**✨ 1. Introduction**

**Fineprint** is a cozy, user-centric online bookstore platform where readers can discover, add, and purchase books — all wrapped in a soft, BookTok-inspired aesthetic.

It supports both **user** and **admin** roles:

* 📖 **Users** can browse books, manage their cart, checkout orders, and view their order history.
* 🛠️ **Admins** can manage the book catalog, user accounts, and monitor all orders via a beautiful admin dashboard.

Fineprint isn’t just about e-commerce — it’s a **reader’s safe haven**, combining beautiful UI with powerful backend management.

**🧰 2. Tech Stack Used**

**🖥️ Frontend (React.js)**

* **React Router DOM** – for seamless page routing
* **Tailwind CSS** – for a sleek and responsive design
* **Axios** – for making API requests
* **LocalStorage** – for persisting cart and session data client-side
* **Framer Motion** & custom animations – for delightful interactions

**⚙️ Backend (Express.js + Node.js)**

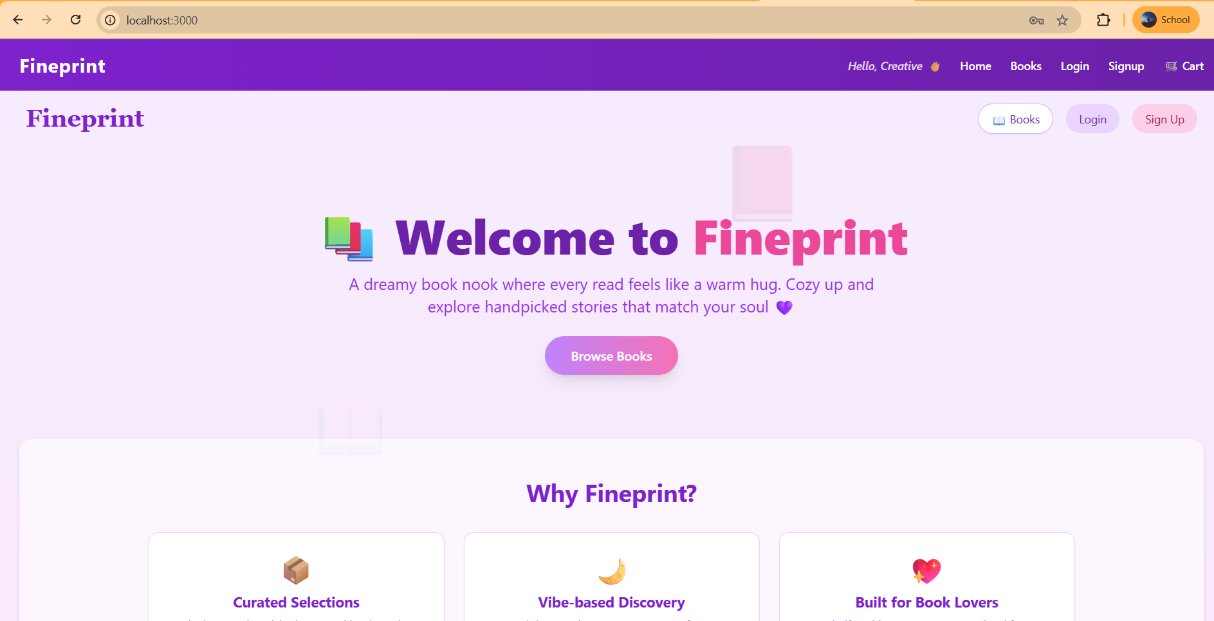
* **Express** – for building RESTful APIs
* **Mongoose** – for modeling and interacting with MongoDB
* **CORS & Body-Parser** – for handling requests properly
* **Custom Routes** – for modularized API endpoints (/books, /users, /checkout, etc.)

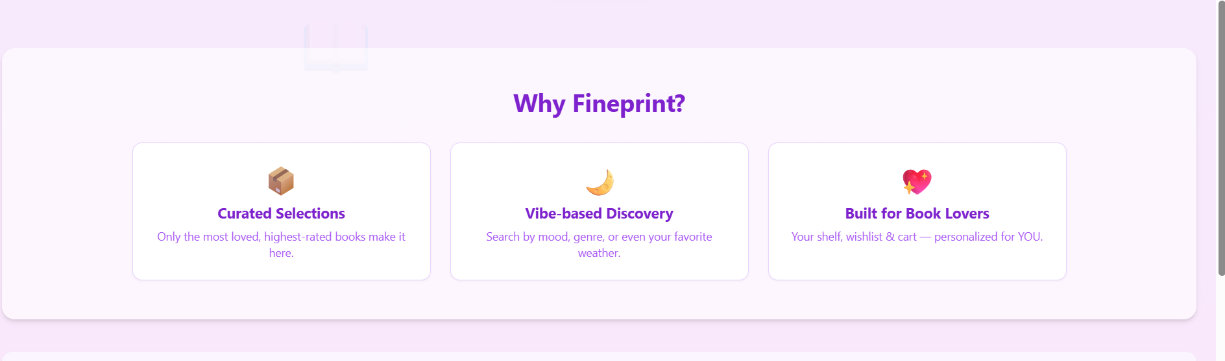
**🗃️ Database (MongoDB)**

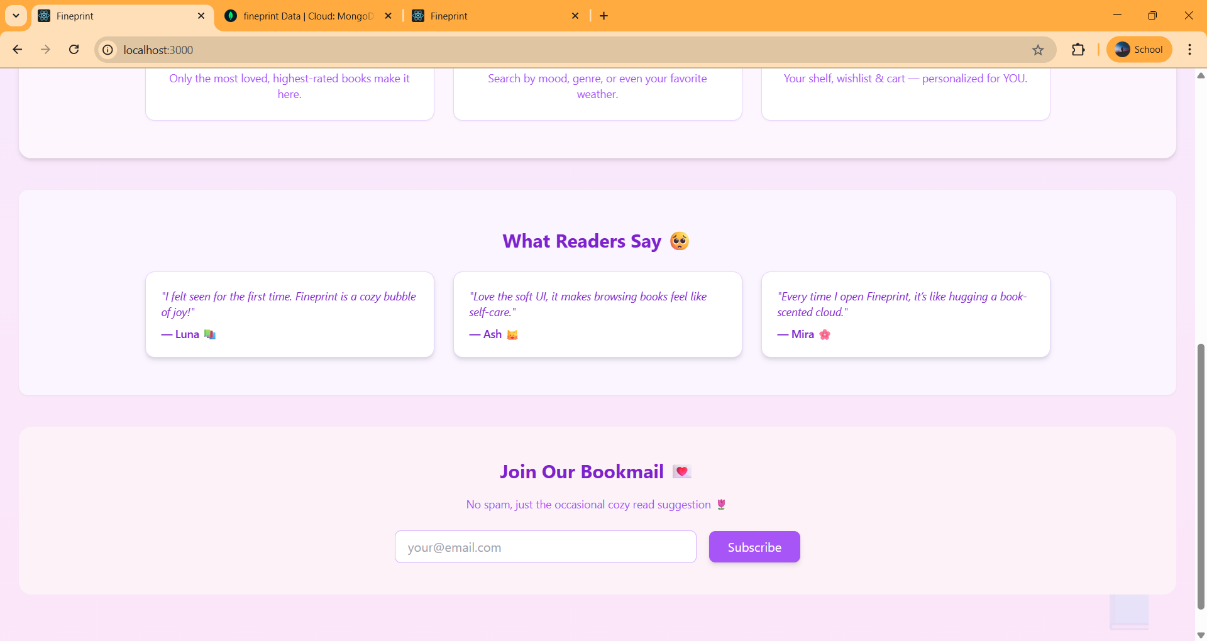
* **Cloud-hosted MongoDB Atlas** or **Local MongoDB** – for storing users, books, and orders
* **Mongoose Models**:
  + User
  + Book
  + Order (checkout)

