

PROJECT REPORT

BOOKSHOP WEB APPLICATION



Y. AMADI ANUTHTHARA
AR/96217
AF/18/14251
DEPARTMENT OF ICT
FACULTY OF HUMANITIES & SOCIAL SCIENCES
UNIVERSITY OF SRI JAYEWARDENEPURA



BOOKSHOP WEB APPLICATION

**Project Proposal submitted in partial fulfillment of the requirement of ICTC3101.3
and Web Application Development of the Degree of BSc in Information Technology**

Y. Amadi Anuththara

AR/96217

AF/18/14251

Department of ICT

Faculty of Humanities & Social Sciences

University of Sri Jayewardenepura

16/07/2022

Table of Contents

1. Introduction.....	3
2. Motivations	4
3. Problem and Objectives	5
3.1. Problem.....	5
3.2. Objectives	6
4. Requirement Specifications.....	7
4.1. Main Requirements	7
4.2. Critical Use Case Requirements.....	7
5. System overview	8
<u>5.1. ER-Diagram for Bookshop Web Application</u>	8
<u>5.2. Background</u>	8
<u>5.3. GUI Design</u>	11
6. Database Structure	18
7. A detailed explanation of each API implementations and respective Endpoints.....	21
8. Summary.....	28
9. Reference	29

1) INTRODUCTION

The main objective of the project is to create an online book shop that allows users to search and purchase a book online based on title, author and subject. The selected books are displayed in a tabular format and the user can order their books online through credit card payment. Using this Website, the user can purchase a book online instead of going out to a book store and wasting time.

I develop a website using “Mern Stack” with MongoDB, Express, React, NodeJS. Online Book shop is an online web application where the customer can purchase books online. Through a web browser the customers can search for a book by its title or author, later can add to the shopping cart and finally purchase using credit card transaction. The user can login using his account details or new customers can set up an account very quickly. They should give the details of their name, contact number and shipping address. The books are divided into many categories based on subject like Biography, Children and Fiction etc.

The Online Book Shop Website provides customers with online shopping through a web browser. A customer can, create, sign in to his account, place items into a shopping cart and purchase using his credit card details. The Administrator will have additional functionalities when compared to the common user. He can add, delete and update the book details, book categories, member information and also confirm a placed order. The Master page, data sets, data grids, user controls are used to develop the Online Bookshop.

2) MOTIVATIONS

The existing system transactions are not performing properly and overcome some of the issues in entire system modules such as Purchase, Sales, Stock Management and Report generations are inaccurate, time wasting, inefficient & unreliability of the semi-automated system due to increase of new customers and sales through existing website of wisdom.

The above-mentioned issues in existing system, so need to eliminate therefore Book shop Management for Wisdom taken decision to make fully computerized the system then it will be help to provide better customer solution and lead the business to a success in a competitive business market. These problems occur during last years because of the large number of data volume processed due to increase of new customers, books and branches.

Considering this problem, proposed a web-based solution to bookstore management system for Wisdom to reach the finest level to fulfil the quality & standard of requirement to the customer's due to unavailability of new books in the store operations. This solution is built in a way to cover the overall administration process of store within all modules.

Finally, there is a special interest and willingness to make use of the web-based technologies and development of web applications. Provide the searching facilities based on various factors. Such as Book, Order, Order Status, Stock, Online Book Store also manage the Book Store Details online for Order Status details, Stock details, Books, Manage the information of Customer, Shows the information and description of the Books, Order, to increase efficiency of managing the Books, Customer and It deals with monitoring the information and transaction of Order Status are major parts of this web application.

3) PROBLEM AND OBJECTIVES

Problem

- Online Book Shop is a specific requirement of the client that integrates the buying and selling services specifically to their customers.
- Reports can be generated at any time within few seconds, so that manual labor is not required, and also analysis can be performed much more frequently which helps in taking decision.
- The details regarding all users, books can also be maintained as their information is very helpful and sometimes becomes a critical requirement.
- Allows user to get registered from their places and transact for the required product.
- To overcome these problems, we develop “Online Book Shop”.

