**Full Stack Development with MERN**

**Project Documentation**

**1. Introduction**

* **Project Title: BOOKNEST**
* **Team Members:**
  + Harshith Kumar (Frontend Developer)
  + Sravan Sai(Frontend Developer)
  + Abijit Prasanth (Backend Developer)
  + Tulasi Vardhan Reddy (Backend Developer)

**2. Project Overview**

* **Purpose:** To create a comprehensive online bookstore allowing users to browse, purchase, and manage books, while providing dedicated portals for sellers and administrators.
* **Features:**
  + User registration and login
  + Book browsing
  + wishlist management
  + Order management
  + Seller and admin portals for inventory and order management
  + Admin functionalities for user and seller management

**3. Architecture**

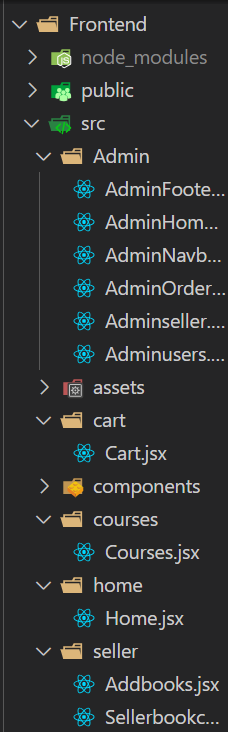
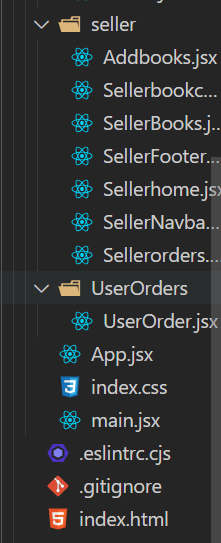
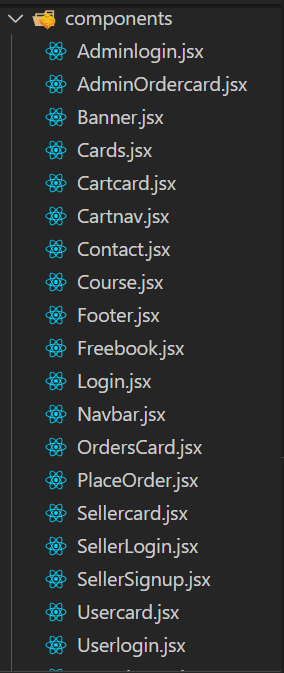
* Frontend: Built with React.js and styled using Tailwind CSS.
* Backend: Developed using Node.js and Express.js.
* Database: MongoDB for storing user data, book details, and order information.

**4. Setup Instructions**

* **Prerequisites:**
  + Node.js
  + Reactjs
  + Expressjs
  + MongoDB
* **Installation:**
  + **Clone the repository.**
  + **Install dependencies for the backend:**
    - cd backend
    - npm i
    - node server.js
  + **Install dependencies for the frontend:**
    - cd frontend
    - npm i
    - npm run dev

**5. Folder Structure**

**Client (frontend):**

**** **** 

* **Server (backend):**
* ****

**6. Running the Application**

* **Frontend:**

cd frontend

npm i

npm run dev

* **Backend:**

cd backend

npm i

node server.js

**7. API Documentation**

* **Endpoints:**
* **POST /usersignup** - Register a new user.
* **POST /userlogin** - Authenticates a user and returns a token.

**User Management**

* **GET /user/-** Retrieves user information by ID.
* **PUT /user/**- Updates user information by ID.

**Seller API Endpoints**

* **POST/sellersignup**- Registers a new Seller
* **POST /sellerlogin** – Authenticates a seller and returns a token.

**Seller Management**

* **GET /seller/** - Retrieves seller information by ID
* **POST /addbooks** – updates seller information

**Admin API Endpoints**

* **POST/ausers**- Registers a new admin
* **POST /adminlogin**– Authenticates a admin and returns a token.

**Admin Management**

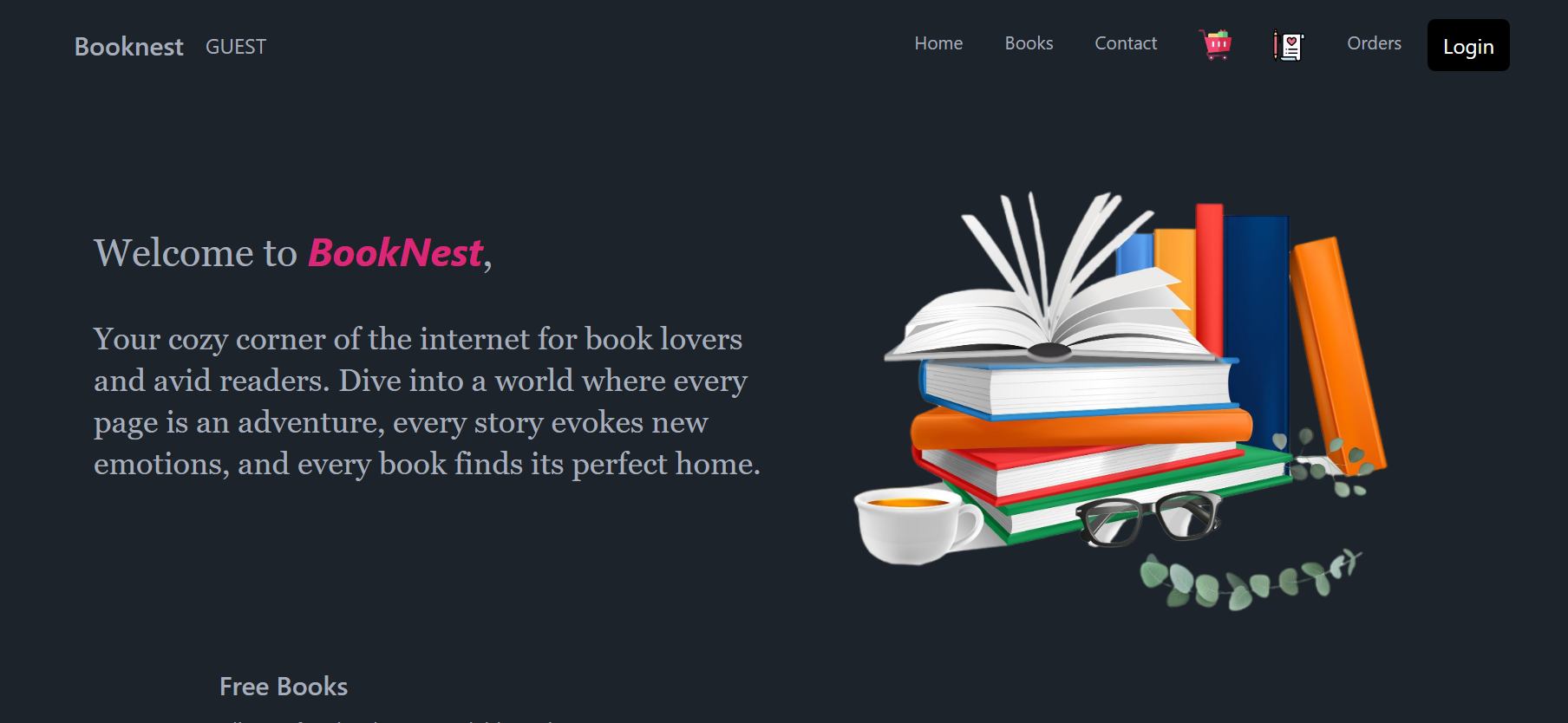
* **GET /ausers** - Retrieves seller information by ID