**WEBSITE OVERVIEW**

**1)HOME PAGE (** aka landingpage**)**







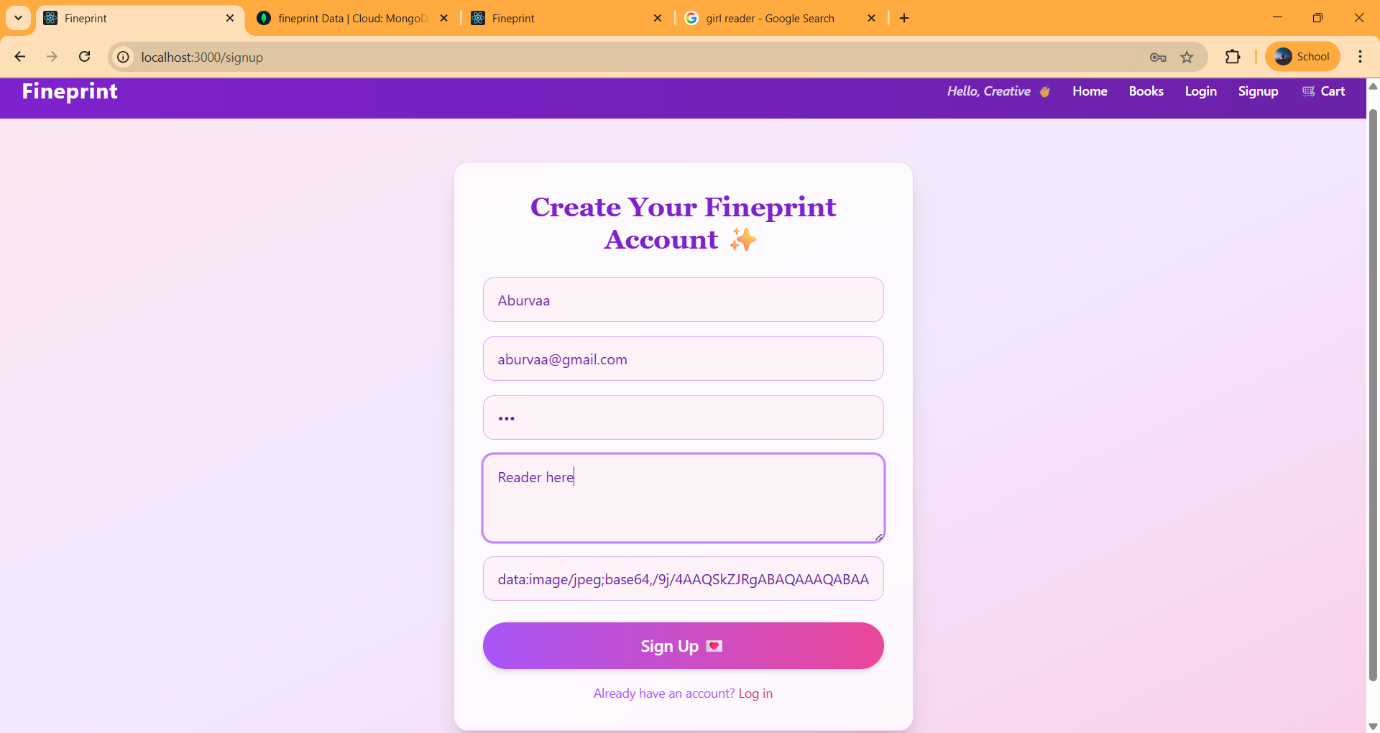
**a) Header**

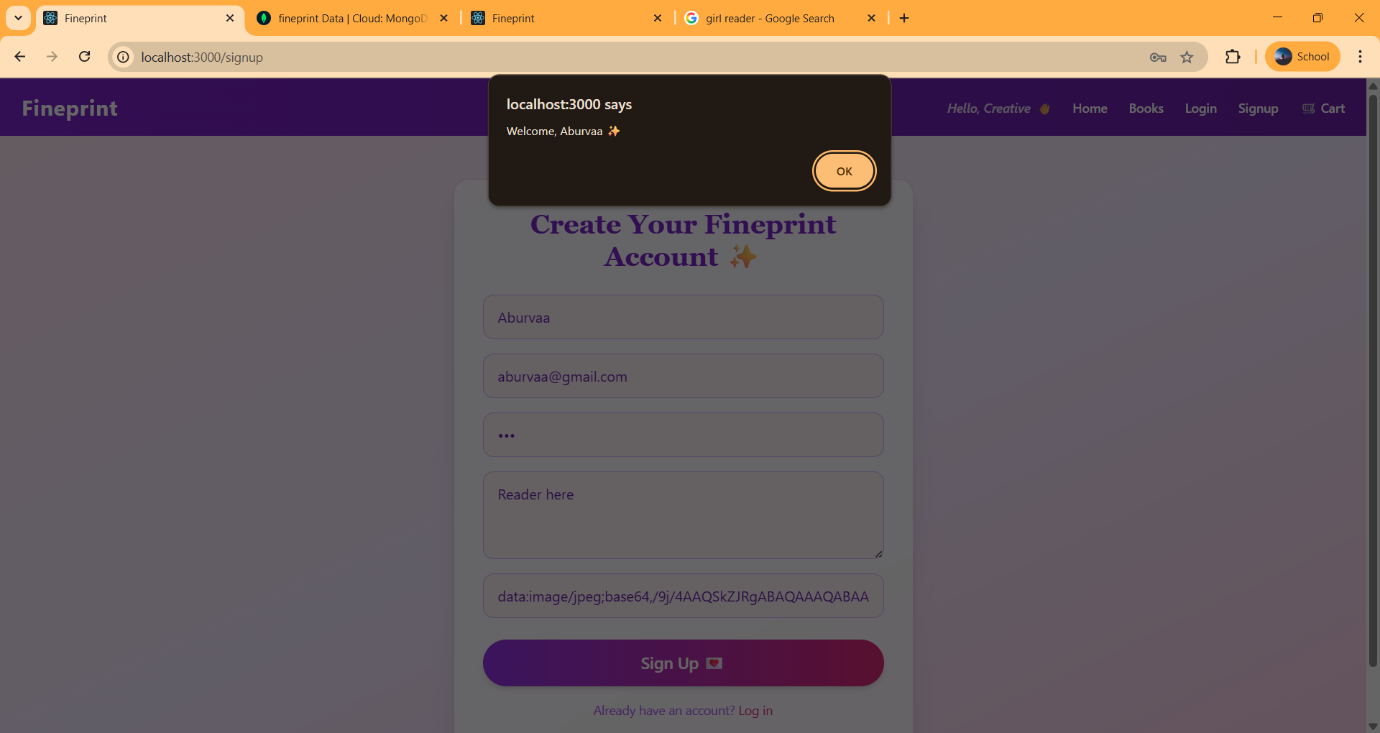
If not logged in the header will be 

**2)SIGNUP**

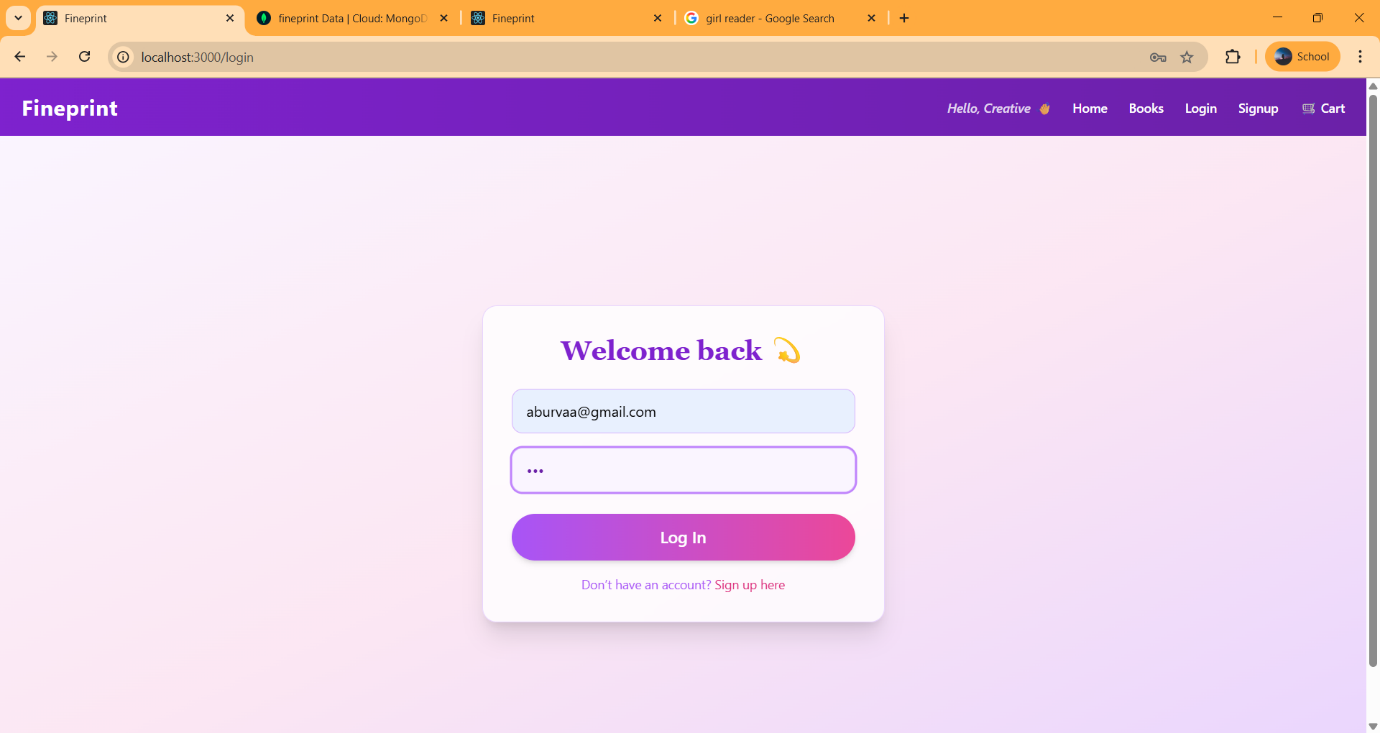


If avatar is not provided automatically inserted in the database randomly. If not the given avatar will be visible