Objectives

The overall objective of the project is to enhance and improve the effectiveness and efficiency of the service provided by the business by introducing a full web-based system which can be divided further as follows,

- To improve the efficiency of daily business process such as purchasing, sales and stock controlling by eliminating the considerable delays and bottlenecks.
- To enhance the accuracy of the business process such as Purchase, Sales, Stock Management via automated Web Based Bookstore Management System for bookshop.
- To eliminate the delay and inaccuracy in providing details needed to the external parties such as customers and suppliers, through which customer business-supplier relationship can be strengthened.
- To enhance the business decision making process by providing the necessary reports timely when needed.
- To preserve the historical business data related to purchasing and sales.
- To decrease the work load and responsibilities by decentralizing the work load and responsibilities among the staff members.
- To eliminate unnecessary paper working & workload in the daily business activities.
- To enhance the security and recoverability of the business-critical data by introducing critical security features such as access control mechanism, authentication and accountability.

4) REQUIREMENT SPECIFICATIONS

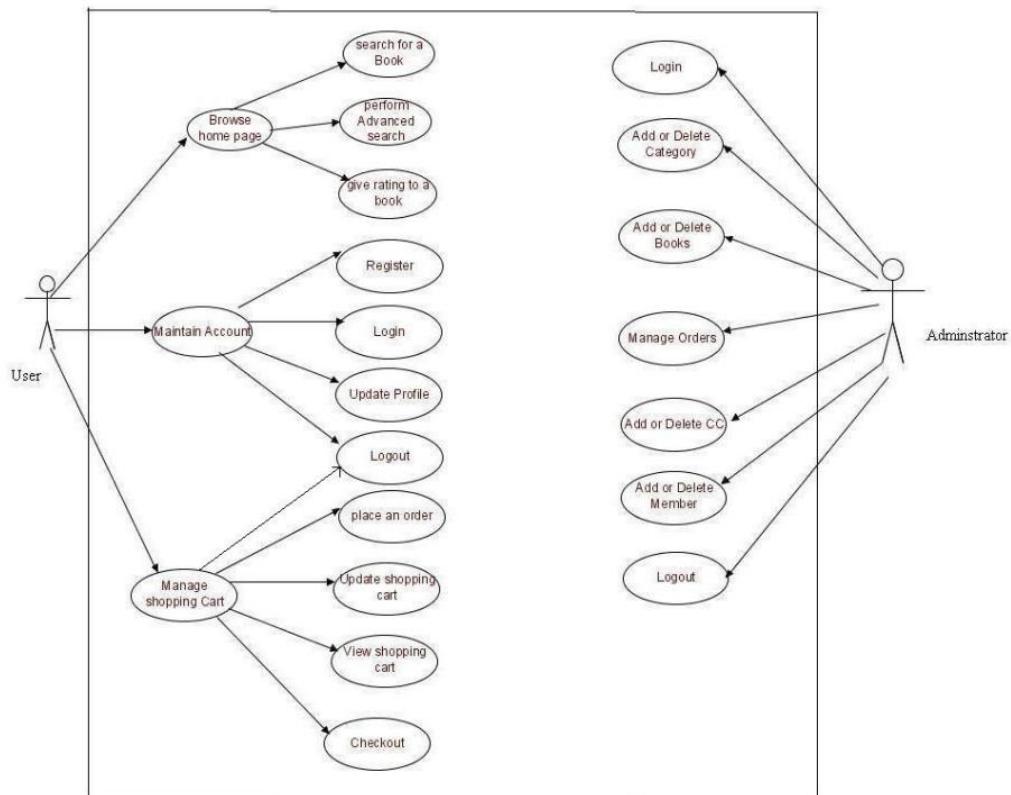
Main Requirements

The Main Requirements include Microsoft Visual Studio to develop the web application, mongo DB to design the database and Google Chrome as a main web browser to run the website.

