

Library Store (E-Commerce Web)

Name: Rhythm Paudel

Email: rhythmpaudel12@gmail.com

Github: github.com/rhythm-paudel

LinkedIn: linkedin.com/in/rhythmpaudel

Table of Contents

1. Project Overview.....	1
1.1. Purpose.....	1
1.2. Scope	1
1.3. Objectives	1
2. Features and Functionalities	2
2.1. Login	2
2.2. Browsing Paginated Book Catalogue.....	4
2.3. View Book Details	7
2.4. Applying Search, Sort and Filters.....	9
2.5. Registering Users to Members.....	11
2.6. Add to Whitelist (Bookmark).....	12
2.7. Add to Cart	16
2.8. Place Cancellable Order	19
2.9. Review Purchased Books	21
2.10. Discount System	23
2.10.1. 5% Bulk Order Discount on Order of 5+ Books	23
2.10.2. 10% Stackable Discount After 10 Successful Orders.....	26
2.11. Receive Claim Code and Order Details in Registered Email	29
2.12. Staff Order Fulfilment Via Member Claim Code	30
2.13. Admin Book Catalogue and Inventory Management (CRUD).....	32
2.14. Admin Timed Discounts with Optional “On Sale” Flag	36
2.15. Admin Timed Banner Announcement	38
2.16. Real-Time Successful Order Broadcasting	39
3. Proof of Work.....	41
3.1. Entity Relationship Diagram.....	41

3.2.	UI Design	42
3.2.1.	Admin Pages UI Design	44
3.2.2.	User Pages UI	57
3.3.	UML Diagrams	64
3.3.1.	Use Case Diagram	64
3.3.2.	Sequence Diagram.....	65
3.4.	Git Hub Repository.....	70
3.5.	Migration History	71
3.6.	Milestone Chart	72
3.7.	API Endpoints	73
3.7.1.	Announcement Endpoints	75
3.7.2.	Announcement User Endpoints.....	75
3.7.3.	Auth Endpoints	75
3.7.4.	Author Endpoints	75
3.7.5.	Bookmark Endpoints	76
3.7.6.	Books Endpoints.....	76
3.7.7.	Cart Endpoints	76
3.7.8.	Dashboard Endpoint.....	76
3.7.9.	Homepage Endpoint.....	76
3.7.10.	Order Endpoint	76
3.7.11.	Review Endpoint.....	76
3.7.12.	Searchs Endpoint	77
3.7.13.	UserDetails Endpoint.....	77
3.7.14.	Users Endpoint	77
3.8.	Testing	78
3.8.1.	API Testing	78
3.8.2.	Frontend Unit/System Testing	87
4.	Individual Reflections.....	97
4.1.	Roles and Responsibilities.....	97
4.2.	Personal Insights	98
4.3.	Challenges	98
4.4.	Learnings	98

4.5. Growth	98
4.6. Impact on Future Work	98
4.7. Personal Evaluation	99
5. Conclusion	100

Table of Figures

Figure 1: Login Process	2
Figure 2: Successful Login and Redirection for Admin	3
Figure 3: Categories Page	5
Figure 4: Second Page for Categories	6
Figure 5: List of Books in Home Page	7
Figure 6: Respective Book Description	8
Figure 7: Categories Page and its Filters	9
Figure 8: Successful Searching of Book	9
Figure 9: Registration Form	11
Figure 10: Successful Addition of Member	12
Figure 11: Select Desired Book and Click It	13
Figure 12: Add Bookmark	14
Figure 13: Successfully Bookmarked Message	14
Figure 14: Addition of Bookmark in My Bookmarks Page	15
Figure 15: Bookmark Removal	15
Figure 16: Choosing Book for Add to Cart	16
Figure 17: Adding to Cart	17
Figure 18: Successfully Added to Cart Message	17
Figure 19: Successful Addition to Carts Sub-Section of Orders Tab	18
Figure 20: Checkout from Carts	19
Figure 21: Order in Pending Tab of Member	19
Figure 22: Order in Pending Tab of Admin	20
Figure 23: Before Leaving Review and Rating	21
Figure 24: Successfully Left Review and Rating	22
Figure 25: Cart ready to checkout with 6 unique books	23
Figure 26: View Pending Orders Tab	24
Figure 27: Discount Details	25
Figure 28: Ten Confirmed Orders	26
Figure 29: Placing 11 th Order	27
Figure 30: Viewing 11 th Order	27
Figure 31: 10% Stackable Discount After 10 Successful Orders	28
Figure 32: Order Placed Successfully	29
Figure 33: Code and Order Details in Registered Email	29
Figure 34: Claim Code in Mail for Staff Order Fulfilment	30
Figure 35: Staff Order Fulfilment via Member Claim Code	30
Figure 36: Successful Staff Order Fulfilment via Member Claim Code	31
Figure 37: Add Book Button	32
Figure 38: Filling Book Details and Adding the Book	33
Figure 39: Successful Addition of Book	33
Figure 40: Edit Book Details Button	33
Figure 41: Updating Book Name	34

Figure 42: Book Updated Successfully	34
Figure 43: Delete Book Button	34
Figure 44: Confirm Deletion Button	34
Figure 45: Adding a New Book with Discount.....	36
Figure 46: Successful Display of Book in User Side with On Sale Tag	37
Figure 47: Adding a Timed Banner Announcement.....	38
Figure 48: Successful Addition of Timed Banner Announcement in Admin Side	38
Figure 49: Successful Addition of Timed Banner Announcement in User Side	38
Figure 50: Proceeding to Checkout for Real-Time Successful Order Broadcasting	39
Figure 51: Successful Real-Time Successful Order Broadcasting	40
Figure 52: Entity Relationship Diagram	41
Figure 53: Sign up Page	42
Figure 54: Sign In Page	42
Figure 55: Admin Dashboard	44
Figure 56: Borrowed Books	44
Figure 57: Overdue Borrowers	45
Figure 58: Book Management	45
Figure 59: User Management	46
Figure 60: Change Credential Popup	47
Figure 61: Catalogue Action	49
Figure 62: Overdue Books View Popup.....	49
Figure 63: Update User Popup	50
Figure 64: Add Book Popup.....	52
Figure 65: Add User Popup	52
Figure 66: Update Book Popup	54
Figure 67: View Book Popup	54
Figure 68: View User Popup	55
Figure 69: Delete Confirmation Popup	55
Figure 70: View Branch Popup	56
Figure 71: Order Page	57
Figure 72: Browse Book Catalog	58
Figure 73: Cart Page	59
Figure 74: Home Page.....	60
Figure 75: Confirmed Order.....	61
Figure 76: User Profile Page	62
Figure 77: Book Details Page	63
Figure 78: Use Case Diagram	64
Figure 79: Sequence Diagram: User Registration	65
Figure 80: Sequence Diagram: Catalog Management	65
Figure 81: Sequence Diagram: Discount and Announcement.....	66
Figure 82: Sequence Diagram: Bookmark.....	67
Figure 83: Sequence Diagram: Browse and Filter.....	67
Figure 84: Sequence Diagram: Order Processing.....	68

Figure 85: Sequence Diagram: Order Validation	68
Figure 86: Sequence Diagram: Review	69
Figure 87: Git Hub Repository	70
Figure 88: Git Log History as Proof of Work	70
Figure 89: Migration History.....	71
Figure 91: API Endpoints 1	73
Figure 92: API Endpoints 2	73
Figure 93: API Endpoints 3	74
Figure 94: API Endpoints 4	75
Figure 95: API Testing: Test 1	78
Figure 96: API Testing: Test 1	79
Figure 97: API Testing: Test 2	80
Figure 98: API Testing: Test 3	82
Figure 99: API Testing: Test 4	83
Figure 100: API Testing: Test 4	84
Figure 101: API Testing: Test 4	85
Figure 102: API Testing: Test 5	86
Figure 103: Unit/System Testing: Test 1	87
Figure 104: Unit/System Testing: Test 1	88
Figure 105: Unit/System Testing: Test	88
Figure 106: Unit/System Testing: Test 2	89
Figure 107: Unit/System Testing: Test 2	90
Figure 108: Unit/System Testing: Test 3	91
Figure 109: Unit/System Testing: Test 3	91
Figure 110: Unit/System Testing: Test 3	92
Figure 111: Unit/System Testing: Test 4	93
Figure 112: Unit/System Testing: Test 4	94
Figure 113: Unit/System Testing: Test 5	95
Figure 114: Unit/System Testing: Test 5	96

Table of Tables

Table 1: Milestone Chart.....	72
Table 2: API Testing: Test 1	78
Table 3: API Testing: Test 2	80
Table 4: API Testing: Test 3	81
Table 5: API Testing: Test 4	83
Table 6: API Testing: Test 5	86
Table 7: Unit/System Testing: Test 1.....	87
Table 8: Unit/System Testing: Test 2.....	89
Table 9: Unit/System Testing: Test 3.....	90
Table 10: Unit/system Testing: Test 4	93
Table 11: Unit/System Testing: Test 5.....	95

1. Project Overview

1.1. Purpose

The main purpose of this system is to digitalize the book library by creating a web-based system which enables users to place order, create wish list and browse the library. The system enhances the order processing and provides the users with role-based features for members, admin and staff. It also supports pick up system using a secure claim code. And maintains effective inventory which is managed by the administrators.

1.2. Scope

This project is a full stack web development which will be built using C# .NET Technologies. The application supports user registration, browsing book catalog, filtering, sorting or searching books, bookmark and cart features, placement of orders including discounts, reviewing books and order pickup with claim code verification. There is role-based access to the users where the admin is responsible for inventory management and staff for processing the orders.

1.3. Objectives

- The system should enable members to register, manage bookmarks and make orders with discounts.
- The system should have smooth user experience with search, filter and sort options while browsing the books.
- It must implement secure in-store pick up by using the claim codes which are verified by the staff.
- The system should allow members to review the books that they have previously purchased.
- Admin must be allowed to manage the book records, authors, discounts and announcements.

2. Features and Functionalities

2.1. Login

- **Description:** The user first registers in the system, and after that, they use the valid email and password to login.
- **Functionality:** The project uses ASP .NET Identity for login process. The password is hashed and checked automatically from the framework. After successful login, the user is redirected to respective homepage depending on their roles.
- **Visual Guide:**
 - Step 1:** Enter valid credentials.
 - Step 2:** Click “SIGN IN” button.

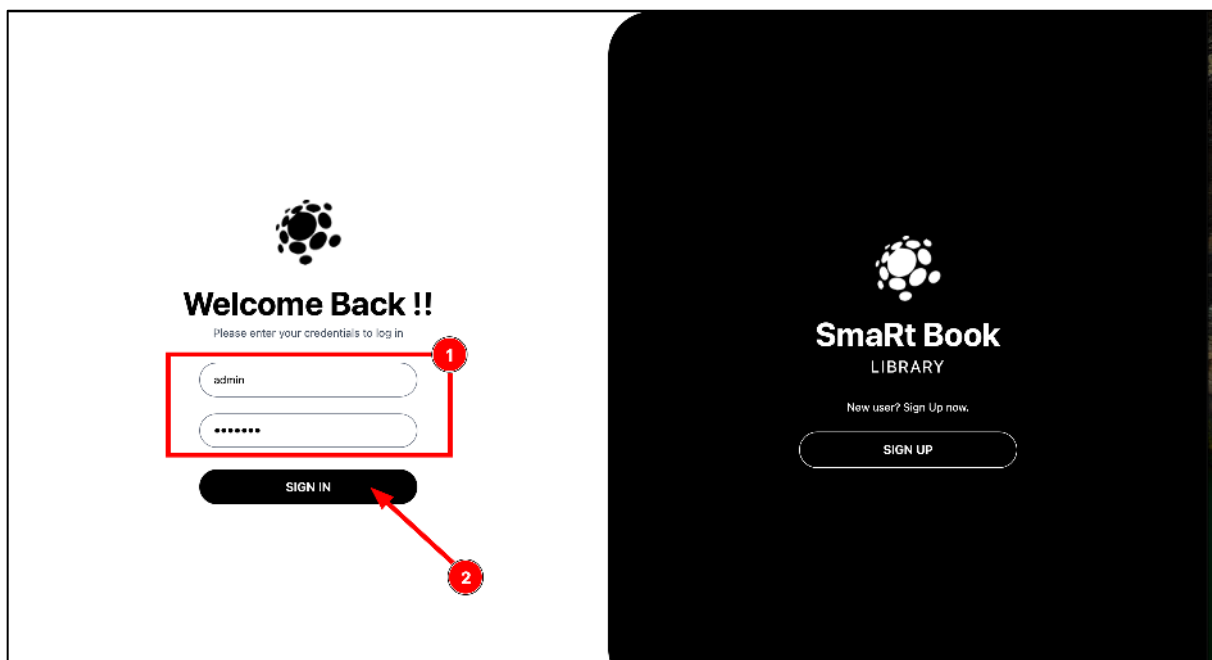


Figure 1: Login Process

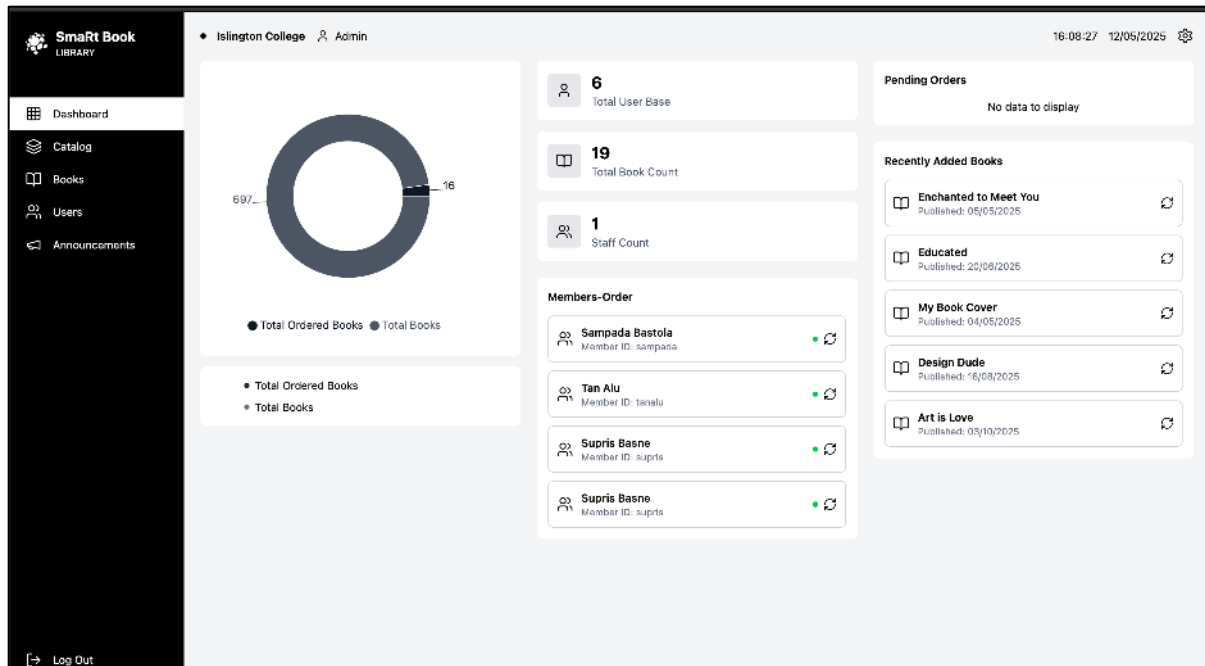


Figure 2: Successful Login and Redirection for Admin.

2.2. Browsing Paginated Book Catalogue

- **Description:** This feature helps the users to navigate through the books catalogue, which improves the load time and usability.
- **Functionality:** The page initially loads 24 books at once. Upon navigating to the next page, it loads another set of 24 books from the database. The pagination logic is applied via the backend.

- **Visual Guide:**

Step 1: Go to “**Categories**” page.

Step 2: Find pagination buttons at the bottom of the page and click to navigate to another page.



Figure 3: Categories Page

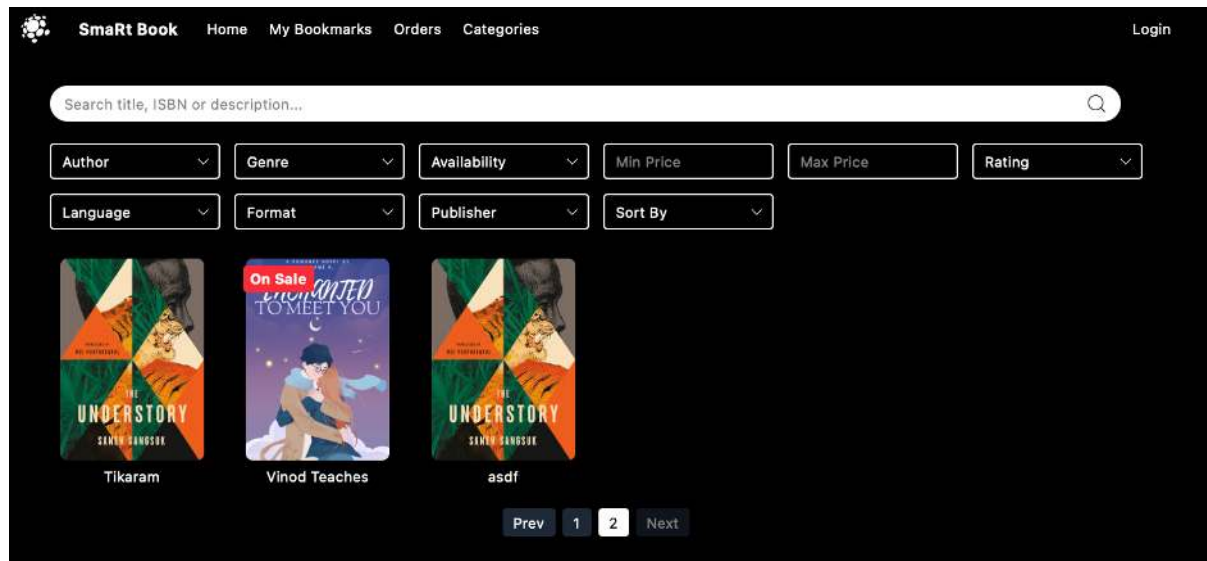


Figure 4: Second Page for Categories

2.3. View Book Details

- **Description:** This feature lets the users view the description of any book of their choice.
- **Functionality:** When the user clicks any book, the frontend passes the book ID of the respective book to the backend, and then the backend responds with the respective book details, and the data is ultimately displayed in the frontend again.

- **Visual Guide:**

Step 1: Go to “Home” page.

Step 2: Select any book of your choice and click on it.

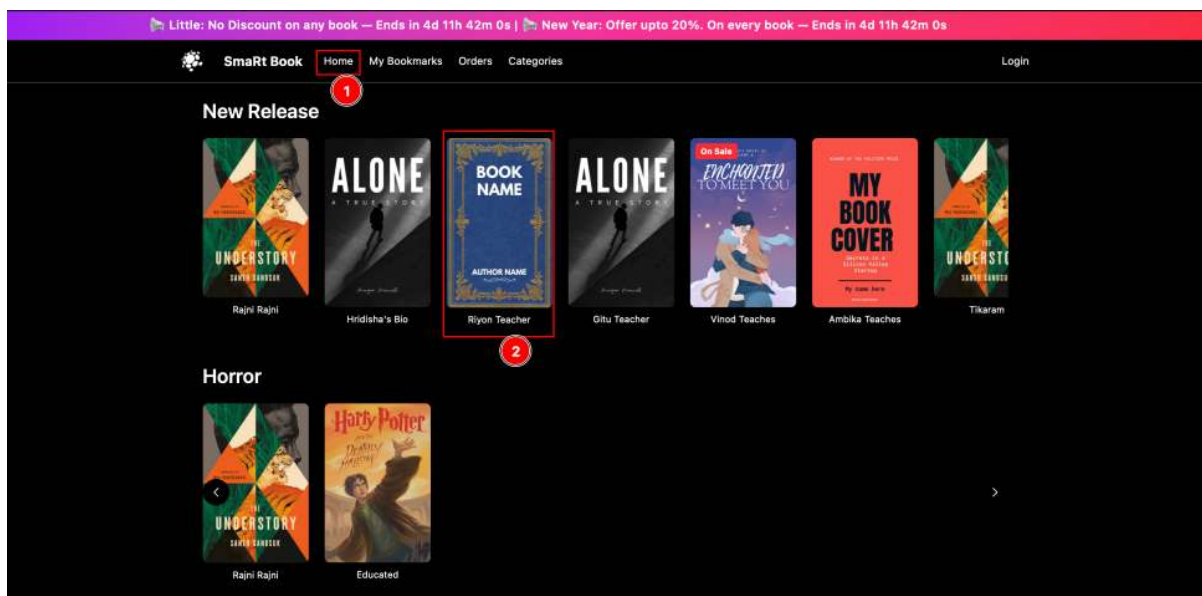


Figure 5: List of Books in Home Page

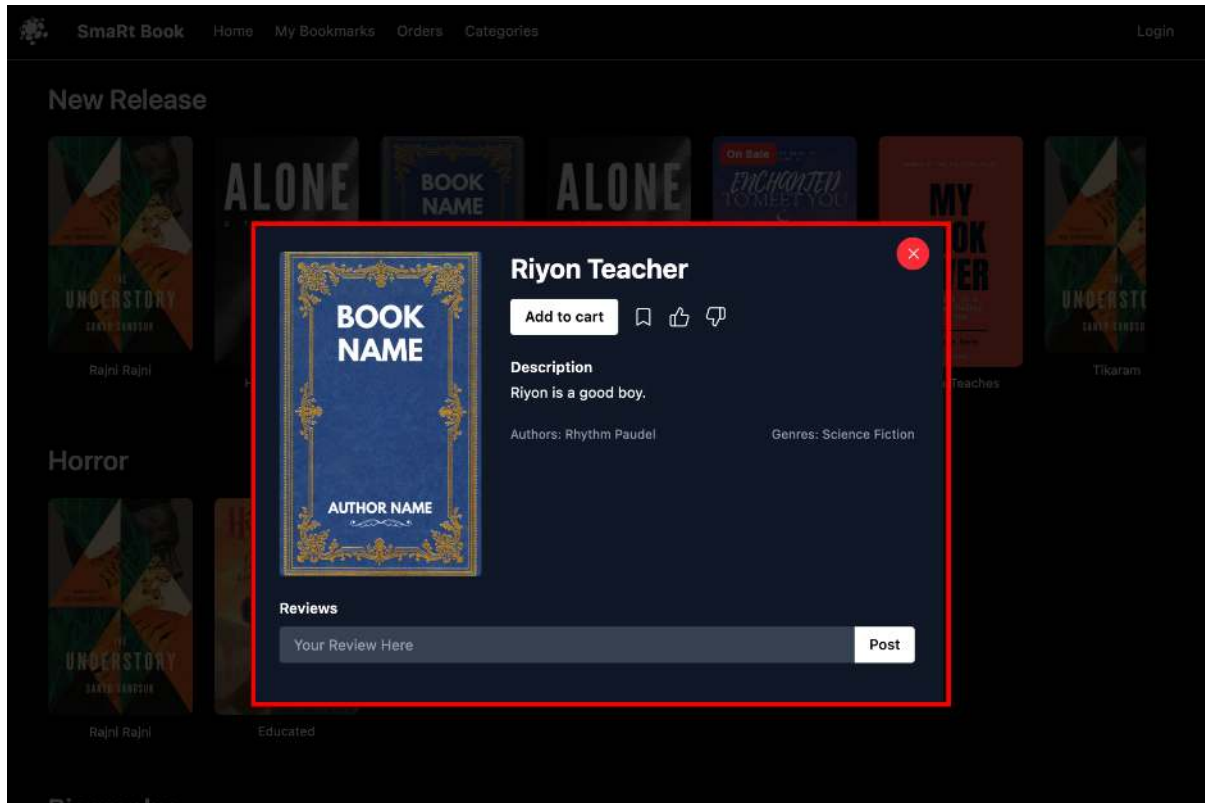


Figure 6: Respective Book Description

2.4. Applying Search, Sort and Filters

- **Description:** This feature lets the users filter, search and sort books according to availability, genre, author, rating, title, ISBN, language, etc.
- **Functionality:** When the user goes to the Categories page, it initially displays five books. Then, the user may choose any filter, search, or sorting of their choice, which passes the value to the backend, and the books get fetched and filtered respectively.
- **Visual Guide:**

Step 1: Go to “Categories” page.

Step 2: Apply any filter, search, or sorting of your choice to find the desired category of books.

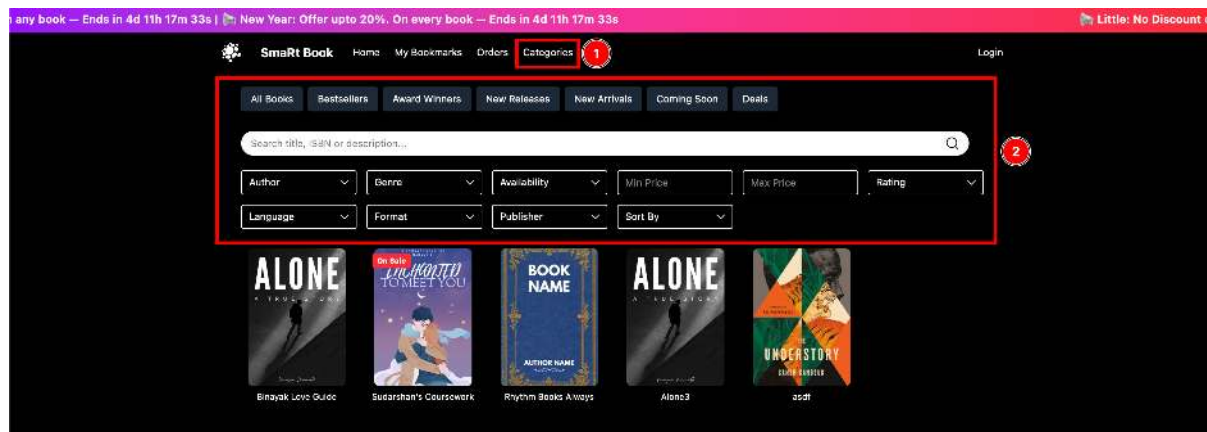


Figure 7: Categories Page and its Filters

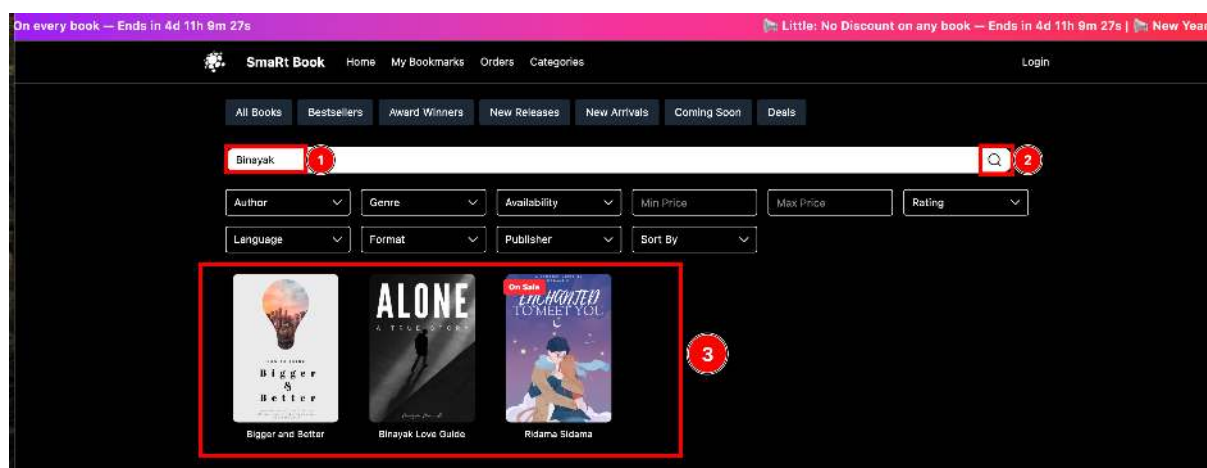


Figure 8: Successful Searching of Book

2.5. Registering Users to Members

- **Description:** This feature lets the user become members of the system to access more exclusive features of the platform, like bookmarking, placing orders, etc.
- **Functionality:** The project uses ASP .NET Identity for register process. The user inputs default fields along with some custom fields for registration. The password is hashed automatically from the framework. After successful validation of input fields, the user is successfully registered into the platform.
- **Visual Guide:**
 - Step 1:** Go to “**Register**” page.
 - Step 2:** Fill up necessary fields and click on “**SIGN UP**” button.
 - Step 3:** Go to Admin Portal to verify the addition of Member.

