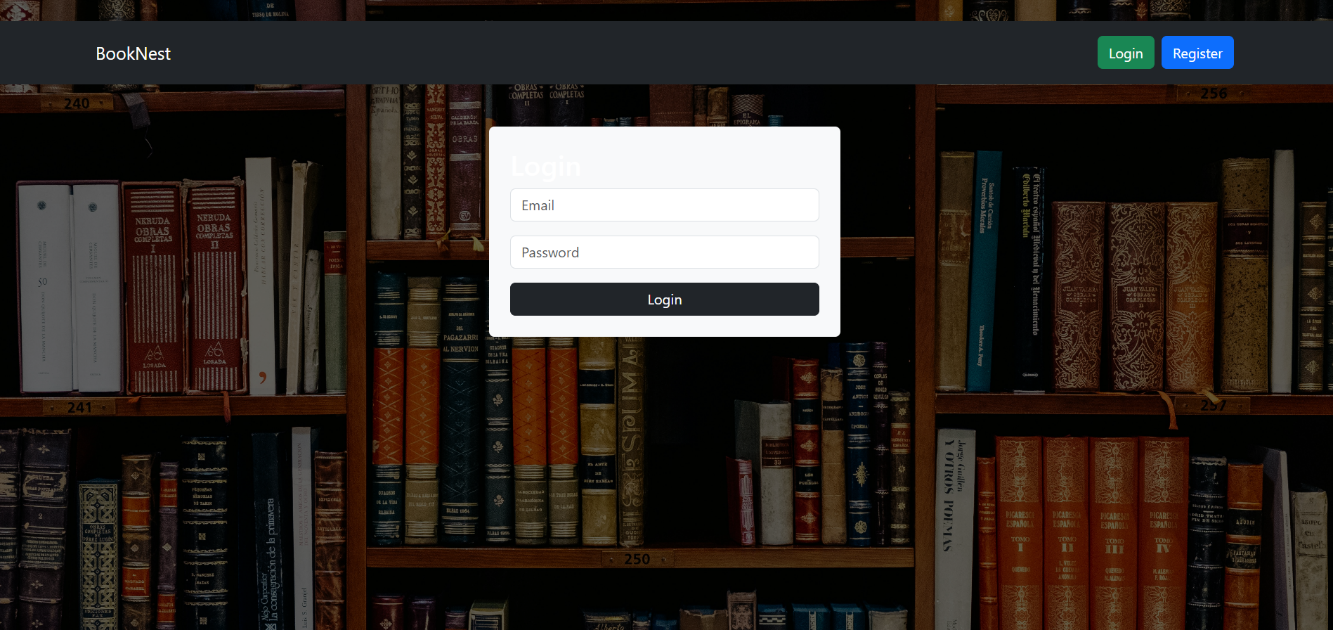
**🔗 Live Demo:**

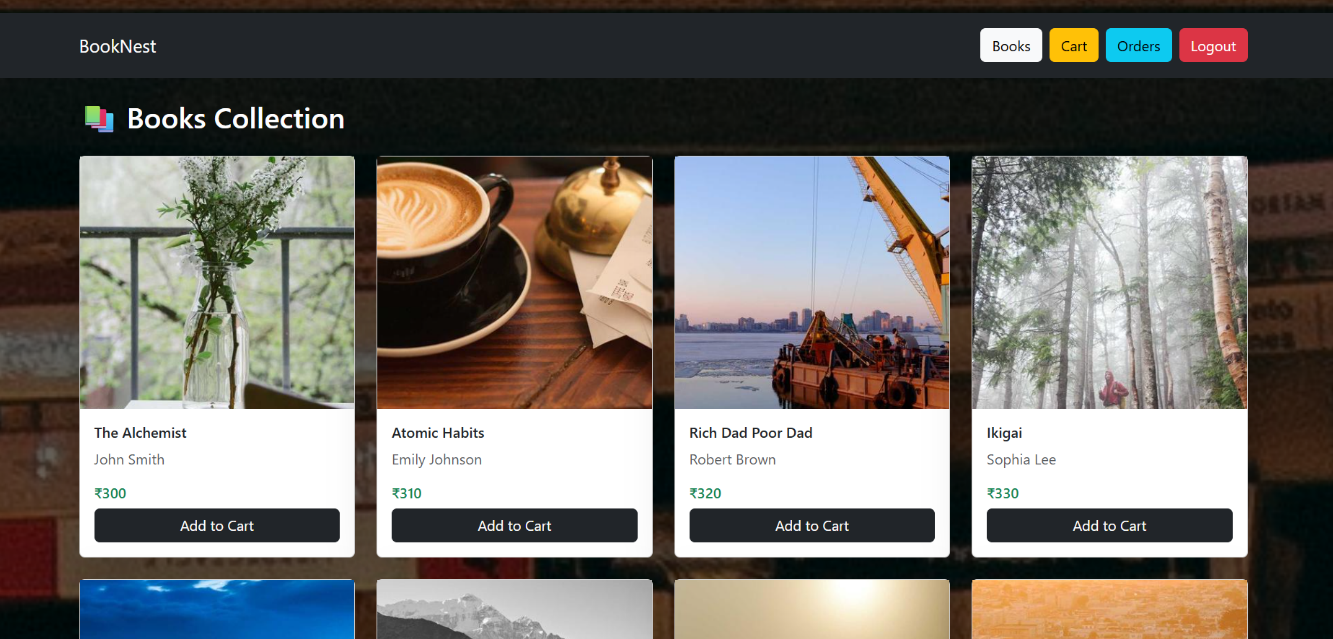
**Frontend:** [https://book-nest-six-topaz.vercel.app/](%20%20https:/book-nest-six-topaz.vercel.app/)