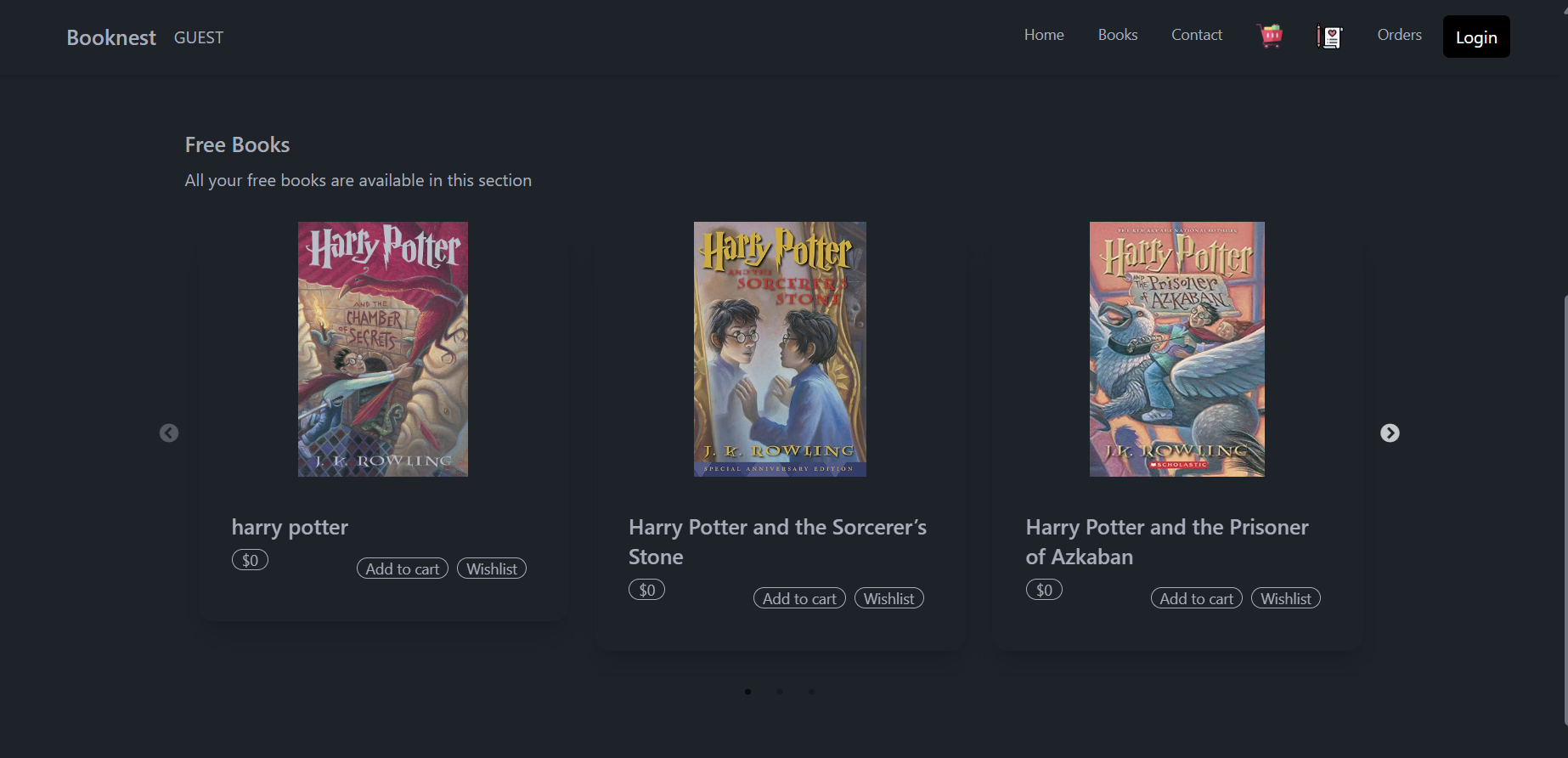
**8. Authentication**

* Process:
  1. User logs in or registers

**9. User Interface**

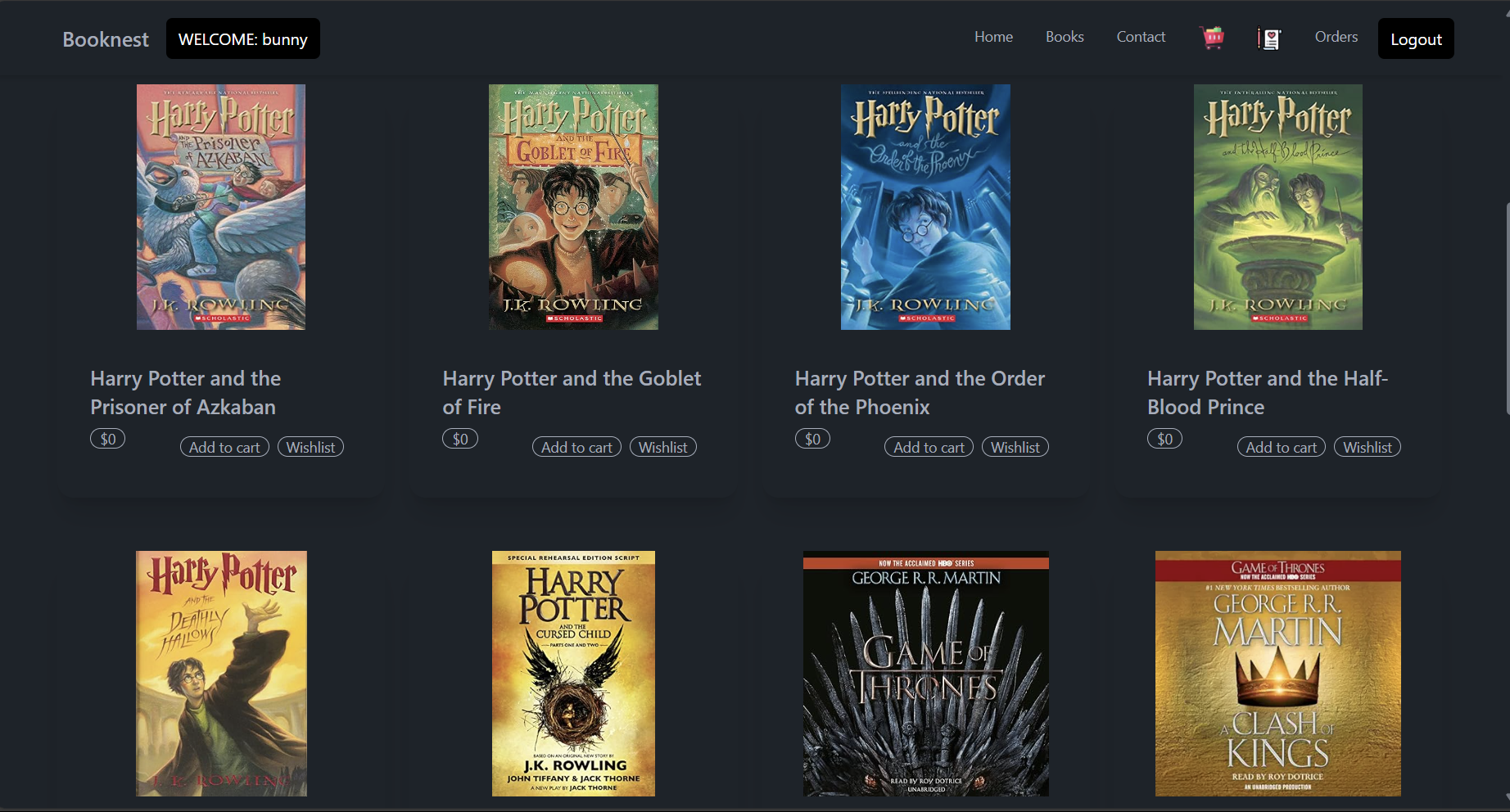
* **Landing Page:**

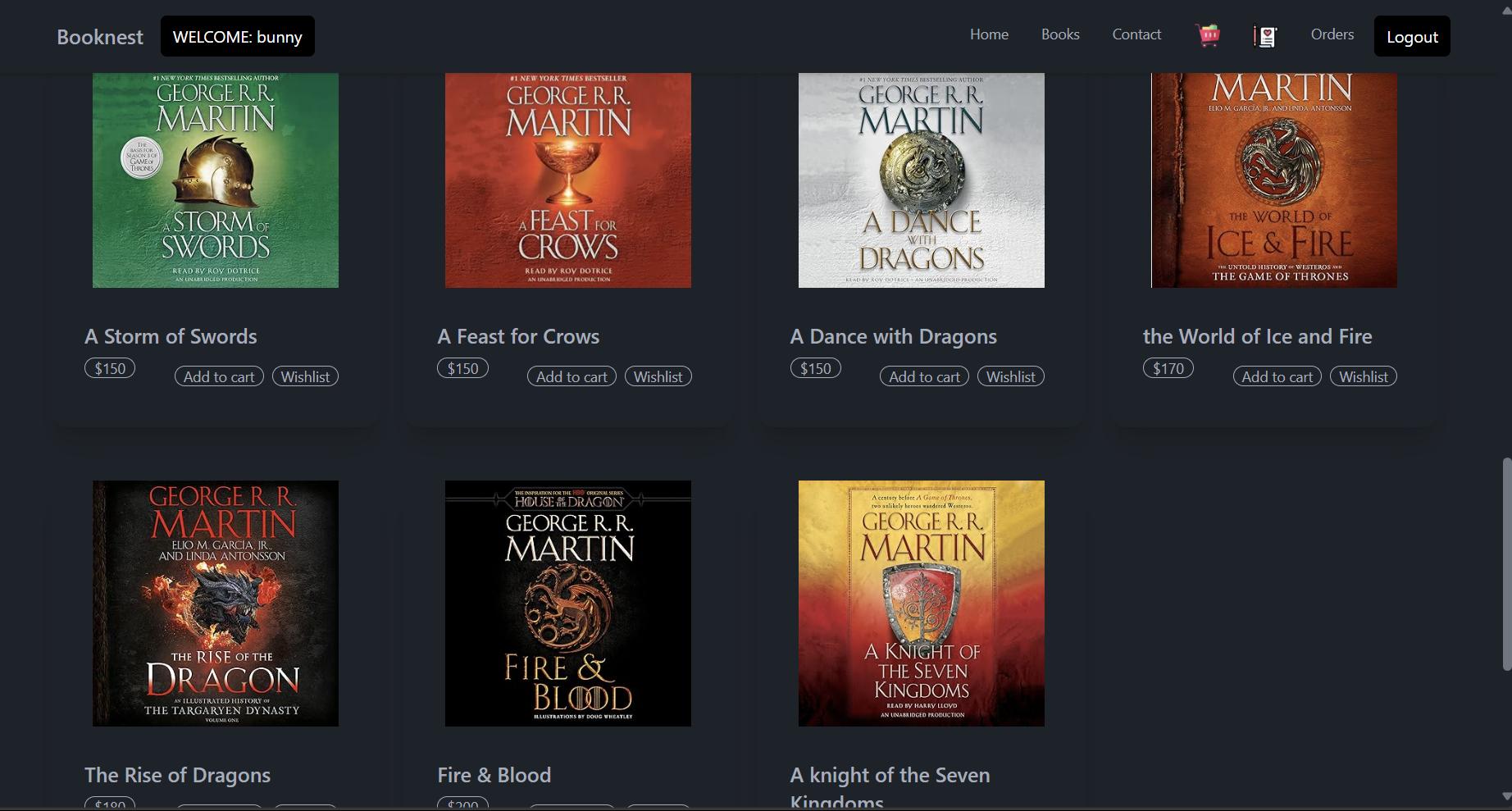