The image shows a web application interface for 'Smart Book LIBRARY'. On the left, a dark sidebar contains the library logo and a 'SIGN IN' button. The main area is white and features a 'Sign Up' section. A red box labeled '1' encloses the registration form fields: Name (Bikram), Surname (Poudel), Phone Number (9867435261), Email (bikram.poudel@gmail.com), Username (bikram), and Password (masked with dots). A red arrow labeled '2' points to the 'SIGN UP' button below the form.

Figure 9: Registration Form

ID	First Name	Last Name	Phone	Email	Membership ID	Role	Action
tarala	Tari	Ahi	9233212321	rhymauead30@gmail.com	MEM250806TA	User	
haradim	Haris	Alim	9895007232	haradim@gmail.com	MEM250810-A	User	
sompada	Sompada	Bastola	2921923219	sompacabastola0@gmail.com	MEM250810GB	User	
suoris	Suoris	Bond	9822222222	suorisoonet@gmail.com	MEM2508043B	User	
bikram	Bikram	Poudel	9867435201	bikram.poudel@gmail.com	MEM250812BP	User	
samp	Samp	Bast	9811111111	sampbast@gmail.com	MEM2508043B	User	

Figure 10: Successful Addition of Member

2.6. Add to Whitelist (Bookmark)

- Description:** This feature lets the member to bookmark any book of their choice, and the book is displayed in “**My Bookmarks**” tab.
- Functionality:** The platform first checks if the user is a registered member or not. If it is a member, then it checks if the bookmark exists or not in the database with the help of a Boolean value. If it exists, then the bookmark is removed; if not, it adds the bookmark. Then the bookmark is validated via the “**My Bookmarks**” tab.
- Visual Guide:**
 - Step 1:** Select any book and view its detail.
 - Step 2:** Click on Bookmark icon of the book.
 - Step 3:** The bookmark gets added or deleted based on the current bookmark status of the member.

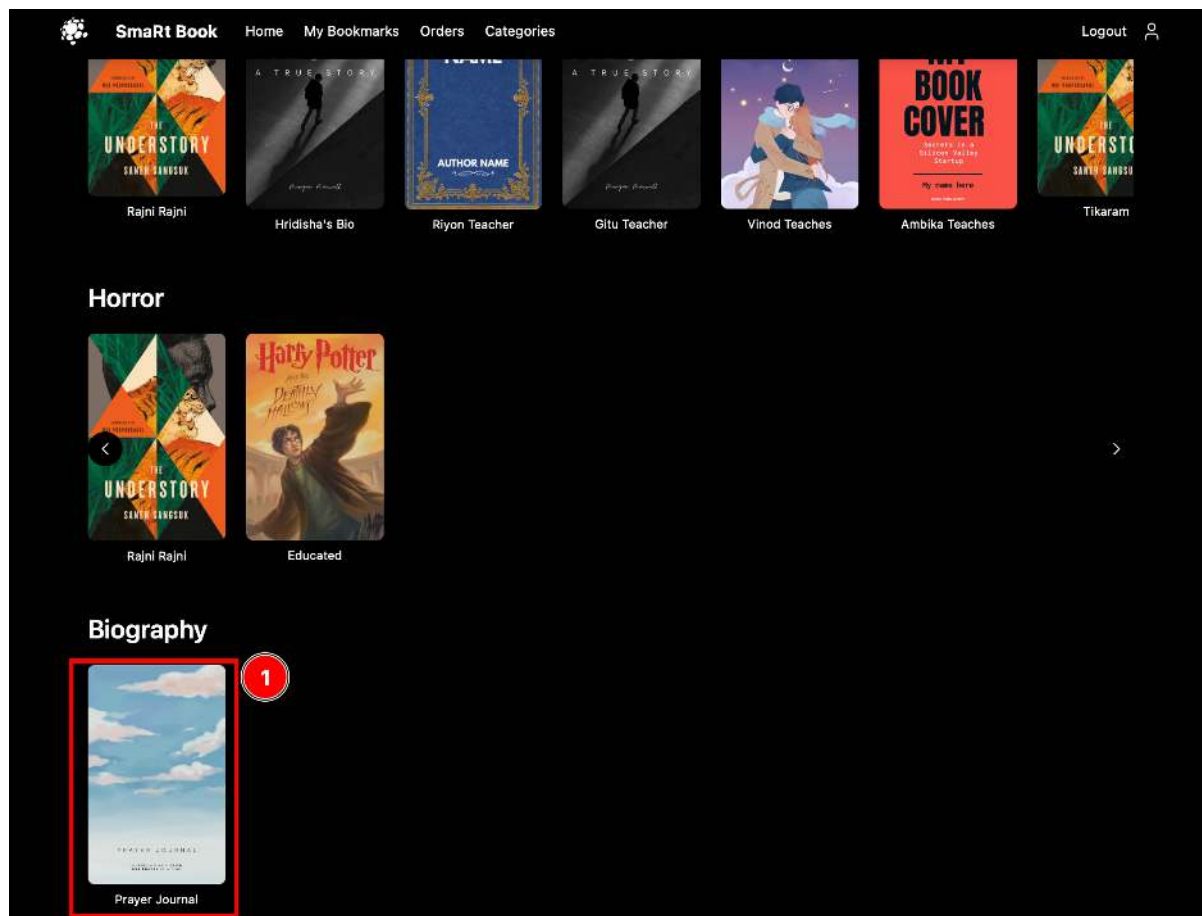


Figure 11: Select Desired Book and Click It

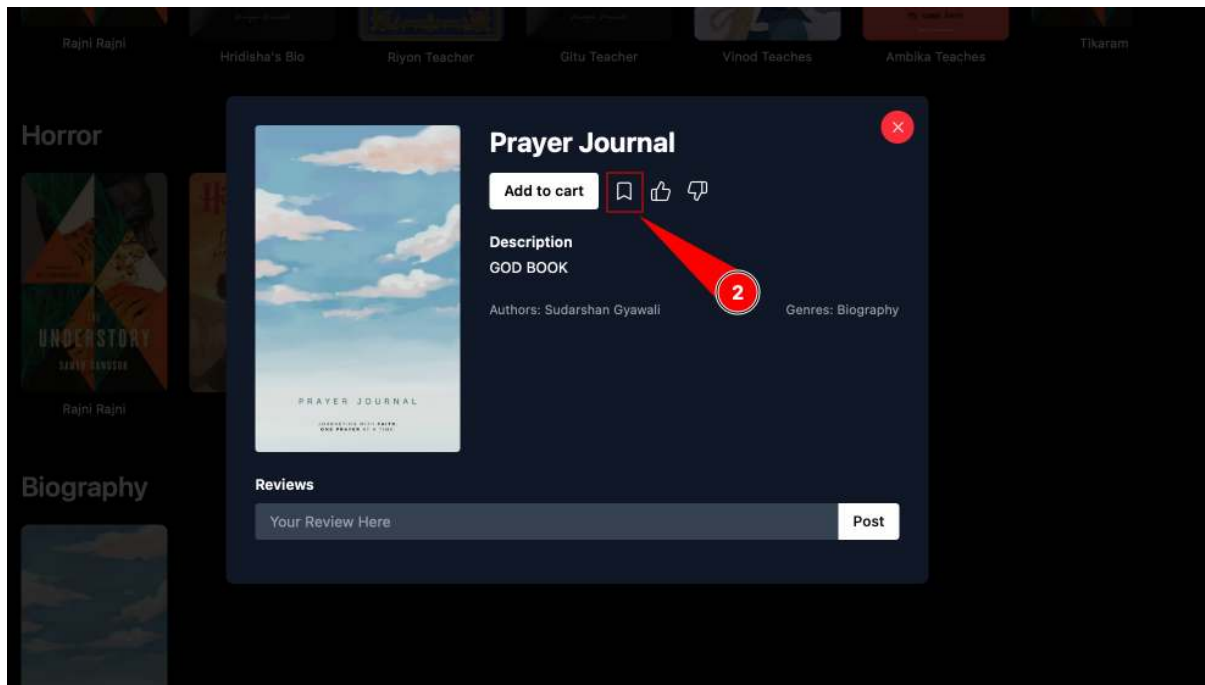


Figure 12: Add Bookmark

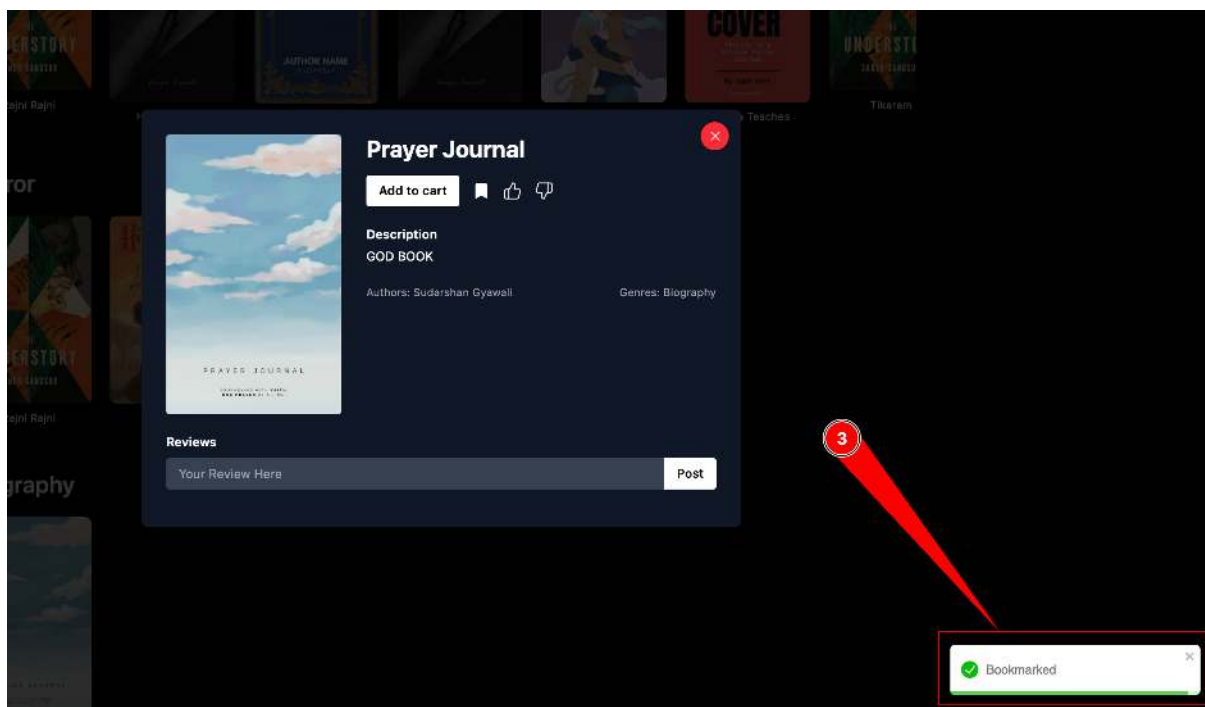


Figure 13: Successfully Bookmarked Message



Figure 14: Addition of Bookmark in My Bookmarks Page

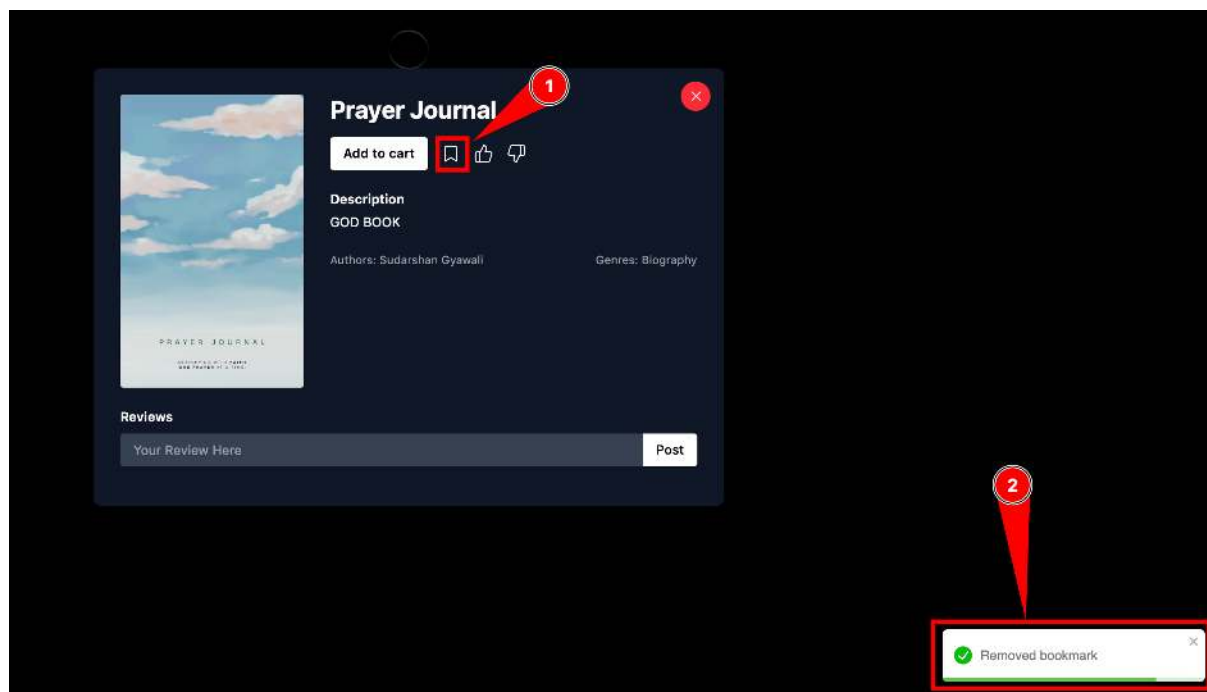


Figure 15: Bookmark Removal

2.7. Add to Cart

- **Description:** This feature lets the member to order any book of their choice, and the book is displayed in “**Orders**” tab.
- **Functionality:** The platform first checks if the user is a registered member or not. If it is a member, then they can add any book of their choice to the cart and the book goes to **Carts** sub-section of **Orders** tab. If the book is already in cart, and the button is changed to “**Remove from cart**”. If the member wants to change the quantity of books, they may do it in **Carts** sub-section.
- **Visual Guide:**
 - Step 1:** Select any book and view its detail.
 - Step 2:** Click on “**Add to cart**” button.
 - Step 3:** The book gets added to the **Carts** sub-section of **Orders** tab.

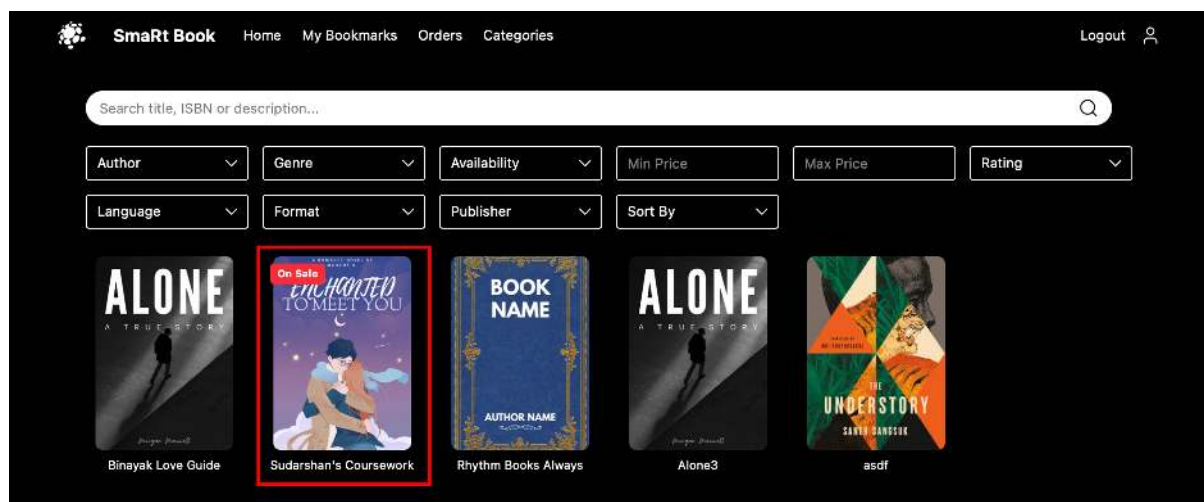


Figure 16: Choosing Book for Add to Cart

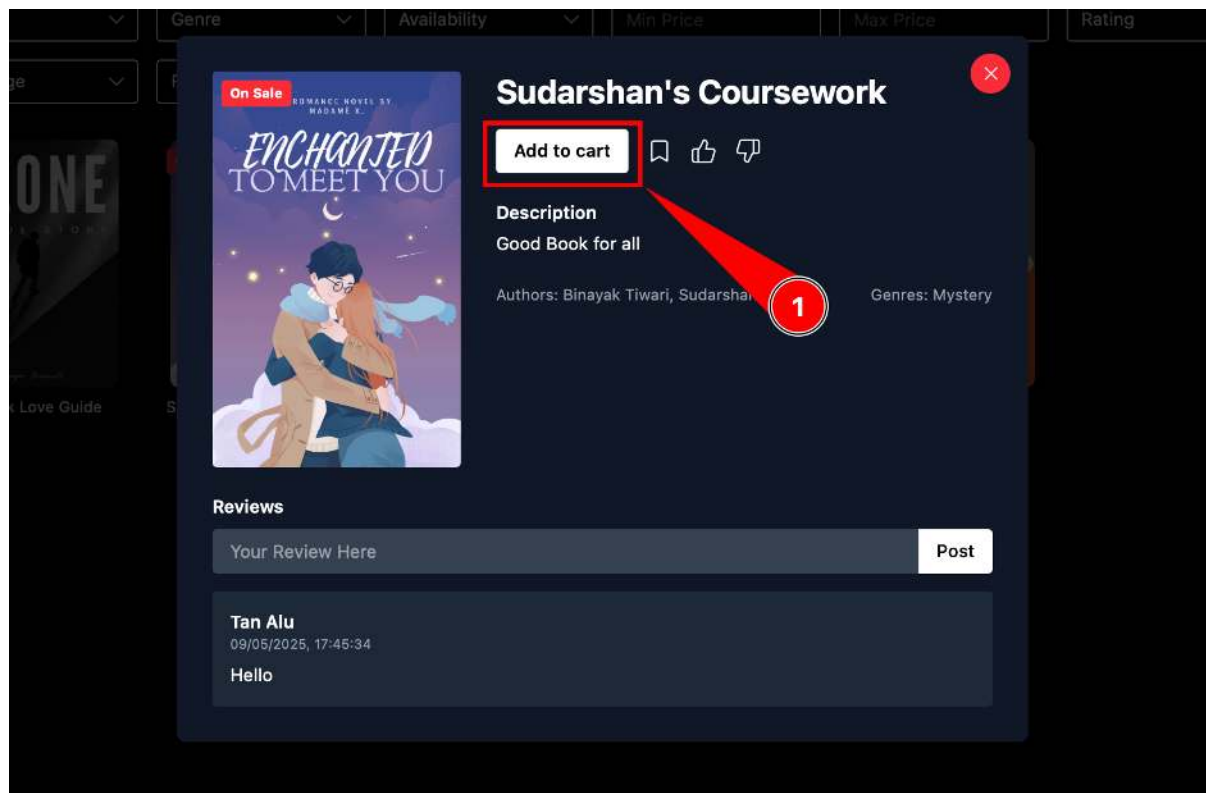


Figure 17: Adding to Cart

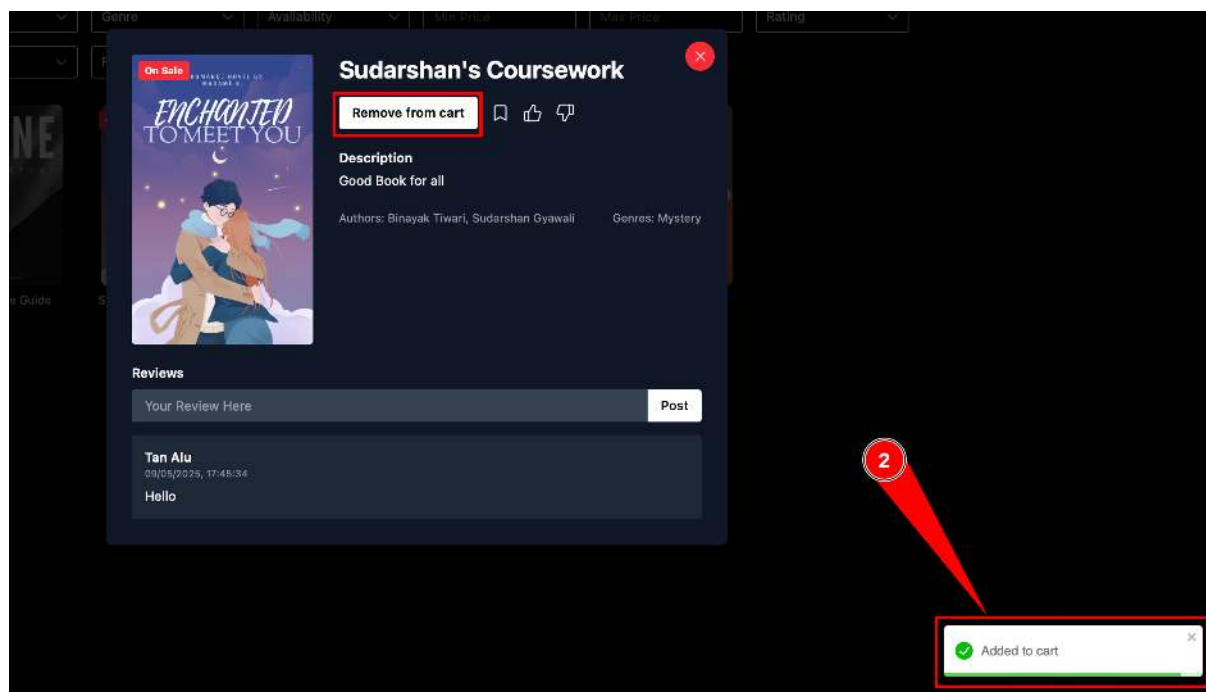


Figure 18: Successfully Added to Cart Message

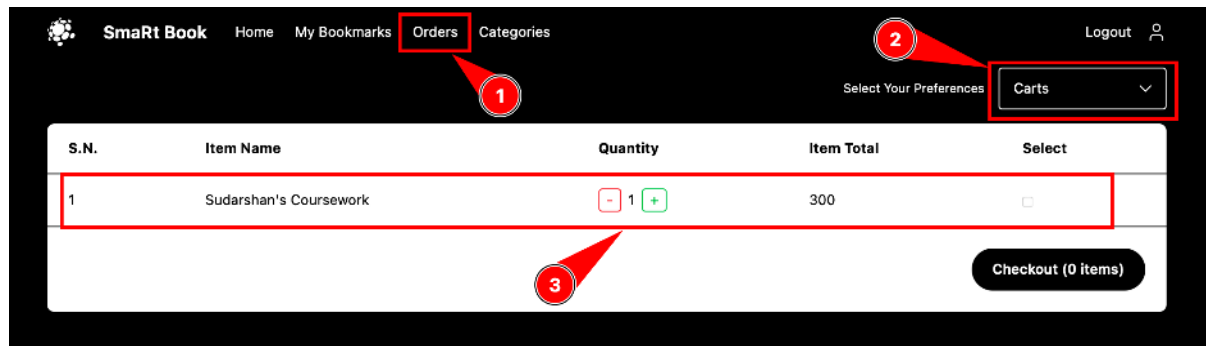


Figure 19: Successful Addition to Carts Sub-Section of Orders Tab

2.8. Place Cancellable Order

- **Description:** This feature lets the member to place multiple cancellable orders.
- **Functionality:** The member must add at least one book to their cart to place an order. Then, they must select the items from cart that need to be included in an order and after adding all desired items, they click “**Checkout**” button. The order is then marked as pending initially. The user can cancel the order if they want. The pending order then gets fetched to admin portal, where the admin may approve or cancel the order, and the status is highlighted accordingly in the database.
- **Visual Guide:**

Step 1: Select all desired items from the cart and click “**Checkout**” button.

Step 2: Filter orders based on their status.