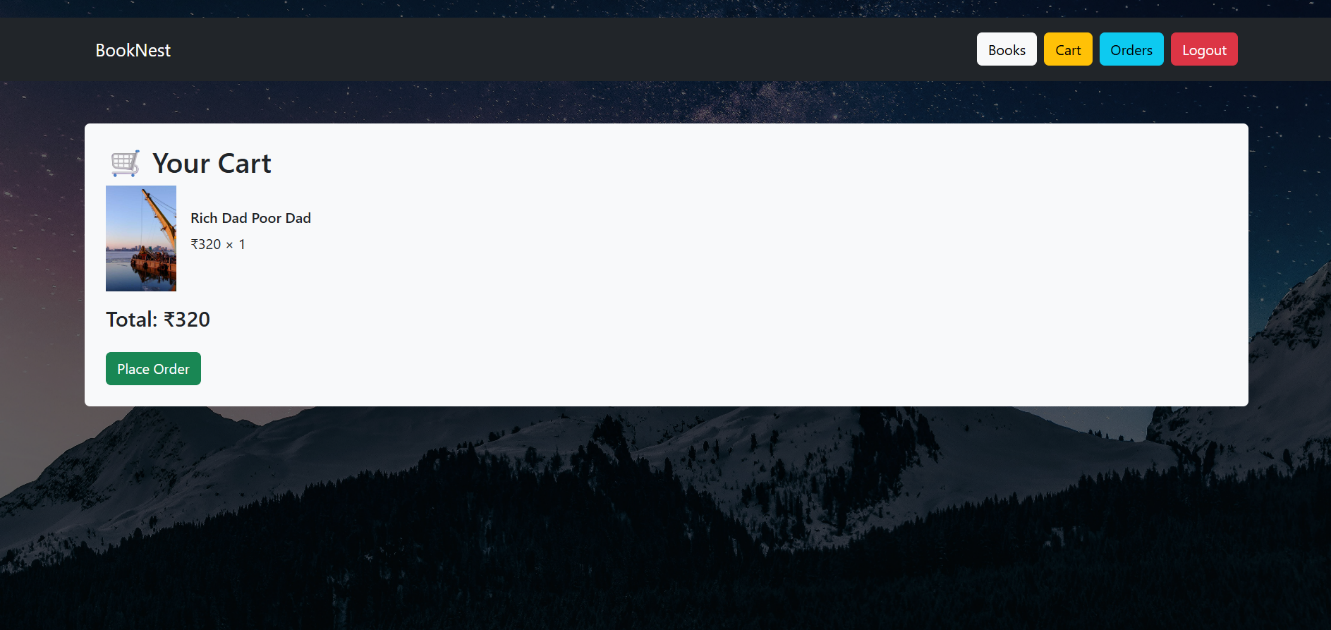
**Backend :**  <https://booknest-backend-0buw.onrender.com>