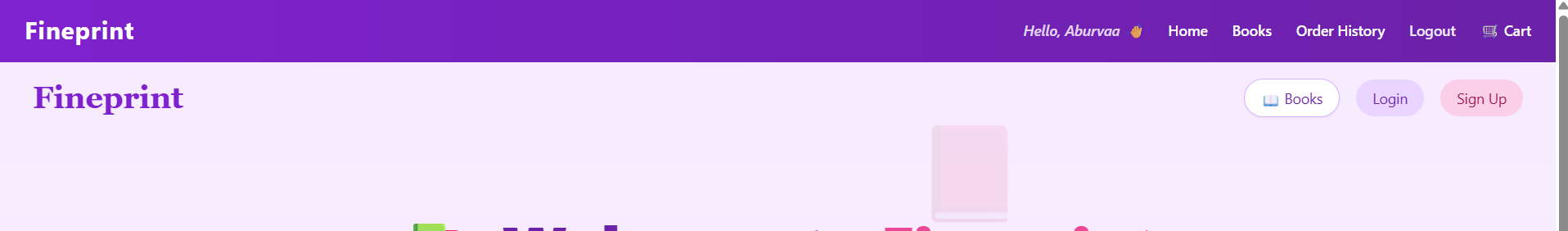


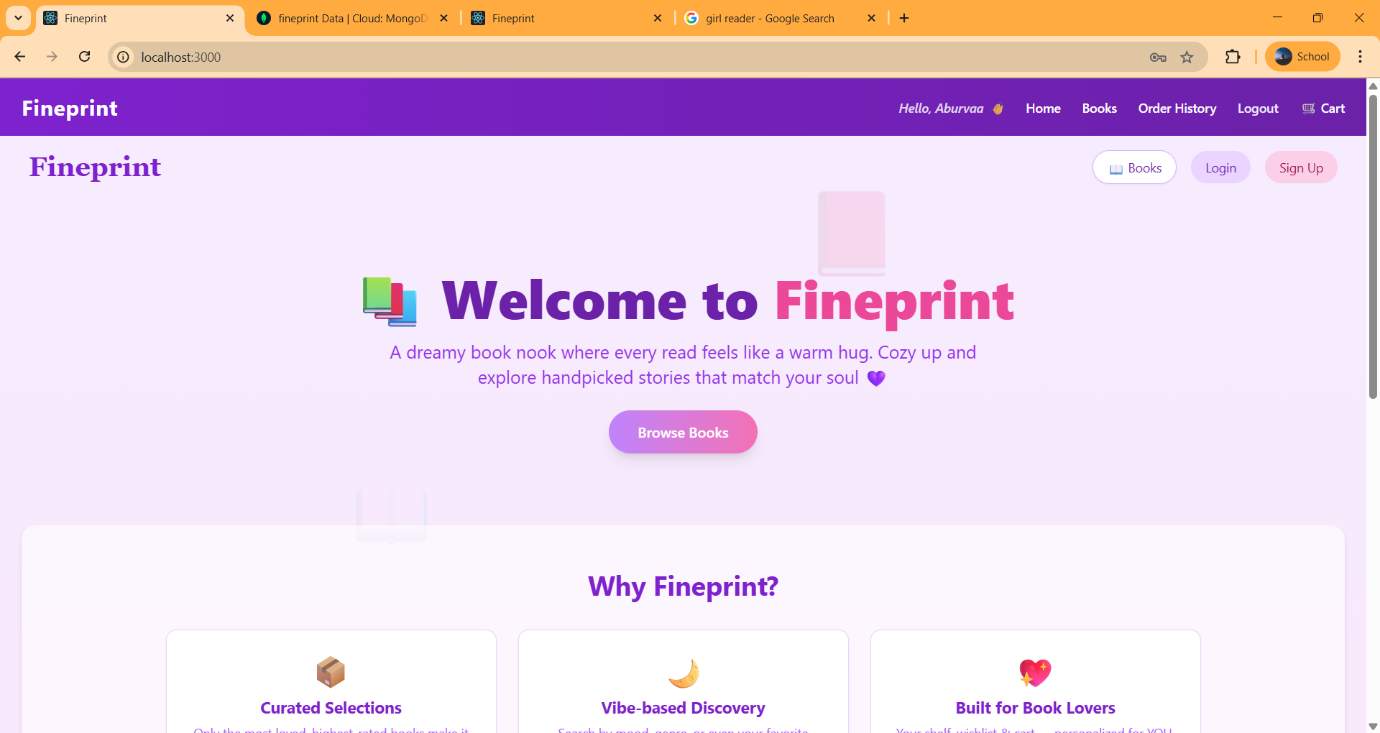


**3)LOGIN**



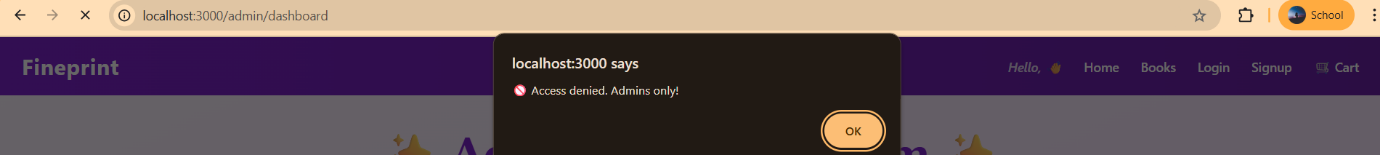
Header will be changed to

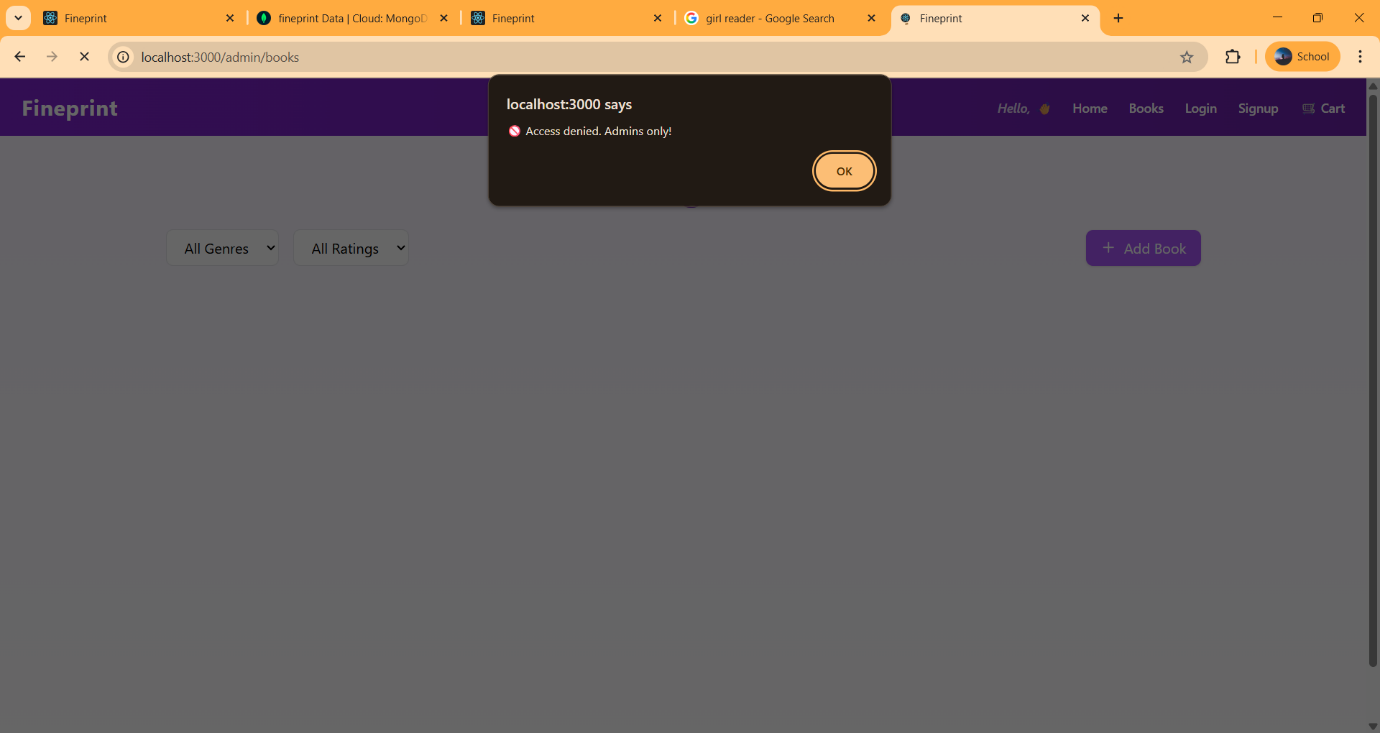
 and landed to the homepage

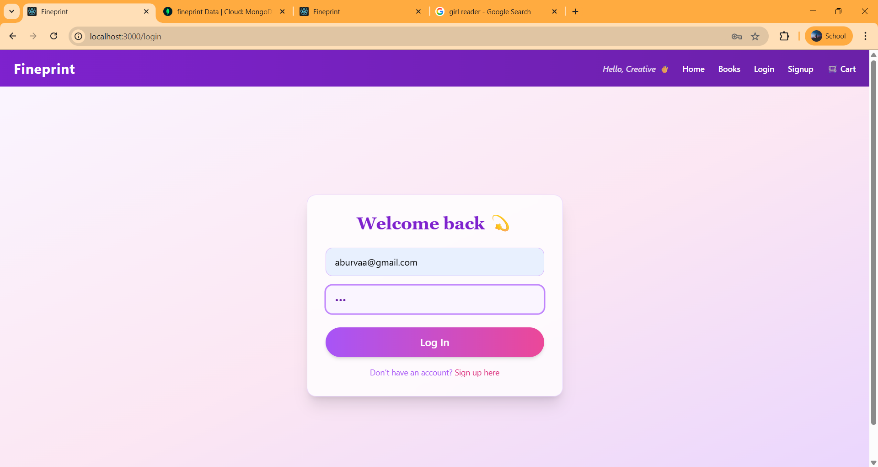


**4) Admin Login**

One cannot go through admin pages via routes



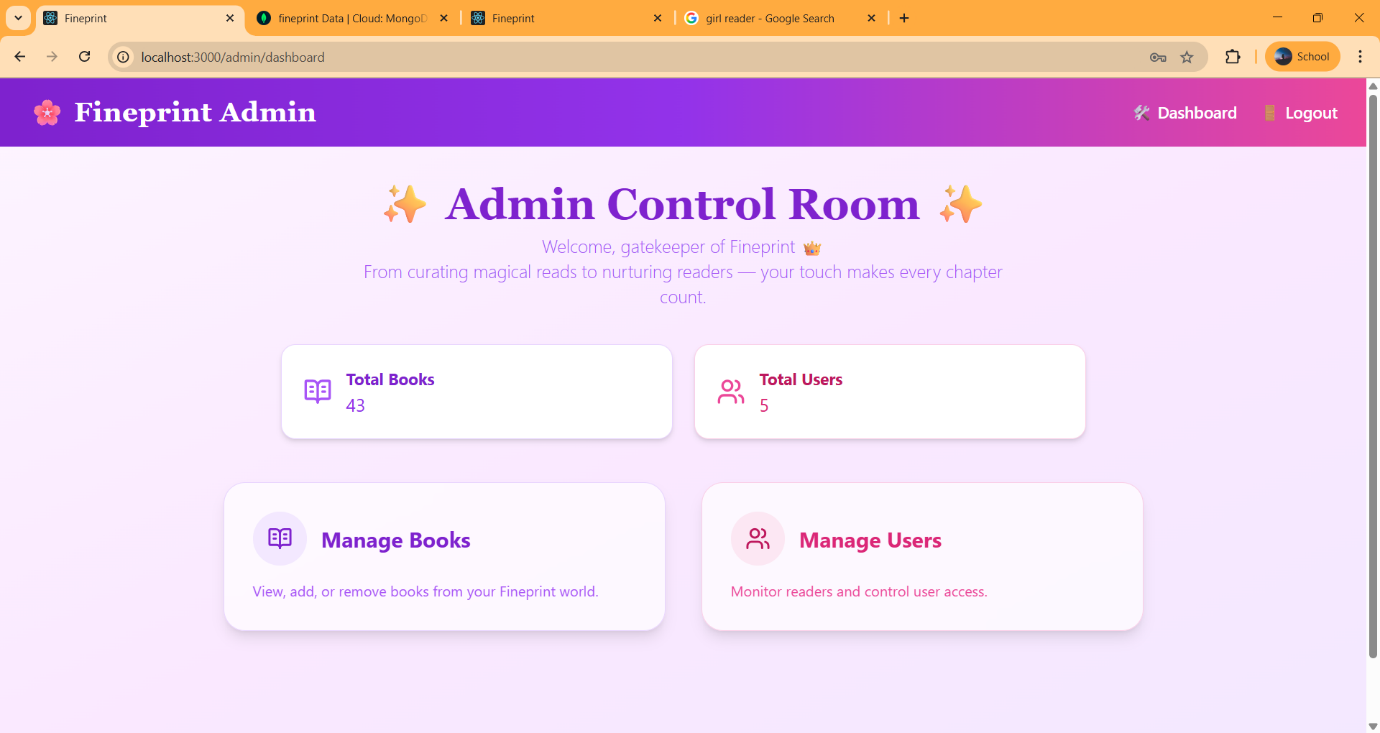




Once Logged in..

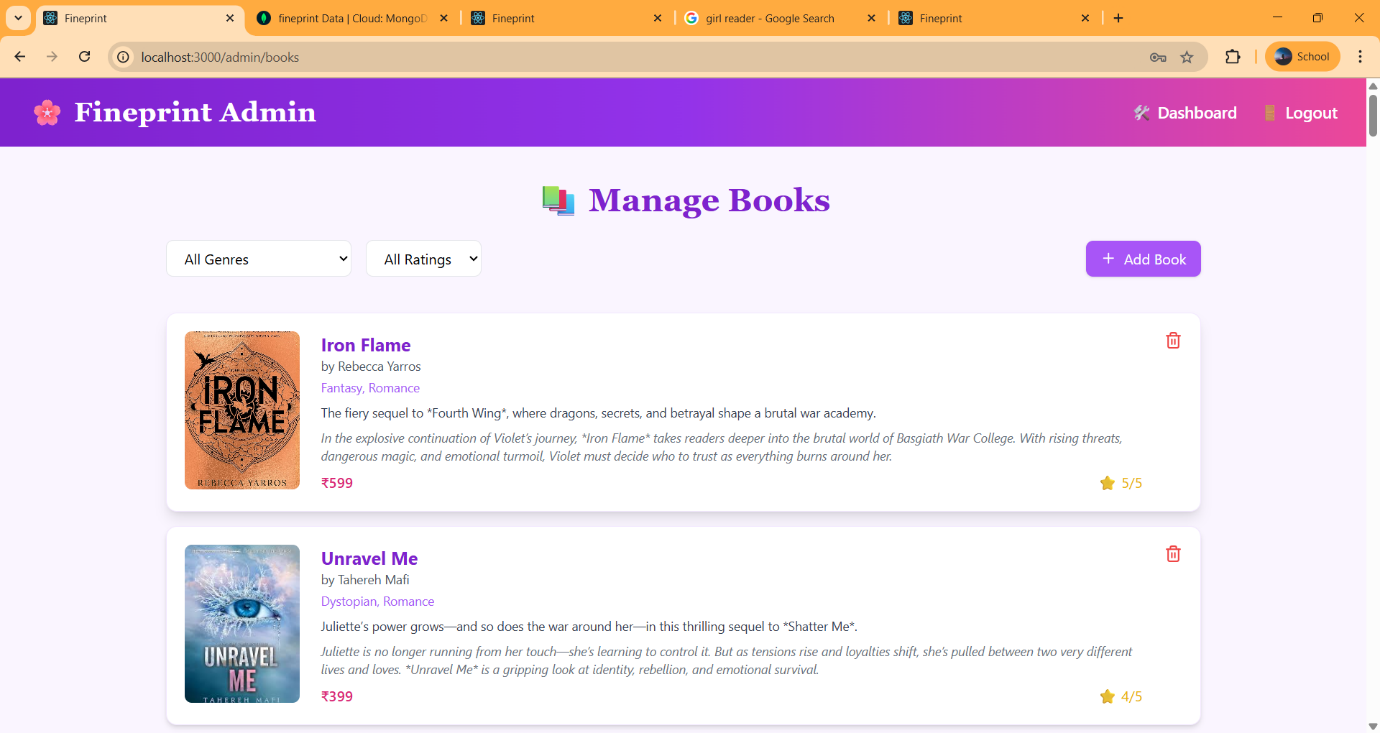
**4.a) THE ADMIN DASHBOARD**

The Admin Dashboard provides a central hub for managing users and books, displaying key analytics like total users and books. With intuitive navigation cards, it empowers admins to monitor and control the Fineprint platform seamlessly.

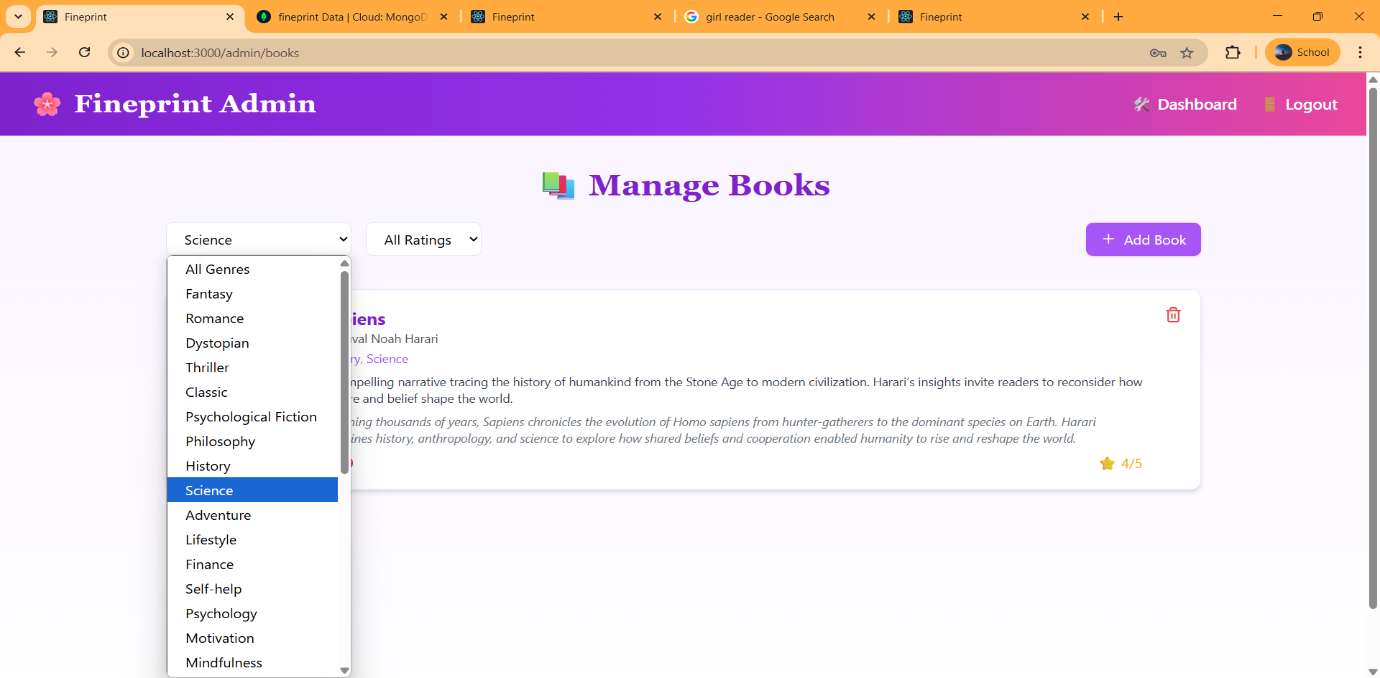


**4.b) MANAGE BOOKS**

The Manage Books page allows admins to seamlessly view, add, and organize books stored in MongoDB. With genre tags, images, and inline details, it ensures smooth control over Fineprint’s ever-growing library.