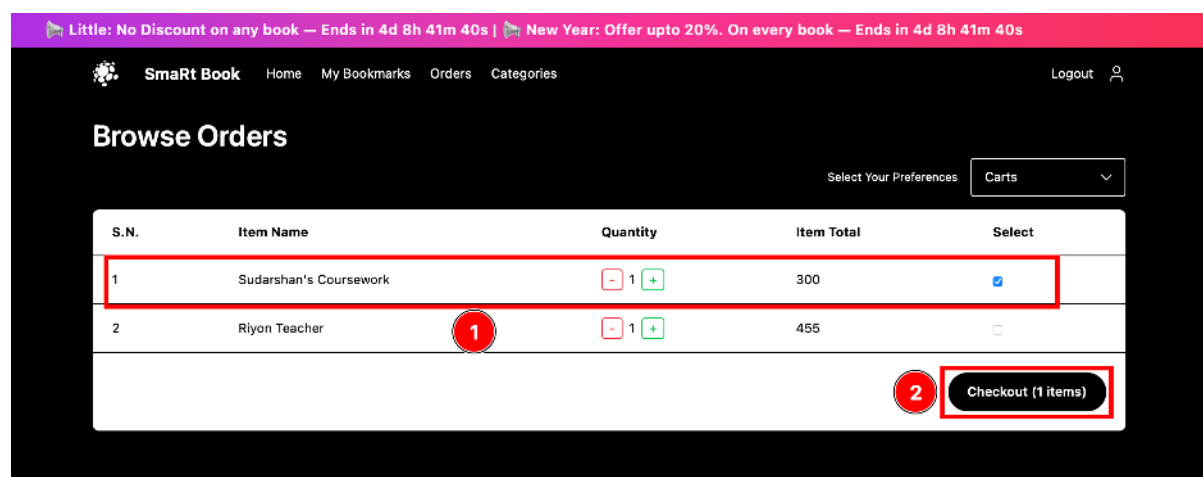


Figure 20: Checkout from Carts

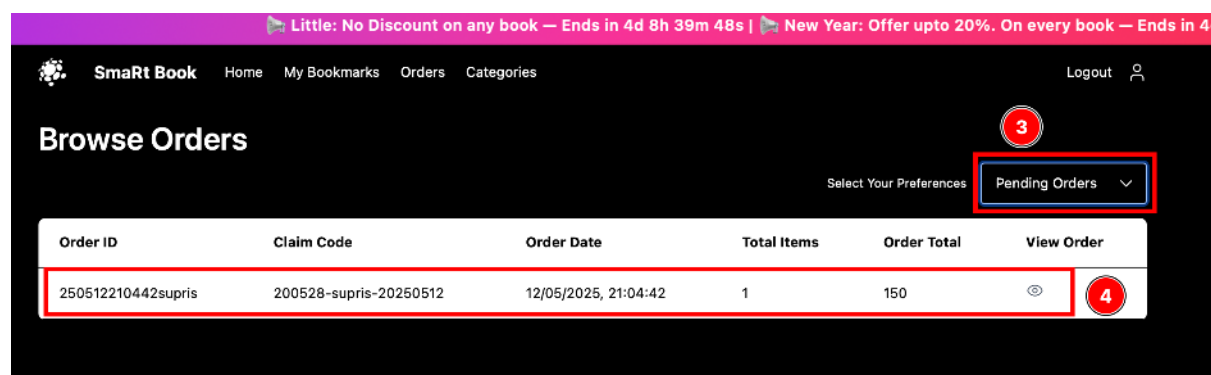


Figure 21: Order in Pending Tab of Member

The screenshot shows the 'Smart Book LIBRARY' admin interface. On the left is a dark sidebar with a menu containing 'Dashboard', 'Catalog', 'Books', 'Users', and 'Announcements'. The 'Catalog' item is highlighted with a red box and a red circle containing the number 1. The main content area has three tabs: 'Completed', 'Pending', and 'Cancelled'. The 'Pending' tab is selected and highlighted with a red box and a red circle containing the number 2. Below the tabs is a table with columns: 'Username', 'Full Name', 'Order ID', 'Order Total', and 'Action'. A single row is present in the table, representing an order, and is highlighted with a red box and a red circle containing the number 3. The row contains the following data: 'supri' for Username, 'supri basnet' for Full Name, '2309122164426supri' for Order ID, and '159' for Order Total. The 'Action' column contains a book icon.

Username	Full Name	Order ID	Order Total	Action
supri	supri basnet	2309122164426supri	159	

Figure 22: Order in Pending Tab of Admin

2.9. Review Purchased Books

- **Description:** This feature lets the member to place a review after purchasing the book.
- **Functionality:** The system first checks if the member has bought that book or not before leaving a review. If purchased, they may leave a review. The review given is then highlighted in the frontend and saved in the database. The member may also leave a rating for the book. The rating is appended in the same record where that book's review is stored in the database. The rating and review can be appended interchangeably depending on which one was done first.
- **Visual Guide:**
 - Step 1:** Open any purchased book.
 - Step 2:** Write the desired review text and click “**Post**” button. You may also like or dislike the book.

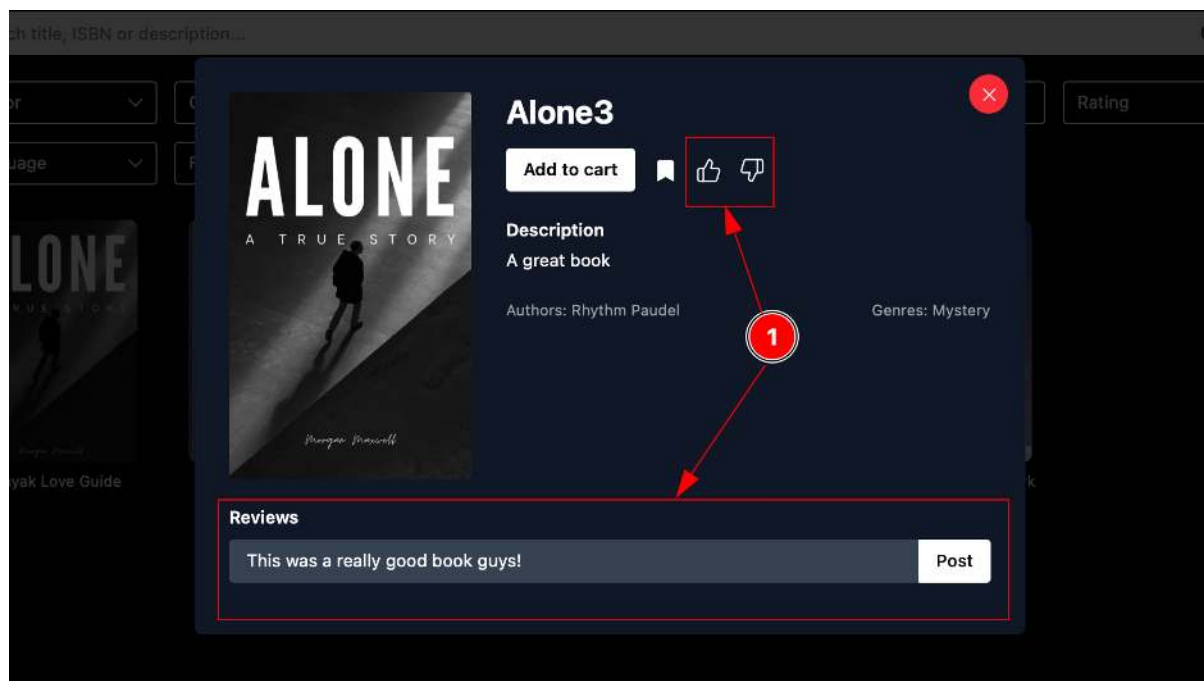


Figure 23: Before Leaving Review and Rating

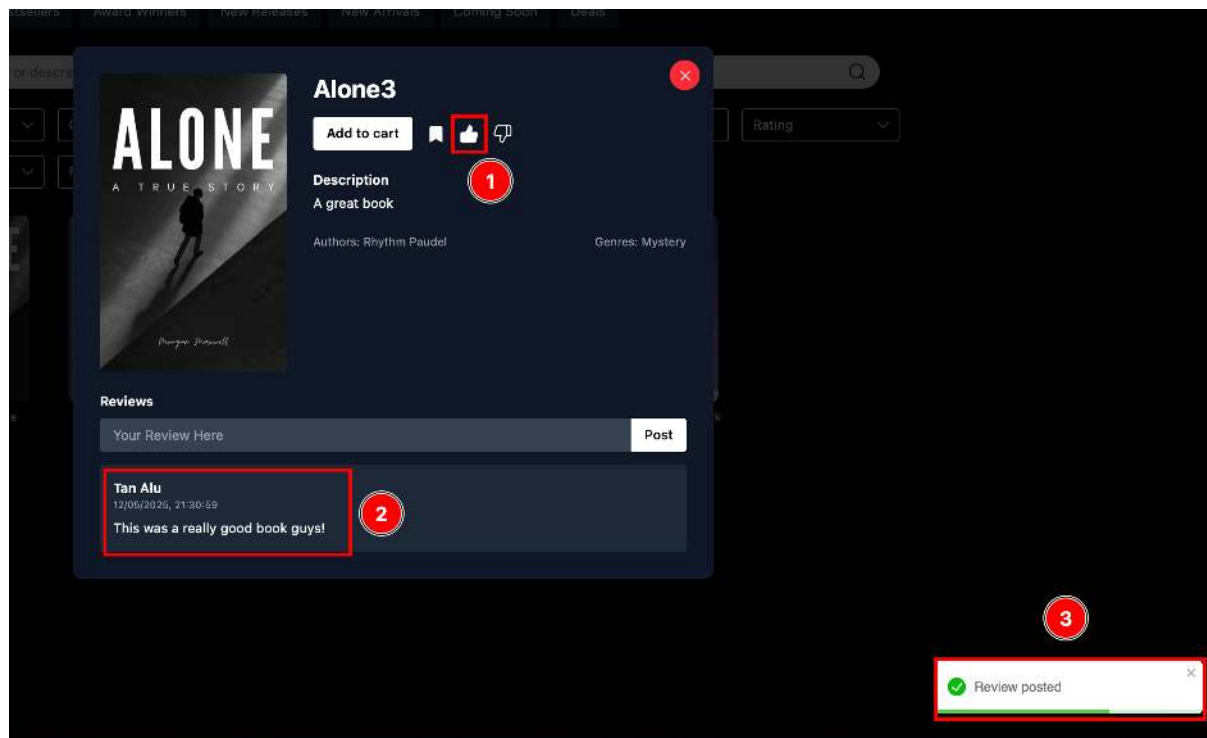


Figure 24: Successfully Left Review and Rating

2.10. Discount System

2.10.1. 5% Bulk Order Discount on Order of 5+ Books

- **Description:** This feature lets the member to get discount on ordering unique 5+ books at once.
- **Functionality:** First, it is checked that if ordered books are 5 or more unique books. If yes, then 5% discount is applied on the total amount of the purchased quantity of books after applying individual book discounts.
- **Visual Guide:**

Step 1: Place an order of 5 or more unique books.

Step 2: Go to Pending Orders and view order details.

Step 3: The 5% bulk order discount along with individual book discount can be seen.

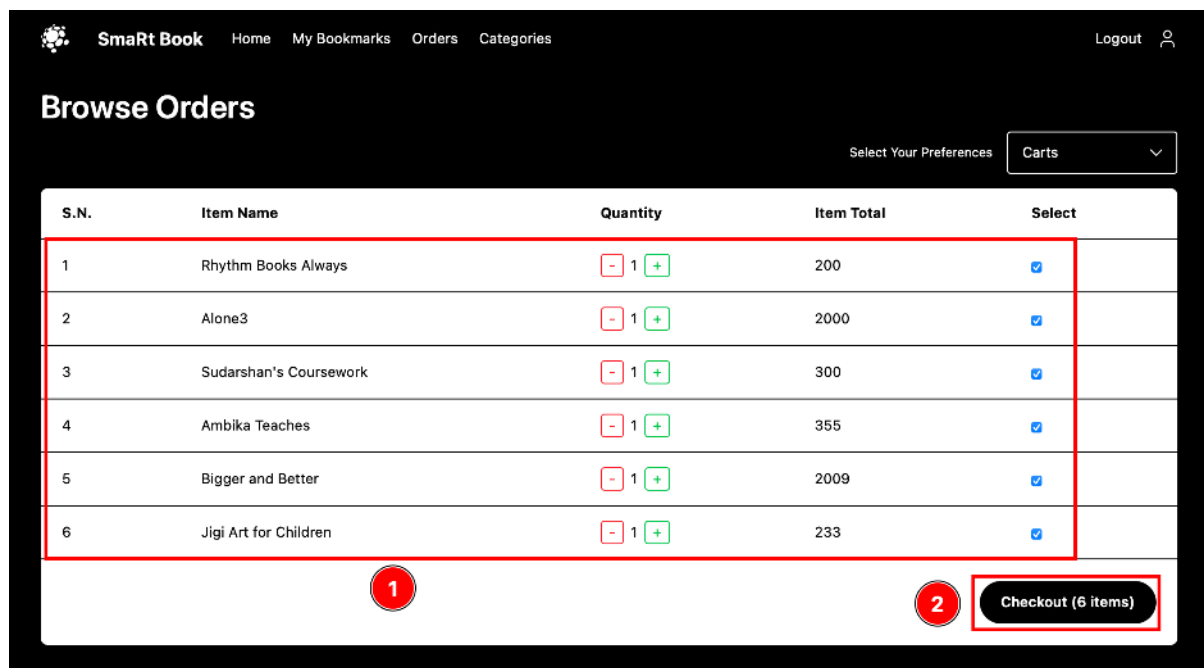
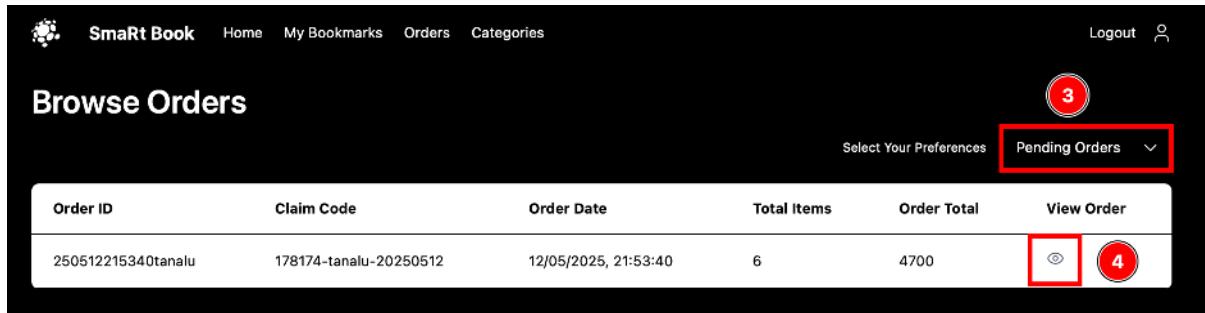


Figure 25: Cart ready to checkout with 6 unique books.



Smart Book Home My Bookmarks Orders Categories Logout

Browse Orders

Select Your Preferences **Pending Orders** ▼

Order ID	Claim Code	Order Date	Total Items	Order Total	View Order
250512215340tanalu	178174-tanalu-20250512	12/05/2025, 21:53:40	6	4700	

Figure 26: View Pending Orders Tab

Order No. 250512215340tanalu

Claim Code: 178174-tanalu-20250512

Status: Pending

[Cancel Order](#)

S.N.	Item Name	Quantity	Unit Price	Line Total
1	Sudarshan's Coursework	1	300	300
2	Bigger and Better	1	2009	2009
3	Jigi Art for Children	1	233	233
4	Ambika Teaches	1	355	355
5	Rhythm Books Always	1	200	200
6	Alone3	1	2000	2000
Order Total				5097
Discount				397
After Discount				4700

Figure 27: Discount Details

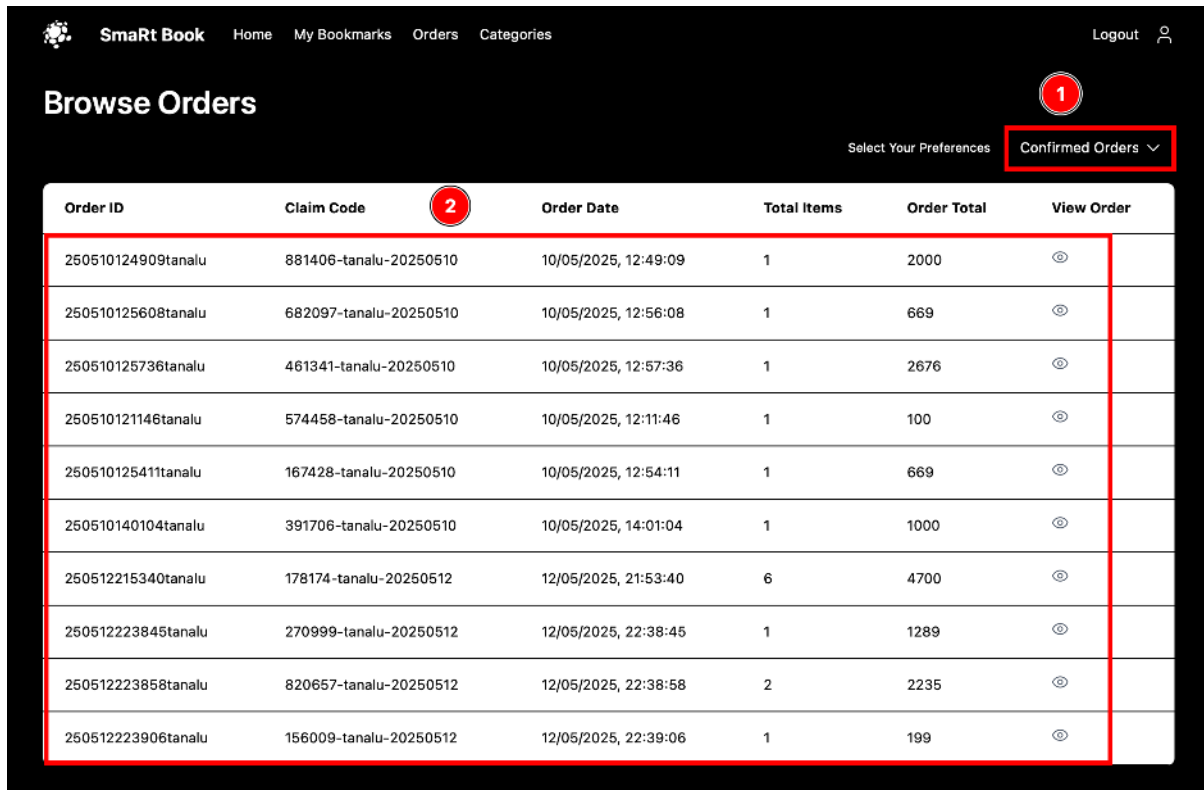
2.10.2. 10% Stackable Discount After 10 Successful Orders

- **Description:** This feature lets the member to get discount 11th order after the processing of 10 successful orders.
- **Functionality:** First, it is checked that if the member has completed the processing of 10 successful orders. If yes, then the future orders will receive a 10% stackable discount. The sum is always rounded up.
- **Visual Guide:**

Step 1: Complete the processing of 10 successful orders.

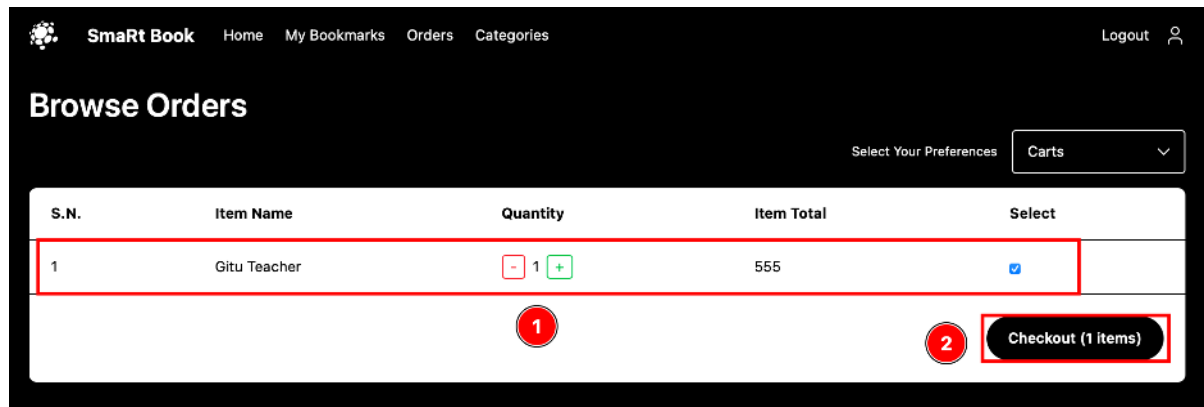
Step 2: Process a new order.

Step 3: The 10% stackable discount along with individual book discount can be seen.



Order ID	Claim Code	Order Date	Total Items	Order Total	View Order
250510124909tanalu	881406-tanalu-20250510	10/05/2025, 12:49:09	1	2000	
250510125608tanalu	682097-tanalu-20250510	10/05/2025, 12:56:08	1	669	
250510125736tanalu	461341-tanalu-20250510	10/05/2025, 12:57:36	1	2676	
250510121146tanalu	574458-tanalu-20250510	10/05/2025, 12:11:46	1	100	
250510125411tanalu	167428-tanalu-20250510	10/05/2025, 12:54:11	1	669	
250510140104tanalu	391706-tanalu-20250510	10/05/2025, 14:01:04	1	1000	
250512215340tanalu	178174-tanalu-20250512	12/05/2025, 21:53:40	6	4700	
250512223845tanalu	270999-tanalu-20250512	12/05/2025, 22:38:45	1	1289	
250512223858tanalu	820657-tanalu-20250512	12/05/2025, 22:38:58	2	2235	
250512223906tanalu	156009-tanalu-20250512	12/05/2025, 22:39:06	1	199	

Figure 28: Ten Confirmed Orders

Figure 29: Placing 11th OrderFigure 30: Viewing 11th Order

Order #250512224825tanalu Details

Order No. 250512224825tanalu

Claim Code: 980109-tanalu-20250512

Status: Pending

Cancel Order

S.N.	Item Name	Quantity	Unit Price	Line Total
1	Gitu Teacher	1	555	555
Order Total				555
Discount				55
After Discount				500

Figure 31: 10% Stackable Discount After 10 Successful Orders

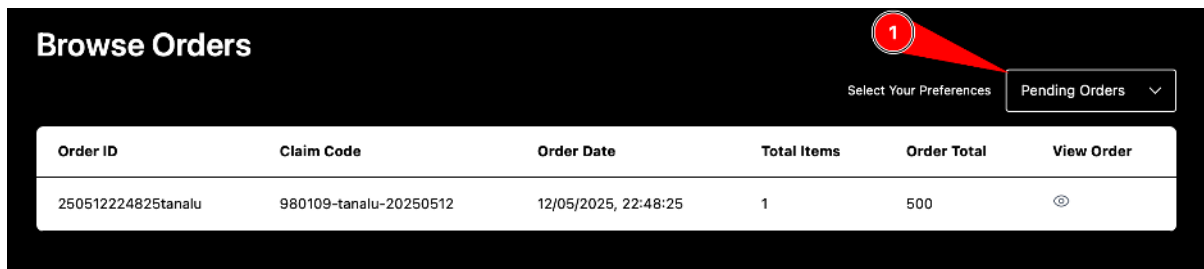
2.11. Receive Claim Code and Order Details in Registered Email

- **Description:** This feature lets the member to get claim code and order details in their respective emails.
- **Functionality:** Members must select the items from cart that need to be included in an order and after adding all desired items, they click “**Checkout**” button. Then they receive their claim code and order details in their respective emails.

- **Visual Guide:**

Step 1: Select all desired items from the cart and click “**Checkout**” button.

Step 2: Check the mail to get claim code and order details.



Order ID	Claim Code	Order Date	Total Items	Order Total	View Order
250512224825tanalu	980109-tanalu-20250512	12/05/2025, 22:48:25	1	500	

Figure 32: Order Placed Successfully

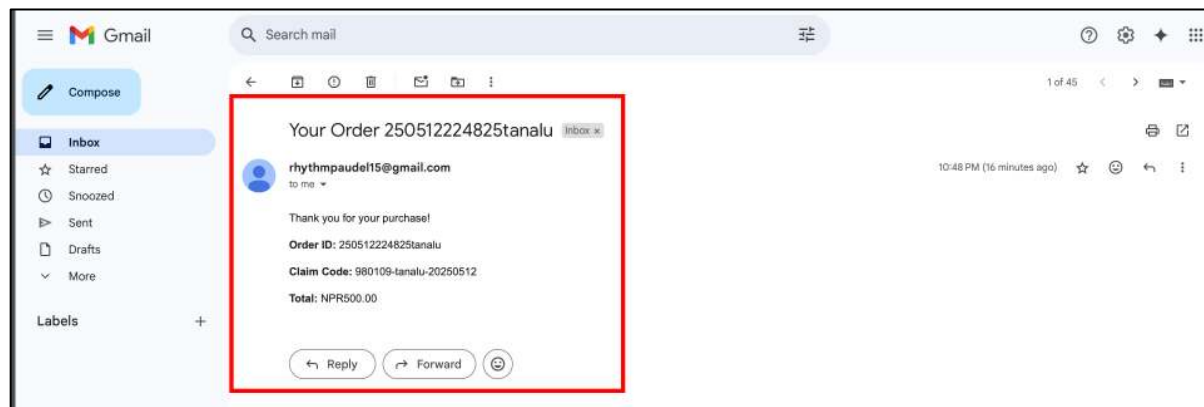


Figure 33: Code and Order Details in Registered Email

2.12. Staff Order Fulfilment Via Member Claim Code

- **Description:** This feature lets the staff to fulfil the order via processing of member claim code.
- **Functionality:** Members after placing a successful order, visit the library physically and provide the claim code from the email to the staff for marking the order as completed.
- **Visual Guide:**

Step 1: Place a successful order and get the claim code from mail.

Step 2: Staff enters the claim code and approves the order.