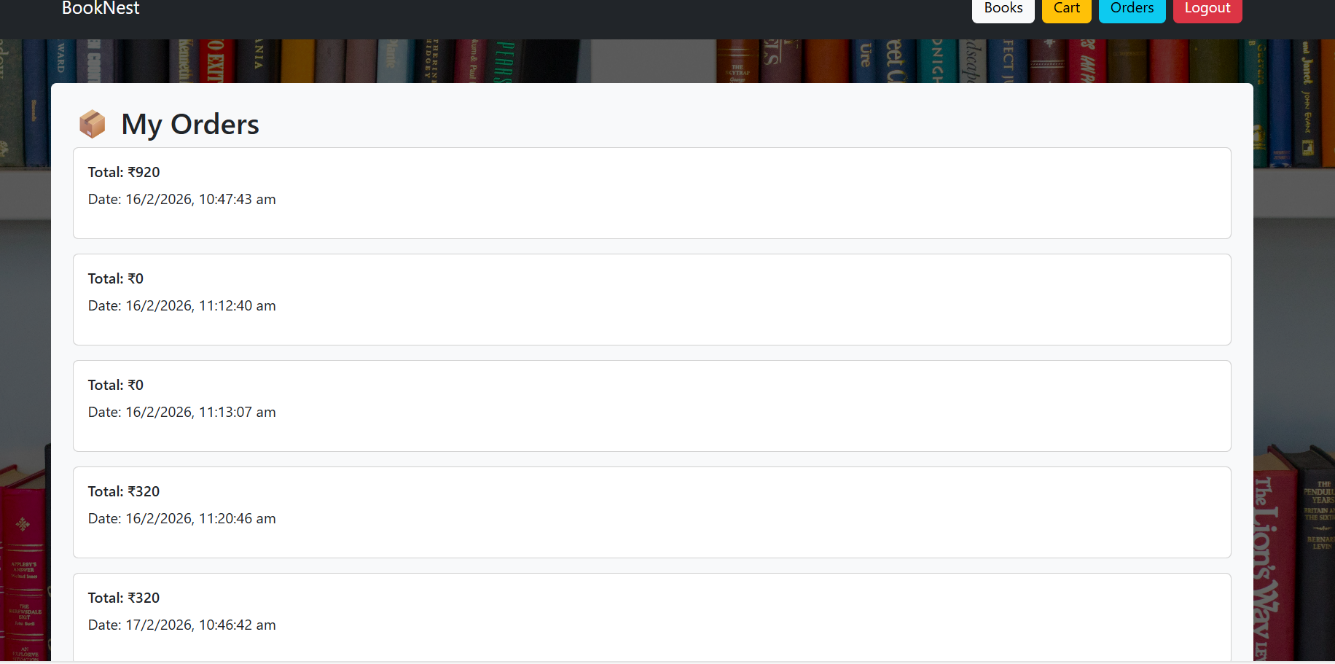












**Known Issues**

* No payment gateway integration (currently demo order placement)
* No email verification
* Limited admin dashboard features
* Basic search (no AI recommendation yet)

**Future Enhancements**

* Payment Gateway Integration (Stripe/Razorpay)
* AI-based Book Recommendation System
* Admin Dashboard
* Order Tracking
* Mobile Application (React Native)
* Wishlist Feature
* Review & Rating System

**Conclusion :** BookNest is a scalable MERN stack application that provides a secure, modern online bookstore platform. It demonstrates full-stack development, cloud deployment, authentication, and REST API integration using modern web technologies.

**APPENDIX:**

Source Code(if any):

import Navbar from "../components/Navbar";

import PageWrapper from "../components/PageWrapper";

function Home() {

  return (

    <PageWrapper bg="https://images.unsplash.com/photo-1512820790803-83ca734da794">

      <Navbar />

      <div className="container text-center mt-5">

        <h1 className="display-4 fw-bold">📚 Welcome to BookNest</h1>

        <p className="lead">Discover Your Next Favorite Book</p>

      </div>

    </PageWrapper>

  );

}

export default Home;

import { useState } from "react";

import axios from "../api/axios";

import { useNavigate } from "react-router-dom";

import Navbar from "../components/Navbar";

import PageWrapper from "../components/PageWrapper";

function Register() {

  const [form, setForm] = useState({

    name: "",

    email: "",

    password: ""

  });

  const navigate = useNavigate();

  const submitHandler = async (e) => {

    e.preventDefault();

    try {

      await axios.post("/users/register", form);

      alert("Registered Successfully ✅");

      navigate("/login");

    } catch (error) {

      alert("Registration Failed ❌");

      console.log(error.response?.data);

    }

  };

  return (

    <PageWrapper bg="https://images.unsplash.com/photo-1495446815901-a7297e633e8d">

      <Navbar />

      <div className="container mt-5 bg-light p-4 rounded" style={{ maxWidth: "400px" }}>

        <h2>Register</h2>

        <form onSubmit={submitHandler}>

          <input

            className="form-control mb-3"

            placeholder="Name"

            value={form.name}

            onChange={(e) =>

              setForm({ ...form, name: e.target.value })

            }

          />

          <input

            className="form-control mb-3"

            placeholder="Email"

            value={form.email}

            onChange={(e) =>

              setForm({ ...form, email: e.target.value })

            }

          />

          <input

            type="password"

            className="form-control mb-3"

            placeholder="Password"

            value={form.password}

            onChange={(e) =>

              setForm({ ...form, password: e.target.value })