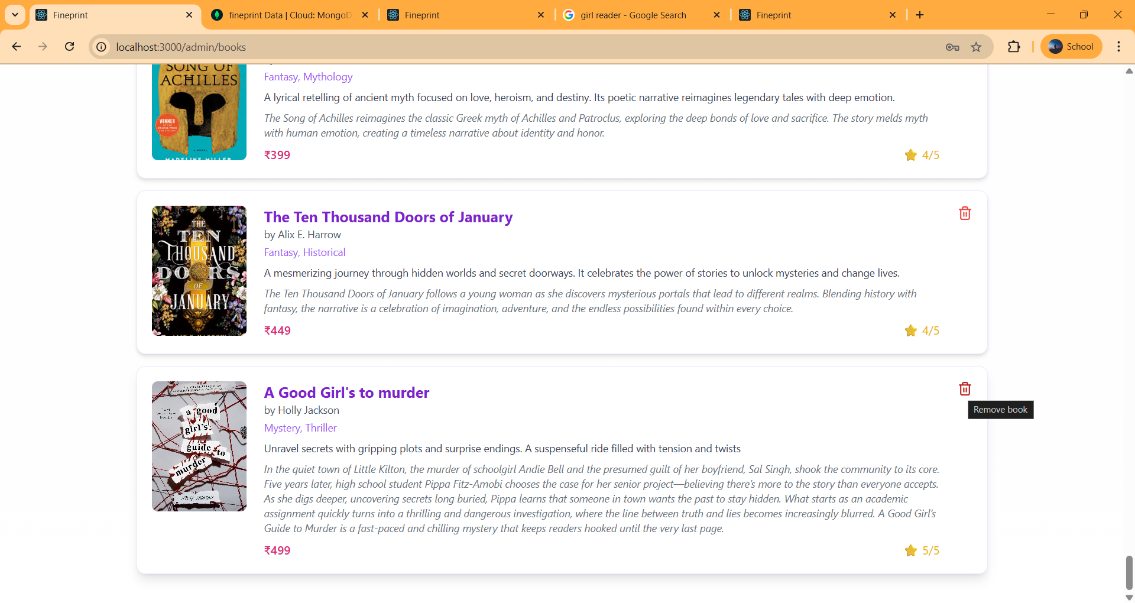


**FILTER OPTIONS**

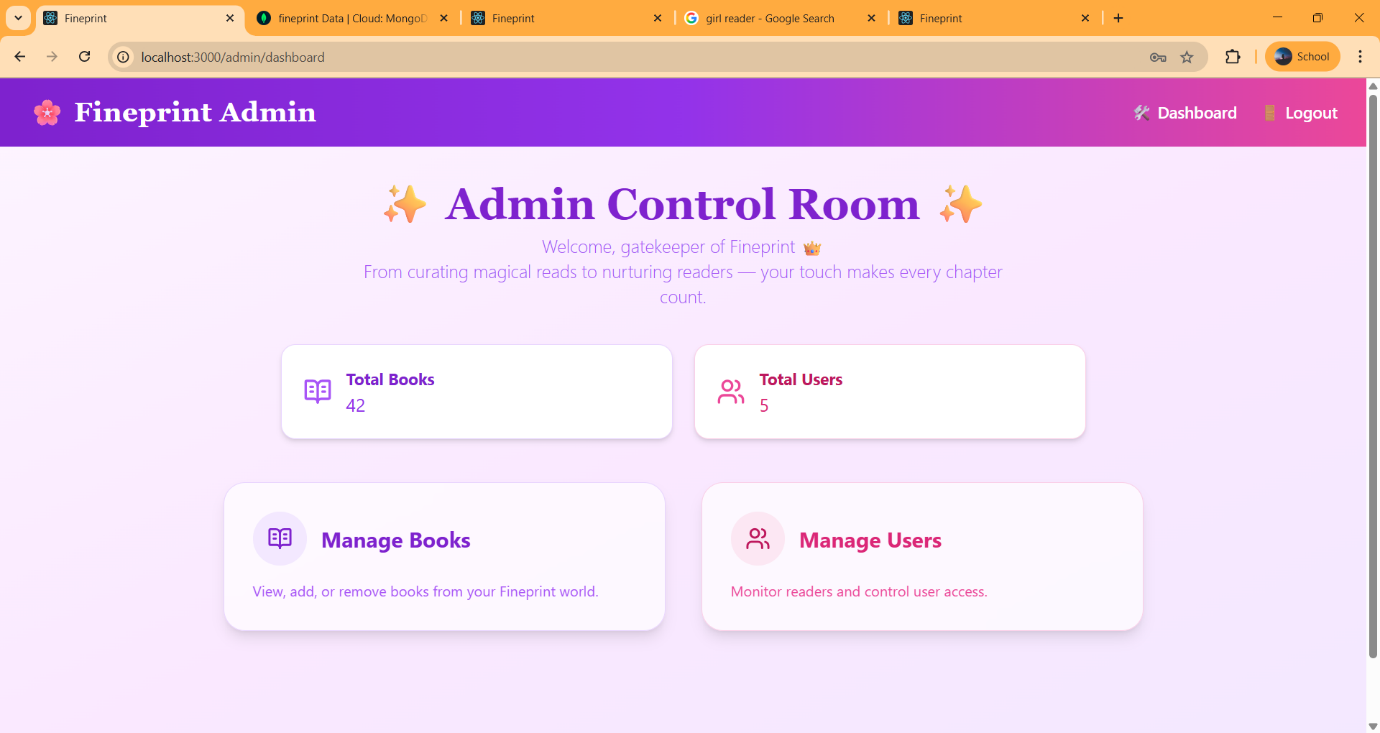




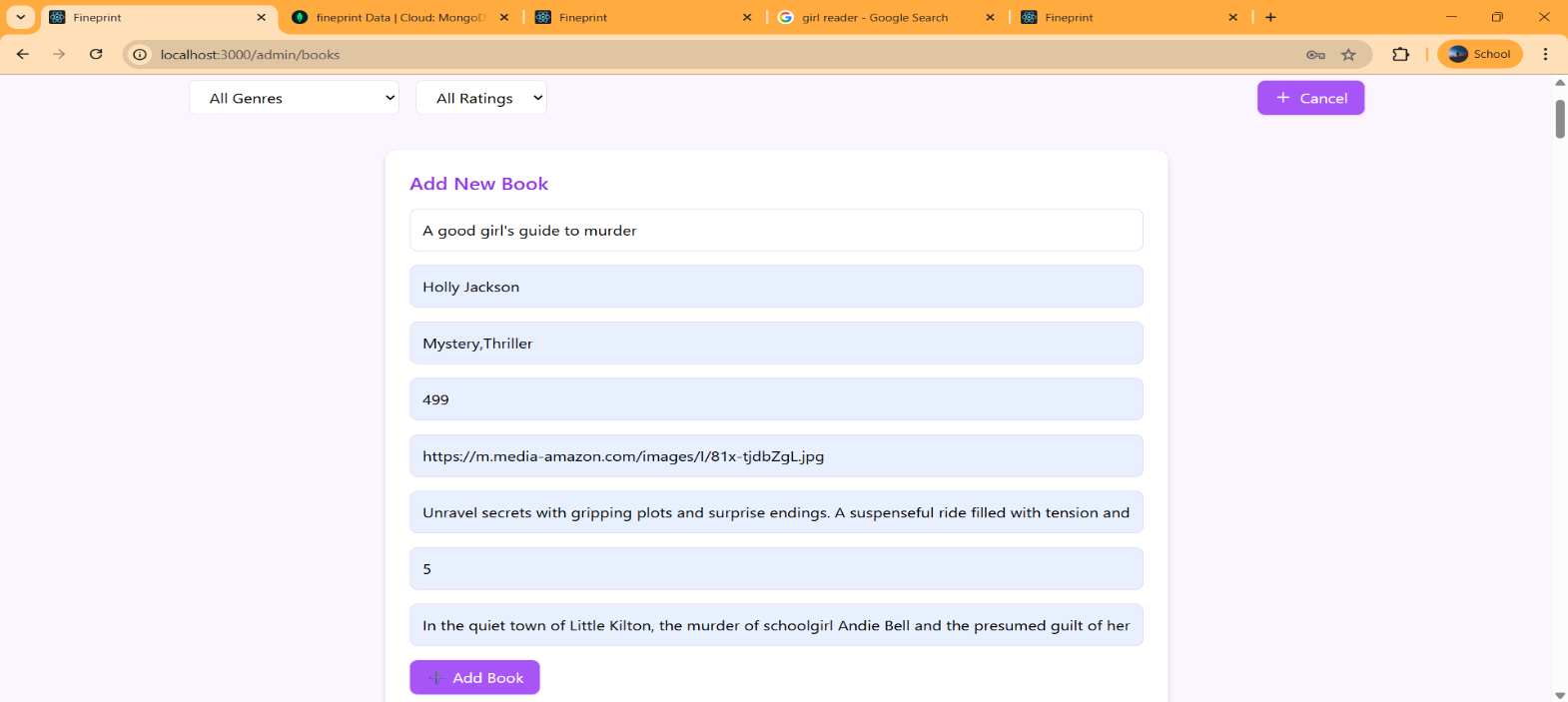
**REMOVE BOOK**



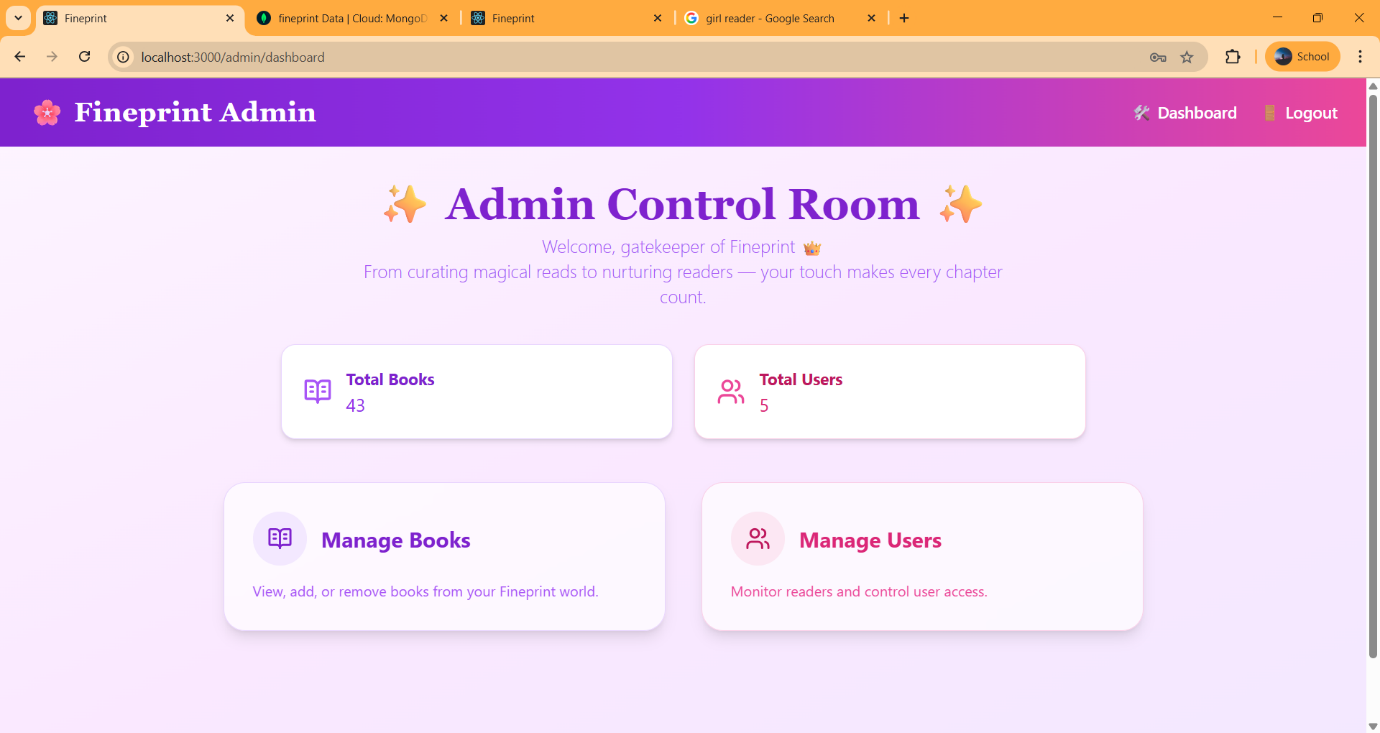
The number also reduces



**ADDING BOOKS**



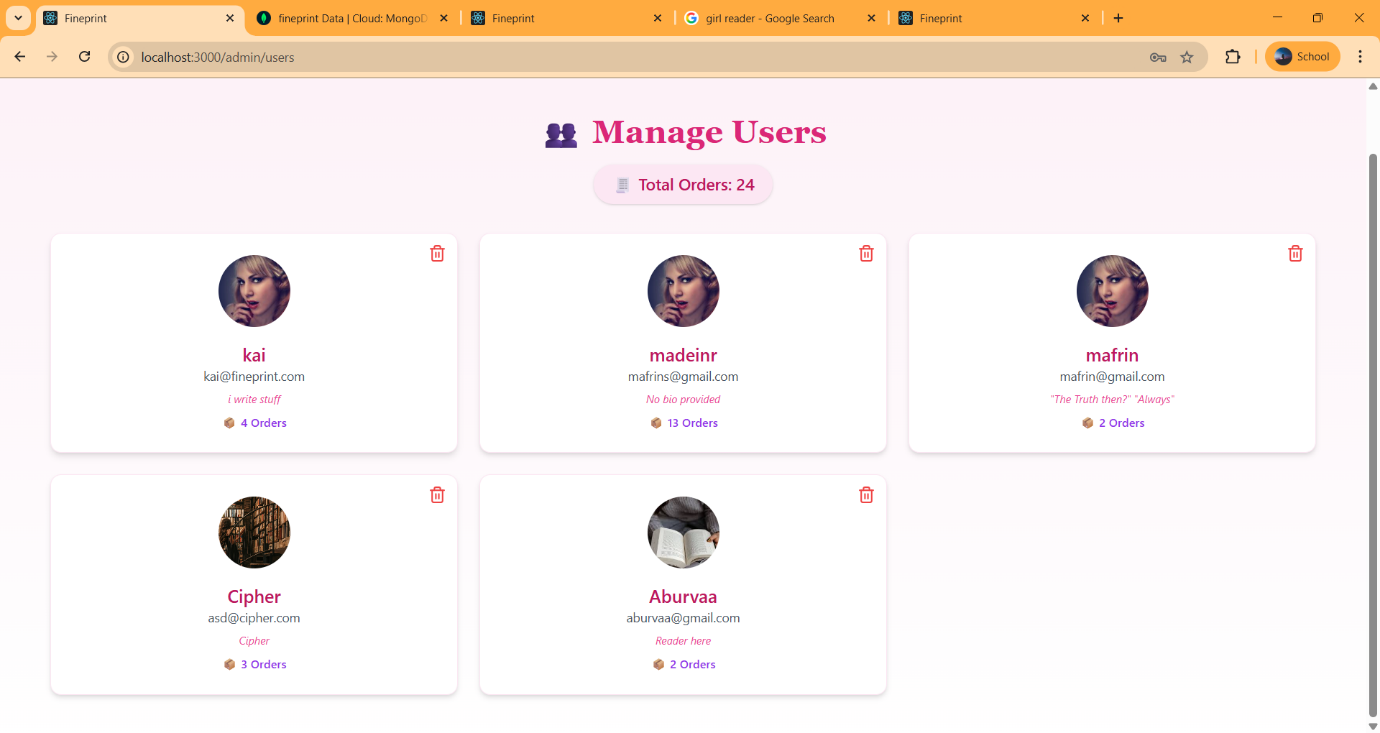
Number gets increased



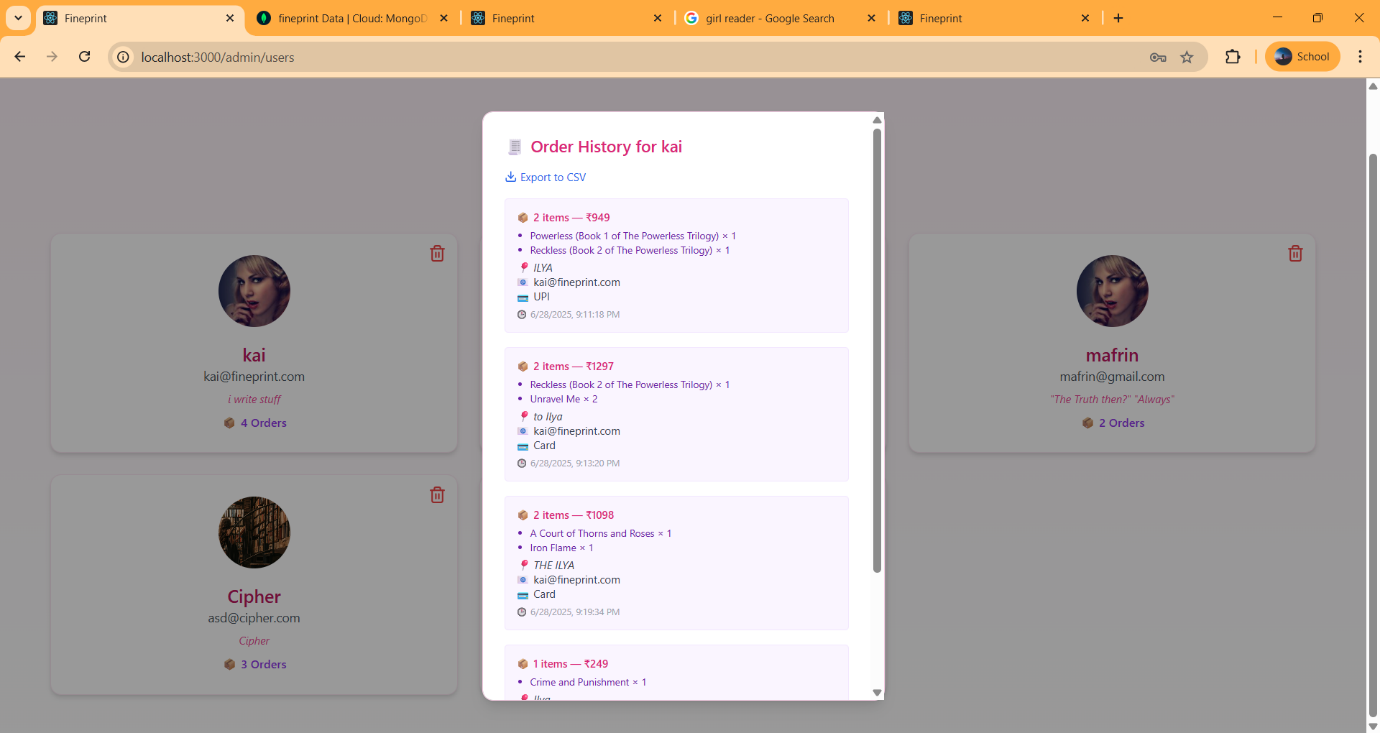
**4.c) MANAGE USERS**

The Manage Users page lets admins view all registered users along with their details, roles, and order history. It offers clean user cards, role badges, and the ability to manage or remove users (excluding self-deletion).