Critical Use Case Requirements

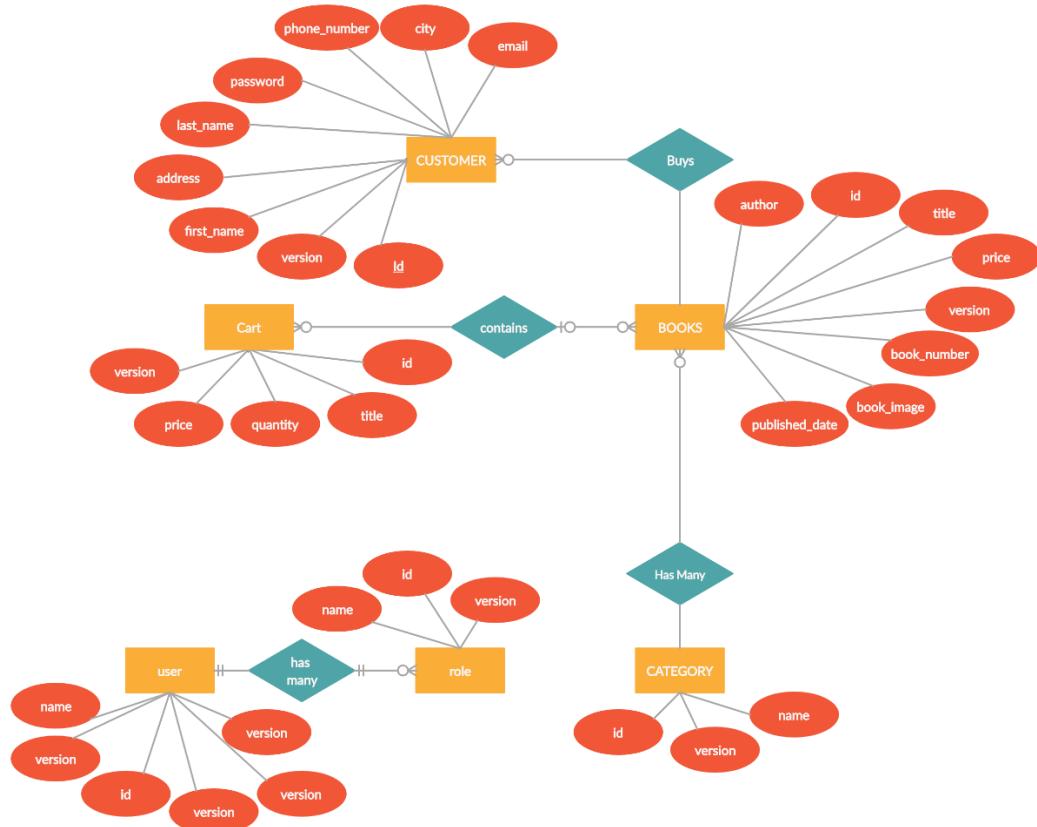
User and Administrator are the two actors included in the Online Book Store. Below image shows the use case diagram for this website.

Use Case Diagram



5) SYSTEM OVERVIEW

ER-Diagram for Bookshop Web Application



Background

Online Book Shop is an online web application where the customer can purchase books online. Through a web browser the customers can search for a book by its title or author, later can add to the shopping cart and finally purchase using credit card transaction. The user can login using his account details or new customers can set up an account very quickly. They should give the details of their name, contact number and shipping address. The books are divided into many categories based on subject Like Biography, Children and Fiction etc.

This project has the following functionalities:

1. A Home page with product catalogue

This is the page where the user will be navigated after a successful login. It will display all the book categories and It also includes some special sections like recommended titles, weekly special books.

2. Filter

A filter by keyword option is provided to the user using a select box. User can filter product by Order and Arrivals.

3. Advanced search

Advanced search helps the user to search for a book based on Title, Author, Category and price range. All the books which match the particular search criteria and their total count will be displayed. From here the user can select a book and add to the shopping cart.

4. Book Description

If the user would like to know details about a book he can click on the title from where he will be directed to a Book description page. It includes the notes on the book content.

5. Shopping Cart

The user can manage a shopping cart which will include all the books he selected. The user can edit, delete and update his shopping cart. A final shopping cart summary is displayed which includes all the items the user selected and the final total cost.

6. Managing user accounts

Each user should have an account to access all the functionalities of website. User can login using login page and logout using the logout page.

7. Administration

The Administrator will be provided with special functionalities like

- Add or delete a book category
- Add or delete a member

- Manage member orders

8. Logout

If the user wants to end his session and sign out of the website then he can use the logout option. The user will click the logout button. The user's account session comes to an end and he should login again if he wants to enter into the website.

9. Manage Shopping Cart

The user will enter the quantity required and click the add to shopping cart button. The order will be added to the user's shopping cart. If the user wants to change the quantity of a book or change a book then he can update his shopping cart. The user will click the book in the category item page to edit and update his order details. The updated order details are reflected in the shopping cart summary.

10. View Shopping Cart

If the user wants to view the items he added to the shopping cart, then he can click the shopping cart link at the top of the page. The user will click the shopping cart link at the top of every page. The user's shopping cart summary will be displayed in the form of a tabular format with all the books and their quantity. A total cost of all the items is also displayed at the bottom.