****

****

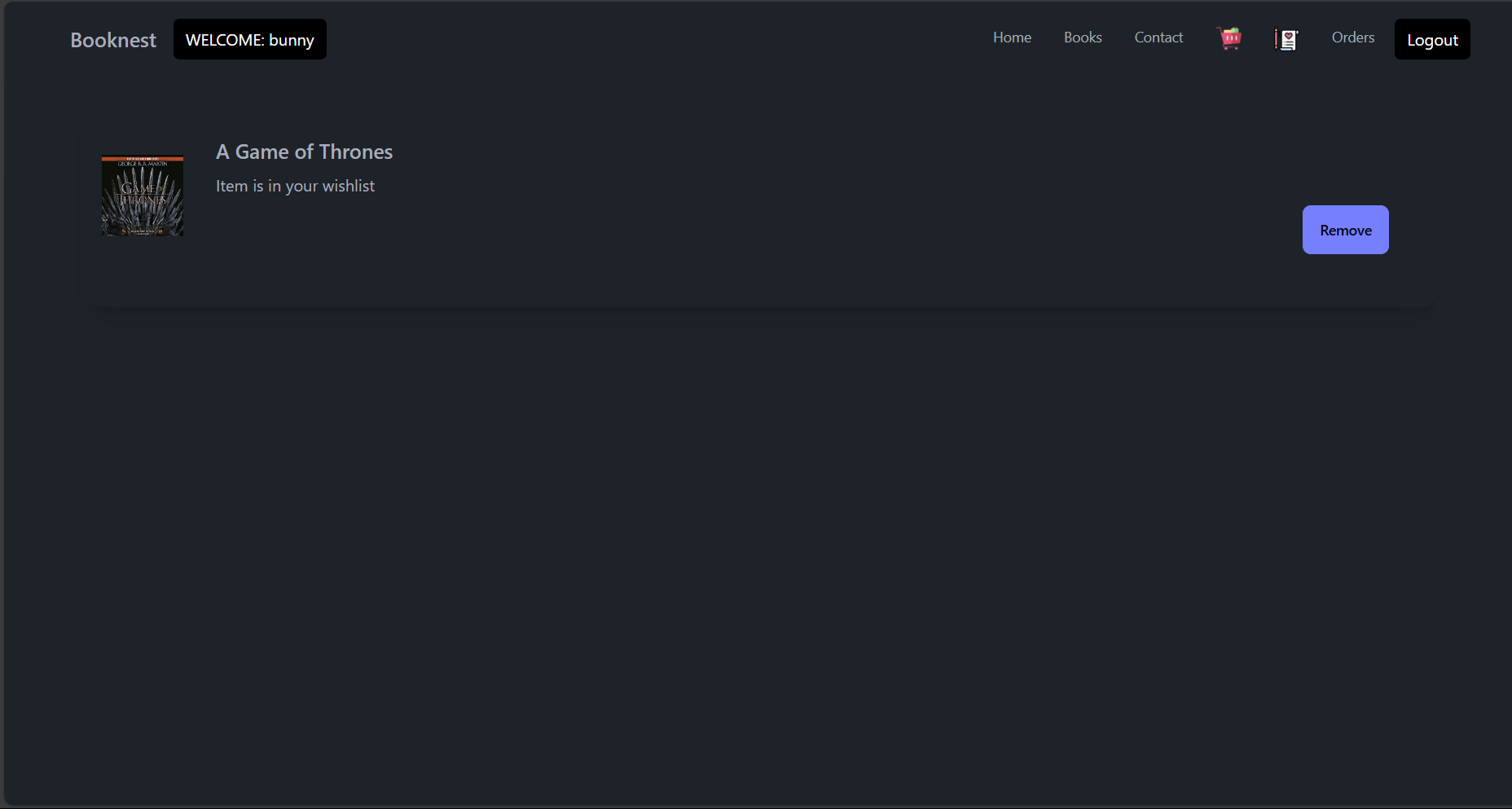
* **User Page:**

****

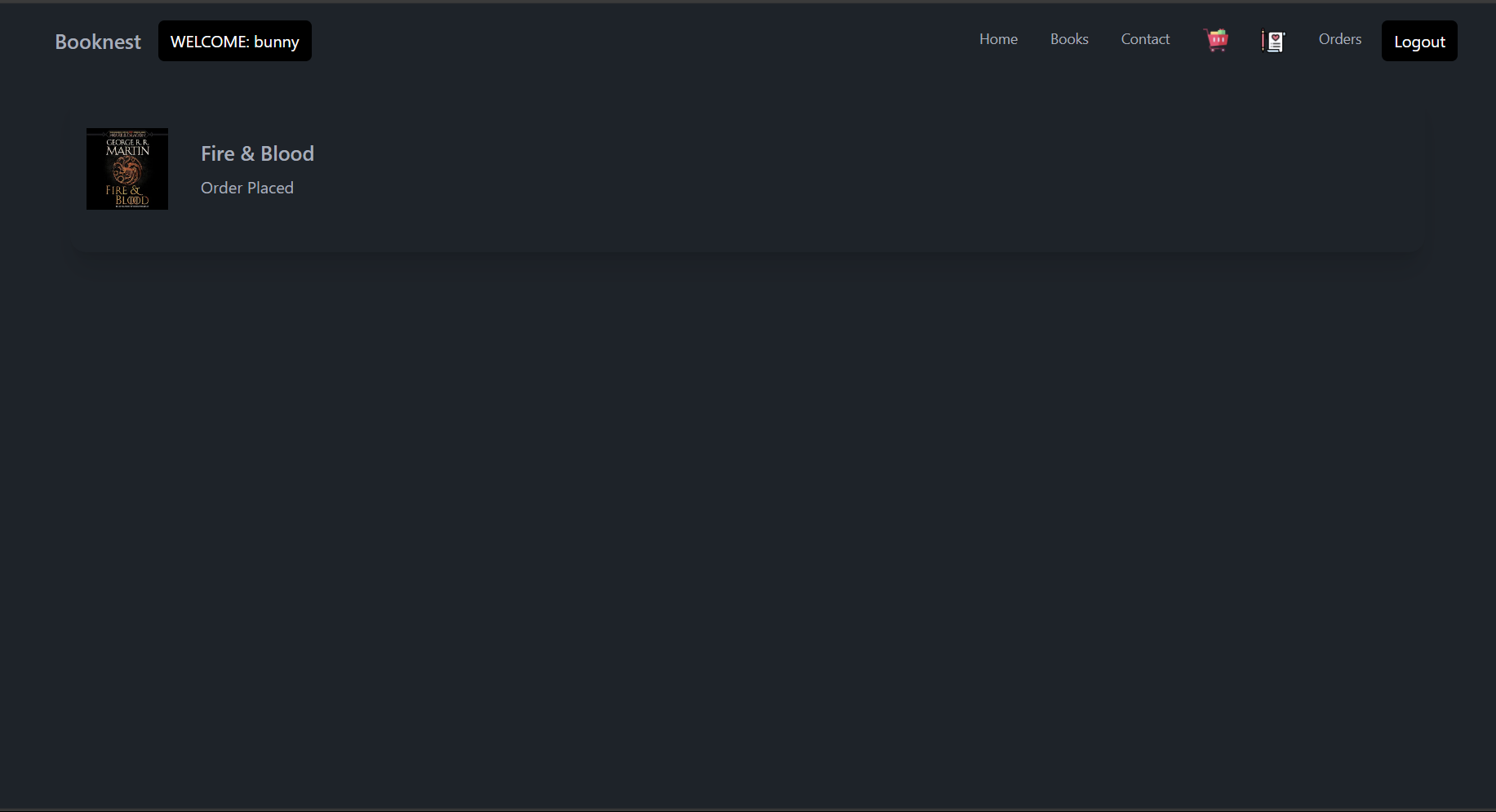
****

****

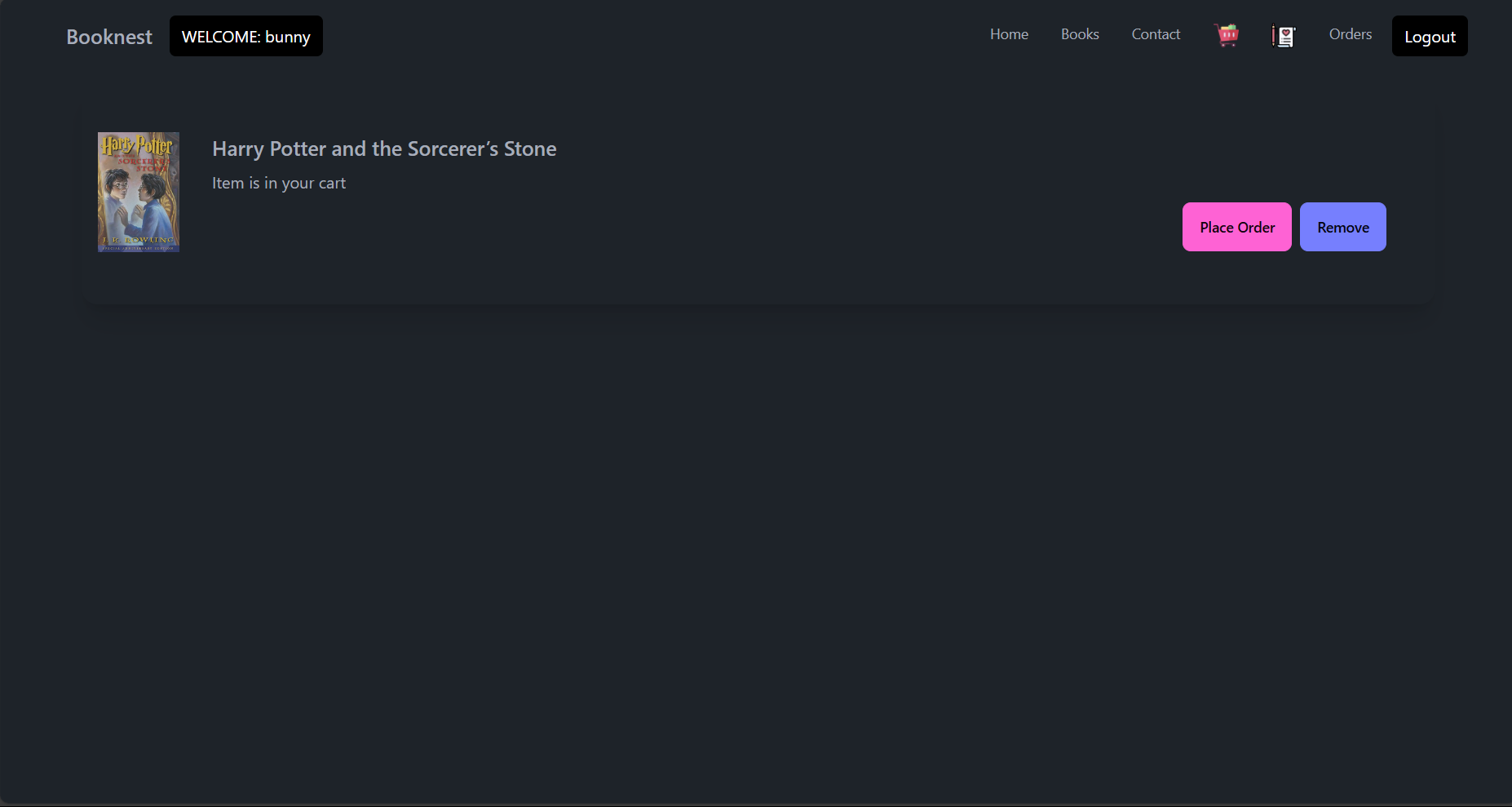
**Wishlist page:**



**User Order page:**



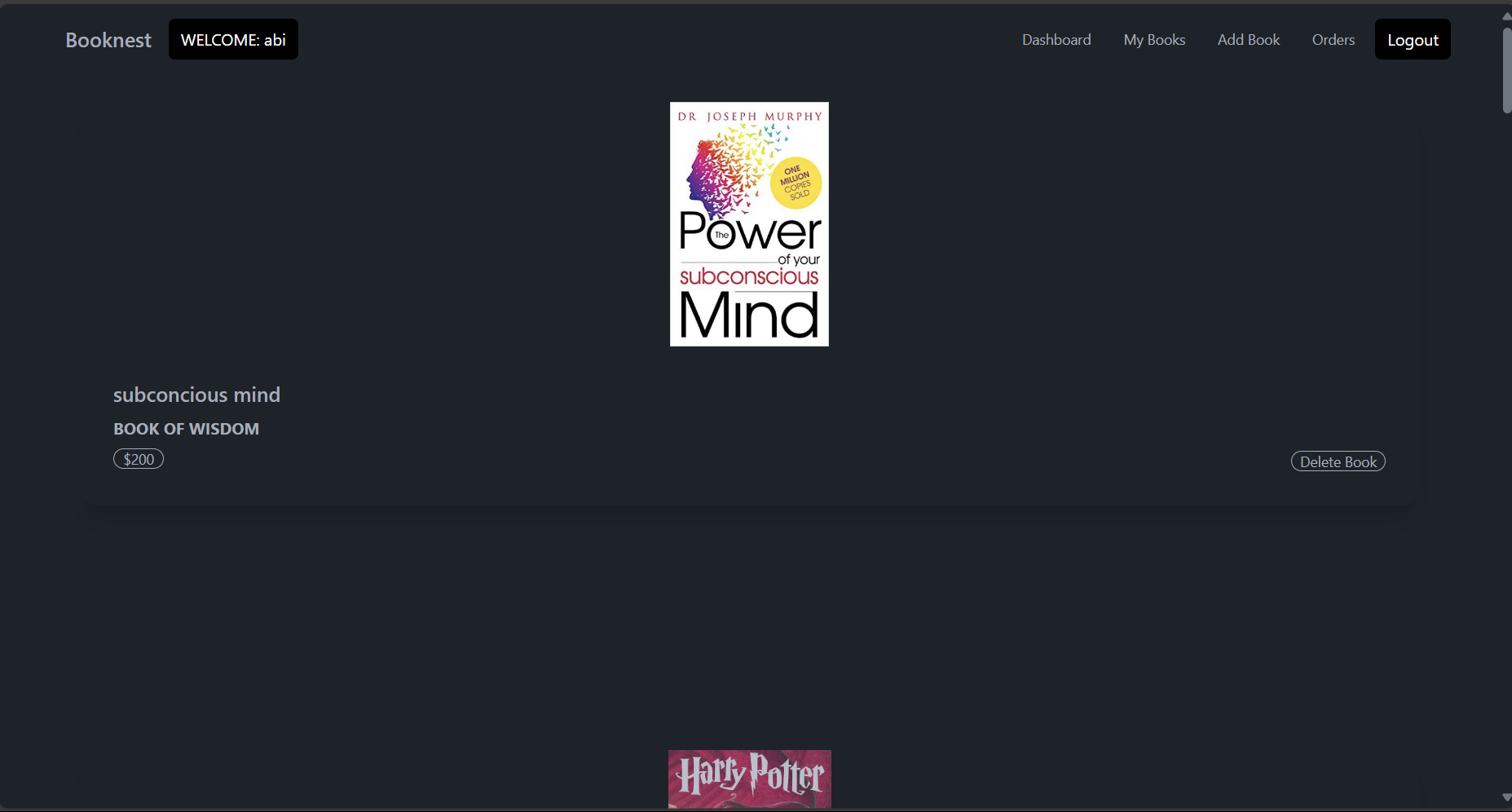