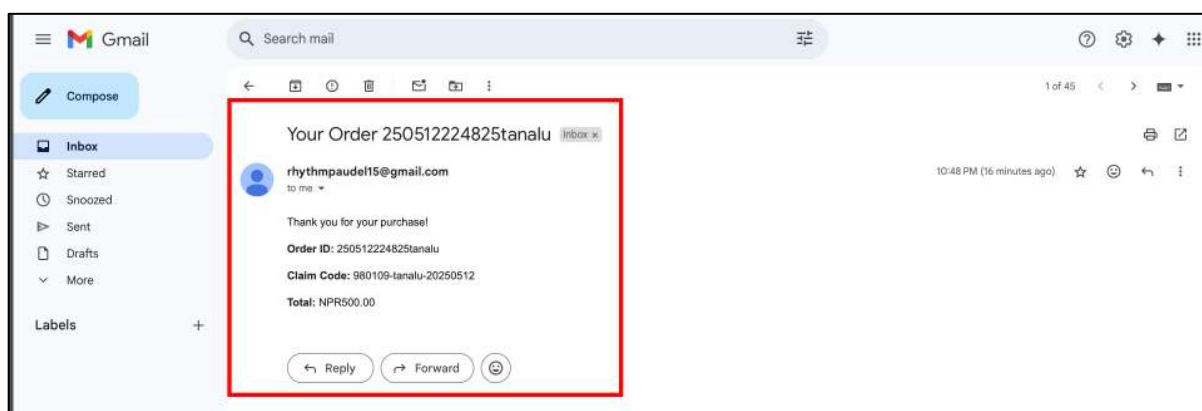


Figure 34: Claim Code in Mail for Staff Order Fulfilment

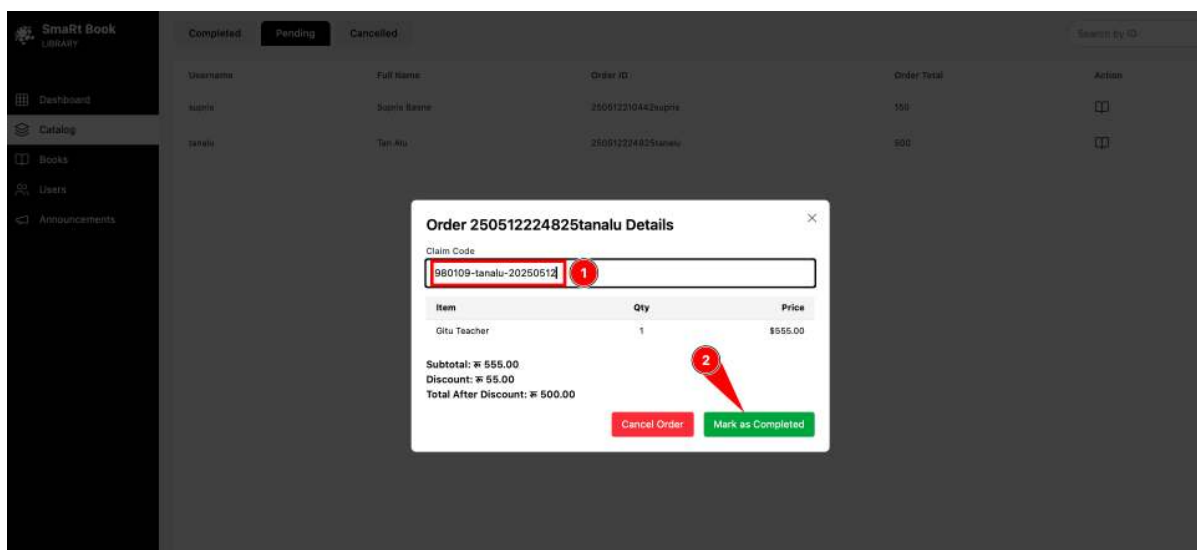
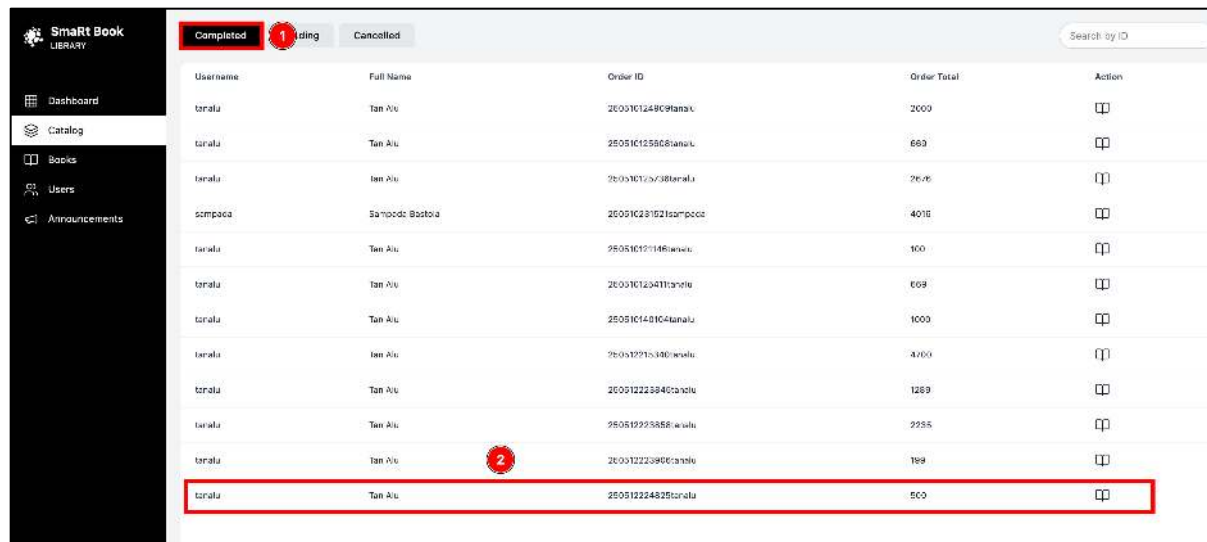


Figure 35: Staff Order Fulfilment via Member Claim Code



Smart Book LIBRARY

Dashboard
Catalog
Books
Users
Announcements

Completed Pending Cancelled

Search by ID

Username	Full Name	Order ID	Order Total	Action
tanalu	Tan Ali	200510124809tanalu	2000	
tanalu	Tan Ali	250510125068tanalu	660	
tanalu	Tan Ali	280510125380tanalu	2606	
sonpaca	Sonpode Bostola	200510125102sonpaca	4016	
tanalu	Tan Ali	280510125146tanalu	100	
tanalu	Tan Ali	200510125411tanalu	609	
tanalu	Tan Ali	250510140104tanalu	1000	
tanalu	Tan Ali	2805122915300tanalu	4000	
tanalu	Tan Ali	200512223940tanalu	1269	
tanalu	Tan Ali	280512223958tanalu	2256	
tanalu	Tan Ali	200512223960tanalu	189	
tanalu	Tan Ali	250512224825tanalu	500	

Figure 36: Successful Staff Order Fulfilment via Member Claim Code

2.13. Admin Book Catalogue and Inventory Management (CRUD)

- **Description:** This feature lets the admin to manage book catalogue.
- **Functionality:** Admin can perform CRUD in book catalogue via this feature.

- **Visual Guide:**

Step 1: Add a book.

Step 2: View the added book.

Step 3: Edit the added book.

Step 4: Delete the added book.

ID	Title	Genre	Language	In Stock	Price	Action
FGO23D094B01156A41	Art is Love	Art	Russian	221	\$236b	Edit Delete Add
PNQ25C509C816C3B1N	Binu Binu	Fantasy	Arabic	4	\$777	Edit Delete Add
JGLQ9865086R00C7JH3	Jigi Food Guide	Cookbook	Hindi	22	\$9999	Edit Delete Add
TVQ2605080900050DE5	Design Dude	School	Russian	100	\$600	Edit Delete Add
DMQ2505121050C04ME	Amika Teacher	Thriller	French	33	\$555	Edit Delete Add
TK...2306121363208RY	Riyon Teacher	Science Fiction	French	64	\$45b	Edit Delete Add
LEQ200512105025RAJ	Rahy Rajni	Horror	German	25	\$209	Edit Delete Add
N...Q9865086R1329DAM	Dama's Swange	Religion	German	24	\$400b	Edit Delete Add
PUB2306090798333BN	Binayak Love Guide	Romance	English	19	\$100b	Edit Delete Add
J5W25050908229HID	Ridania Scama	Children's	Arabic	7	\$09b	Edit Delete Add
ULQ26051210504361R1	Hidishana Ito	Travel	French	64	\$100b	Edit Delete Add
WQ420008090814BEDU	Edusted	Horror	German	8	\$800b	Edit Delete Add
IFU250512164708EAR	Baba's Gyon	Historical Fiction	French	47	\$19b	Edit Delete Add
4...92500121048971IK	Tinaram	Self-help	Russian	12	\$199	Edit Delete Add
CCK250509081012HAR	Harry Potter	Mystery	Portuguese	32	\$533b	Edit Delete Add
90XR260512137175H1Q	Emperated in Mweel You	Romance	French	26	\$10b	Edit Delete Add

Figure 37: Add Book Button

Smart Book LIBRARY

Book Management

CRUD - Creating a Book

Author:

Select Author:

Selected Authors:

Submitter's Copyright:

Genre:

ISBN:

Test:

Date:

Language:

Audiobook:

Collector's:

Award Winner: ☒ Bestseller ☐ Coming Soon

Cancel Add

Table Data:

ID	Title	Genre	Language	In Stock	Price	Action
190230009001195441	Art is Love	Historical Fiction	French	221	\$2445	
7N0200009008700391N	Shu Shu	Self-help	Russian	20	\$777	
10102000900000000000	Light Food Guide	Mystery	Portuguese	22	\$2222	
TVQ20000900000000000	Design Guide	Thriller	English	100	\$500	
CMG20001210000000000	Artistic Touches	Business	German	33	\$555	
1KJ20000100000000000	Olivia Teacher	Autobiography	Arabic	44	\$555	
LEQ20001210000000000	Rainy Rain	Drama	Spanish	25	\$299	
7N0200009008700391N	Dama's Revenge	Mystery	Portuguese	24	\$4000	
7UQ20000900000000000	Prayer Journal	Biography	German	18	\$1000	
4BW20000900000000000	Rhythm Books Analysis	Mystery	English	7	\$999	
1HJ20001210000000000	Alone3	Mystery	English	44	\$1888	
WQ49000000000000000	Enchanted to Meet You	Romance	French	8	\$3000	
FU20000121000000000	My Book Cover	Mystery	French	47	\$299	
4LQ20000121000000000	CRUD - Creating a Book	Thriller	Spanish	12	\$199	
QCK20000000000000000	CRUD - Creating a Book	Thriller	Spanish	25	\$555	
WQ49000000000000000	CRUD - Creating a Book	Thriller	Spanish	25	\$100	

Figure 38: Filling Book Details and Adding the Book

Smart Book LIBRARY

Book Management

Table Data:

ID	Title	Genre	Language	In Stock	Price	Action
1KJ20000100000000000	Baba's Gym	Historical Fiction	French	47	\$248	
4LE20001210000000000	Tikaram	Self-help	Russian	12	\$198	
QCK20000900000000000	Harry Potter	Mystery	Portuguese	32	\$2323	
WXK200007162120000	Enchanted to Meet You	Romance	French	25	\$100	
QAE20000900000000000	Prayer Journal	Biography	German	37	\$385	
1Y12000000000000000	Submitter's Copyright	Mystery	English	0	\$388	
8QK20000000000000000	Dama Lama Same	Business	French	20	\$1308	
6752000000000000000	acer	Thriller	English	12	\$12	
V3H200007162240MYB	My Book Cover	Mystery	French	18	\$1201	
FOE20001210000000000	Vince Touches	Business	German	25	\$177	
4Q02000191000000000	Olivia Teacher	Autobiography	Arabic	88	\$555	
8A12000000000000000	Rigger and Heller	Historical Fiction	Chinese	37	\$1308	
D9K20000900000000000	Jagi Art for Children	Drama	Spanish	11	\$233	
E0K20000900000000000	Rhythm Books Analysis	Mystery	English	28	\$220	
R0K20000700000000000	Alone3	Mystery	English	5	\$2300	
FRG200012173820CRU	CRUD - Creating a Book	Thriller	Spanish	1	\$1111	

Figure 39: Successful Addition of Book

Smart Book LIBRARY

Book Management

Table Data:

ID	Title	Genre	Language	In Stock	Price	Action
FRG200012173820CRU	CRUD - Creating a Book	Thriller	Spanish	1	\$1111	

Figure 40: Edit Book Details Button

Update Book

BOOK NAME

Change Cover

CRUD - Updating a Book

Select Author

Rhythm Paudel

Sudershan Gyawali

Thriller

1 1111

111111112345

Test

Test

09/05/2025

Spanish

Audiobook

Figure 41: Updating Book Name

PRG250512173820CRU	CRUD - Updating a Book	Thriller	Spanish	1	\$1111	  
--------------------	------------------------	----------	---------	---	--------	---

Figure 42: Book Updated Successfully

PRG250512173820CRU	CRUD - Updating a Book	Thriller	Spanish	1	\$1111	  
--------------------	------------------------	----------	---------	---	--------	---

Figure 43: Delete Book Button

Delete Confirmation

"Delete book PRG250512173820CRU?"

CONFIRM

Figure 44: Confirm Deletion Button

2.14. Admin Timed Discounts with Optional “On Sale” Flag

- **Description:** This feature lets the admin to add timed discounts to individual books.
- **Functionality:** Admin can add timed discounts to individual books. The books will show a tag “On Sale” after its addition.
- **Visual Guide:**
Step 1: Add a book with discount.
Step 2: View the added book with its On Sale tag.

Book Discount Test

Authors

Select Author

Selected Authors:

Sudarshan Gyawali

Romance

44

455

3434343456756

Test for On Sale

1

DODO

15/05/2025

German

PDF

2

Author's

25

08/05/2025

30/05/2025

☒ Award Winner ☒ Bestseller ☒ Coming Soon

Cancel

3

Add

Figure 45: Adding a New Book with Discount

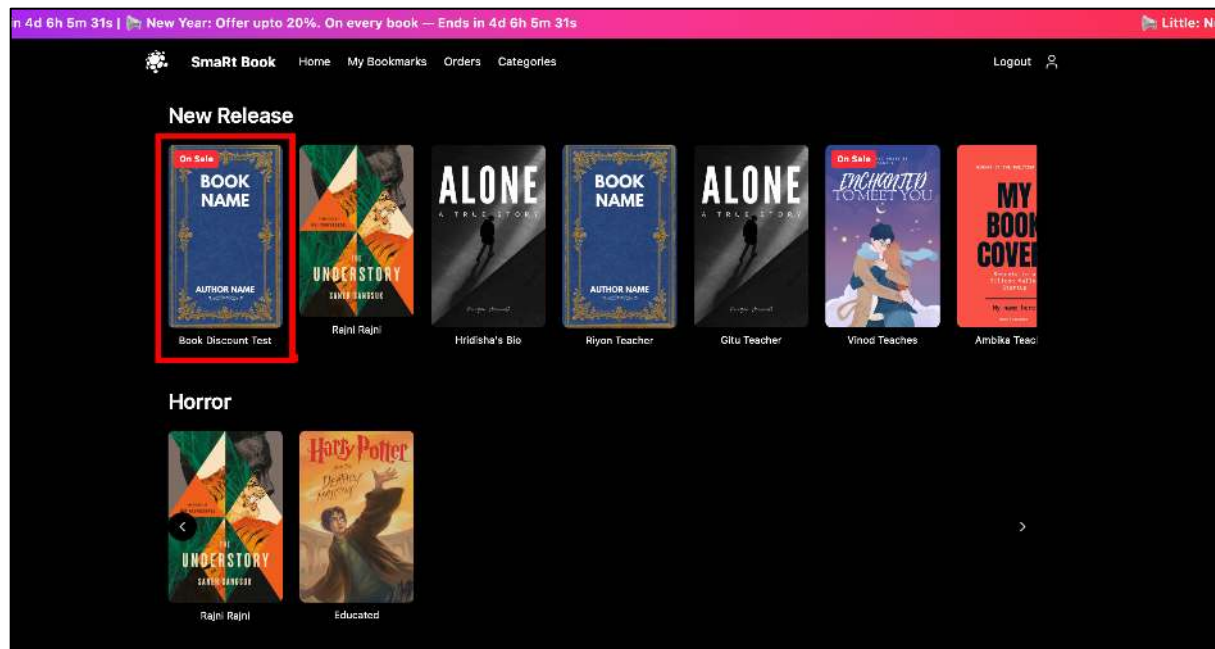


Figure 46: Successful Display of Book in User Side with On Sale Tag

2.15. Admin Timed Banner Announcement

- **Description:** This feature lets the admin to add timed banner announcements.
- **Functionality:** Admin can add timed banner announcements for the users to see.

The announcement gets saved in database after addition and is fetched to platform user's each page.

- **Visual Guide:**

Step 1: Add a timed banner announcement.

Step 2: View the added timed banner announcement in user side pages.

Announcement Management

New Announcement

Title: New Year Sale

Message: 50% OFF on all books for this year!!!

Start Date: 15/05/2025

End Date: 31/05/2025

+ Save

ID	Title	Message	Start	End	Action
No announcements found					

Figure 47: Adding a Timed Banner Announcement

Announcement Management

+ Add

Search: Announcement added

ID	Title	Message	Start	End	Action
c794e8e6-af27-45b9-b30-b536bd1d5792	New Year Sale	50% OFF on all books for this year!!!	2025-05-15T00:00:00Z	2025-05-31T00:00:00Z	

Figure 48: Successful Addition of Timed Banner Announcement in Admin Side

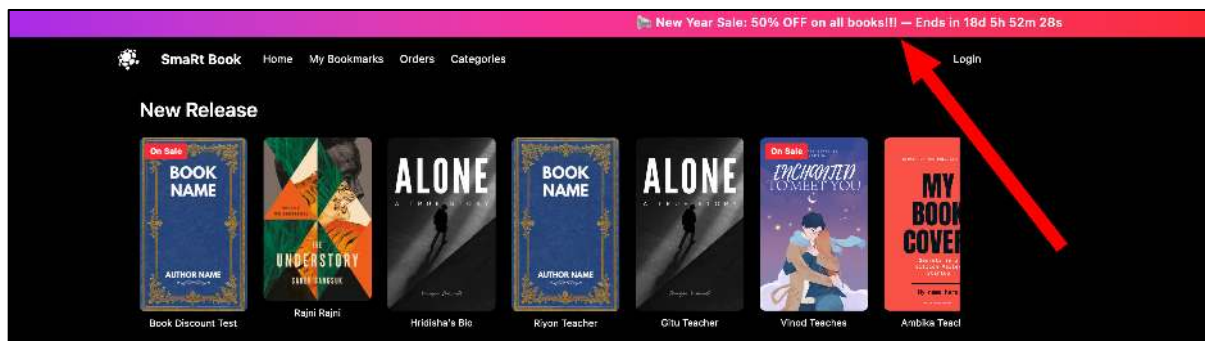


Figure 49: Successful Addition of Timed Banner Announcement in User Side

2.16. Real-Time Successful Order Broadcasting

- **Description:** This feature lets the users to see real-time successful order broadcasting.
- **Functionality:** First, any member must place an order for processing. Then, the order is saved in the database as a pending order. Immediately after that, a broadcast message is shown to all users.

- **Visual Guide:**

Step 1: Proceed to checkout after adding item in the cart.

Step 2: View the real-time successful order broadcasting message from any user account.

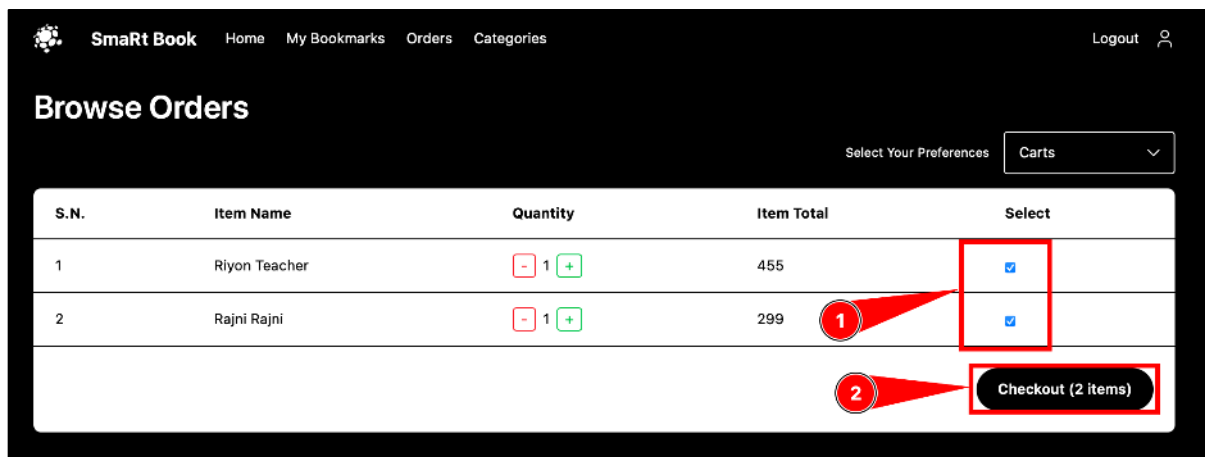


Figure 50: Proceeding to Checkout for Real-Time Successful Order Broadcasting

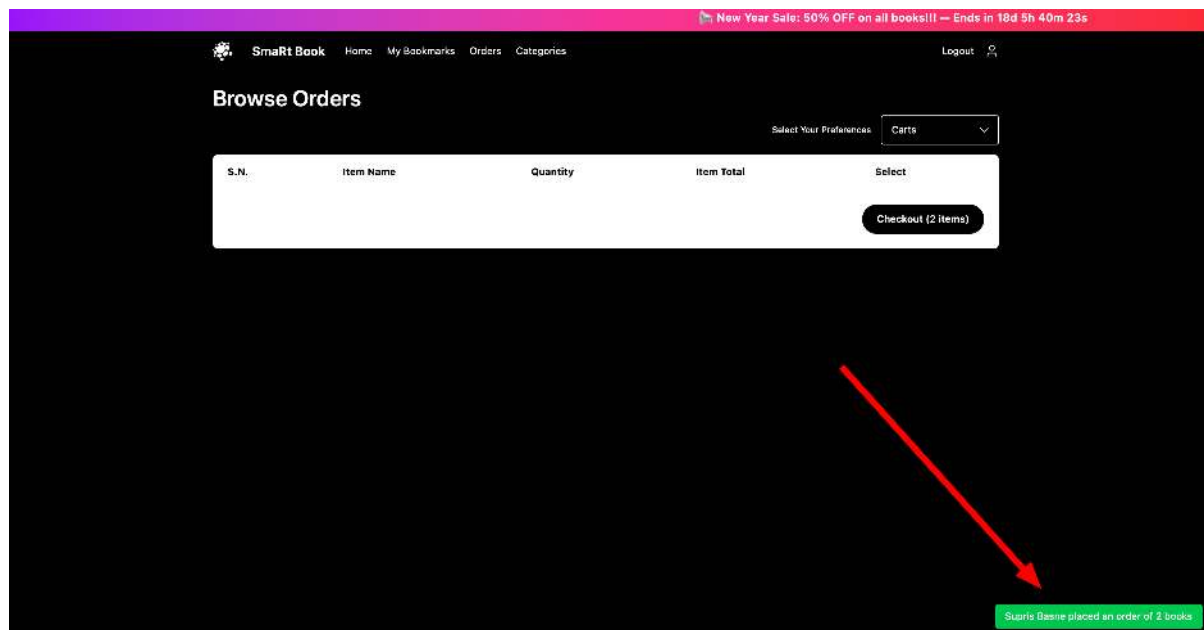
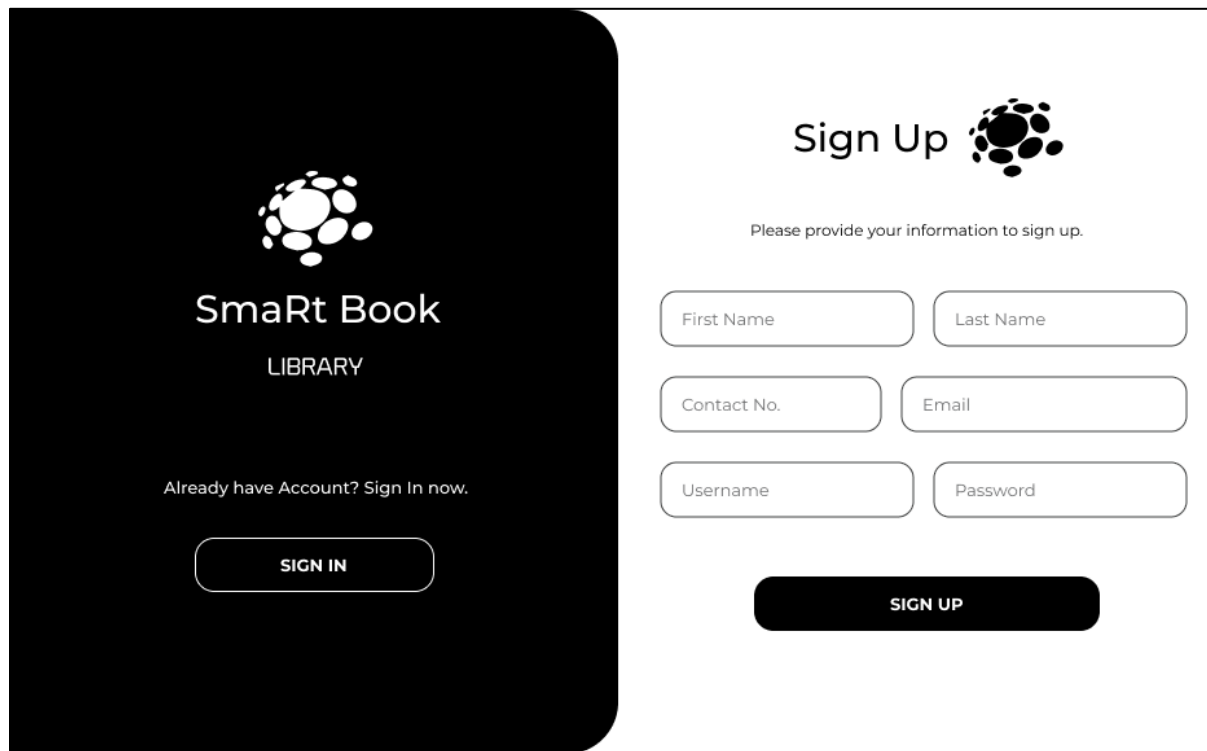


Figure 51: Successful Real-Time Successful Order Broadcasting

3.2. UI Design



The Sign Up page features a dark blue sidebar on the left with the 'SmaRt Book LIBRARY' logo and a 'SIGN IN' button. The main content area is white and contains a 'Sign Up' heading with a logo, a prompt to provide information, and six input fields: First Name, Last Name, Contact No., Email, Username, and Password. A large black 'SIGN UP' button is at the bottom right.

Sign Up 

Please provide your information to sign up.