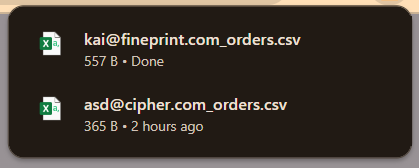
**Auto-inserted Avatar**



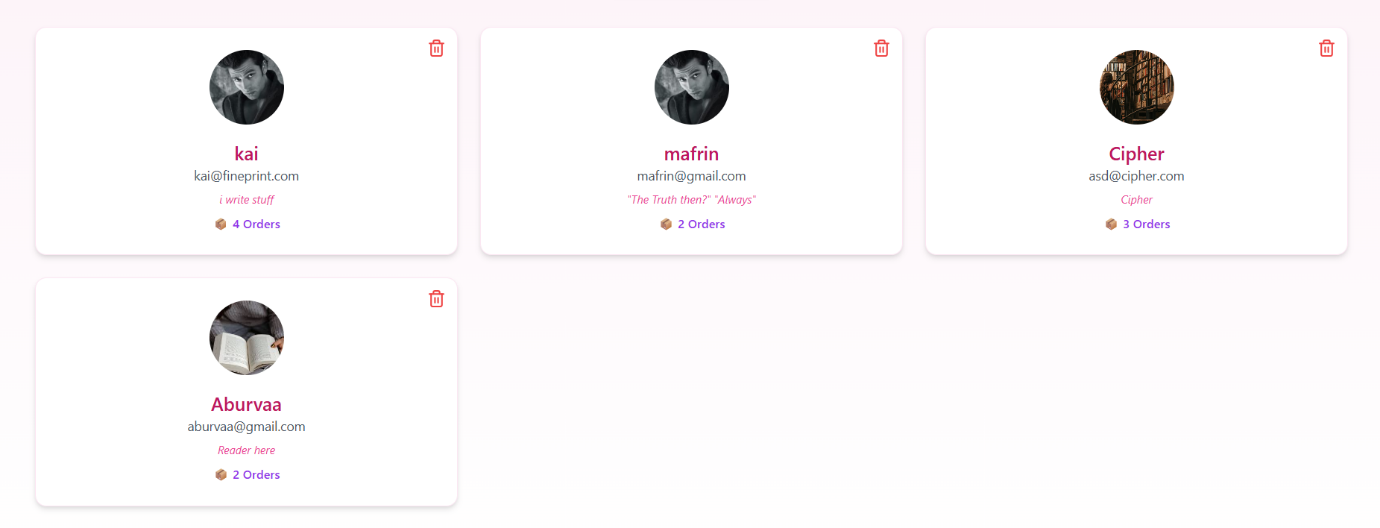
**Order History of Each User**

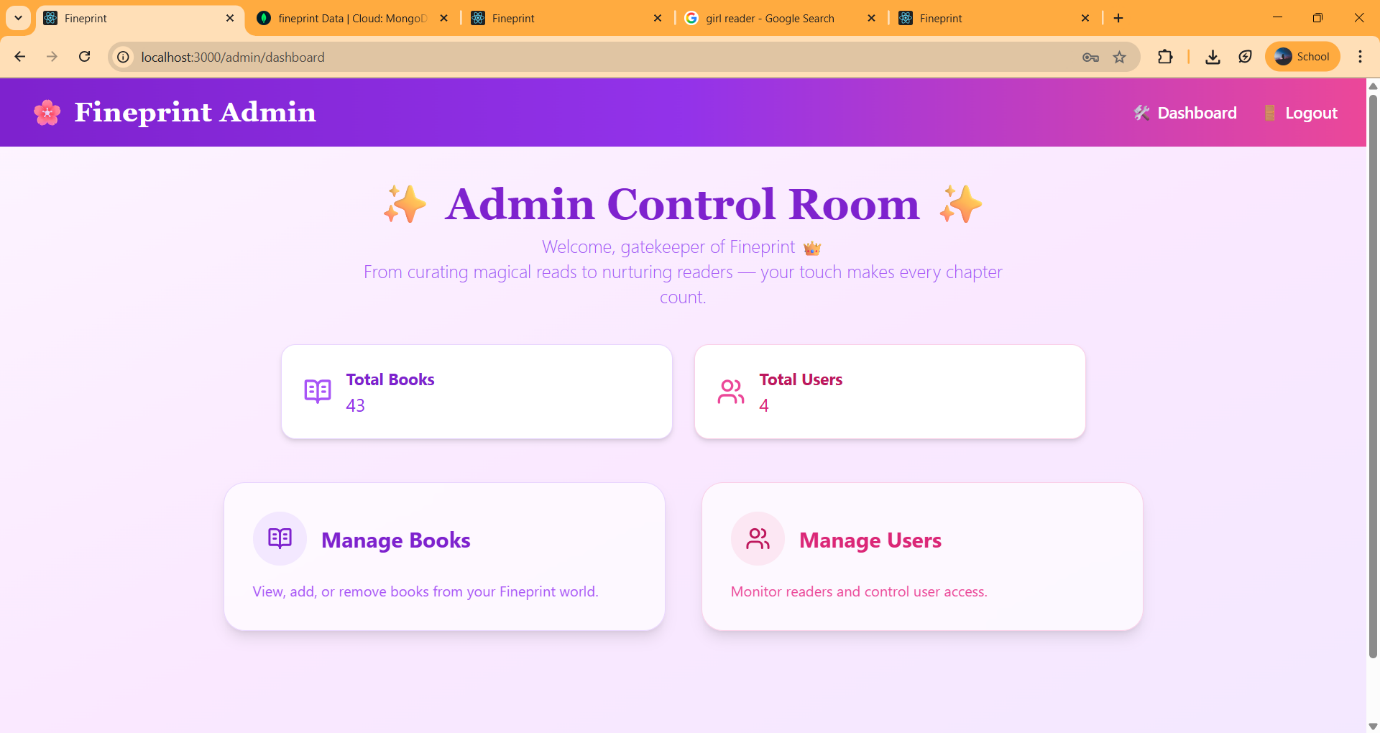


**OrderList of each user can be downloaded in csv**



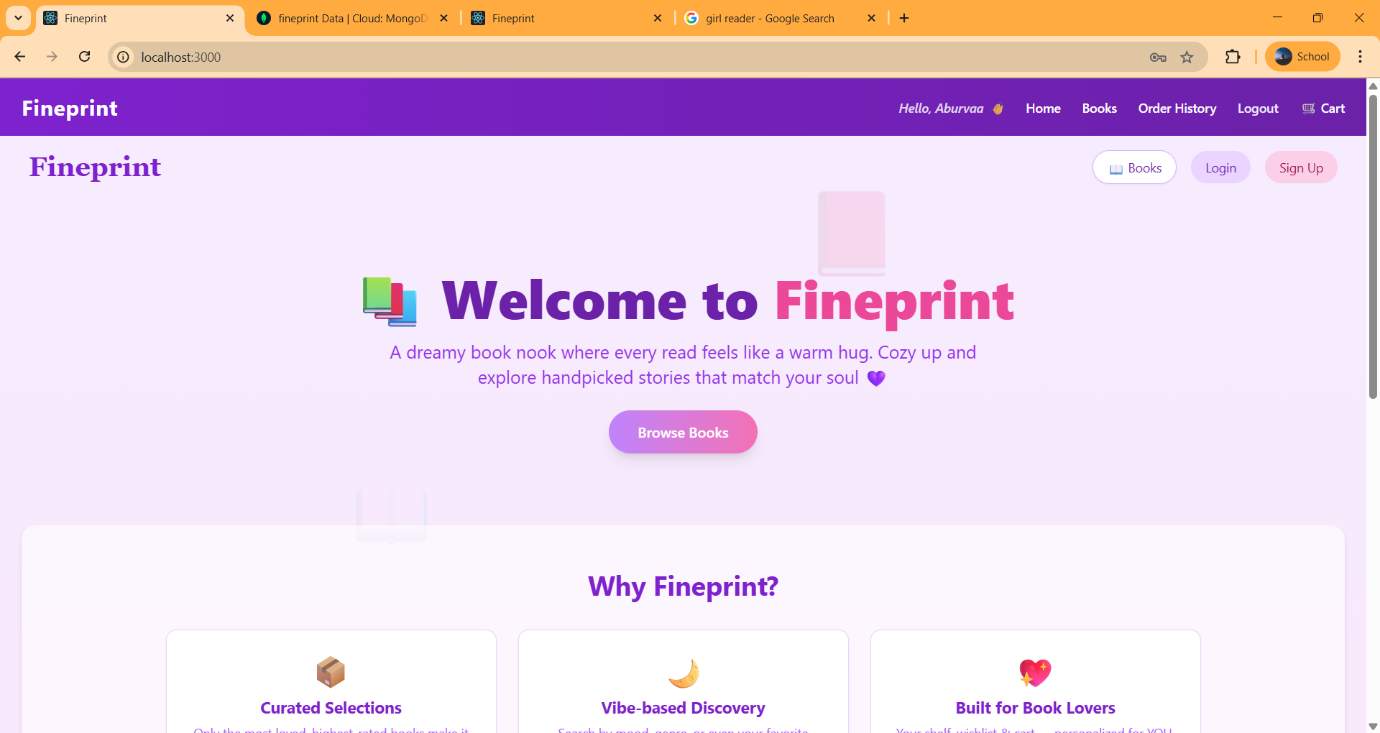
**Deleting USER (**madeinr is deleted)





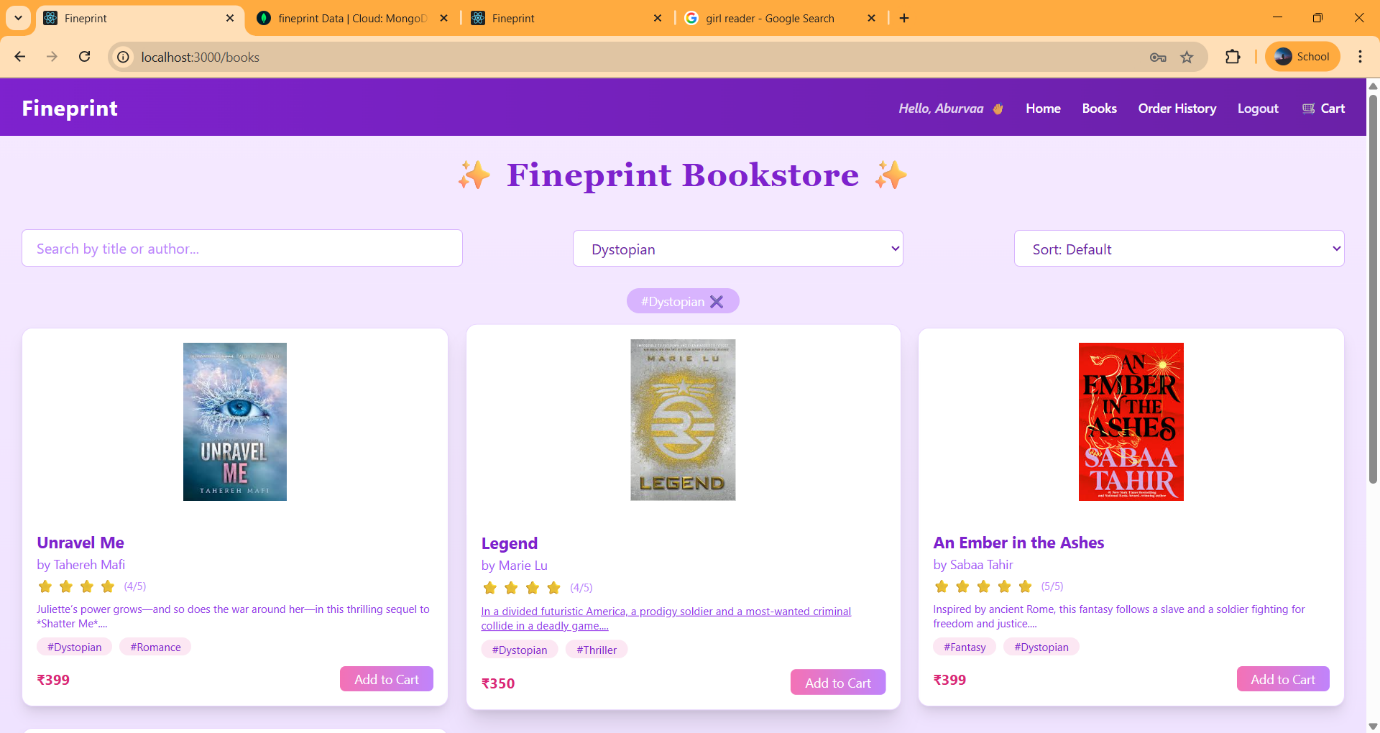
**5) USER ROUTES**

The User Role interface provides a streamlined and personalized experience, allowing users to browse the catalog, manage their cart, proceed with secure checkout, and view their order history. It ensures an intuitive and engaging environment tailored for seamless book discovery and purchase.