            }

          />

          <button className="btn btn-dark w-100">

            Register

          </button>

        </form>

      </div>

    </PageWrapper>

  );

}

export default Register;

import { useEffect, useState } from "react";

import axios from "../api/axios";

import Navbar from "../components/Navbar";

import BookCard from "../components/BookCard";

import PageWrapper from "../components/PageWrapper";

function Books() {

  const [books, setBooks] = useState([]);

  useEffect(() => {

    const fetchBooks = async () => {

      const { data } = await axios.get("/books");

      setBooks(data);

    };

    fetchBooks();

  }, []);

  return (

    <PageWrapper bg="https://images.unsplash.com/photo-1507842217343-583bb7270b66">

      <Navbar />

      <div className="container mt-4">

        <h2 className="mb-4">📚 Books Collection</h2>

        <div className="row">

          {books.map((book) => (

            <div className="col-md-3 mb-4" key={book.\_id}>

              <BookCard book={book} />

            </div>

          ))}

        </div>

      </div>

    </PageWrapper>

  );

}

export default Books;

import { useState, useEffect } from "react";

import Navbar from "../components/Navbar";

import axios from "../api/axios";

import { useNavigate } from "react-router-dom";

import PageWrapper from "../components/PageWrapper";

function Cart() {

  const [cart, setCart] = useState([]);

  const navigate = useNavigate();

  useEffect(() => {

    setCart(JSON.parse(localStorage.getItem("cart")) || []);

  }, []);

  const totalPrice = cart.reduce(

    (acc, item) => acc + item.price \* item.quantity,

    0

  );

  const placeOrder = async () => {

    const token = localStorage.getItem("token");

    if (!token) {

      alert("Please Login First");

      navigate("/login");

      return;

    }

    try {

      await axios.post(

        "/orders",

        {

          books: cart.map(item => ({

            book: item.\_id,

            quantity: item.quantity

          })),

          totalPrice

        },

        {

          headers: { Authorization: token }

        }

      );

      alert("Order Placed Successfully 🎉");

      localStorage.removeItem("cart");

      setCart([]);

      navigate("/orders");

    } catch (error) {

      alert("Order Failed ❌");

    }

  };

  return (

    <PageWrapper bg="https://images.unsplash.com/photo-1519681393784-d120267933ba">

      <Navbar />

      <div className="container mt-5 text-dark bg-light p-4 rounded">

        <h2>🛒 Your Cart</h2>

        {cart.length === 0 && <p>Your cart is empty.</p>}

        {cart.map(item => (

          <div

            key={item.\_id}

            className="d-flex align-items-center mb-3"

          >

            <img

              src={item.image}

              alt={item.title}

              width="80"

              className="me-3"

            />

            <div>

              <h6>{item.title}</h6>

              <p>

                ₹{item.price} × {item.quantity}

              </p>

            </div>

          </div>

        ))}

        {cart.length > 0 && (

          <>

            <h4>Total: ₹{totalPrice}</h4>

            <button

              className="btn btn-success mt-3"

              onClick={placeOrder}

            >

              Place Order

            </button>

          </>

        )}

      </div>

    </PageWrapper>

  );

}

export default Cart;

import { useEffect, useState } from "react";

import axios from "../api/axios";

import Navbar from "../components/Navbar";

import PageWrapper from "../components/PageWrapper";

function Orders() {

  const [orders, setOrders] = useState([]);

  useEffect(() => {

    const fetchOrders = async () => {

      const token = localStorage.getItem("token");

      if (!token) return;

      try {

        const { data } = await axios.get("/orders", {

          headers: { Authorization: token }

        });

        setOrders(data);

      } catch (error) {

        console.log(error);

      }

    };

    fetchOrders();

  }, []);

  return (

    <PageWrapper bg="https://images.unsplash.com/photo-1524578271613-d550eacf6090">

      <Navbar />

      <div className="container mt-5 text-dark bg-light p-4 rounded">

        <h2>📦 My Orders</h2>

        {orders.length === 0 && <p>No Orders Yet.</p>}

        {orders.map(order => (

          <div

            key={order.\_id}

            className="card p-3 mb-3"

          >

            <h6>Total: ₹{order.totalPrice}</h6>

            <p>

              Date: {new Date(order.createdAt).toLocaleString()}

            </p>

          </div>

        ))}

      </div>

    </PageWrapper>

  );

}

export default Orders;

**GitHub :** [**https://github.com/Kushulatha-k/BookNest.git**](https://github.com/Kushulatha-k/BookNest.git)

**Project Demo Link :**

<https://drive.google.com/file/d/1K6YkmwtcKwJTDKb7e7lD4rHTD0ums8km/view?usp=drivesdk>