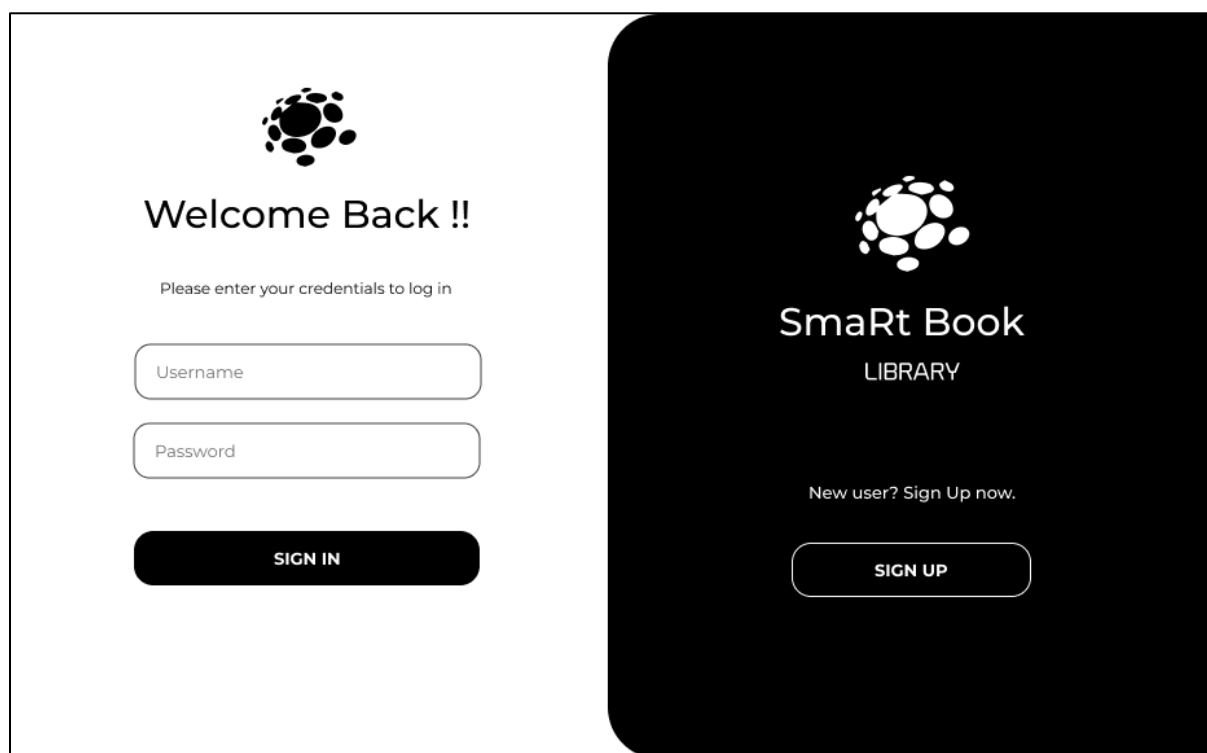
First Name Last Name

Contact No. Email

Username Password

SIGN UP

Figure 53: Sign up Page



The Sign In page features a white sidebar on the left with the 'SmaRt Book LIBRARY' logo and a 'SIGN IN' button. The main content area is white and contains a 'Welcome Back !!' heading with a logo, a prompt to enter credentials, and two input fields: Username and Password. A large black 'SIGN IN' button is at the bottom left. A dark blue sidebar on the right contains the 'SmaRt Book LIBRARY' logo and a 'SIGN UP' button.

 **Welcome Back !!**

Please enter your credentials to log in

Username Password

SIGN IN

SmaRt Book LIBRARY

New user? Sign Up now.

SIGN UP

Figure 54: Sign In Page

3.2.1. Admin Pages UI Design

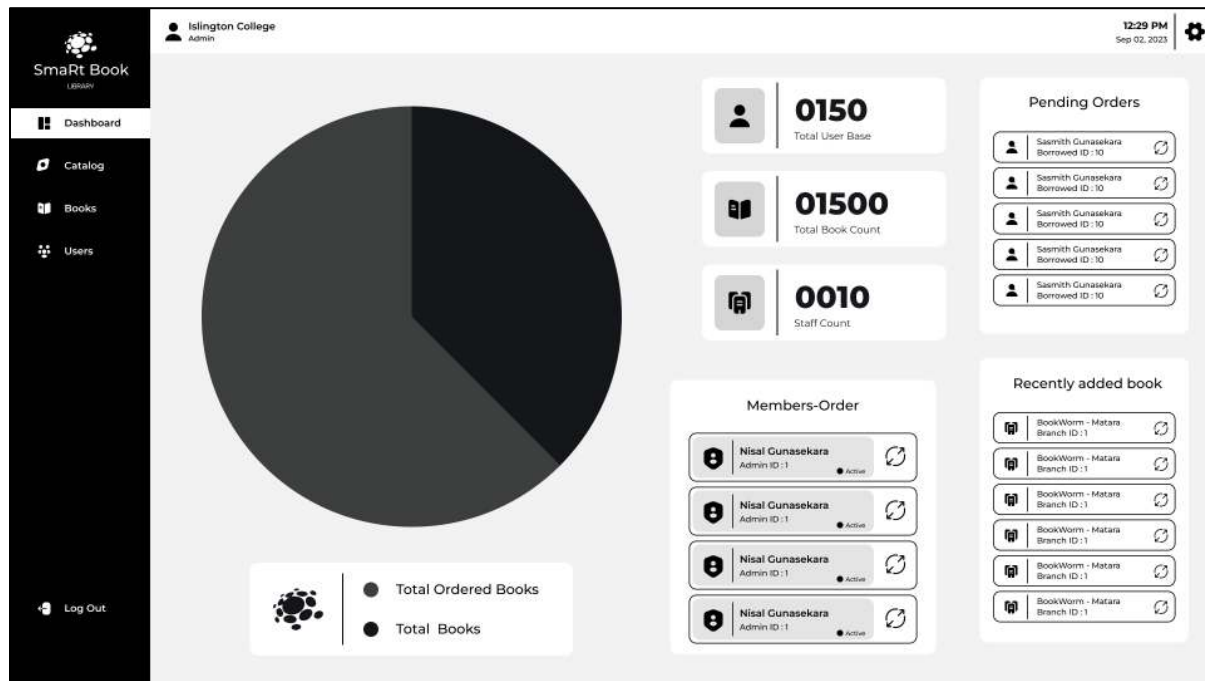


Figure 55: Admin Dashboard

The Borrowed Books section displays a table of all borrowed books. The table has columns for username, Full Name, Book ID, Order Quantity, Date & Time, and Action. The data is filtered by 'Completed' status. The table shows 12 rows of data, all with a status of 'Completed'.

username	Full Name	Book ID	Order Quantity	Date & Time	Action
001	001	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	001	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	001	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	001	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	001	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	001	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	001	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	001	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	001	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	001	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	001	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	001	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	

Figure 56: Borrowed Books

Smart Book Library

Nisal Gunasekara Admin 12:29 PM Sep 02, 2023

Borrowed Books **Overdue Borrowers** Search by ID

ID	User ID	Amount	Due Date	Date & Time	Action
001	1	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	1	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	1	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	1	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	1	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	1	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	1	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	1	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	1	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	1	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	1	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	
001	1	002 Books	13 - 03 - 2024	25-02-2024 10:39:43	

Figure 57: Overdue Borrowers

Smart Book Library

Islington College Admin 12:29 PM Sep 02, 2023

Book Management Add Book Search by ID or Type

ID	Name	Type	Language	Availability	Action
1	Hibernate Core ~11th	Educational	English	Available	 
1	Hibernate Core ~11th	Educational	English	Borrowed	 
1	Hibernate Core ~11th	Educational	English	Available	 
1	Hibernate Core ~11th	Educational	English	Available	 
1	Hibernate Core ~11th	Educational	English	Borrowed	 
1	Hibernate Core ~11th	Educational	English	Available	 
1	Hibernate Core ~11th	Educational	English	Available	 
1	Hibernate Core ~11th	Educational	English	Available	 
1	Hibernate Core ~11th	Educational	English	Borrowed	 
1	Hibernate Core ~11th	Educational	English	Available	 
1	Hibernate Core ~11th	Educational	English	Available	 
1	Hibernate Core ~11th	Educational	English	Available	 

Figure 58: Book Management



Dashboard

Catalog

Books

Users

Log Out

Islington college

Admin

12:29 PM

Sep 02, 2023

Settings

User Management

Add User

Search by ID or Name

ID	Name	Email	Username	Action
1	Prabath Jayasuriya	prabathjayik@gmail.com	prabathjay	  
1	Prabath Jayasuriya	prabathjayik@gmail.com	prabathjay	  
1	Prabath Jayasuriya	prabathjayik@gmail.com	prabathjay	  
1	Prabath Jayasuriya	prabathjayik@gmail.com	prabathjay	  
1	Prabath Jayasuriya	prabathjayik@gmail.com	prabathjay	  
1	Prabath Jayasuriya	prabathjayik@gmail.com	prabathjay	  
1	Prabath Jayasuriya	prabathjayik@gmail.com	prabathjay	  
1	Prabath Jayasuriya	prabathjayik@gmail.com	prabathjay	  
1	Prabath Jayasuriya	prabathjayik@gmail.com	prabathjay	  
1	Prabath Jayasuriya	prabathjayik@gmail.com	prabathjay	  
1	Prabath Jayasuriya	prabathjayik@gmail.com	prabathjay	  
1	Prabath Jayasuriya	prabathjayik@gmail.com	prabathjay	  

Figure 59: User Management

 **Change Password** 

Enter Current Password

Enter Current Password

Enter New Password

Enter New Password

Confirm New Password

Confirm New Password

CANCEL

CONFIRM

Figure 60: Change Credential Popup

 **Update Book** ✕

CANCEL **ADD**

Figure 61: Catalogue Action

Book ID	Name	Type	Language
1	Hibernate - Core	Educational	English
1	Hibernate - Core	Educational	English
1	Hibernate - Core	Educational	English
1	Hibernate - Core	Educational	English
1	Hibernate - Core	Educational	English
1	Hibernate - Core	Educational	English
1	Hibernate - Core	Educational	English

ID

4

Total Books : 04 Books

Due Date : 13 - 12 - 2024

CLOSE

Figure 62: Overdue Books View Popup

 **Update User** 

CANCEL

UPDATE

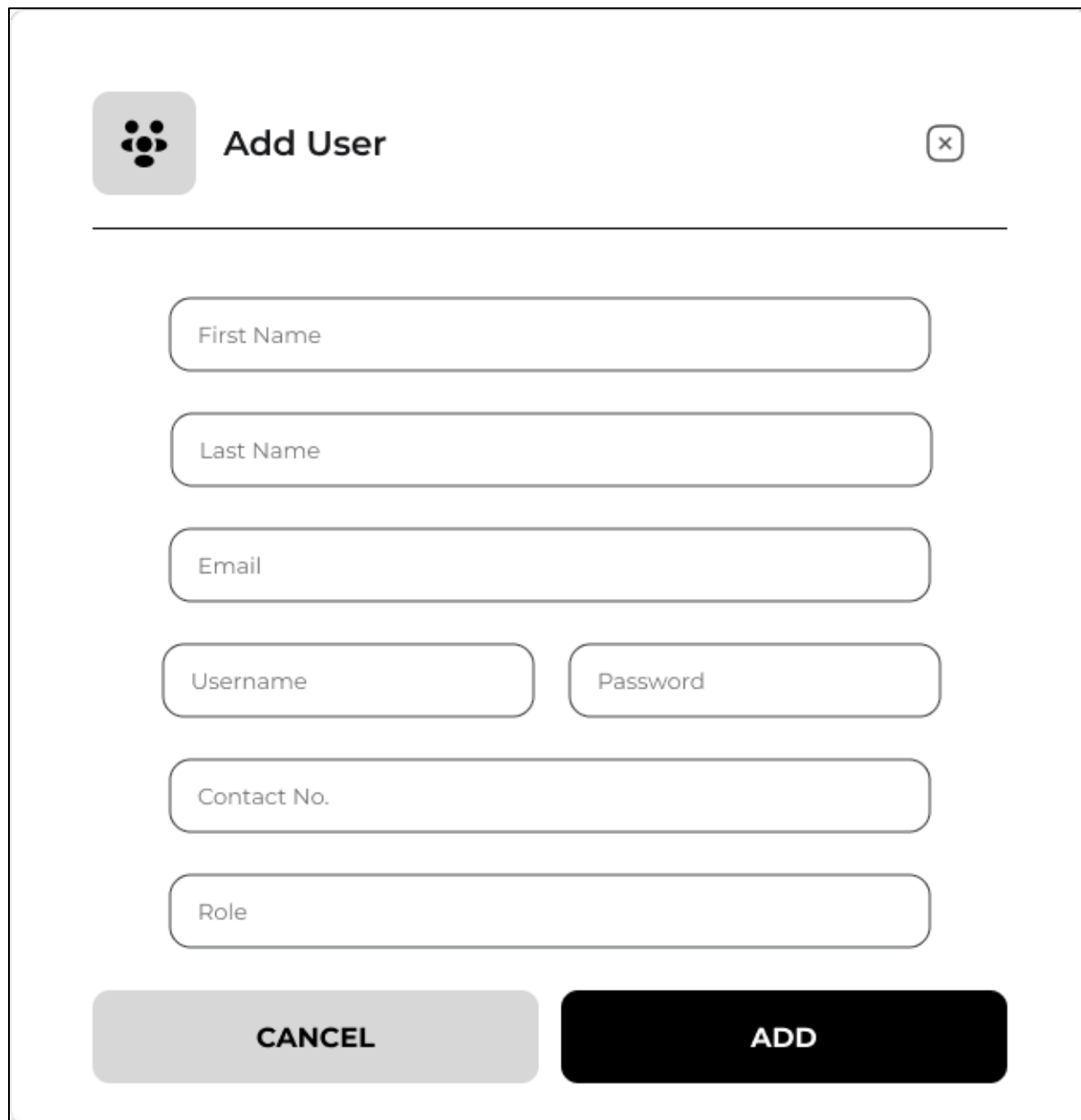
Figure 63: Update User Popup



Add Book

**CANCEL****ADD**

Figure 64: Add Book Popup



The image shows a modal window titled "Add User". At the top left is a logo consisting of four dots arranged in a square. To the right of the logo is the title "Add User". At the top right is a close button represented by a small square with an "x" inside. Below the title bar is a horizontal line. The form contains six input fields: "First Name", "Last Name", "Email", "Username", "Password", "Contact No.", and "Role". The "Username" and "Password" fields are side-by-side. At the bottom are two buttons: a light gray "CANCEL" button and a black "ADD" button.

 **Add User** 

CANCEL

ADD

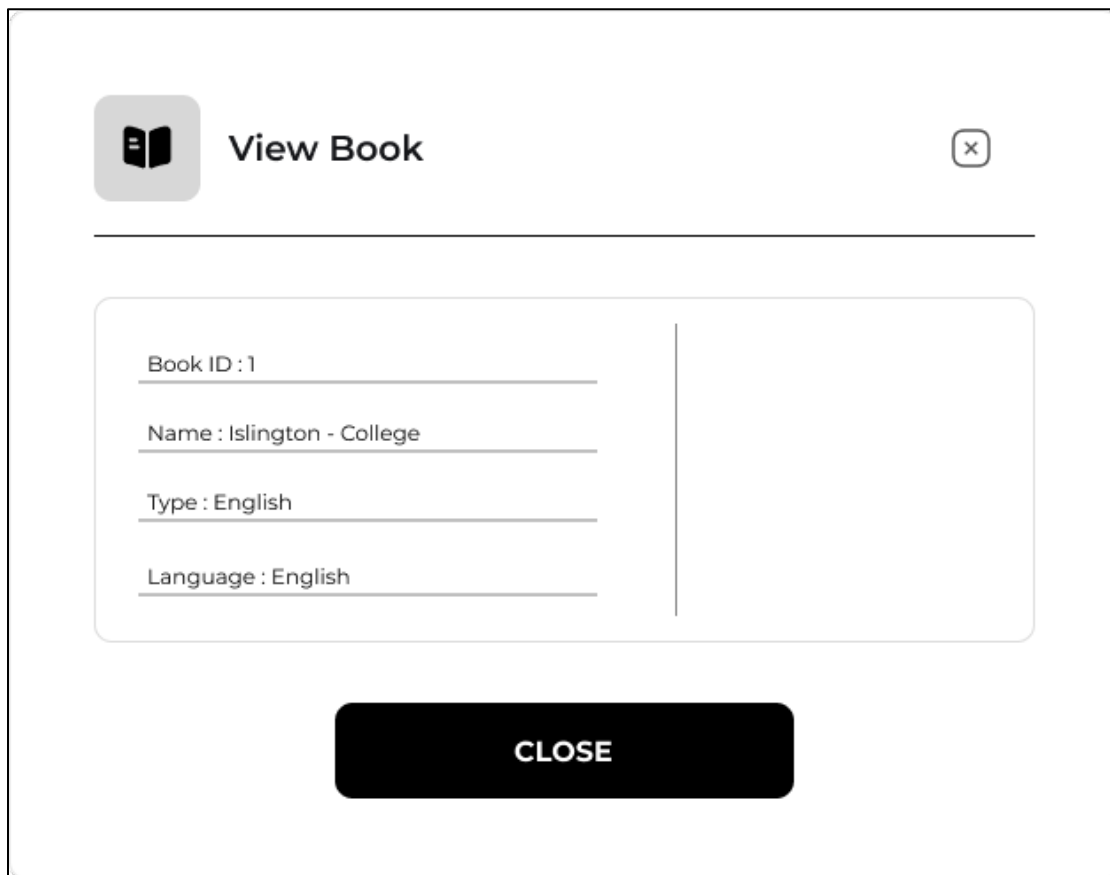
Figure 65: Add User Popup



Update Book

**CANCEL****ADD**

Figure 66: Update Book Popup



The image shows a 'View Book' popup window. At the top left is a book icon, and at the top right is a close button (an 'x' in a circle). The title 'View Book' is centered at the top. Below the title is a horizontal line. Underneath this line is a rounded rectangle containing four text labels: 'Book ID : 1', 'Name : Islington - College', 'Type : English', and 'Language : English'. Each label is followed by a horizontal line. To the right of these labels is a vertical line. At the bottom center of the popup is a large black button with the word 'CLOSE' in white capital letters.

Figure 67: View Book Popup

 **View User** 

User ID : 1

Name : Sulav Budhathoki

Email : SulavBudhathoki@gmail.com

Username : sulav

Saved by :

Islington College
(Admin)

CLOSE

Figure 68: View User Popup

 **Delete Confirmation** 

"Are you certain you wish to proceed
with the deletion of the selected
entry?"

CONFIRM

Figure 69: Delete Confirmation Popup



View Branch

✕

Branch ID : 1

Name : BookWorm - Matara

Contact No : wimalgsh@gamil.com

Location : Matara

Saved by :

Nisal Gunasekara
(Admin)

CLOSE

Figure 70: View Branch Popup

3.2.2. User Pages UI

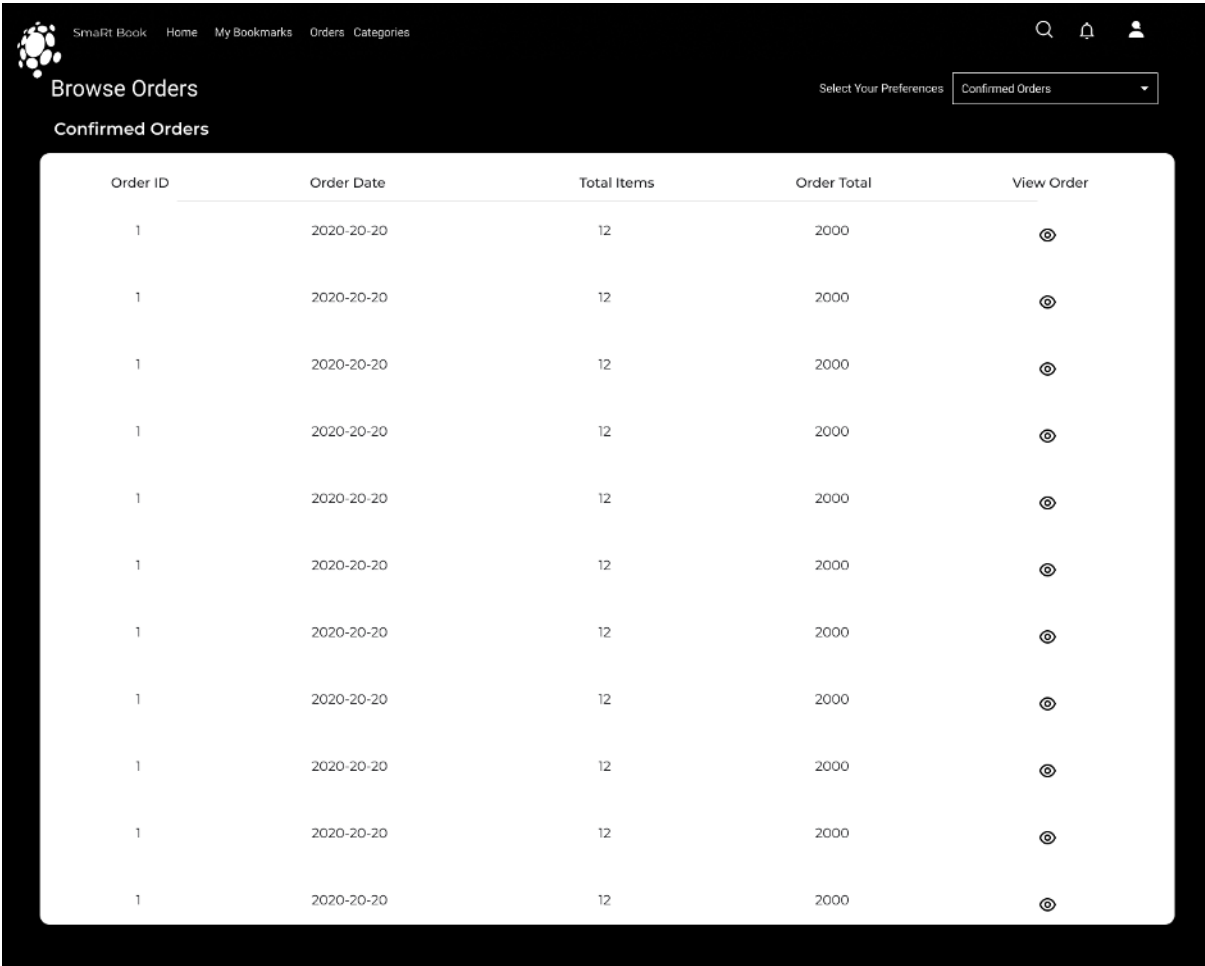


Figure 71: Order Page

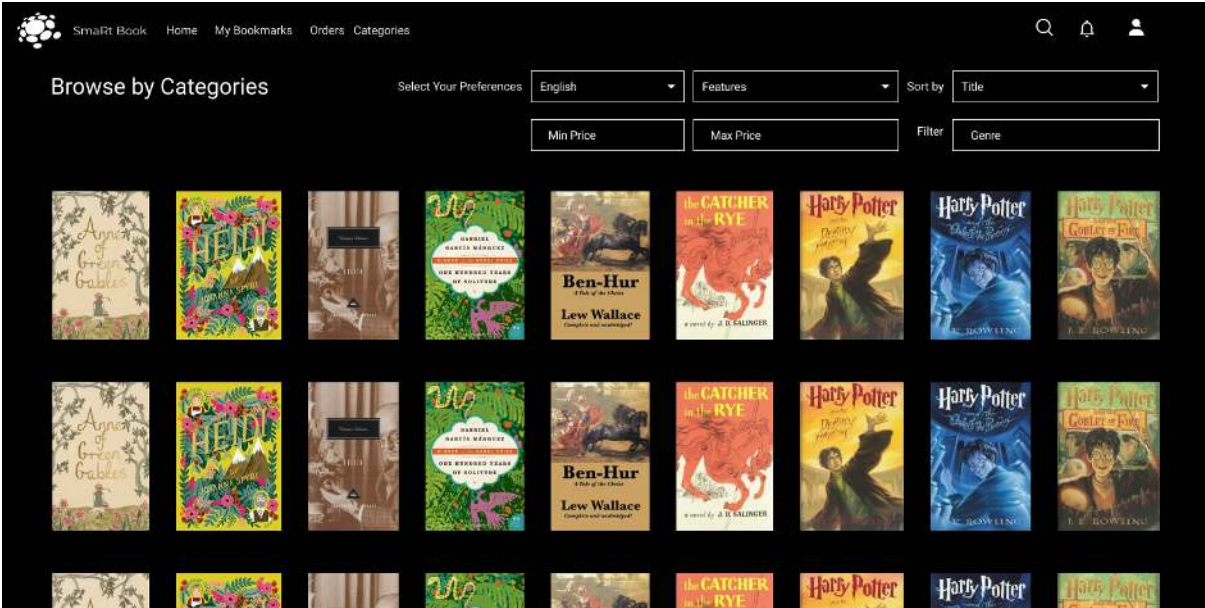


Figure 72: Browse Book Catalog

[illegible]

59

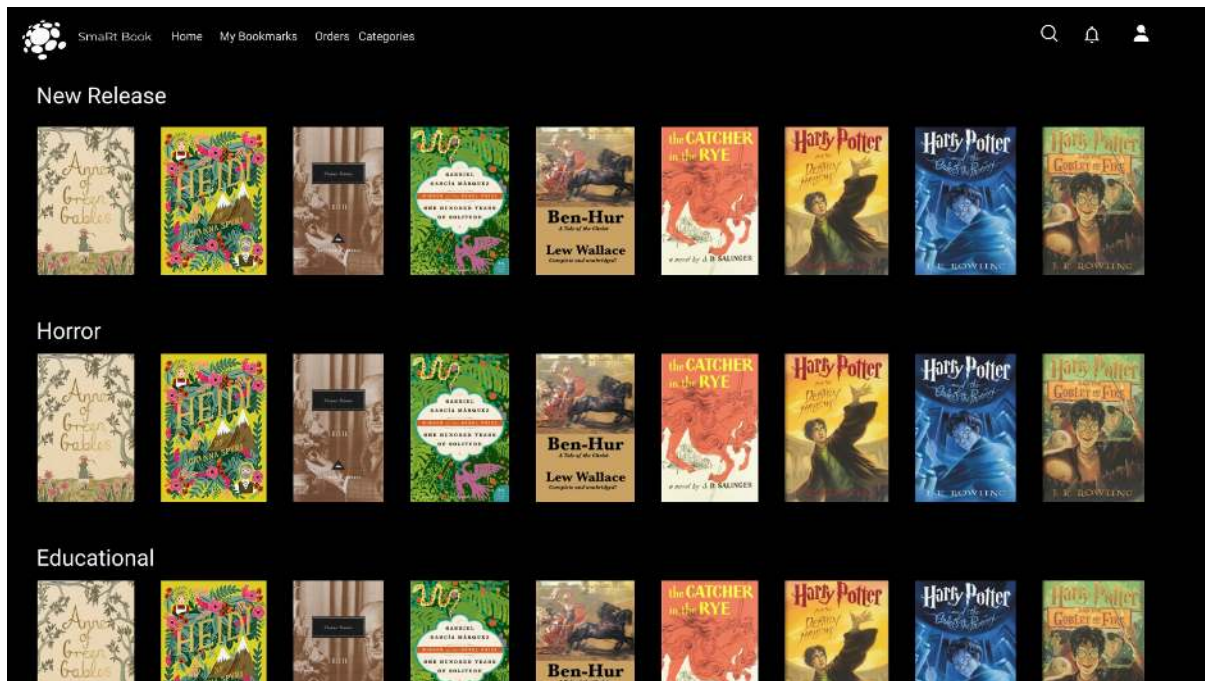
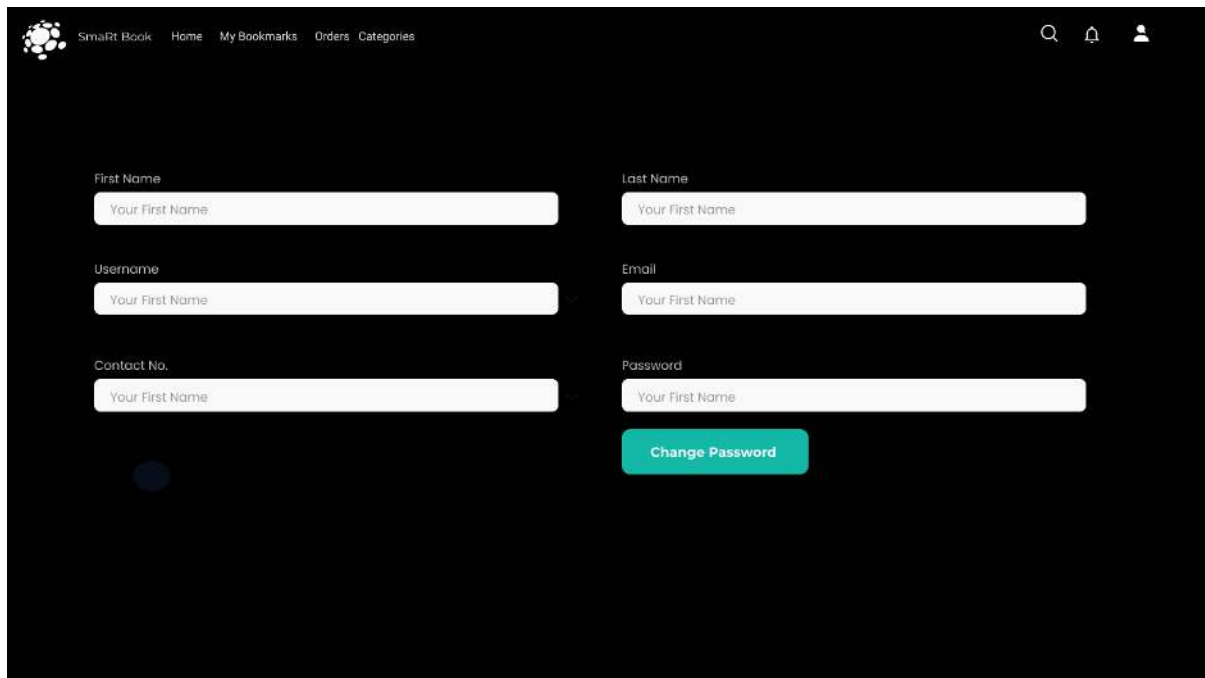


Figure 74: Home Page

Rhythm Paudel



The screenshot shows a user profile page with a dark blue header. The header contains a logo on the left, navigation links (Smart Book, Home, My Bookmarks, Orders, Categories) in the center, and search, notification, and user icons on the right. The main content area is white and contains a profile card. The card has a profile picture placeholder (a blue circle) and a 'Change Password' button. Below the profile card are two columns of form fields for user information: First Name, Last Name, Username, Email, Contact No., and Password. Each field has a placeholder text 'Your First Name'.