**User cart page:**

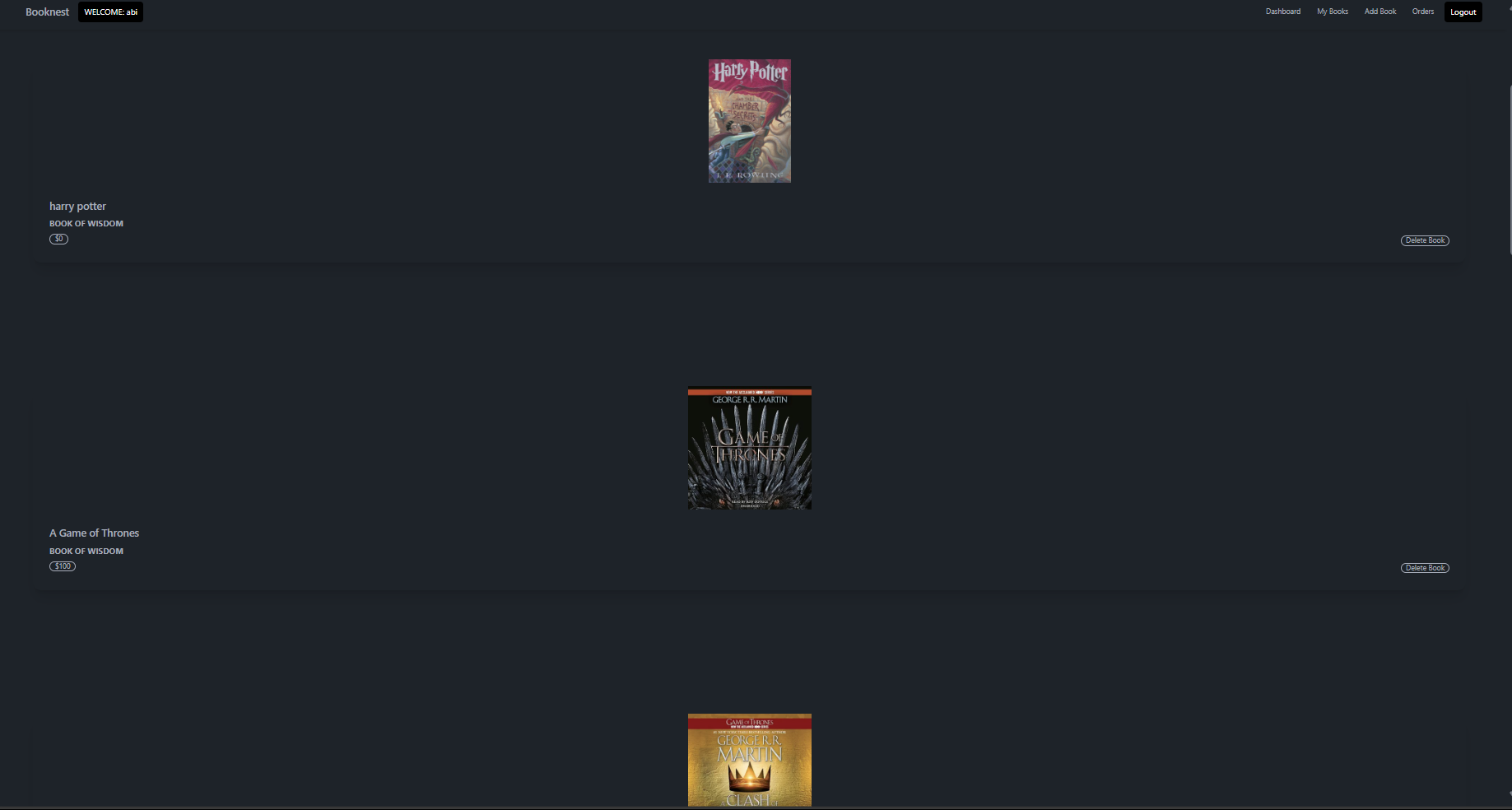


**Seller Dashboard page:**

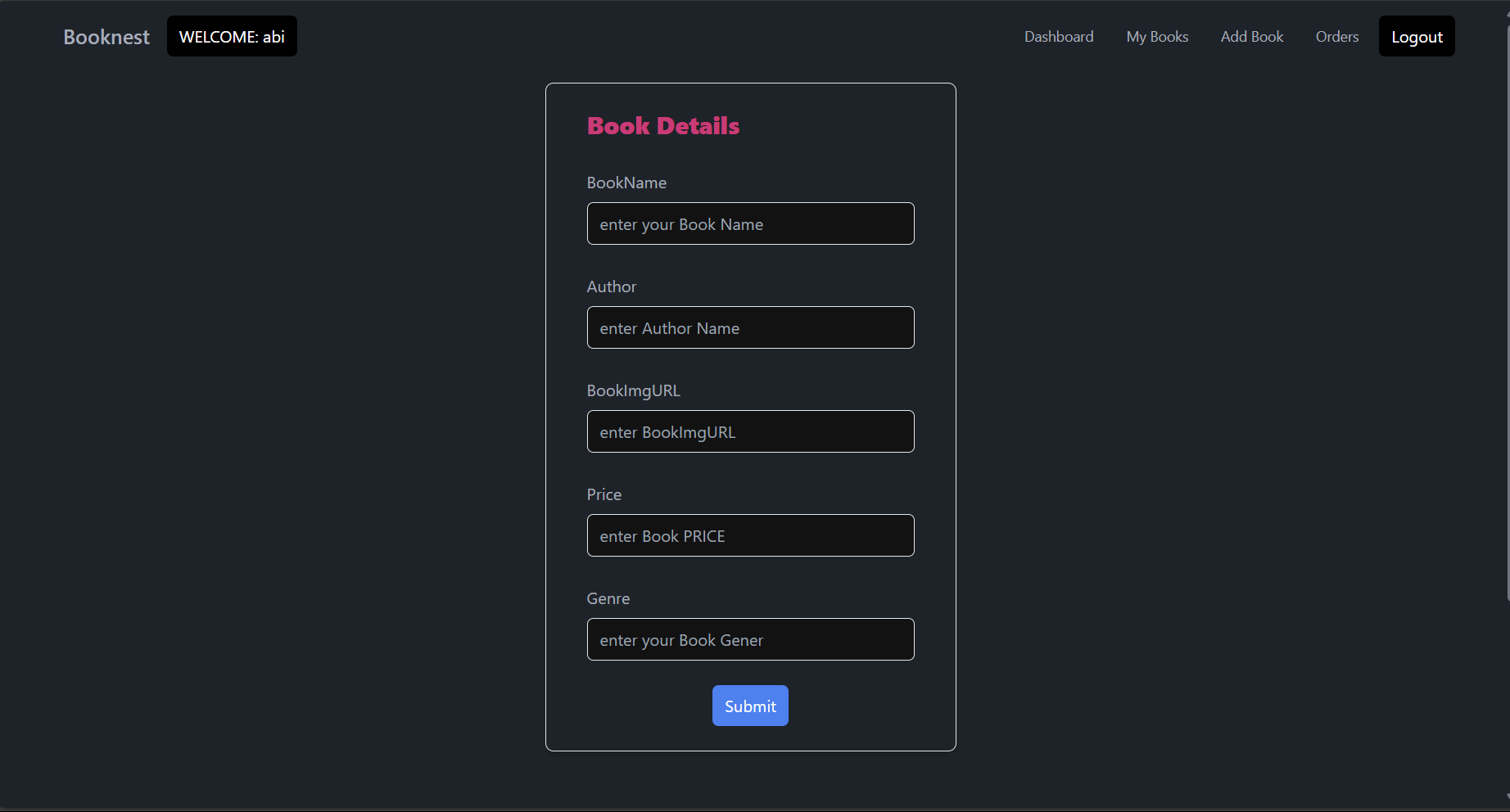


**Seller Books page:**

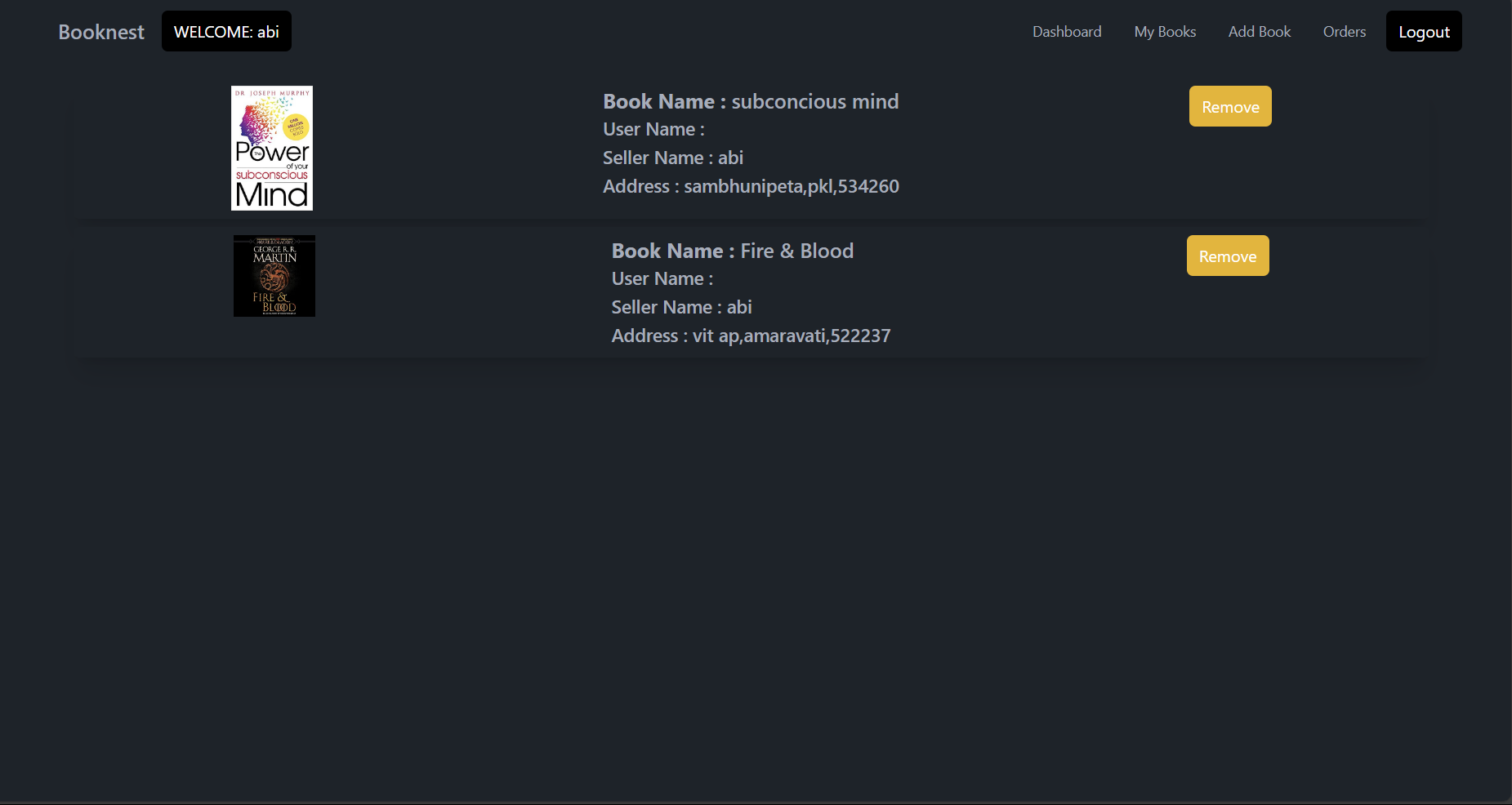