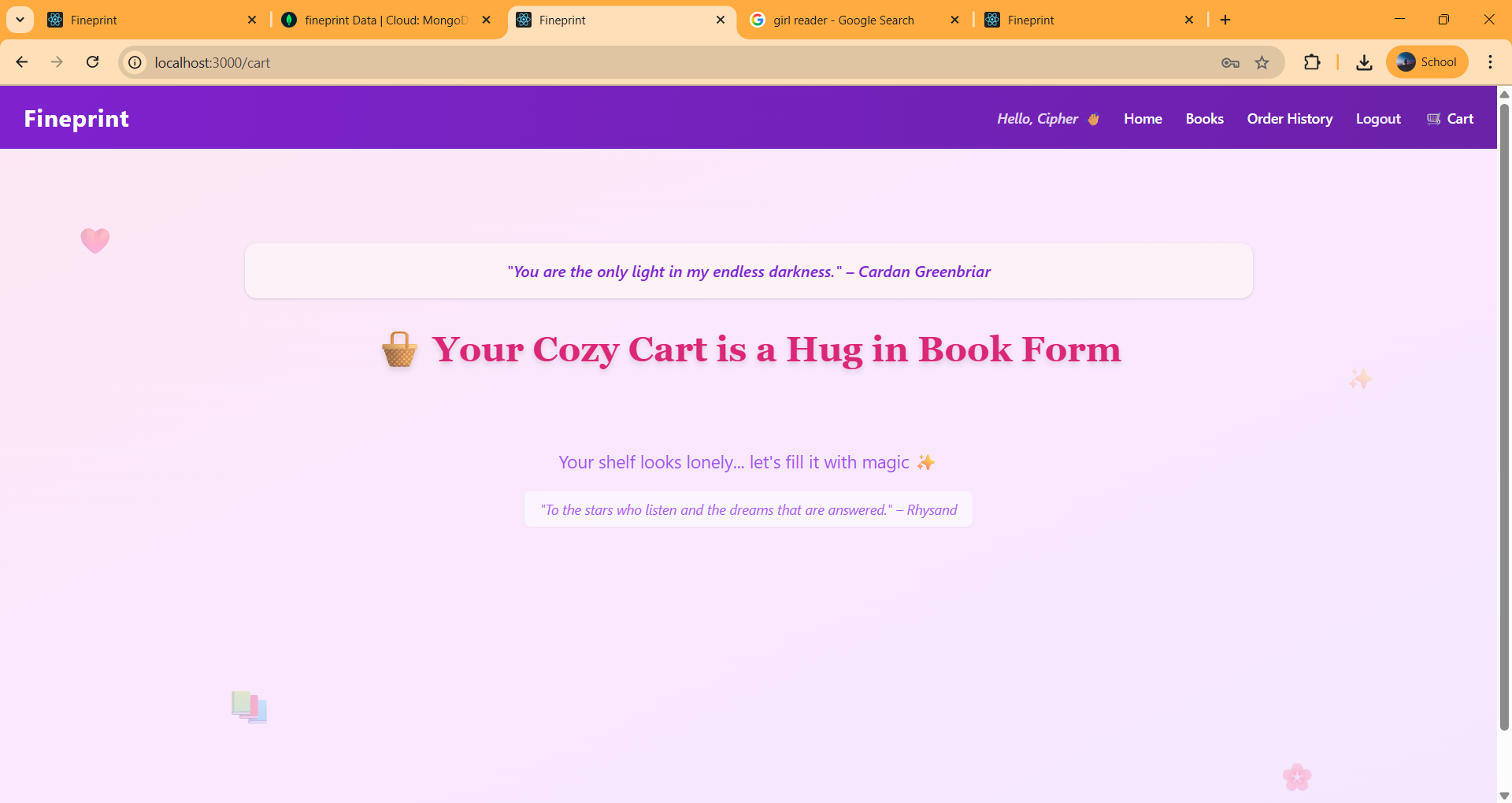
**5.a) PRODUCT DISPLAY**

The Books Page displays the complete catalog of available titles with detailed information including author, genre, price, and ratings. Designed for easy browsing and discovery, it allows users to explore and add books to their cart effortlessly.



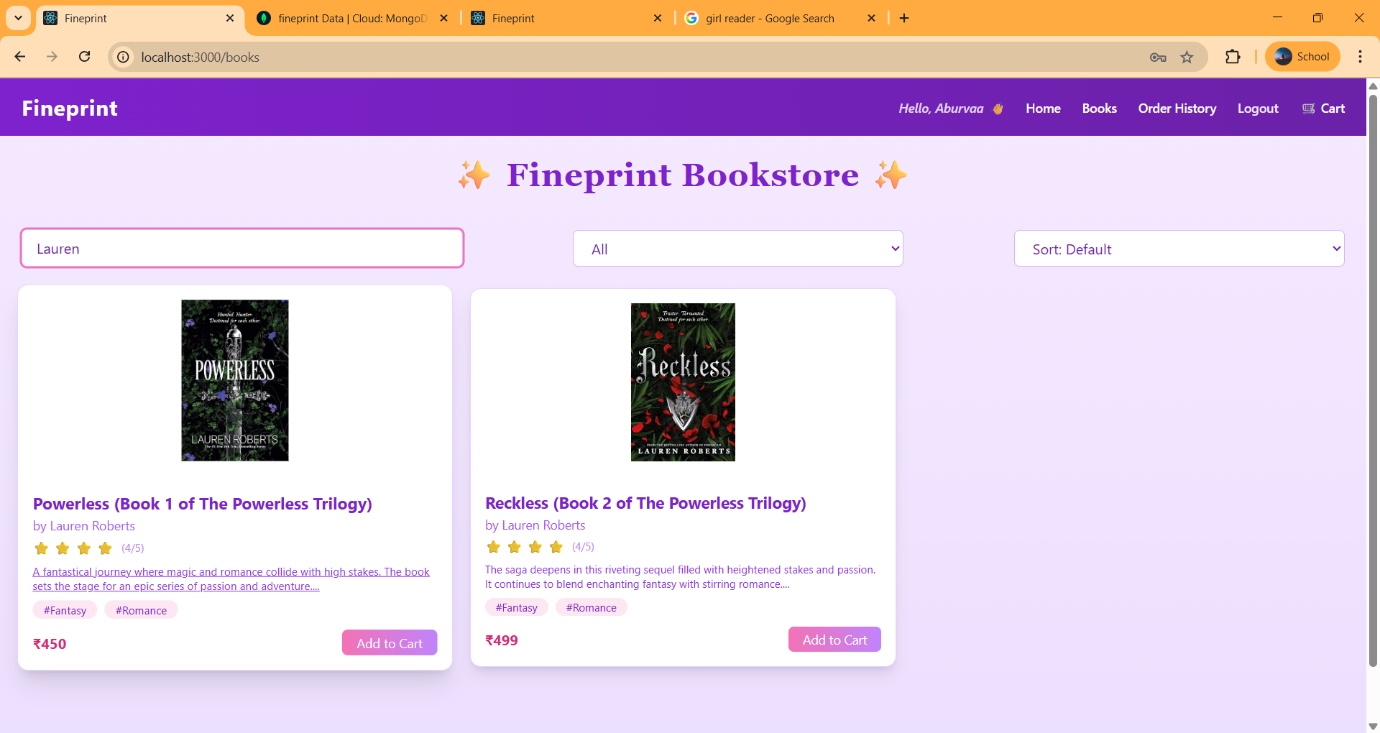
**EMPTY CART**

The Empty Cart view gently informs users that their cart is currently empty, using warm visuals and motivational BookTok quotes. It encourages users to start exploring and adding books, maintaining the cozy and engaging tone of the platform

****

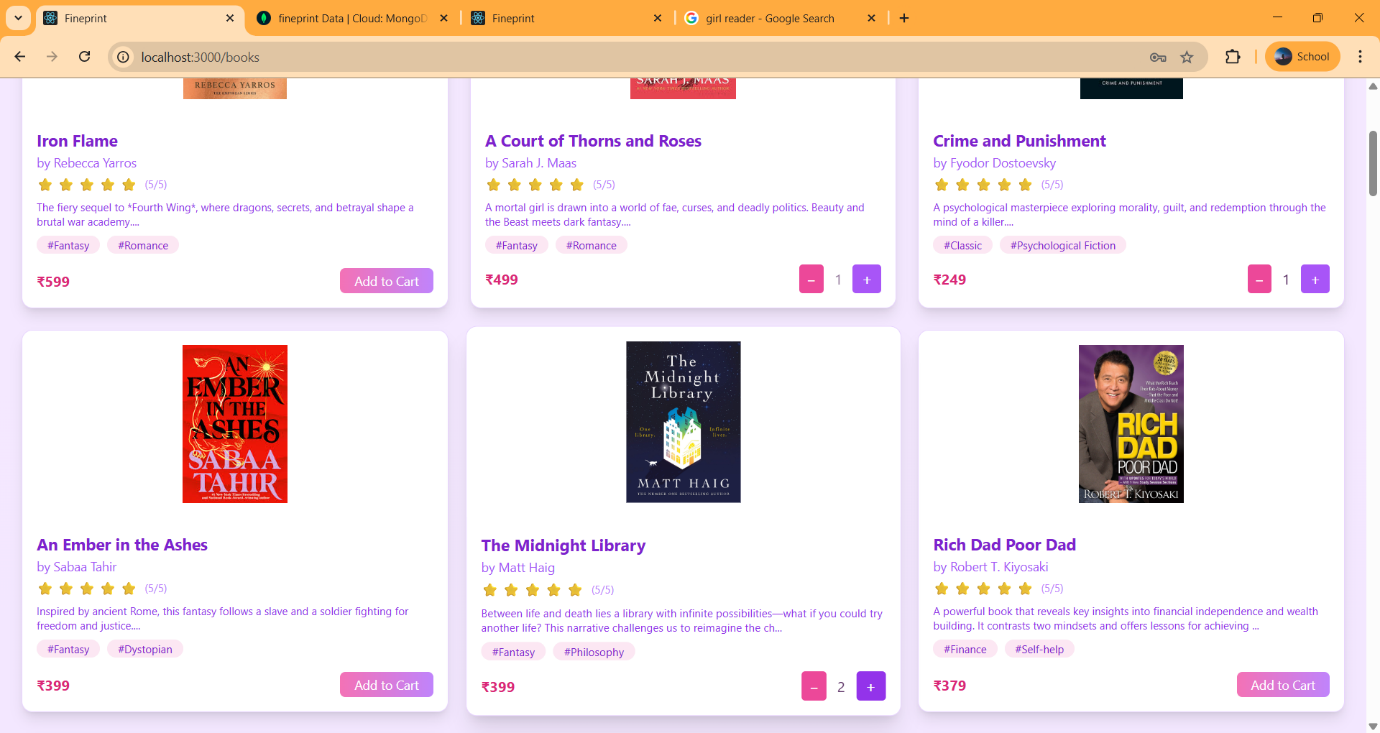
**FILTERS**

The **Books Page** includes intuitive **filters** that let users refine their search by genre, rating, and price range. This ensures a personalized browsing experience, helping users discover books that match their preferences quickly and efficiently.