Smart Book Home My Bookmarks Orders Categories

First Name
Your First Name

Last Name
Your First Name

Username
Your First Name

Email
Your First Name

Contact No.
Your First Name

Password
Your First Name

Change Password

Figure 76: User Profile Page

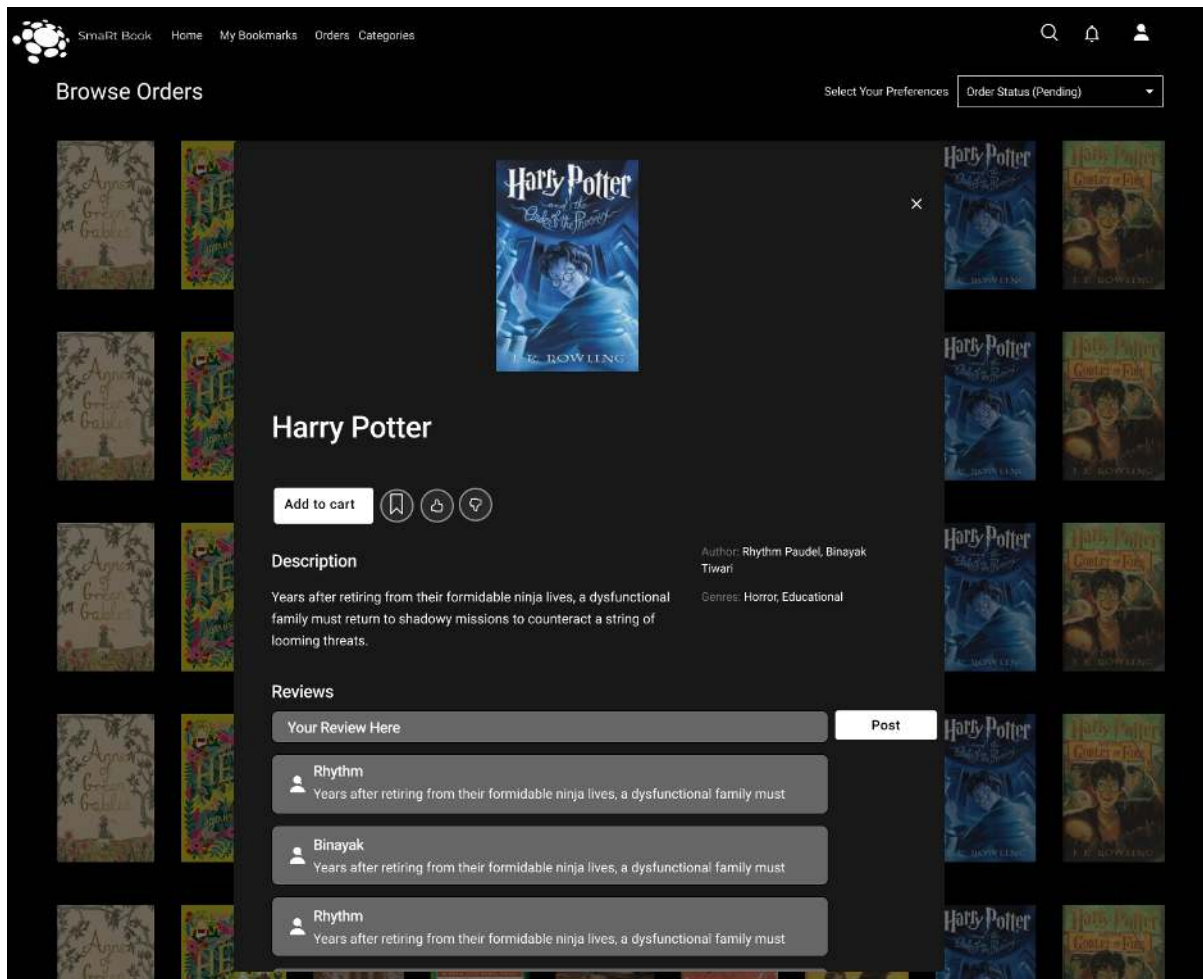


Figure 77: Book Details Page

3.3. UML Diagrams

3.3.1. Use Case Diagram

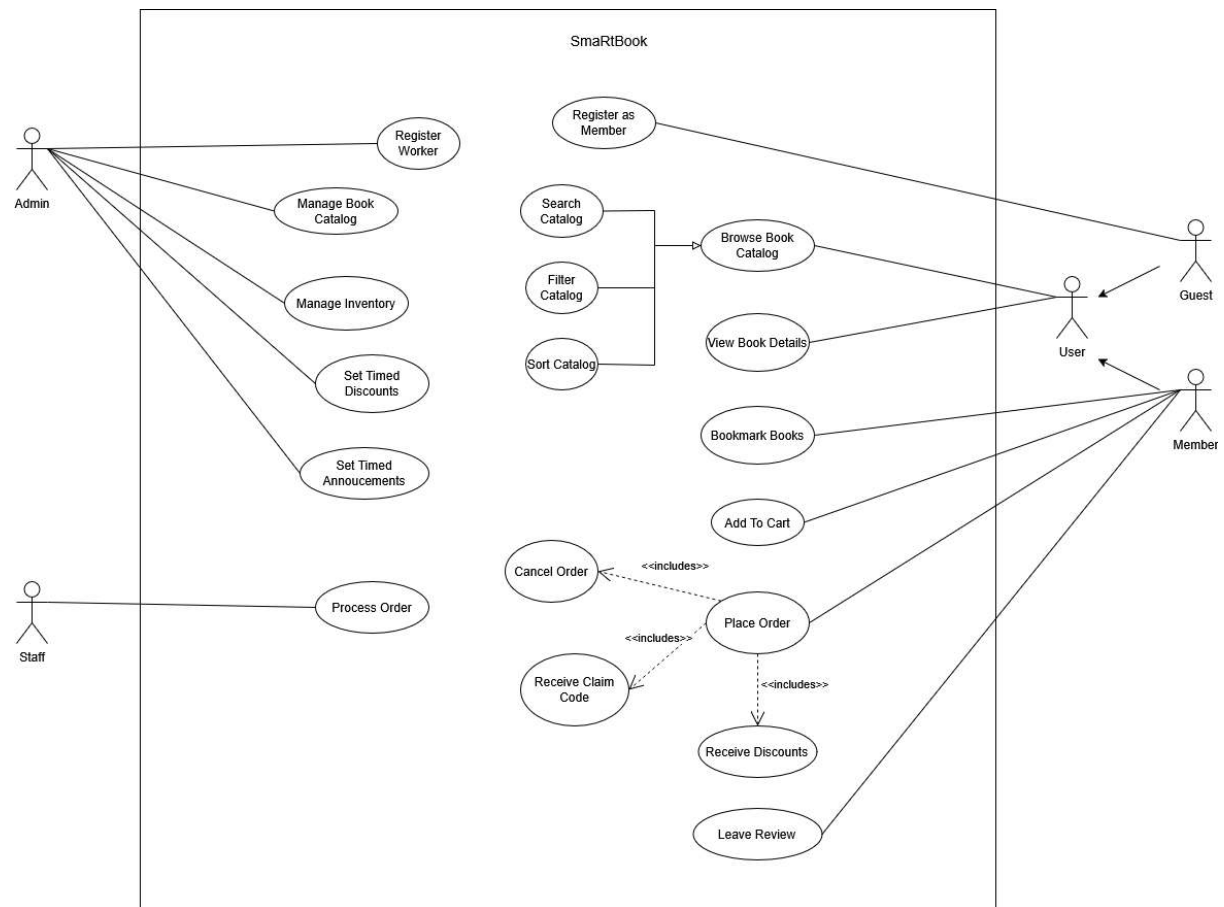


Figure 78: Use Case Diagram

3.3.2. Sequence Diagram

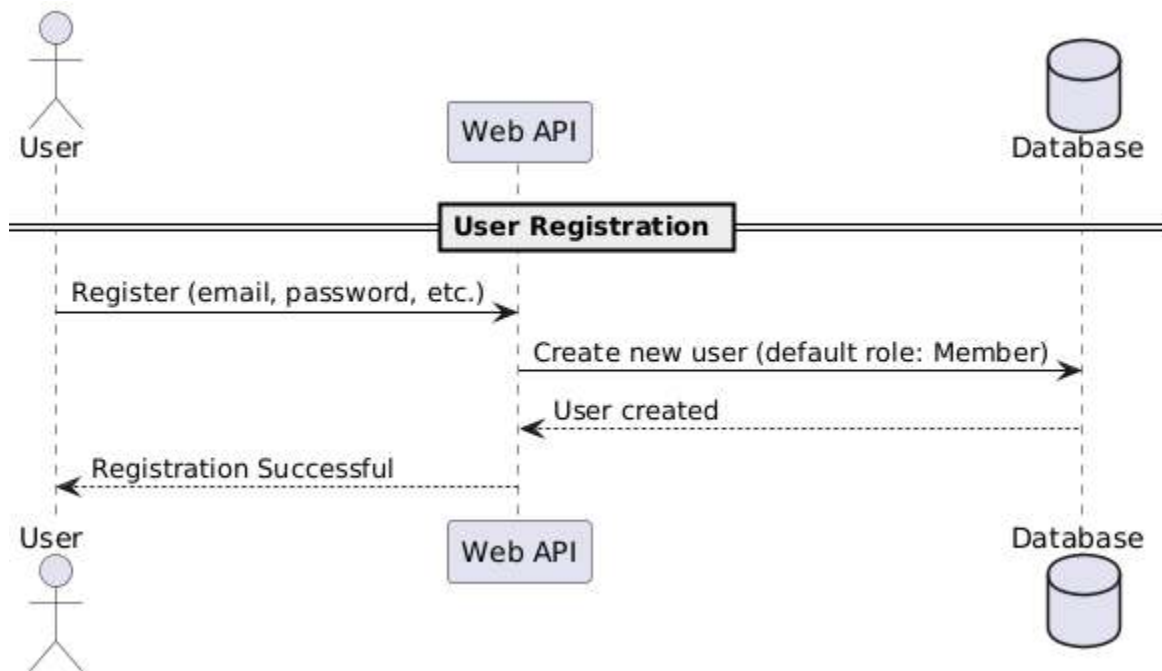


Figure 79: Sequence Diagram: User Registration

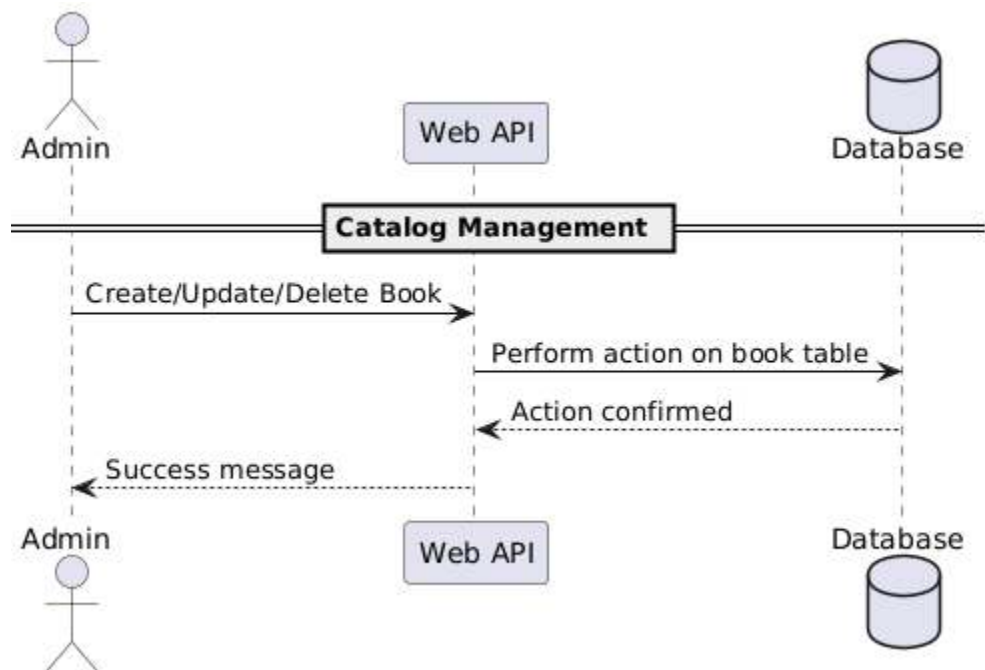


Figure 80: Sequence Diagram: Catalog Management

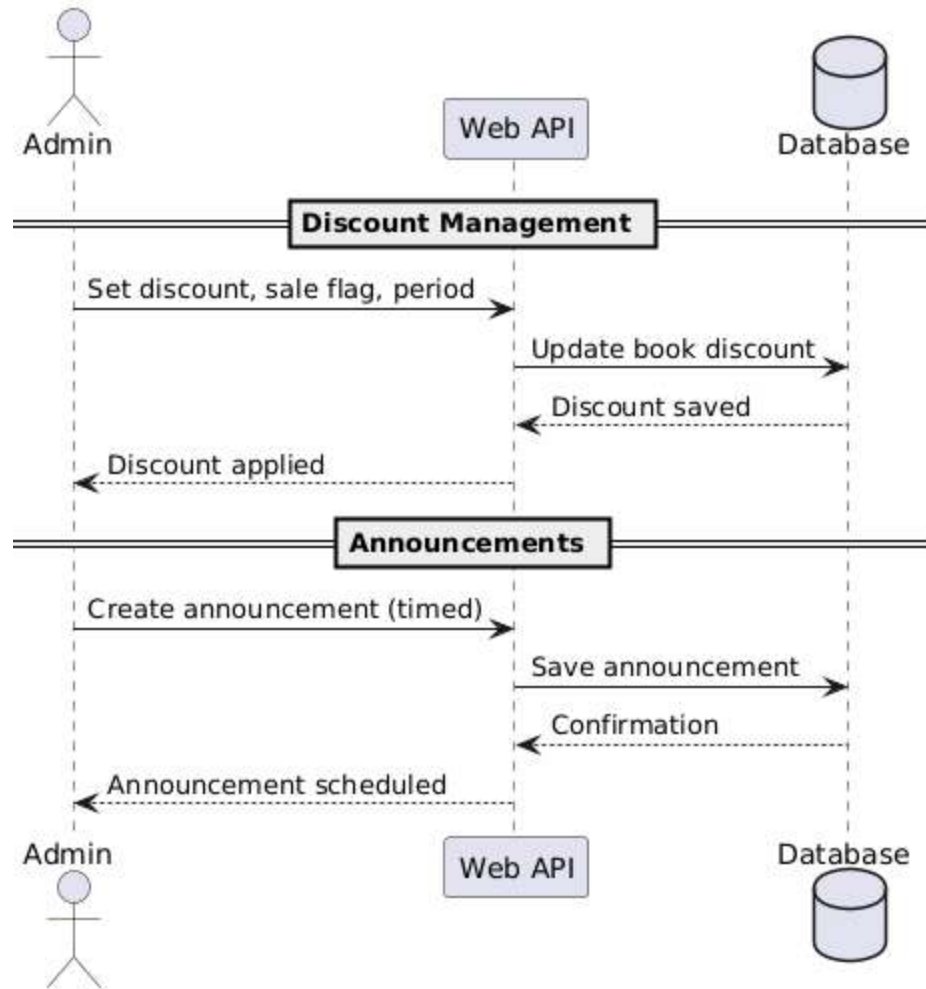


Figure 81: Sequence Diagram: Discount and Announcement

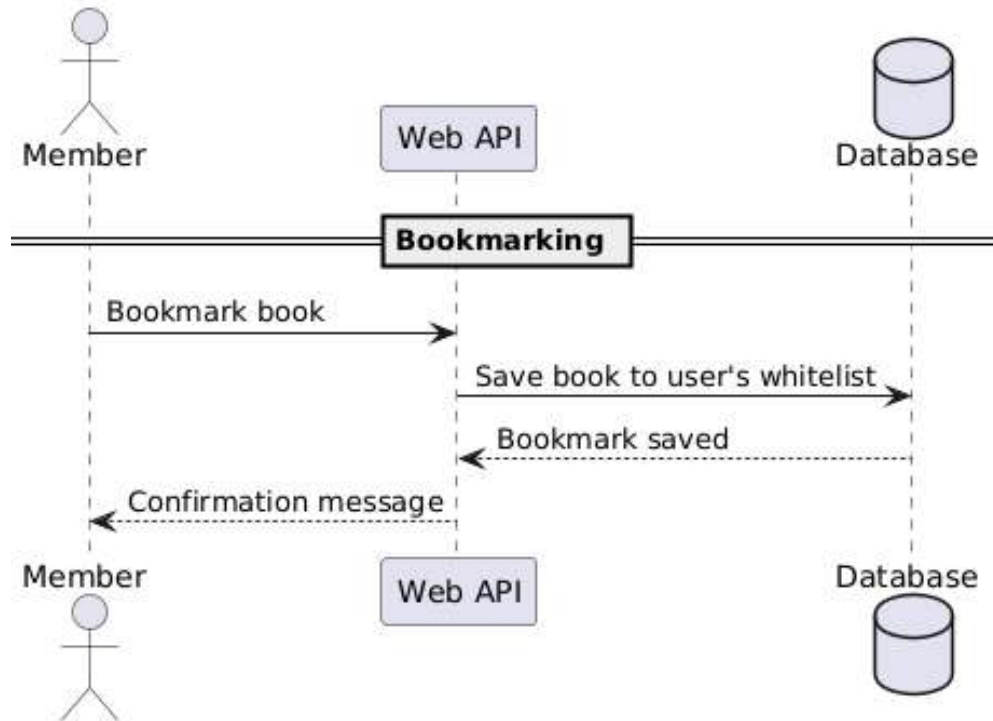


Figure 82: Sequence Diagram: Bookmark

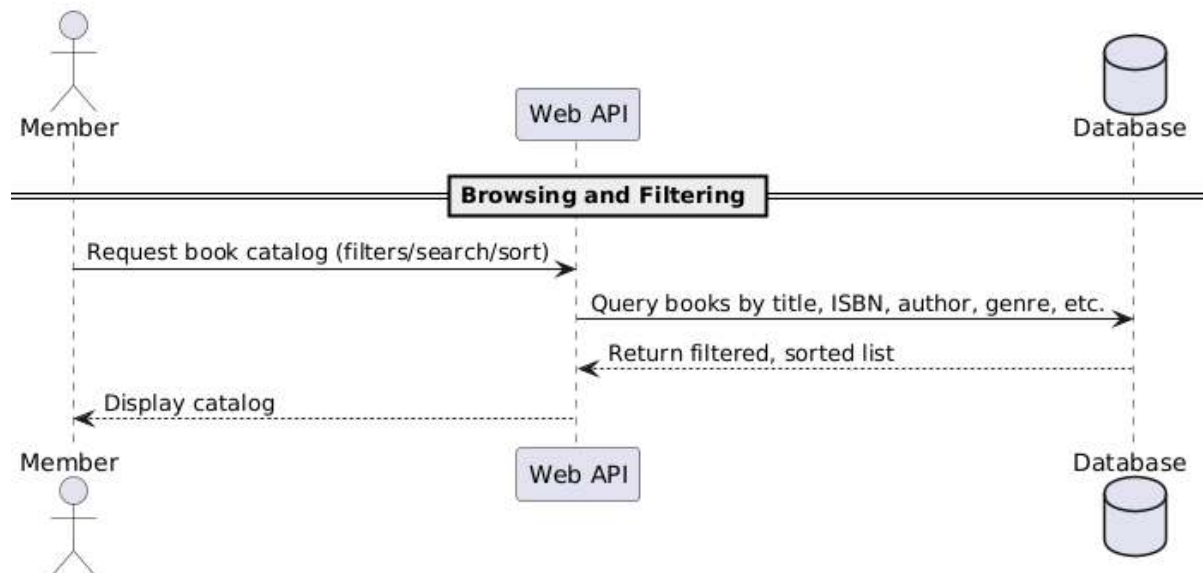


Figure 83: Sequence Diagram: Browse and Filter

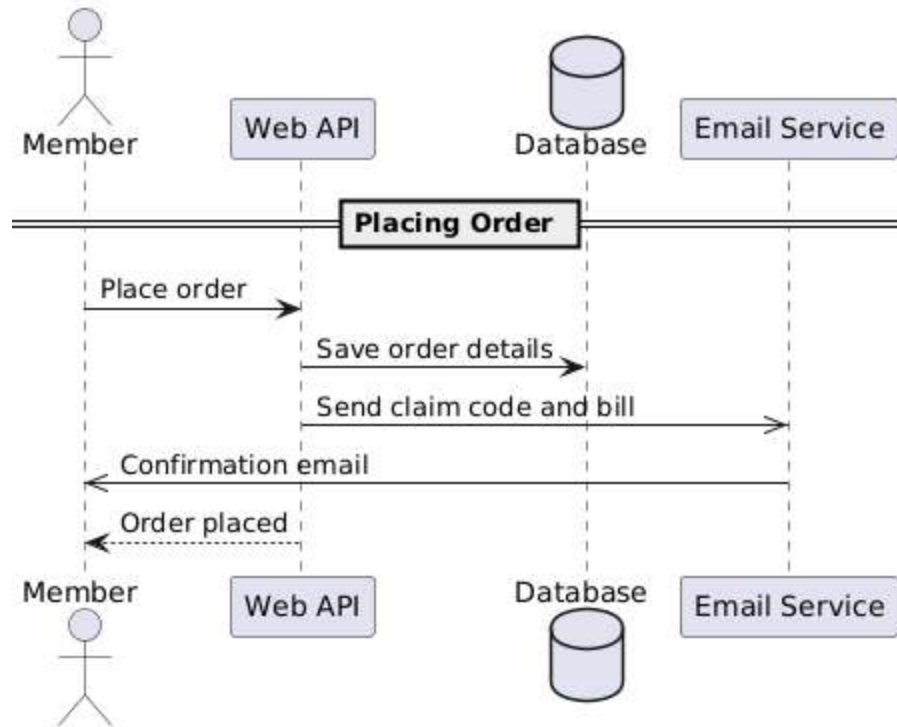


Figure 84: Sequence Diagram: Order Processing

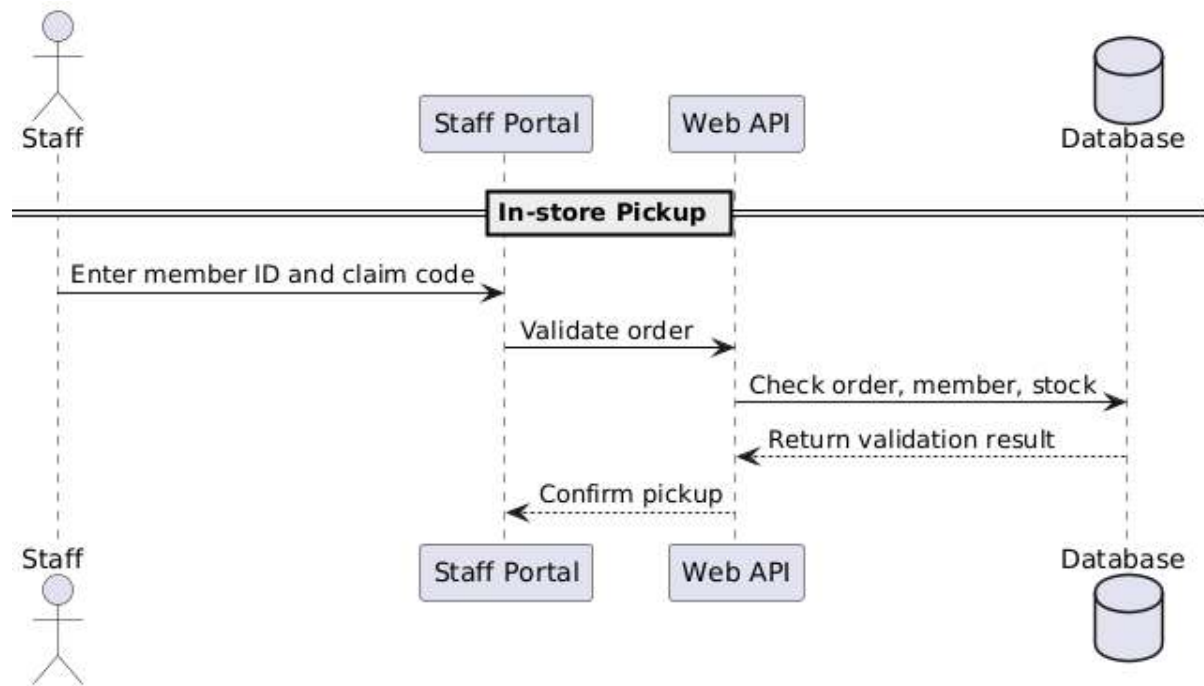


Figure 85: Sequence Diagram: Order Validation

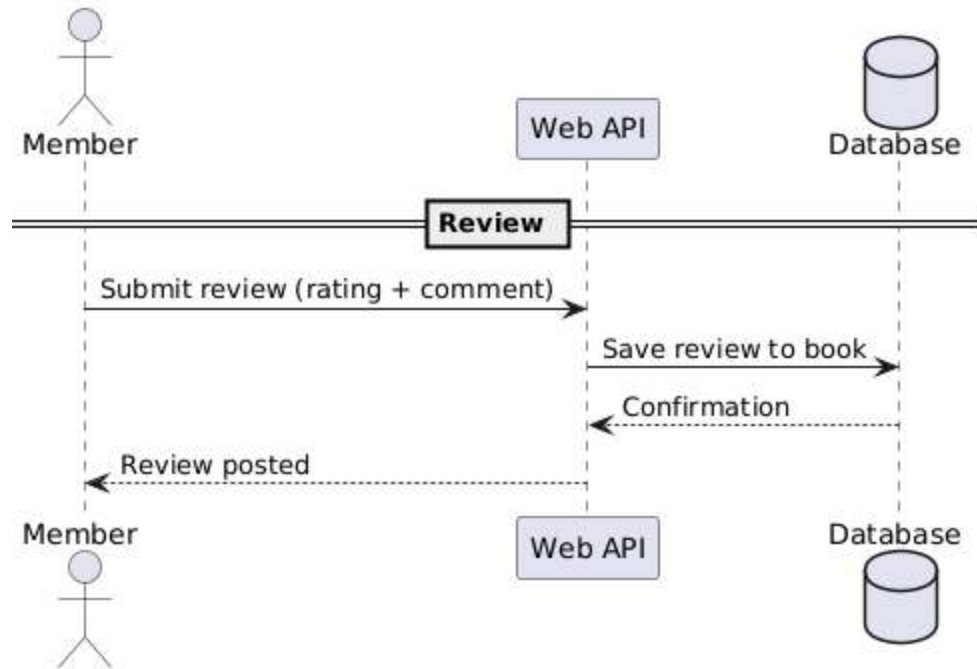


Figure 86: Sequence Diagram: Review

3.4. Git Hub Repository

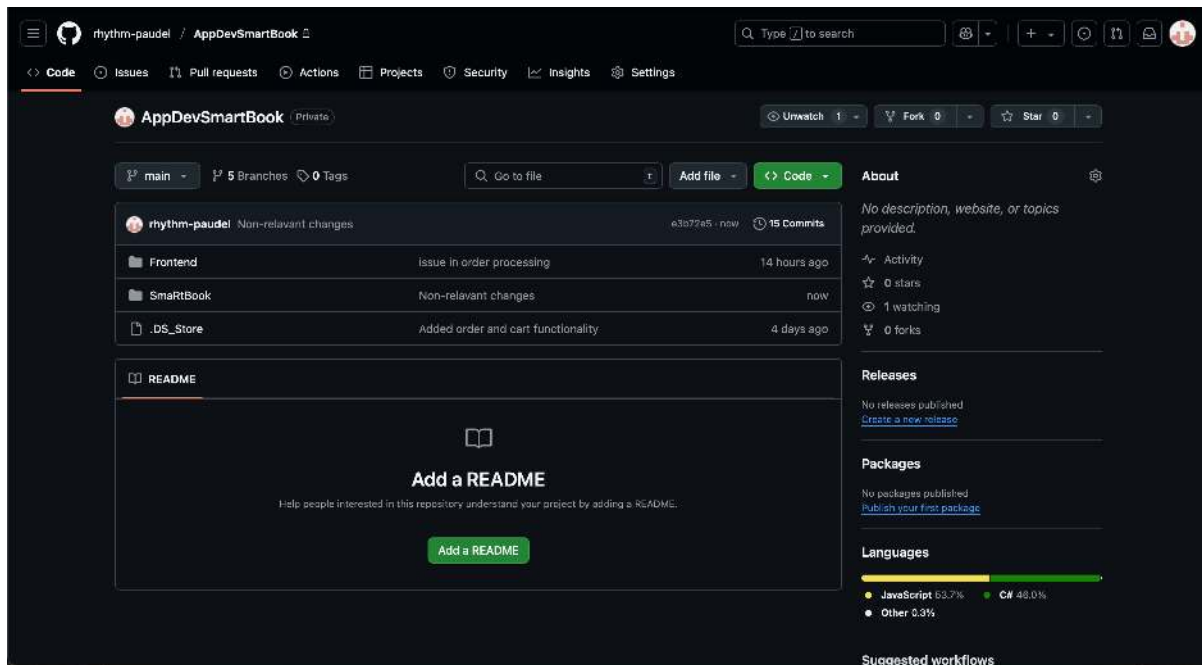


Figure 87: Git Hub Repository

```

rhythm@Rhythms-MacBook-Pro COURSE_WORK_DRAFT_FILES % git log --pretty=format:"%h %s"

2640f0a issue in order processing
068d82e added signalR for message broadcasting
885b754 added email verification and announcement feature
2692566 Added review service too
dca09ff Added order and cart functionality
8f186c2 add to cart / remove to cart feature
f834bf2 Redid everything replaces list of author instead of single author
f8656a6 Search functionality
179fea6 changes in book fields, database migration in frontend and backend
0fbdd0f User crud via admin panel
2e5890d Book Controllers
c21ba08 Added jwt token configurations
474cbb8 update divided
d87cb5f Basic frontend setup and backend configuration
  
```

Figure 88: Git Log History as Proof of Work

3.5. Migration History

	MigrationId [PK] character varying (150) 	ProductVersion character varying (32) 
1	20250502120445_Initial	8.0.15
2	20250502120548_InitialMigration	8.0.15
3	20250502120640_SecondMigration	8.0.15
4	20250502120839_ThirdMigration	8.0.15
5	20250502125850_Removed Password from user model	8.0.15
6	20250502130321_Addded phone number and username in user field	8.0.15
7	20250502131436_Redid User model	8.0.15
8	20250505063104_Added Added date in book field	8.0.15
9	20250505160120_Changes in book field	8.0.15
10	20250507054016_Added Author and bookauthor table	8.0.15
11	20250507054501_a	8.0.15
12	20250507055230_author and authorbook again	8.0.15
13	20250507084951_Change in datatype of author	8.0.15
14	20250507160600_Cart Update	8.0.15
15	20250508125412_Added start and end date in books	8.0.15
16	20250509111508_minor change	8.0.15

Figure 89: Migration History

3.6. Milestone Chart

S.No.	Milestone	Start Date	End Date	Status
1.	Finalize the requirements and features	April 15	April 17	Completed
2.	Frontend UI Components	April 18	April 20	Completed
3.	Backend API + UI Test for registration	April 21	April 24	Completed
4.	API +n UI for bookmark/ cart and order	April 25	April 30	Completed
5.	Backend logic testing and fixes for discount	May 2	May 4	Completed
6.	Email Confirmation with Claim Code	May 5	May 6	Completed
7.	Admin CRUD and Logic Testing and Fixing	May 6	May 8	Completed
8.	Testing and Proof of Work for Documentation	May 7	May 10	Completed
9.	Final Testing, QA and Documentation	May 11	May 12	Completed

Table 1: Milestone Chart

3.7. API Endpoints

Announcement			^
POST	/api/Announcement		🔒 ▼
GET	/api/Announcement		🔒 ▼
DELETE	/api/Announcement/{id}		🔒 ▼
AnnouncementUser			^
GET	/AnnouncementUser		🔒 ▼
Auth			^
POST	/api/Auth/Register		🔒 ▼
POST	/api/Auth/Login		🔒 ▼
GET	/api/Auth		🔒 ▼
GET	/api/Auth/Logout		🔒 ▼
Author			^
GET	/api/Author		🔒 ▼

Figure 90: API Endpoints 1

Bookmark			^
GET	/Bookmark/{bookid}		🔒 ▼
POST	/Bookmark/{bookid}		🔒 ▼
GET	/Bookmark		🔒 ▼
Books			^
POST	/api/Books/book		🔒 ▼
PUT	/api/Books/updatebook/{id}		🔒 ▼
DELETE	/api/Books/delete/{id}		🔒 ▼
GET	/api/Books		🔒 ▼
Cart			^
GET	/Cart/{id}		🔒 ▼
POST	/Cart/{id}		🔒 ▼
GET	/Cart/mycart		🔒 ▼
PUT	/Cart/changeinstock		🔒 ▼

Figure 91: API Endpoints 2

Dashboard			^
GET	/api/Dashboard/chart		🔒 ▼
GET	/api/Dashboard/stats		🔒 ▼
GET	/api/Dashboard/members-orders		🔒 ▼
GET	/api/Dashboard/pending-orders		🔒 ▼
GET	/api/Dashboard/recent-books		🔒 ▼
Homepage			^
GET	/horror	📖 🔒	▼
GET	/biography		🔒 ▼
GET	/newrelease		🔒 ▼
Order			^
POST	/Order		🔒 ▼
GET	/Order		🔒 ▼
GET	/Order/{orderId}		🔒 ▼
PUT	/Order/cancel/{orderId}		🔒 ▼
GET	/api/Order		🔒 ▼
GET	/api/Order/{orderId}		🔒 ▼
PUT	/api/Order/complete/{orderId}/{claimCode}		🔒 ▼
PUT	/api/Order/cancel/{orderId}		🔒 ▼

Figure 92: API Endpoints 3

Review ^		
POST	/Review	🔒 ▼
GET	/Review/{id}	🔒 ▼
Searchs ^		
GET	/Searchs/defaultsearch	🔒 ▼
GET	/Searchs/searchbyparams	🔒 ▼
UserDetails ^		
GET	/UserDetails	🔒 ▼
PUT	/UserDetails	🔒 ▼
PUT	/UserDetails/password	🔒 ▼
Users ^		
PUT	/api/Users/{id}	🔒 ▼
DELETE	/api/Users/{id}	🔒 ▼
GET	/api/Users	🔒 ▼
POST	/api/Users	🔒 ▼

Figure 93: API Endpoints 4

3.7.1. Announcement Endpoints

This API endpoint is responsible for carrying out actions like creating, fetching announcements, and also deleting desired announcements by their ID in admin side.

3.7.2. Announcement User Endpoints

This API endpoint is responsible for fetching announcements to the user side.

3.7.3. Auth Endpoints

This API endpoint is responsible for carrying out Authorization and Authentication operations which includes important steps like login, register, hashing, etc.

3.7.4. Author Endpoints

This API endpoint is responsible for fetching author detail.

3.7.5. Bookmark Endpoints

This API endpoint is responsible for fetching all bookmarks of the user. Additionally, the endpoint is also used for knowing the bookmark status of each book of the logged in user and is also used for adding or removing a bookmark.