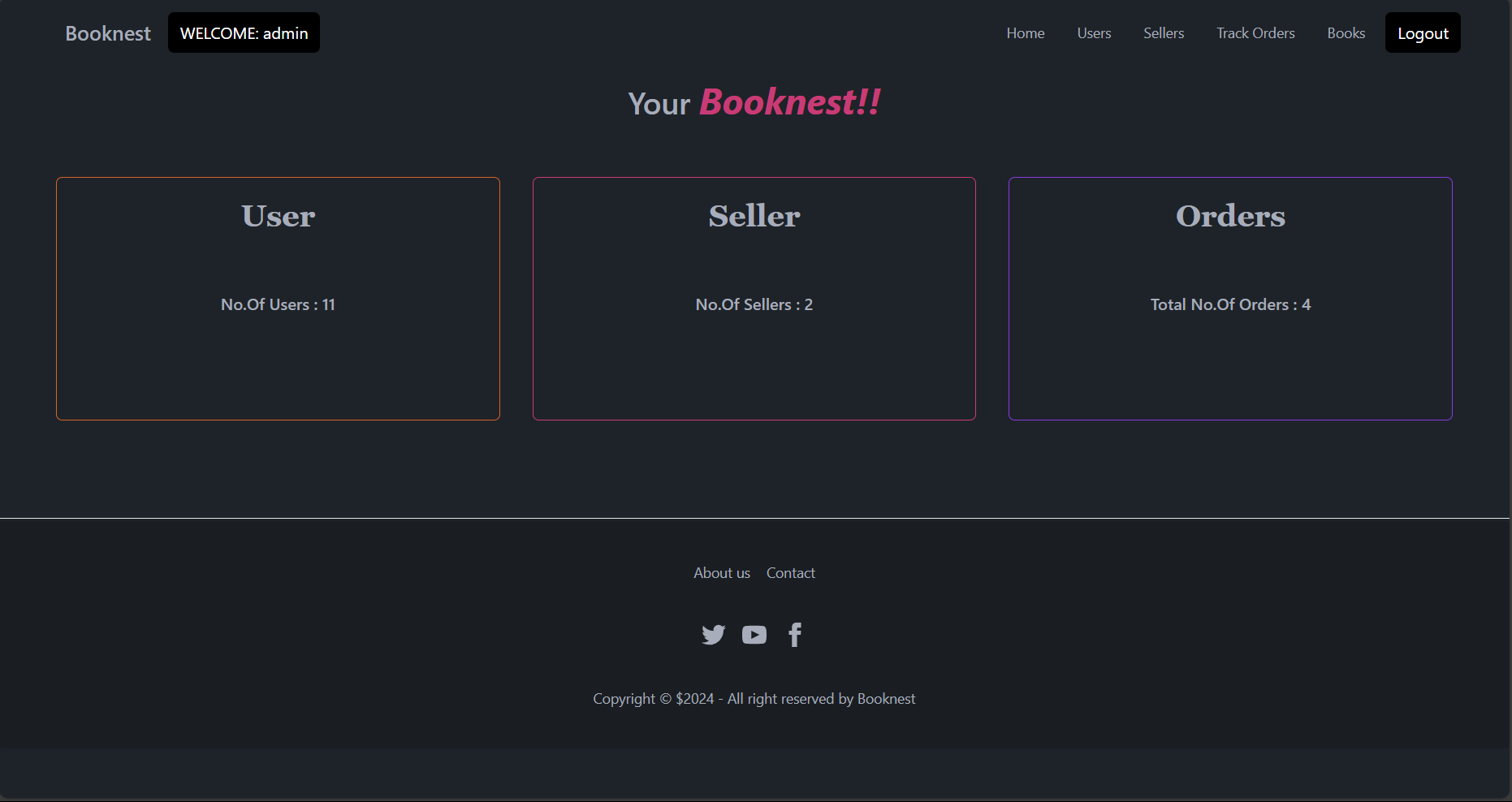
**Seller Add Books page:**

****

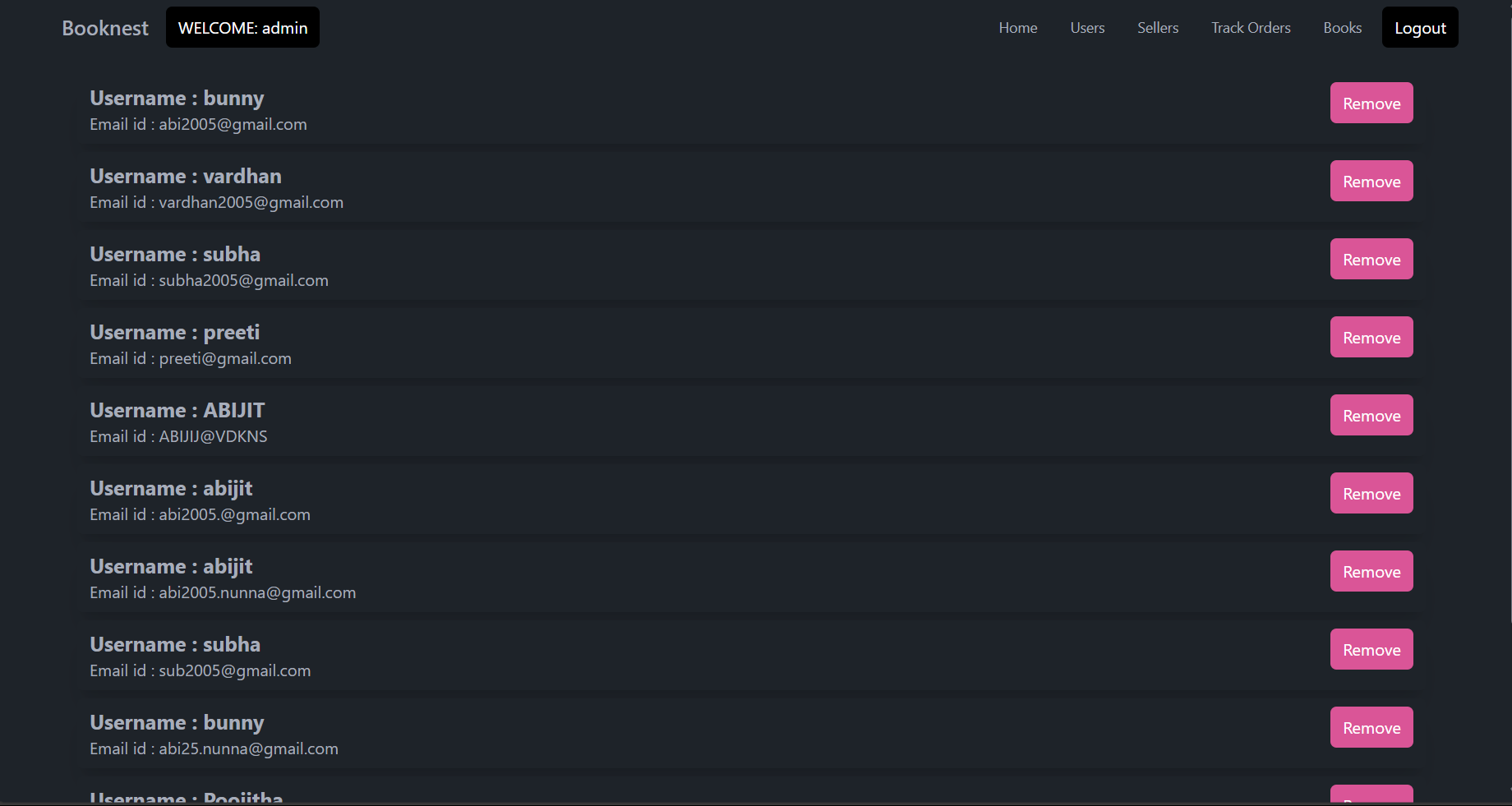
**Seller orders page:**

****

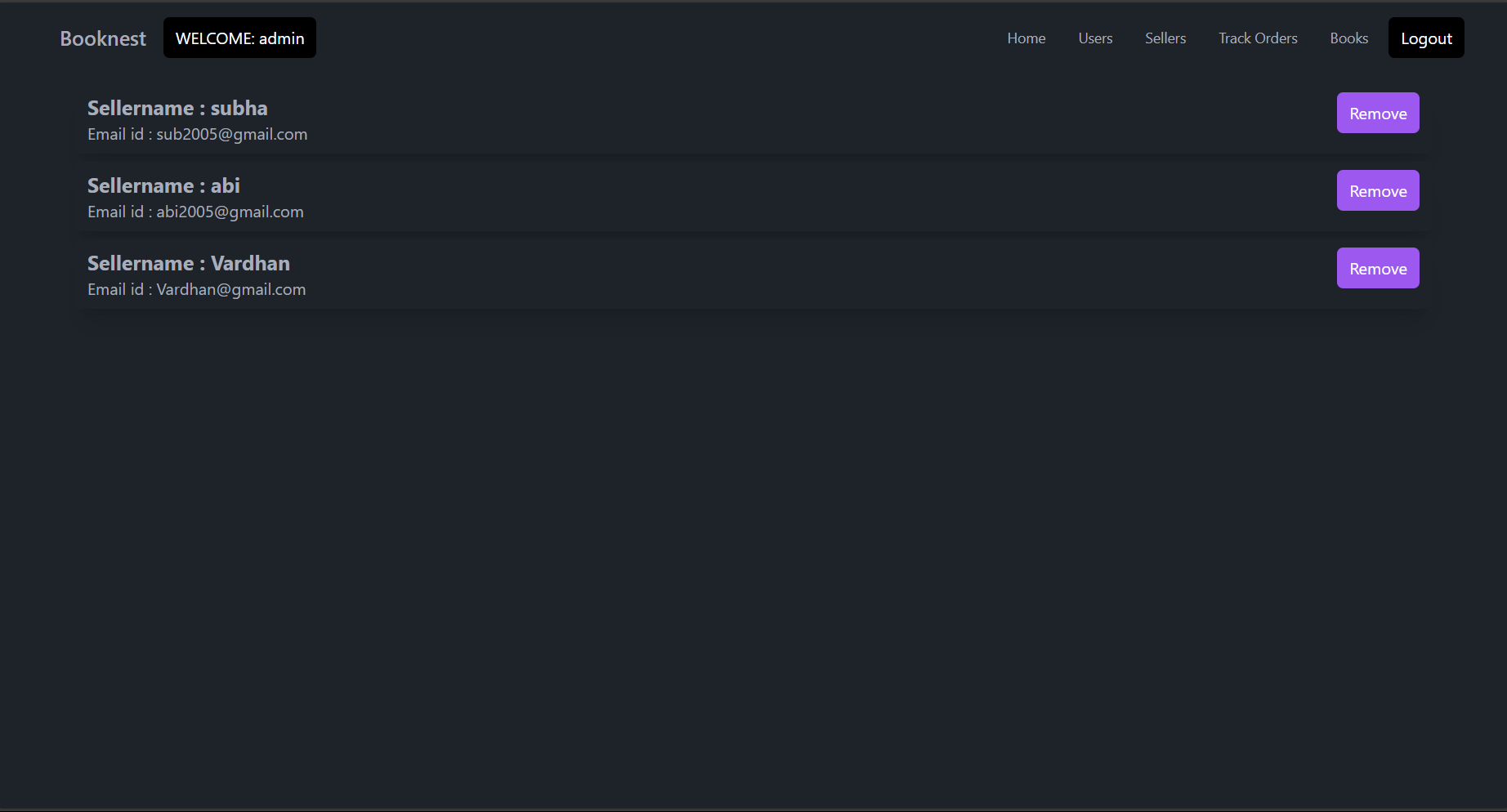
**Admin Home page:**

****

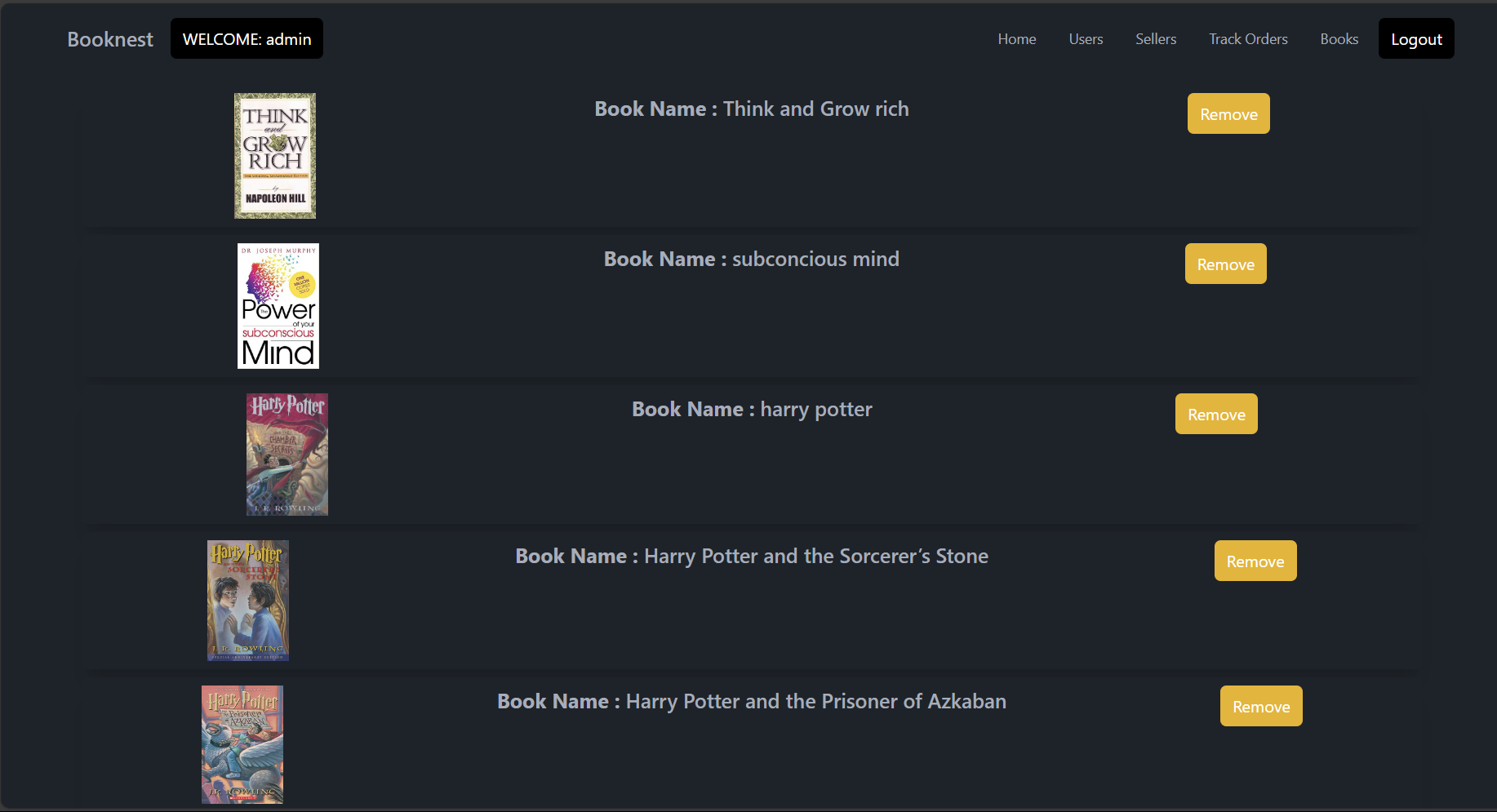