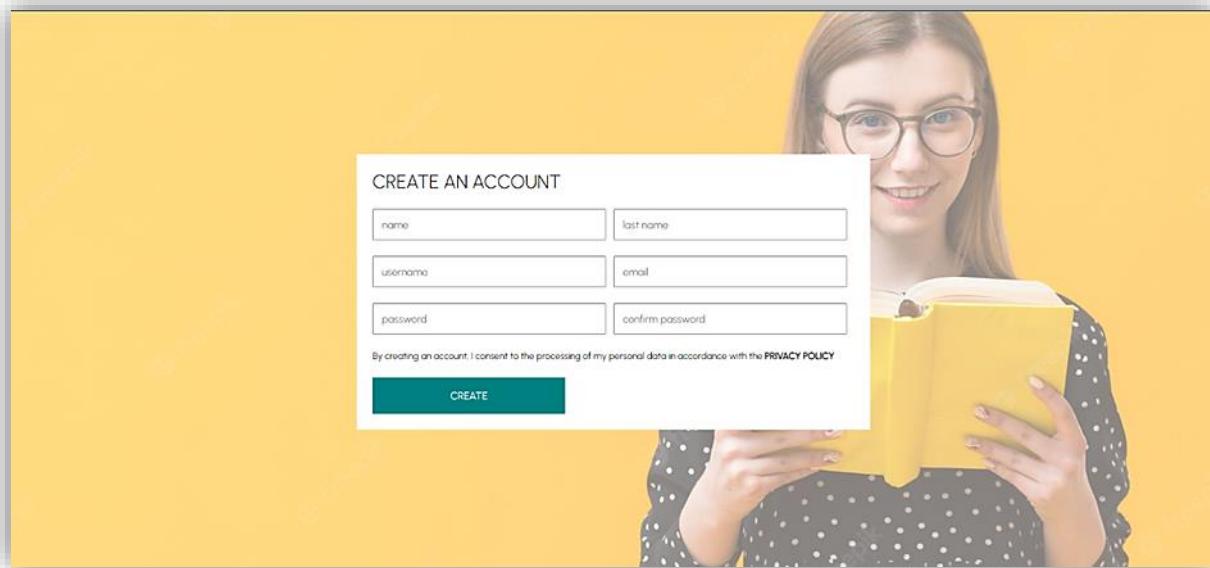
11. Add or Delete Category

If the Administrator wants to add or delete a book category then he can insert or delete a book category using his administration rights and the category table will be updated in the database. If the Administrator wants to add a book category then he should click the insert link button in the category page else he can delete a particular selected book. The updated categories list will be displayed in the main home page.

GUI Design

The GUI Builder makes it possible to build professional-looking GUIs without an intimate understanding of layout managers. I used CSS and HTML to design interfaces of the app in VS code and java script used for do a functionality of the web application. Seven interfaces have included in this app. Home Screen, Login Screen, Register Screen, Product Screen, Product Description Screen and Cart Screen are the Interfaces of this Bookshop web application.

1) Register Screen



If the user doesn't have an account, then he will be asked to register. The user is new to this Web application, they can create the account using "create an account" User Interface. Clicking 'create a new account' text, they can navigate to this interface.

The user will enter details in the registration form according to the required fields.