3.7.6. Books Endpoints

This API endpoint is responsible for creating, updating, fetching, and removing books by admin via admin panel.

3.7.7. Cart Endpoints

This API endpoint is responsible for important tasks in the cart, like getting the cart status of a specific book, adding/removing books from the cart, updating the quantity of books in the cart, and getting all cart items.

3.7.8. Dashboard Endpoint

This API endpoint is responsible for summarizing all operations happening in library in admin panel.

3.7.9. Homepage Endpoint

This API endpoint is responsible for fetching the latest release, and the genre types of horror and biography.

3.7.10. Order Endpoint

This API endpoint is responsible for important tasks like creating, fetching, updating status of order and getting details of each order. Staffs can also approve orders based on the claim code given by user or cancel the order via this API.

3.7.11. Review Endpoint

This API endpoint is responsible for creating and fetching reviews posted by user who has purchased the respective book.

3.7.12. Searches Endpoint

This API endpoint is responsible for searching the books via different parameters, among which some of them are, pagination, title, author, ISBN, language, and genre.

3.7.13. UserDetails Endpoint

This API endpoint is responsible for important tasks like fetching and updating user details along with their password.

3.7.14. Users Endpoint

This API endpoint is responsible for important tasks like creating, updating, fetching, and removing users through admin portal.

3.8. Testing

3.8.1. API Testing

3.8.1.1. Test 1 – Create a new book

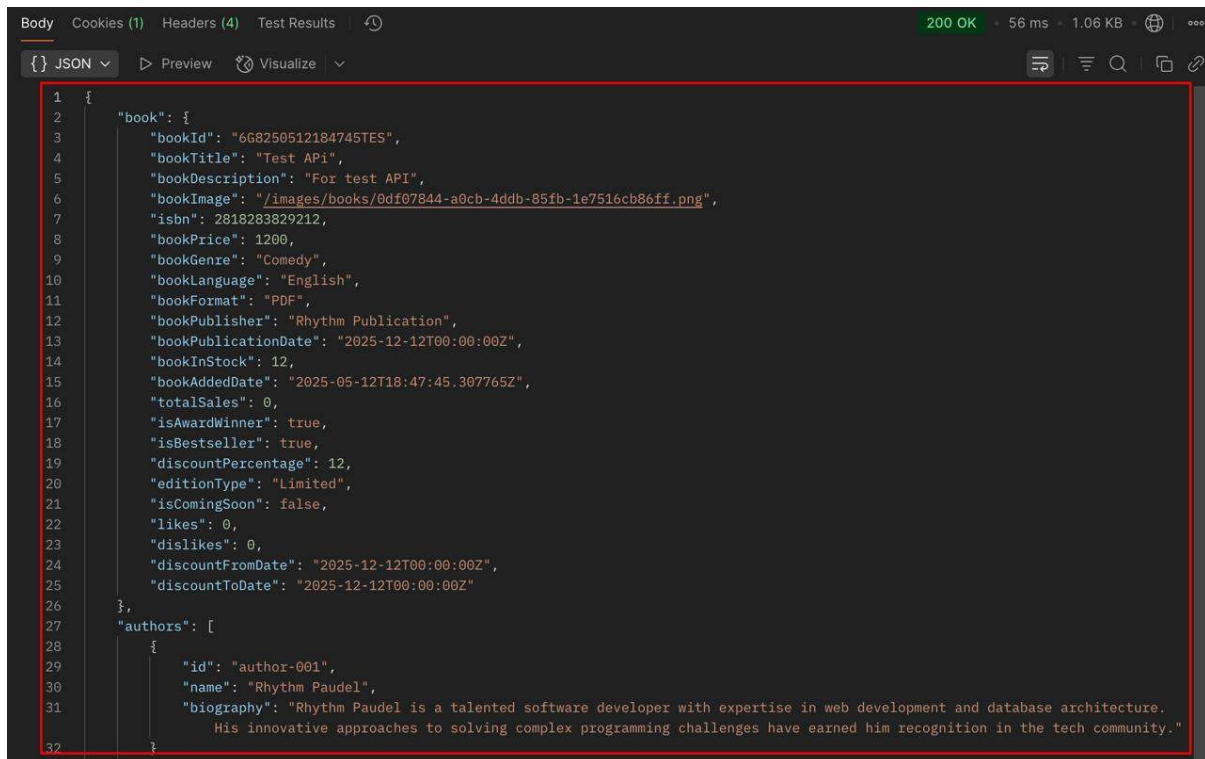
Test	1
Objective	Create a new book
Action	Send POST /api/Books/book with title, author, genre, price, etc. as admin
Expected Result	Book is created with 201 Created
Actual Result	Book created with 201 created
Conclusion	Test was successful.

Table 2: API Testing: Test 1

The screenshot shows a REST client interface with the following details:

- URL:** http://localhost:5012/api/Books/book
- Method:** POST
- Body Type:** form-data
- Fields:**
 - BookTitle: Text, Test API
 - AuthorId: Text, author-001
 - BookDescription: Text, For test API
 - Isbn: Text, 2818283829212
 - BookPrice: Text, 1200
 - BookGenre: Text, Comedy
 - BookLanguage: Text, English
 - BookFormat: Text, PDF
 - BookPublicationDate: Text, 12/12/2025
 - BookInStock: Text, 12
 - BookPublisher: Text, Rhythm Publication
 - IsAwardWinner: Text, true
 - IsBestseller: Text, true
 - DiscountPercentage: Text, 12
 - EditionType: Text, Limited
 - IsComingSoon: Text, false
 - DiscountFromDate: Text, 12/12/2025
 - DiscountToDate: Text, 12/12/2025
 - BookImageFile: File, Screenshot 2025-04-27 at 3.01.38 PM.p...
- Buttons:** Save, Send, Cookies

Figure 94: API Testing: Test 1



The screenshot shows a REST client interface with a dark theme. The top bar indicates a successful 200 OK response with a status of 56 ms and a body size of 1.06 KB. The main area displays a JSON response, which is a list containing one object. This object has a 'book' field and an 'authors' field. The 'book' field is an object with various attributes like bookId, bookTitle, bookDescription, bookImage, isbn, bookPrice, bookGenre, bookLanguage, bookFormat, bookPublisher, bookPublicationDate, bookInStock, bookAddedDate, totalSales, isAwardWinner, isBestseller, discountPercentage, editionType, isComingSoon, likes, dislikes, discountFromDate, and discountToDate. The 'authors' field is an array containing one author object with fields id, name, and biography. The biography field contains a paragraph of text about Rhythm Paudel.

```
1 {
2   "book": {
3     "bookId": "6G8250512184745TES",
4     "bookTitle": "Test API",
5     "bookDescription": "For test API",
6     "bookImage": "/images/books/0df07844-a0cb-4ddb-85fb-1e7516cb86ff.png",
7     "isbn": 2818283829212,
8     "bookPrice": 1200,
9     "bookGenre": "Comedy",
10    "bookLanguage": "English",
11    "bookFormat": "PDF",
12    "bookPublisher": "Rhythm Publication",
13    "bookPublicationDate": "2025-12-12T00:00:00Z",
14    "bookInStock": 12,
15    "bookAddedDate": "2025-05-12T18:47:45.307765Z",
16    "totalSales": 0,
17    "isAwardWinner": true,
18    "isBestseller": true,
19    "discountPercentage": 12,
20    "editionType": "Limited",
21    "isComingSoon": false,
22    "likes": 0,
23    "dislikes": 0,
24    "discountFromDate": "2025-12-12T00:00:00Z",
25    "discountToDate": "2025-12-12T00:00:00Z"
26  },
27  "authors": [
28    {
29      "id": "author-001",
30      "name": "Rhythm Paudel",
31      "biography": "Rhythm Paudel is a talented software developer with expertise in web development and database architecture.
32                  His innovative approaches to solving complex programming challenges have earned him recognition in the tech community."
33    }
34  ]
35 }
```

Figure 95: API Testing: Test 1

3.8.1.2. Test 2 – Update existing book details

Test	2
Objective	Update existing book details
Action	Send PUT /api/Books/updatebook/:id with updated price or stock as admin
Expected Result	Returns 200 OK
Actual Result	Returned 200 OK
Conclusion	Test was successful.

Table 3: API Testing: Test 2

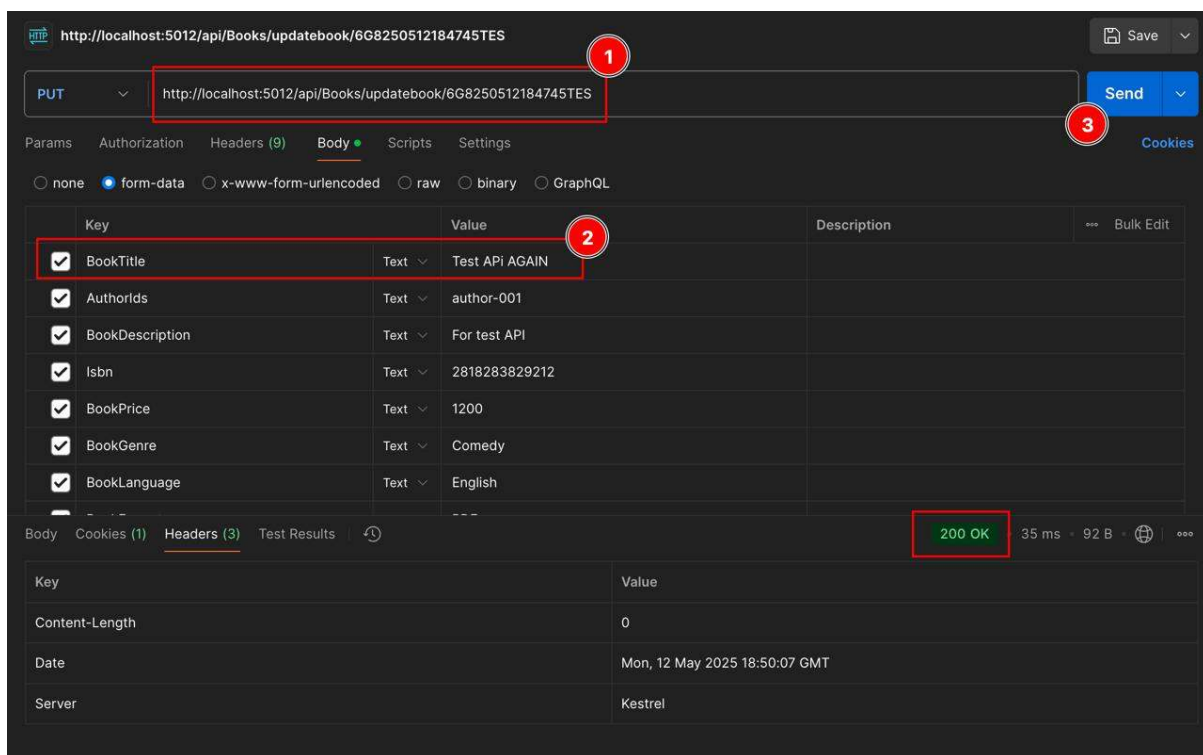


Figure 96: API Testing: Test 2

3.8.1.3. Test 3 – Admin Announcement

Test	3
Objective	Admin Announcement
Action	Send POST /api/Announcement with title, content, and expiry
Expected Result	Returns 201 Created
Actual Result	Returned 201 Created
Conclusion	Test was successful.

Table 4: API Testing: Test 3

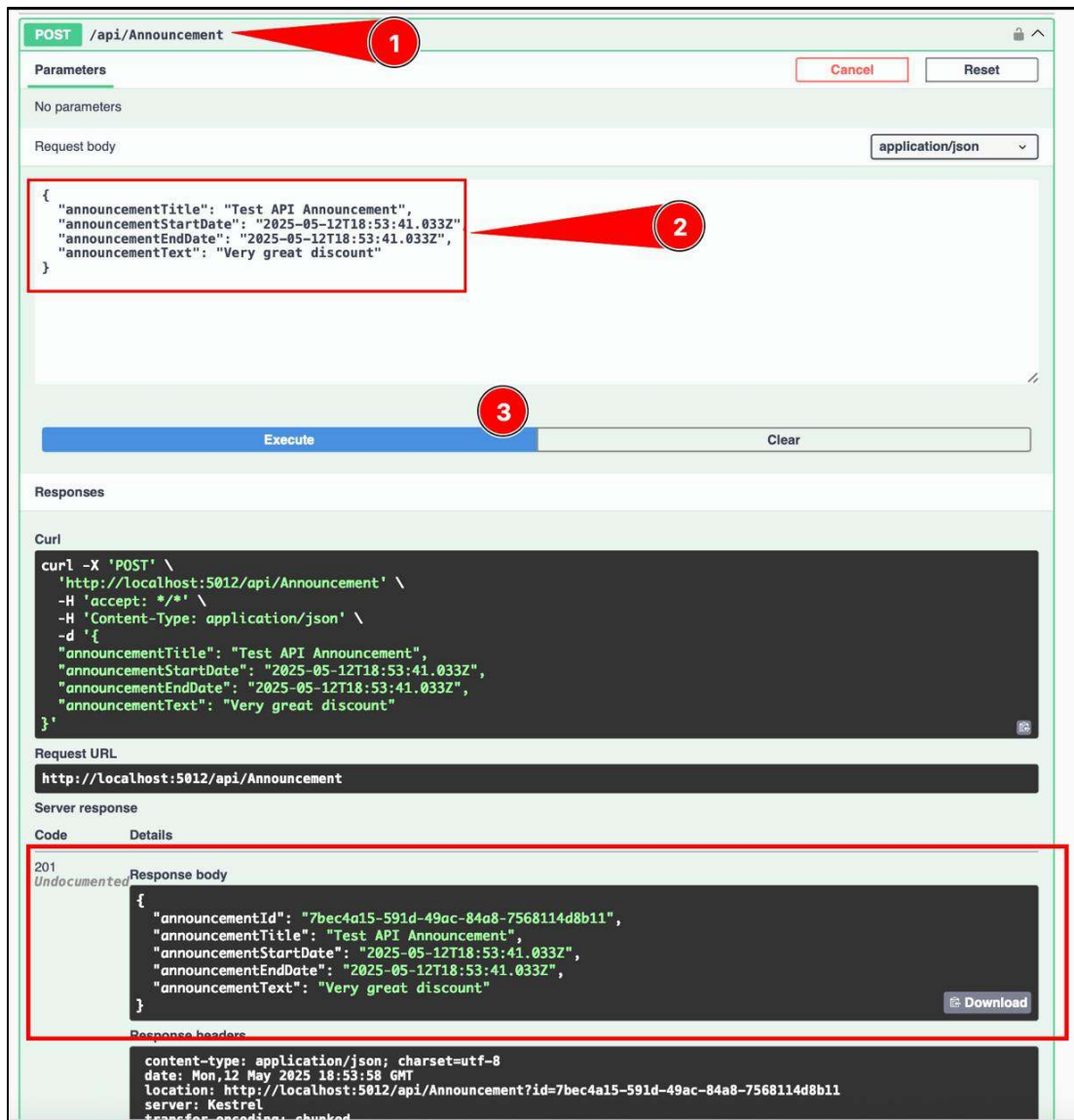


Figure 97: API Testing: Test 3

3.8.1.4. Test 4 – Cancel Placed Order

Test	4
Objective	Cancel Placed Order
Action	Send PATCH /api/Order/cancel/:id
Expected Result	Returns 200 OK and order is marked as canceled
Actual Result	Returned 200 OK and order is marked as canceled
Conclusion	Test was successful.