**Admin users page:**

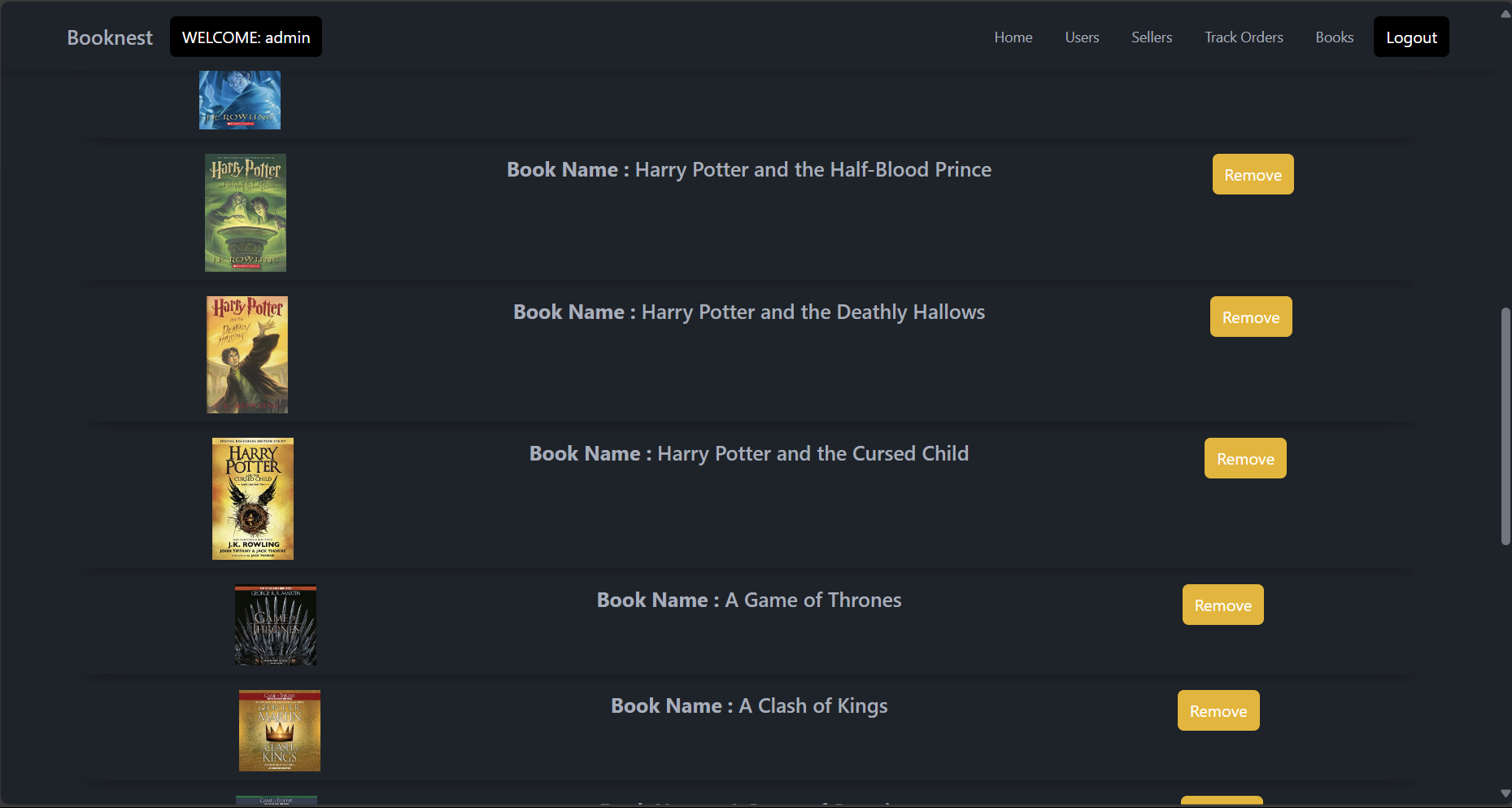
****

**Admin Seller page:**

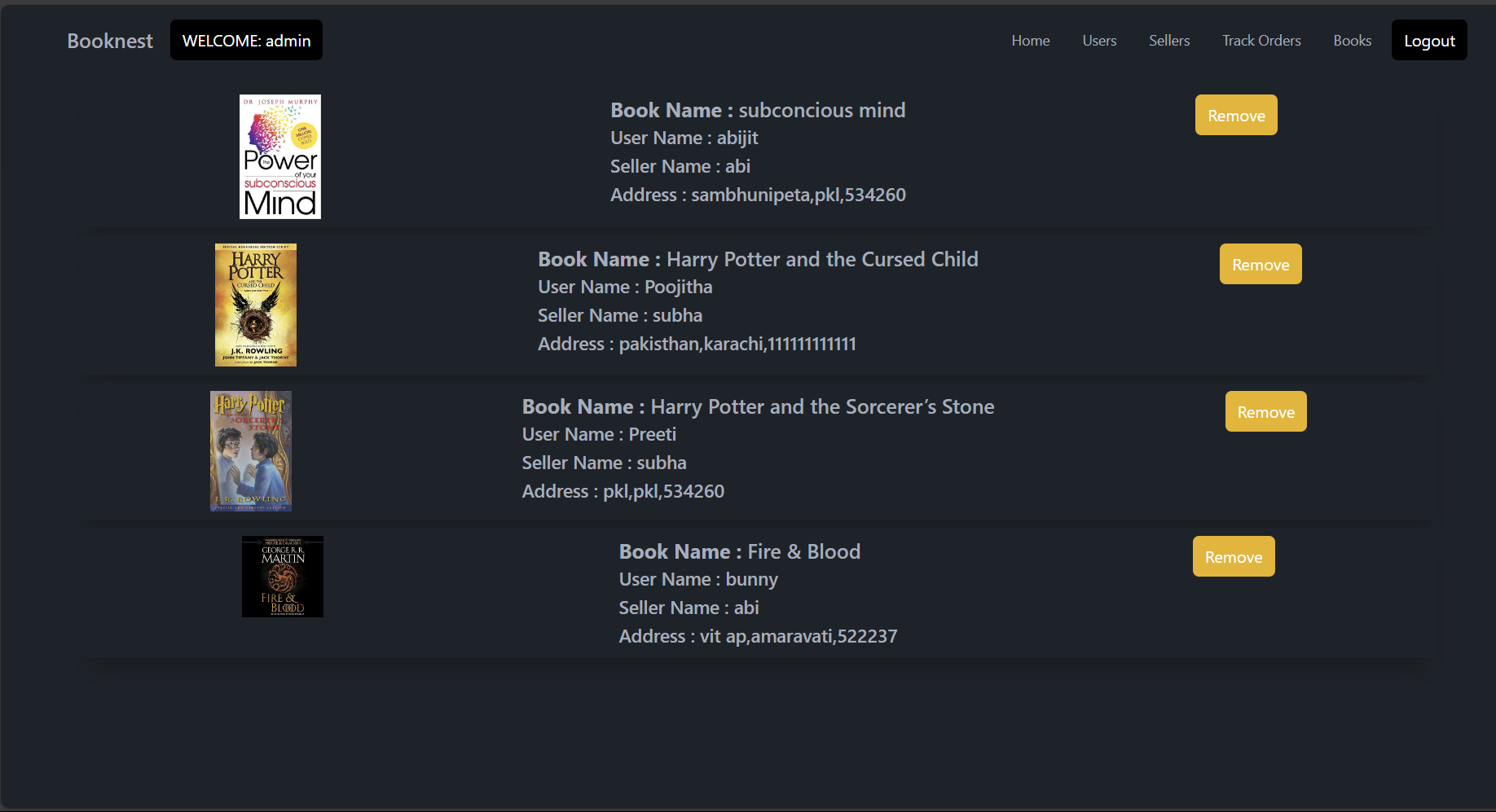
****

**Admin books page:**

****

****

**Admin Orders page:**

****

**10. Testing**

* Strategy: Manual testing of user flows and functionalities.
* Tools: thunder client for API testing.

**11. Screenshots or Demo**

* **Link to Demo: [Demo URL]**

**12. Known Issues**

* Placing an order shows an alert "order placed" as a demo instead of performing a real transaction.

**13. Future Enhancements**

* Implement real payment gateway integration.
* Improve UI/UX with more responsive design elements.
* Add more detailed book descriptions and reviews.