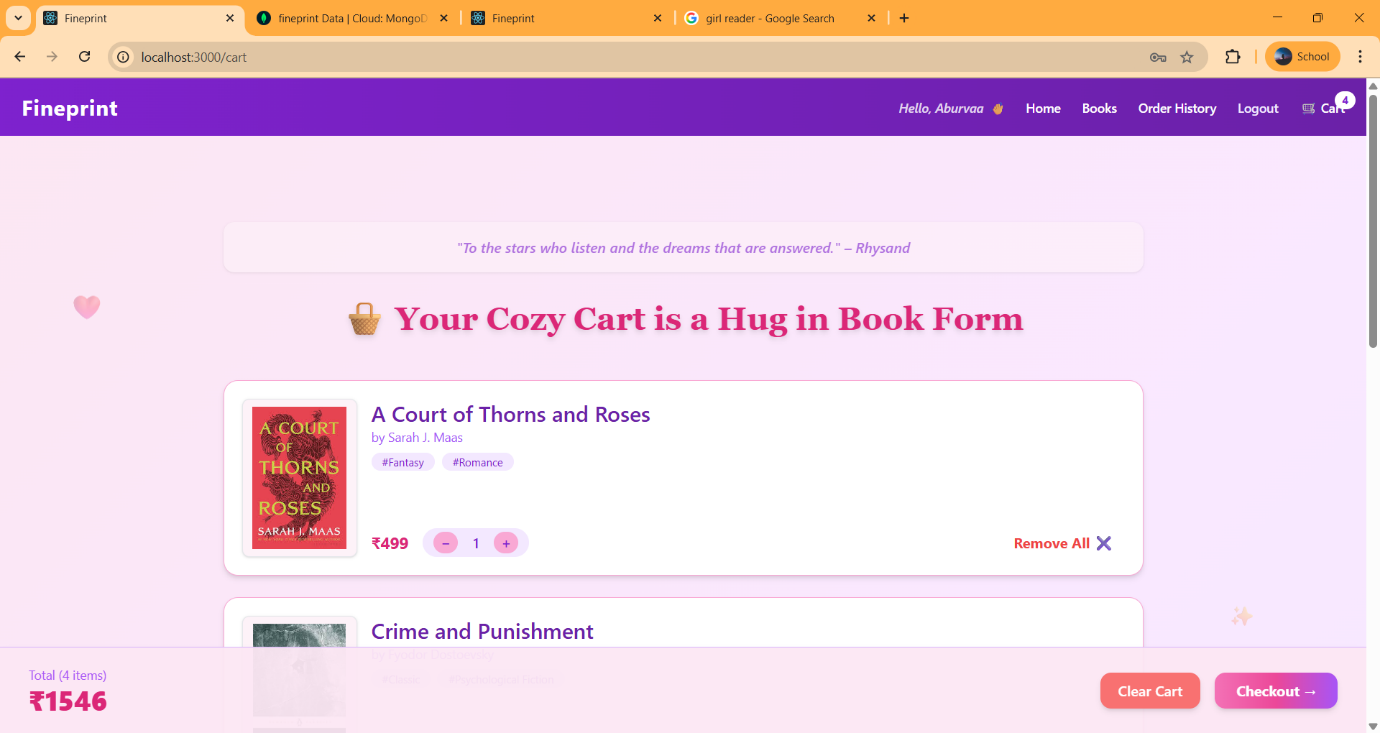


**5.b)ADDING TO THE CART**

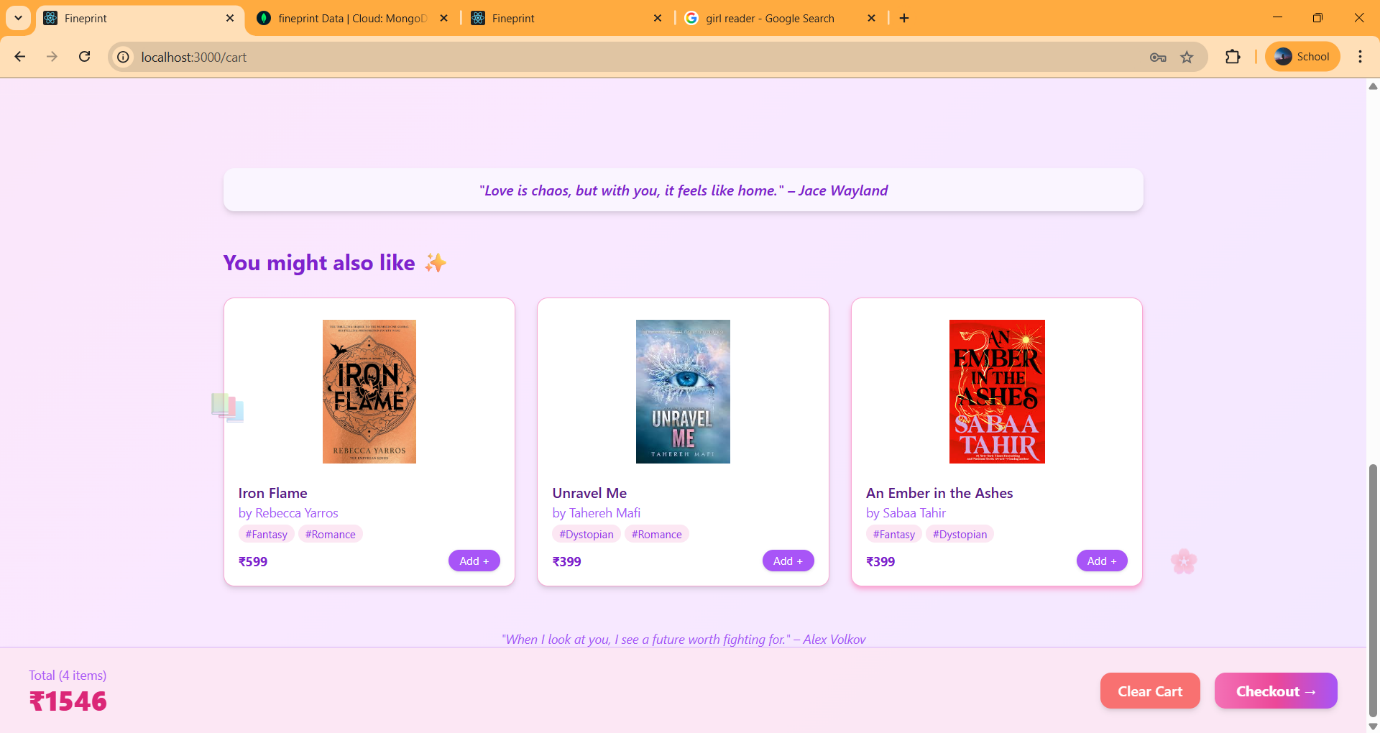


**5.c) CART**

The Cart Page displays all selected books in a grouped, visually engaging layout, showing quantities, prices, and genre tags. Users can adjust quantities, remove items, and proceed to checkout, with cozy animations and rotating BookTok quotes enhancing the experience.

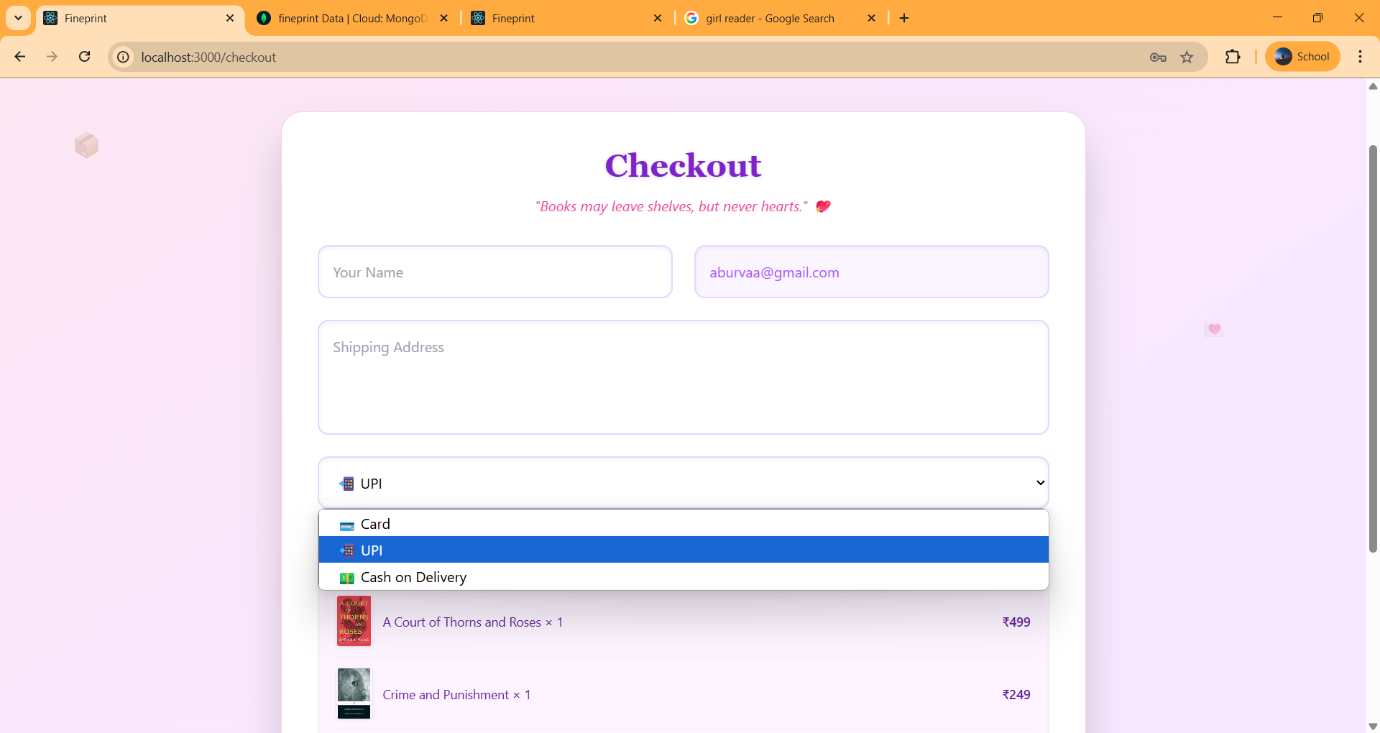


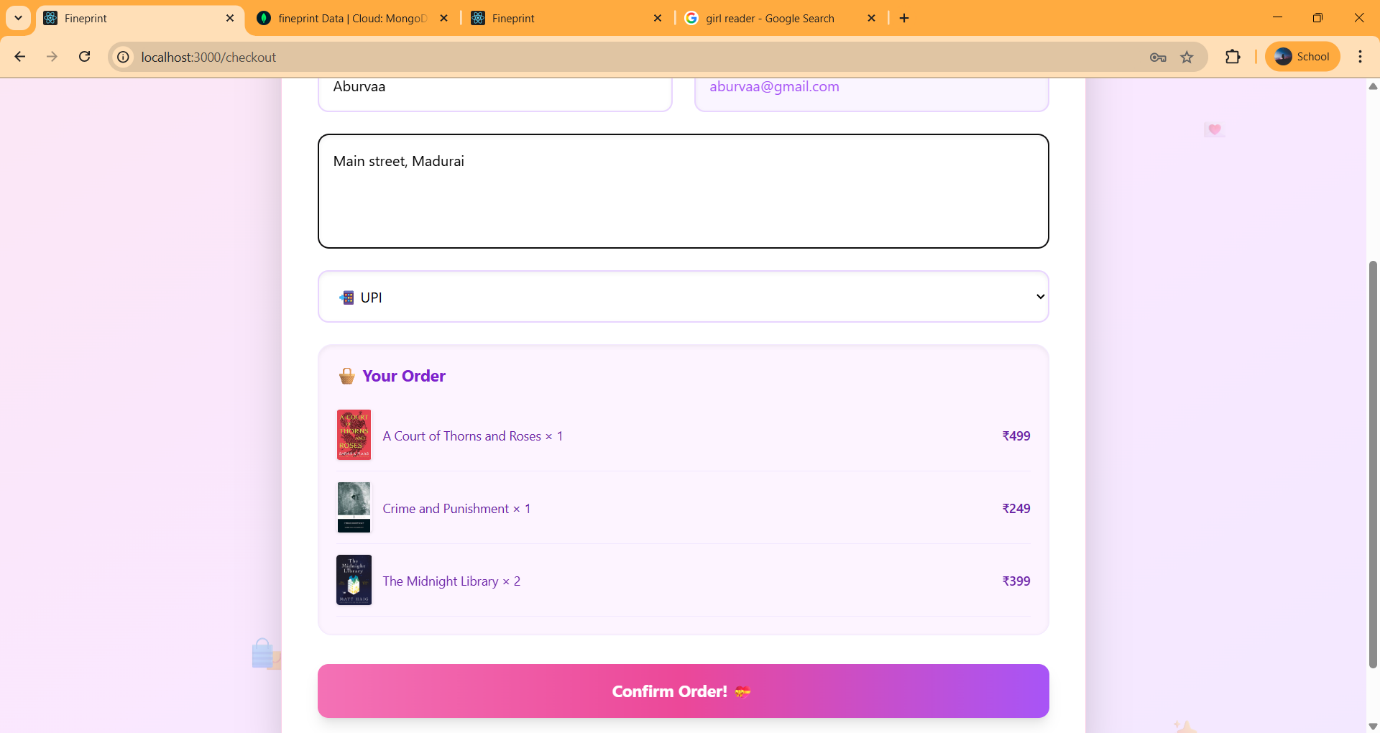
The Cart Page displays all selected books in a grouped, visually engaging layout, showing quantities, prices, and genre tags. Users can adjust quantities, remove items, and proceed to checkout, with cozy animations and rotating BookTok quotes enhancing the experience.



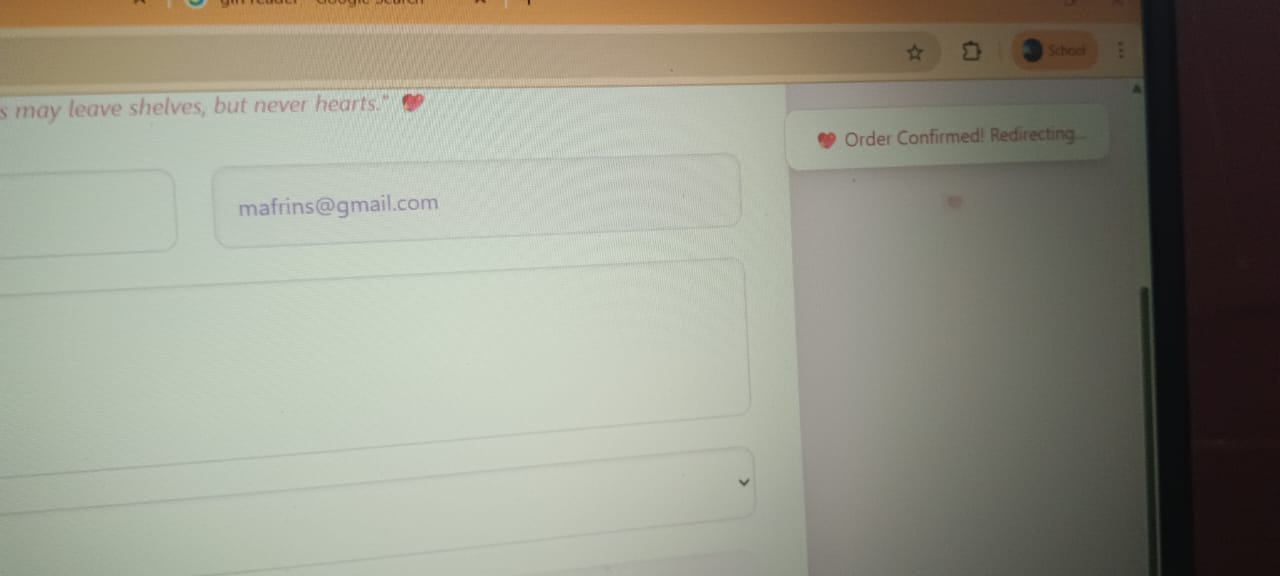
**5.d) CHECKOUT**

The Checkout Page collects essential user details like name, address, and payment method while displaying a summary of the cart items. It ensures a secure and user-friendly order placement process, with a clean interface and confirmation feedback.