Table 5: API Testing: Test 4

The screenshot shows an API testing interface with the following details:

- Method:** GET
- URL:** /api/Order
- Parameters:** No parameters
- Execute:** Button to run the request
- Responses:**
 - Code:** 200
 - Response body:**

```
[
  {
    "username": "tanalu",
    "fullName": "Tan Alu",
    "orderId": "25051224825tanalu",
    "orderTotal": 500,
    "orderStatus": "Completed"
  },
  {
    "username": "supris",
    "fullName": "Supris Basne",
    "orderId": "250513000432supris",
    "orderTotal": 754,
    "orderStatus": "Pending"
  }
]
```

The second object in the array is highlighted with a red box, and a red arrow points to it with the number 1.
 - Response headers:**

```
content-type: application/json; charset=utf-8
date: Mon, 12 May 2025 18:58:20 GMT
server: Kestrel
transfer-encoding: chunked
```
 - Responses table:**

Code	Description	Links
200	OK	No links

Figure 98: API Testing: Test 4

Here in the above screenshot, the order status is pending, this needs to be changed to order status cancel.

The screenshot shows an API testing interface for a PUT request to `/api/Order/cancel/{orderId}`. The `orderId` parameter is set to `250513000432supris`. The `Execute` button has been clicked, resulting in a 200 status code response. The response headers are displayed, indicating a successful cancellation.

Parameters

Name	Description
<code>orderId</code> • required string (path)	250513000432supris

Execute **Clear**

Responses

Curl

```
curl -X 'PUT' \  
  'http://localhost:5012/api/Order/cancel/250513000432supris' \  
  -H 'accept: */*'
```

Request URL

```
http://localhost:5012/api/Order/cancel/250513000432supris
```

Server response

Code	Details
200	Response headers content-length: 0 date: Mon, 12 May 2025 19:01:27 GMT server: Kestrel

Responses

Code	Description	Links
200	OK	No links

Figure 99: API Testing: Test 4

Here the status of the order is changes to cancelled. The order status is shown below while getting the specific order again.

The screenshot shows an API testing interface with the following sections:

- GET /api/Order**: The request method and URL, highlighted with a red box.
- Parameters**: No parameters are present.
- Execute** and **Clear** buttons.
- Responses**:
 - Curl**:

```
curl -X 'GET' \
'http://localhost:5012/api/Order' \
-H 'accept: */*'
```
 - Request URL**: `http://localhost:5012/api/Order`
 - Server response**:
 - Code**: 200
 - Details**:
 - Response body**:

```
{
  "username": "tanalu",
  "fullName": "Tan Alu",
  "orderId": "25051222482Stanalu",
  "orderTotal": 500,
  "orderStatus": "Completed"
},
{
  "username": "supris",
  "fullName": "Supris Basne",
  "orderId": "250513000432supris",
  "orderTotal": 754,
  "orderStatus": "Cancelled"
}
]
```

A red circle with the number 1 points to the "Cancelled" status.
 - Response headers**:

```
content-type: application/json; charset=utf-8
date: Mon, 12 May 2025 19:02:03 GMT
server: Kestrel
transfer-encoding: chunked
```
 - Responses** table:

Code	Description	Links
200	OK	No links

Figure 100: API Testing: Test 4

3.8.1.5. Test 5 – Submit Book Review

Test	5
Objective	Submit Book Review
Action	Send POST /api/Review with review
Expected Result	Returns 200 OK.
Actual Result	Returned 200 OK.
Conclusion	Test was successful.

Table 6: API Testing: Test 5

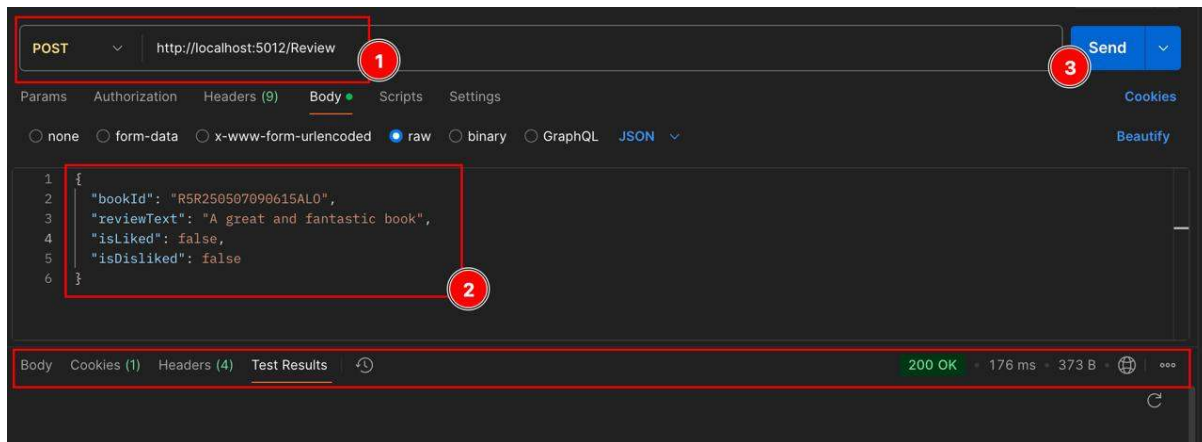


Figure 101: API Testing: Test 5

3.8.2. Frontend Unit/System Testing

3.8.2.1. Test 1 – Update Existing Book Details

Test	1
Objective	Update Existing book details
Action	Admin logs in → Edit a book in UI → Save
Expected Result	Updated info reflects in the catalog and book details
Actual Result	Updated info was reflected in the catalog and book details
Conclusion	Test was successful.

Table 7: Unit/System Testing: Test 1

PRG250512173820CRU	CRUD - Creating a Book	Thriller	Spanish	1	\$1111	  
--------------------	------------------------	----------	---------	---	--------	---

Figure 102: Unit/System Testing: Test 1

After the edit button is clicked, the update book pop up appears where the details can be updated.

Figure 103: Unit/System Testing: Test 1

The title of the book is changed to CRUD - Updating a book, after the book details is updated.

PRG250512173820CRU	CRUD - Updating a Book	Thriller	Spanish	1	\$1111	  
--------------------	------------------------	----------	---------	---	--------	---

Figure 104: Unit/System Testing: Test

3.8.2.2. Test 2 – Filter books by Genre

Test	2
Objective	Filter books by Genre
Action	On catalogue page, select a genre filter
Expected Result	Only books of selected genre are displayed
Actual Result	Only books of selected genre were displayed
Conclusion	Test was successful.

Table 8: Unit/System Testing: Test 2

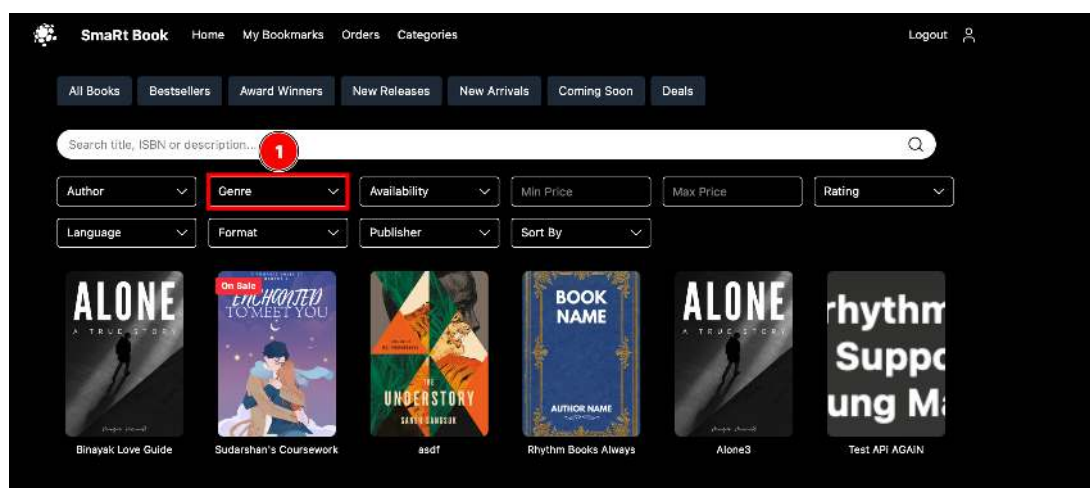


Figure 105: Unit/System Testing: Test 2

In the above screenshot, the genre should be selected, Science genre is selected.

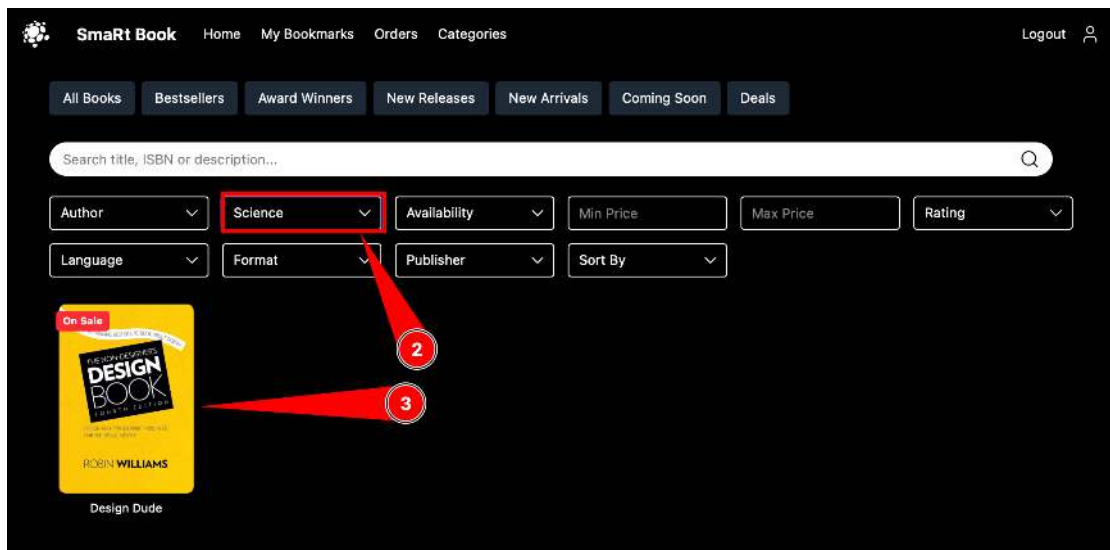


Figure 106: Unit/System Testing: Test 2

3.8.2.3. Test 3 – Place order by applying discount on 5+ books order

Test	3
Objective	Place order by applying discount on 5+ unique book order
Action	Add 5+ unique book to cart → Proceed to checkout
Expected Result	5% discount is shown and applied in total
Actual Result	5% discount was shown and applied in total
Conclusion	Test was successful.

Table 9: Unit/System Testing: Test 3

In the screenshow below, the items are added to cart. There are 6 items added to cart and checkout button is clicked.

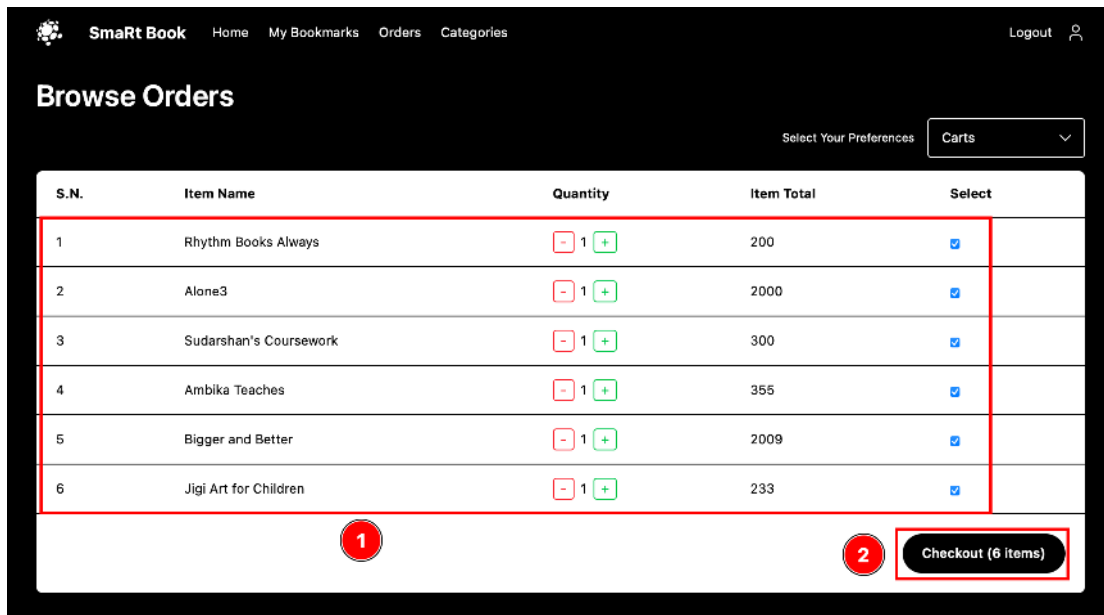


Figure 107: Unit/System Testing: Test 3

Here, after the order is placed, the order is shown to the user in browse orders page. The view button needs to be clicked to view the order details.

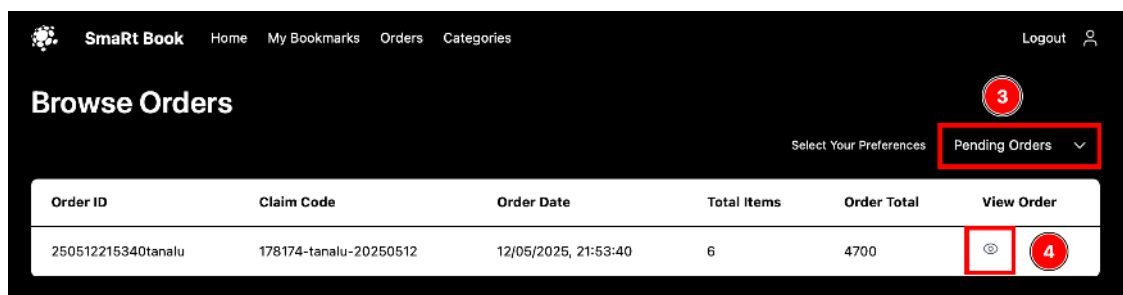


Figure 108: Unit/System Testing: Test 3

Order No. 250512215340tanalu

Claim Code: 178174-tanalu-20250512

Status: Pending

Cancel Order

S.N.	Item Name	Quantity	Unit Price	Line Total
1	Sudarshan's Coursework	1	300	300
2	Bigger and Better	1	2009	2009
3	Jigi Art for Children	1	233	233
4	Ambika Teaches	1	355	355
5	Rhythm Books Always	1	200	200
6	Alone3	1	2000	2000
Order Total				5097
Discount				397
After Discount				4700

Figure 109: Unit/System Testing: Test 3

In the above shown screenshot, the applied discount is shown to the user and the amount after discount too is mentioned.

3.8.2.4. Test 4 – Leave Book Review after purchase

Test	4
Objective	Leave Book Review after purchase
Action	After order delivery, go to book page → Submit review
Expected Result	Review is posted and visible on book details
Actual Result	Review was posted and visible on book details
Conclusion	Test was successful.

Table 10: Unit/system Testing: Test 4

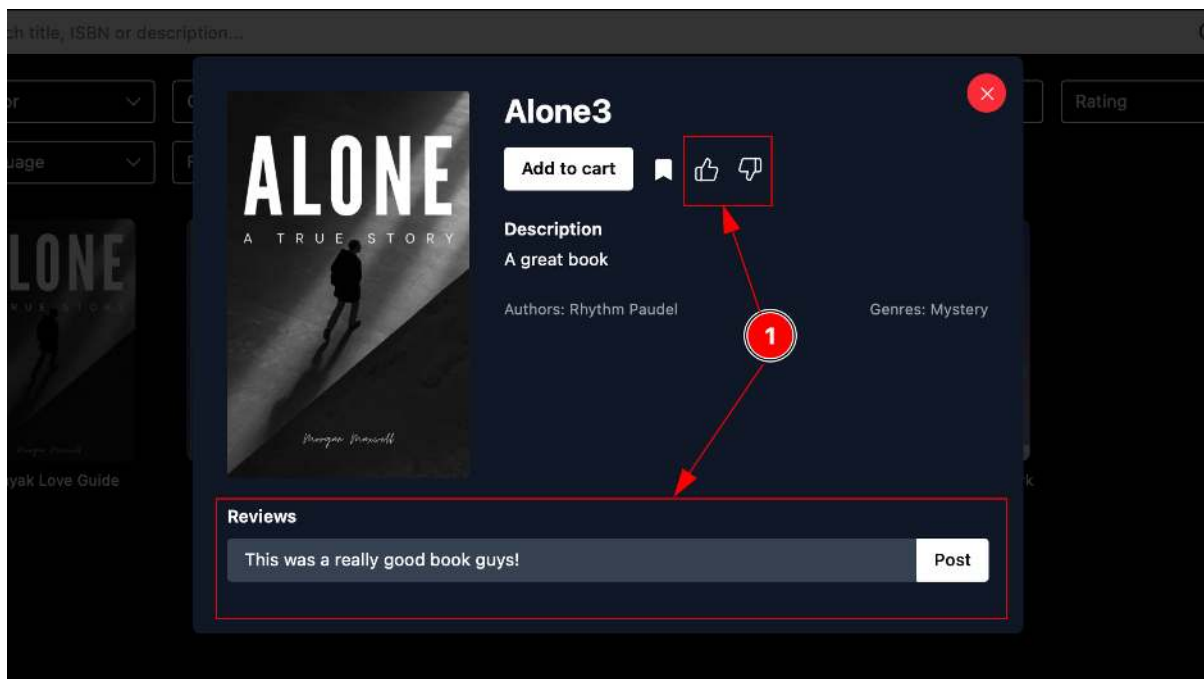


Figure 110: Unit/System Testing: Test 4

The above screenshot shows the member posting review on the book. The screenshot below shows the review on the book details popup after the review is posted.

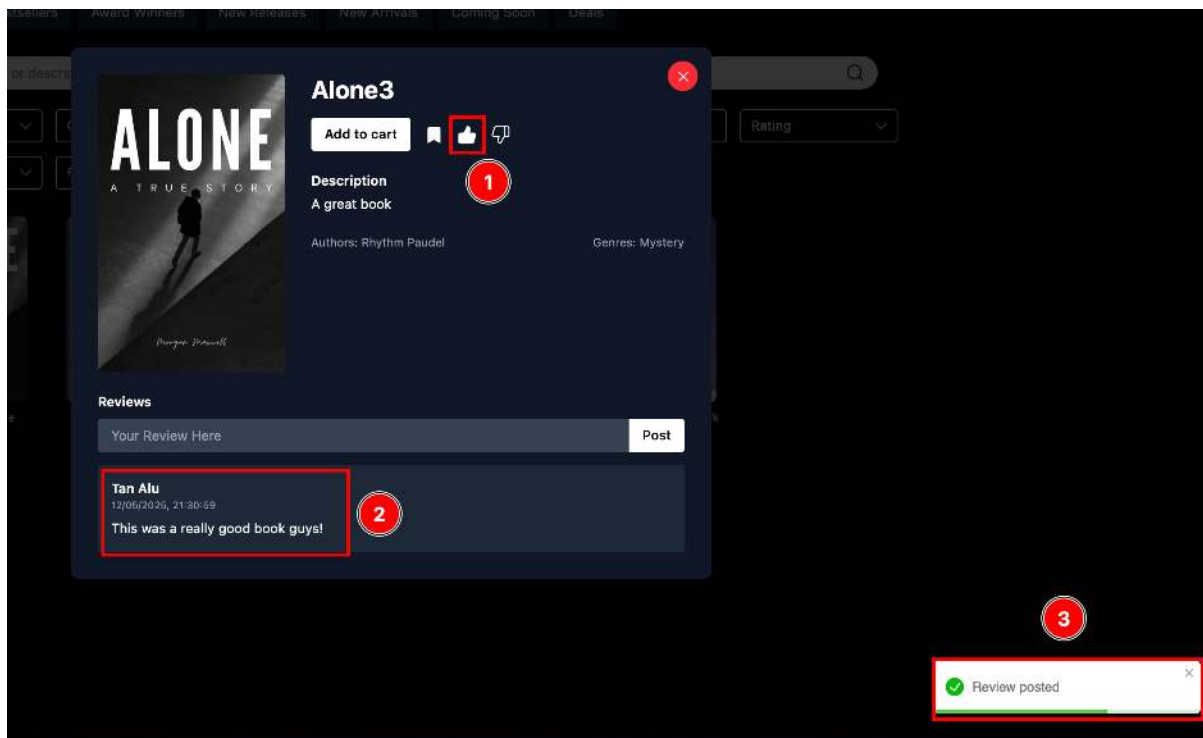


Figure 111: Unit/System Testing: Test 4

3.8.2.5. Test 5 – Apply Discount to Specific Book

Test	5
Objective	Apply Discount to Specific Book
Action	Admin marks a book as "On Sale" with 25% discount
Expected Result	Book shows discounted price and "On Sale" label
Actual Result	Book showed discounted price and "On Sale" label
Conclusion	Test was successful.

Table 11: Unit/System Testing: Test 5

Here the admin can add the discount in particular book by setting the discount to desired percentage and the start date and expiry date of the discount.

Figure 112: Unit/System Testing: Test 5

After adding the discount, in the user portal of the site, the book can be seen with the On Sale banner on top of it.

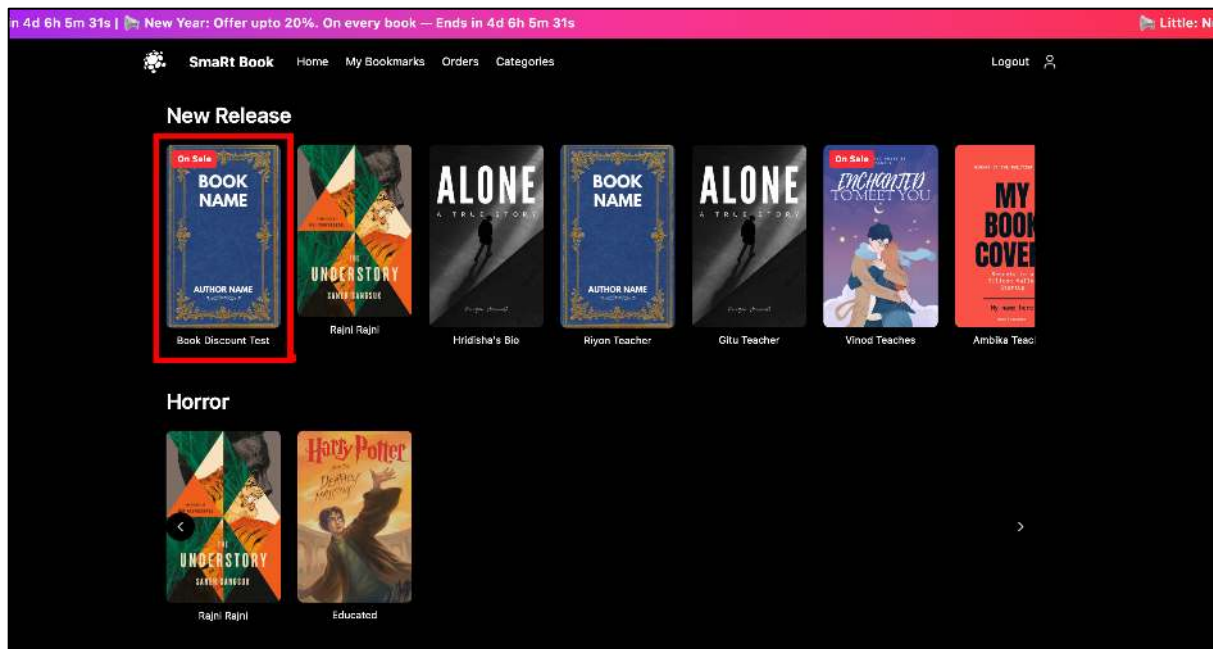


Figure 113: Unit/System Testing: Test 5

4. Individual Reflections

4.1. Roles and Responsibilities

→ Rhythm Paudel

- Solution Architecture
- Development of Admin Panel
- Order Management Features
- Front-End Interface Development
- Catalogue Management Functionality
- API Tester
- Member Account and Services Integration
- Order Workflow Implementation
- Discount Feature Development
- Order Processing Feature
- System Testing
- QA and Documentation

4.2. Personal Insights

During this project each I developed a deeper understanding of how to contribute to a large. I learned integration of two different frameworks (react in frontend and dotnet in backend) and its difficulties. The importance of early designing of database schemas and early visualization of system features and their interactions for better user experience was also realized while working on this project.

4.3. Challenges

- Integration of mail service during the order process.
- Implementation of clean code for proper understanding of all members.
- Proper data abstraction through DTOs.
- Debugging minor errors during the build process.

4.4. Learnings

- Improved my knowledge of full stack development, web API structuring and implementation, modular design and data abstraction.
- Realized the importance of version controlling through git and asynchronous task management for smooth running of the application.
- Understood the use of Identity framework in user authentication and registration, role-based login implementation and layered architectures.

4.5. Growth

I gained confidence in building full stack applications that are stable and scalable. I also learned about API structuring, end points etc. and feel I am better equipped for real work environments. I have managed to hone our ability to adequately divide tasks and manage and integrate complex systems in a modular way.

4.6. Impact on Future Work

Working on this web application project has given me confidence to pursue even more challenging and complex web applications built using API structures. I now have a much clearer idea about full stack development and working in real-time team environments.

This project is a strong push for our portfolio and has molded my thinking toward systematic thinking and agile development processes. Such takeaways will be invaluable for my future endeavors whether that maybe internships, active jobs, future academic pursuits and competitions.

4.7. Personal Evaluation

I was focused, motivated and was able to properly achieve my goals. While there were some occasional hiccups along the way, I tackled problems effectively for the completion of this project.

5. Conclusion

The main aim of this project was to develop a full stack web platform for a book library store to display and retail its book catalogues to users online. The library management system's key functionalities included: book catalogue browsing, filtering based on genre, format etc., role-based login, user/book management, order/ cart, staff order validation, mail order confirmation etc.

During this project I found that features such as role-based login, feature rich catalogues, individual and stackable discounts provide for a better user experience and are factors that determine the user's return to the store. In the future versions, online payment systems, delivery, real-time inventory updates etc. will further help to enhance user experience.

Although this app provides many features, it is also limited by in-store only pickup, lack admin functions/dashboard, more difficult to use by less tech savvy or older users, etc. Hence, in the future enhancing the web applications capabilities by implementing third party APIs such as Google Books, ML recommendation systems and enhanced accessibility to encourage wide base of users can be good future work updates to the application.

To conclude, this project is capable of fully showcasing the market feasibility and future scalability of the .NET Web API framework in other applications as well. It is a fully functional web application that allows users to browse and order books of their choice from a catalogue of multiple books of all types. This project is a good example of proper implementation of full stack development using both frontend frameworks like react and backend frameworks like .NET Web API and their integration.