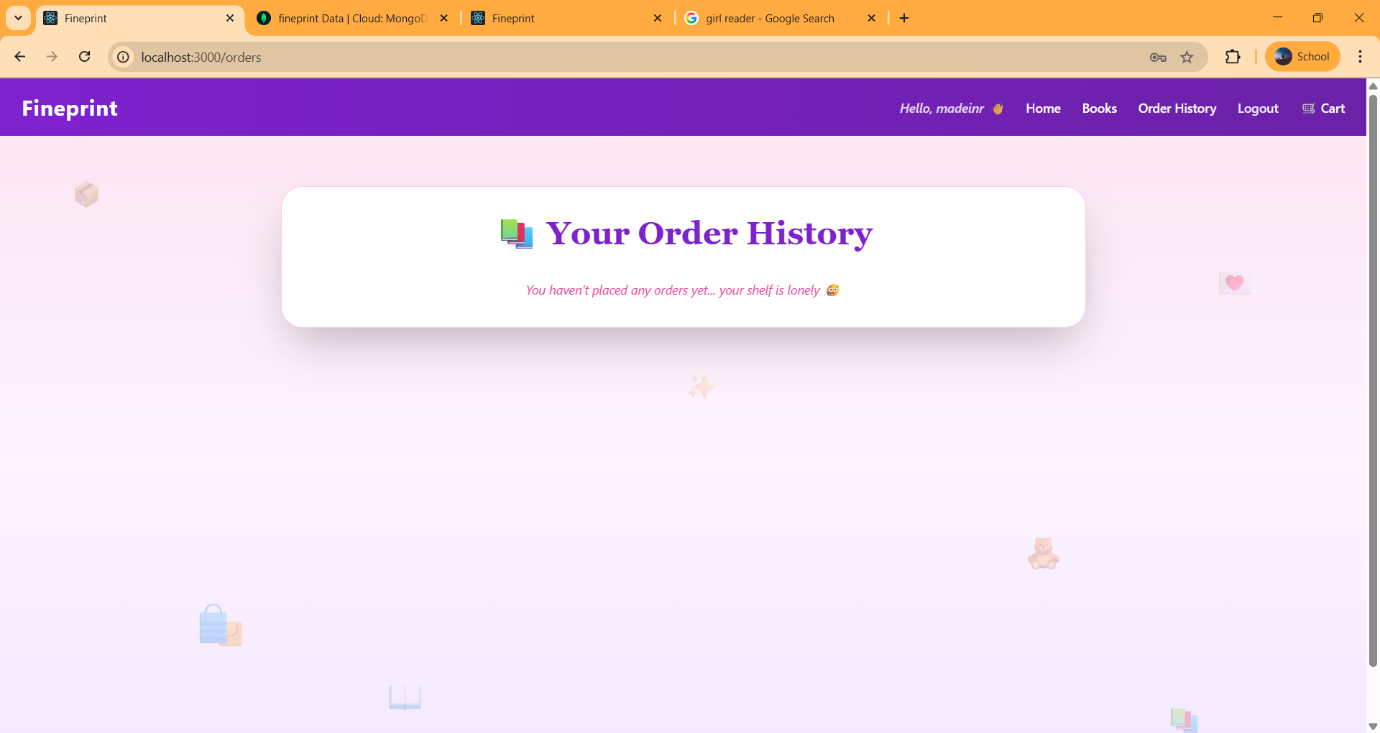




**5.e) ORDER CONFIRMED MESSAGE**

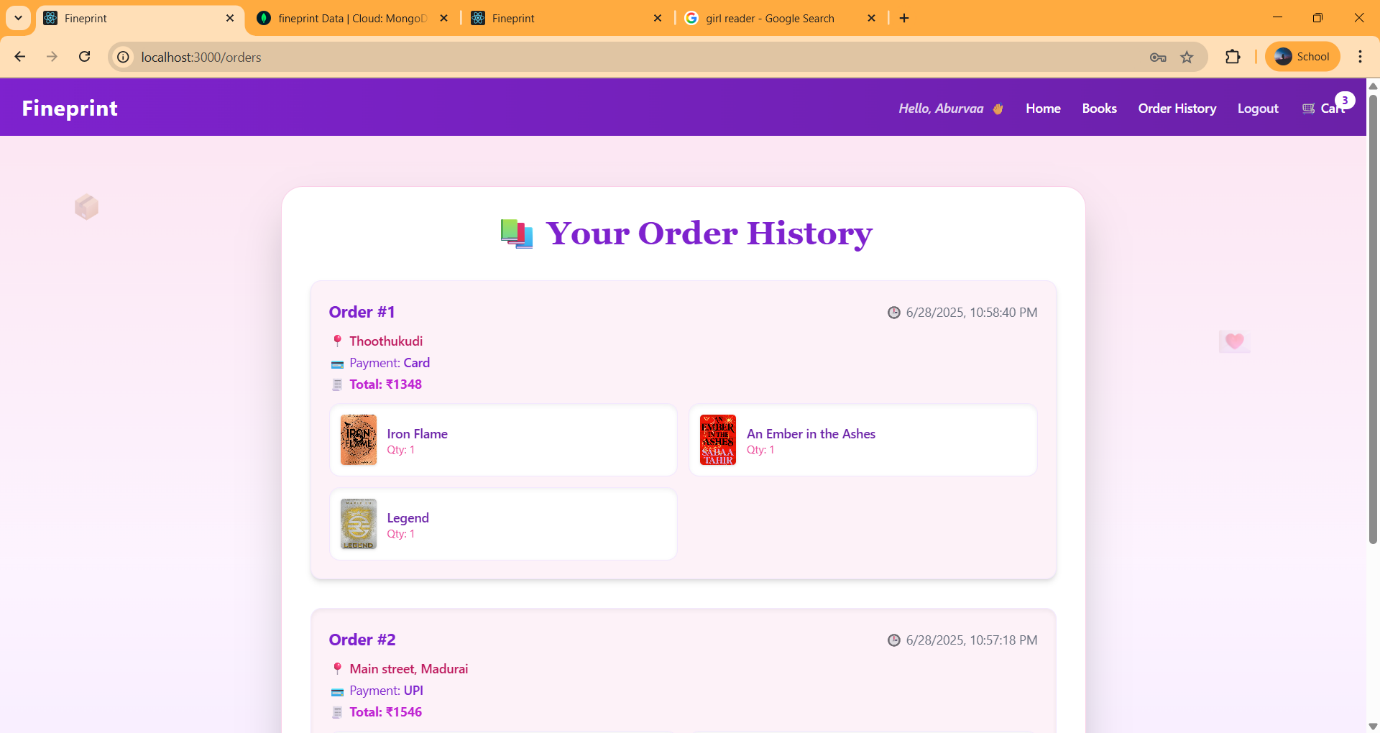


**5.f) INTIAL EMPTY ORDER HISTORY**



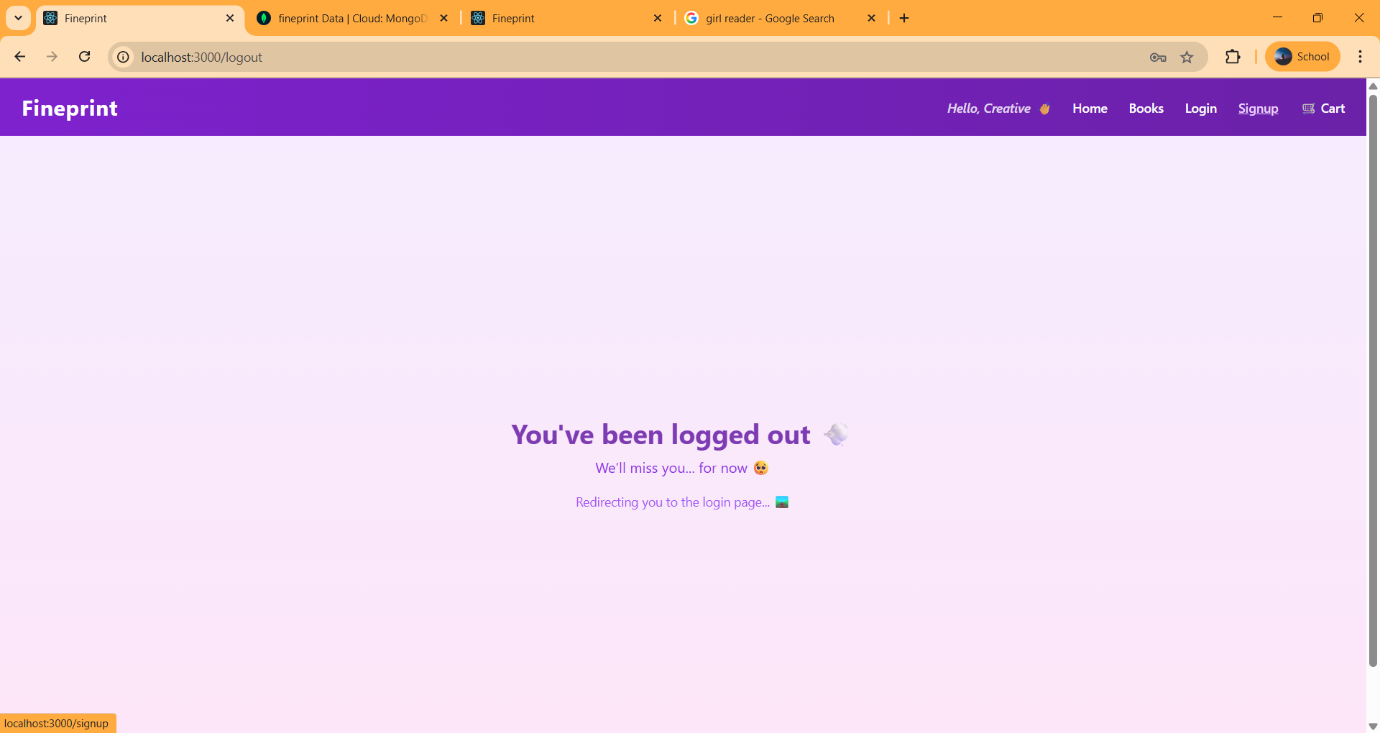
**5.g) ORDER HISTORY**

The Order History page displays a chronological list of all past orders placed by the user, including book details, quantities, total amount, and order date. It provides a clear and organized view of their reading and purchase journey.



**6.LOGOUT**

The Logout functionality securely ends the user's session, clears relevant local data (such as cart and user info), and redirects them to the login page. It ensures user data protection and a smooth exit experience.



**IV. FOLDER STRUCTURE**

fp/

├── client/ # React frontend

│ ├── public/ # Static assets

│ └── src/

│ ├── assets/ # Images, icons, etc.

│ ├── components/ # Reusable UI components

│ │ └── Admin/ # Admin-specific components

│ ├── data/ # Local JSON files (e.g., books\_data.json)

│ ├── pages/ # React pages for routing

│ ├── styles/ # Tailwind config or custom CSS (if needed)

│ ├── App.js

│ └── index.js

│

├── server/ # Node.js + Express backend

│ ├── models/ # Mongoose schemas (User.js, Book.js, Order.js)

│ ├── routes/ # Express routes (user.js, book.js, checkout.js)

│ ├── controllers/ # (optional) Route handler logic

│ ├── config/ # DB config or environment setup

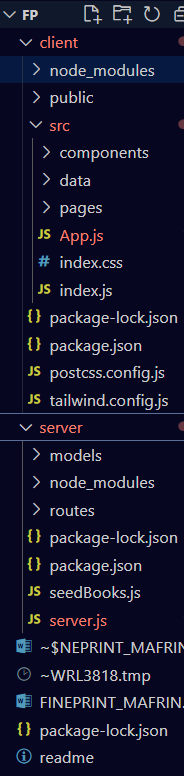
│ ├── server.js # Main Express app

│

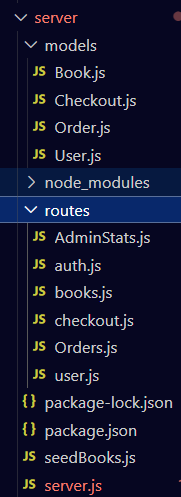
├── package.json # Root dependencies (concurrent scripts)

├── README.md # Project description and setup

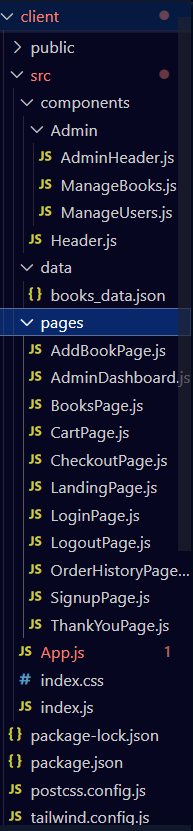
└── README\_DOC.md # Fineprint Documentation

****

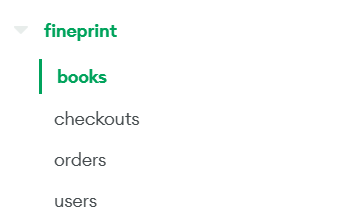
**BACKEND**

****

**FRONT END**



**MONGO DB COLLECTIONS**

****