The fields include

1. Username
2. Password

3. confirm password

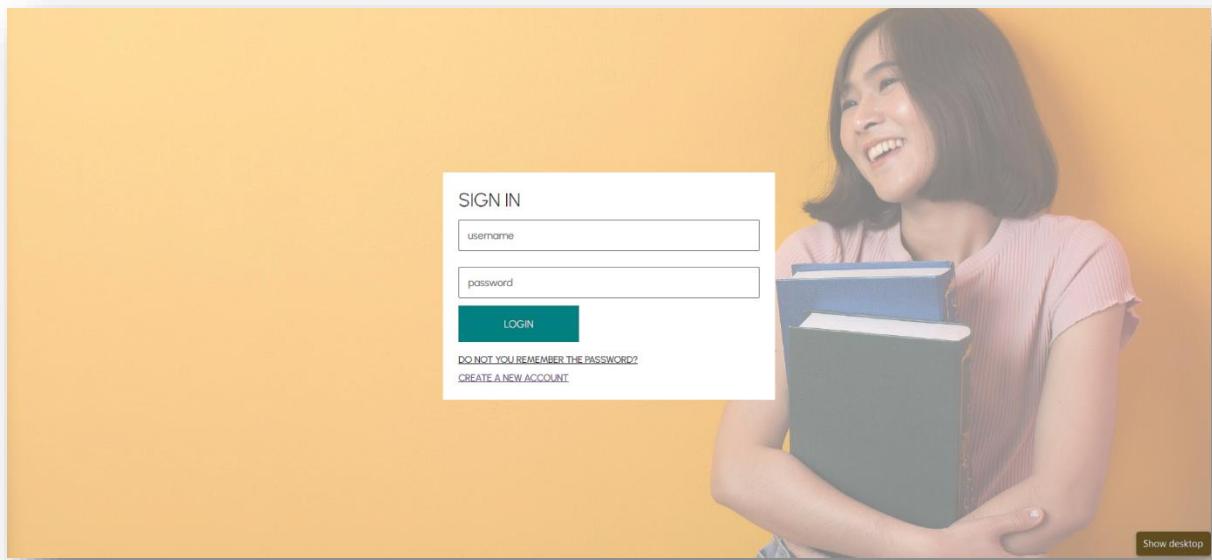
4. first name

5. last name

6. email

After registration the user will be directed to the main home page.

2) Login Screen



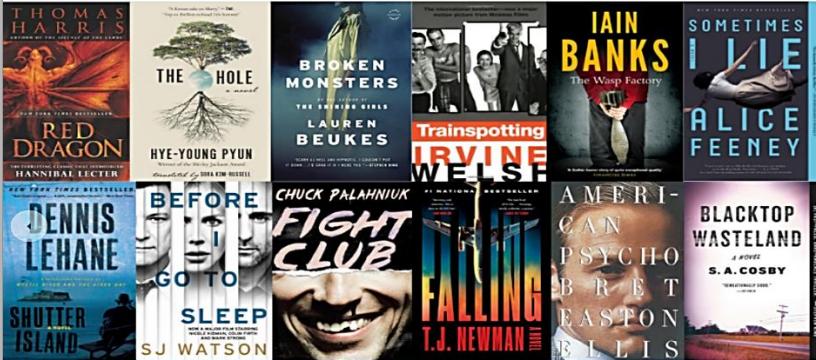
If the user wants to get access to all the functionalities of Online Book Shop, he should login using his username and password. The user will enter his username and password. Output: If it is a successful login the user will be directed to the main home page. Else if the user enters invalid information, he will be asked to check the entered information.

3) Home Screen

New month new 50% off sale! Start saving here.

BOOKDEPOT

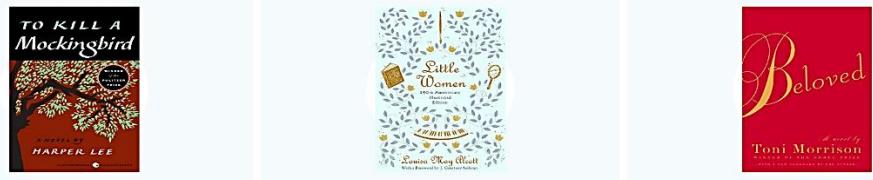
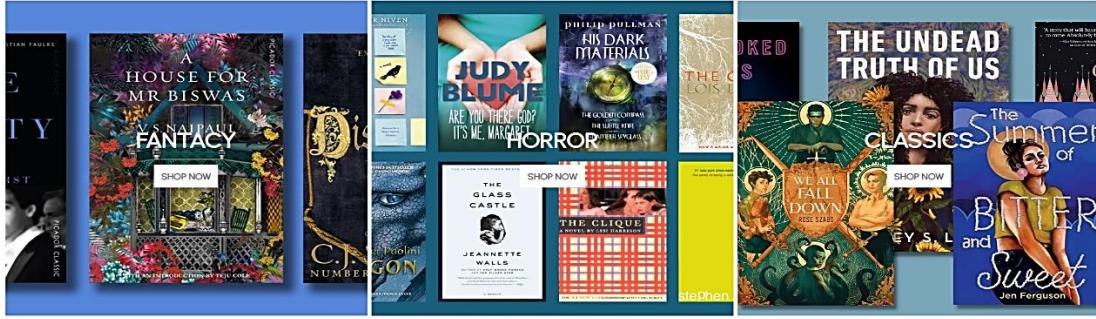
REGISTER SIGN IN 



NEW PRODUCT

Discover new books, reviews, and interviews with your favorite authors.

[SHOW NOW](#)



Newsletter

Get timely updates from your favorite books.

Your email 

BOOKDEPOT

We offer the vast selection of titles, competitive prices, and fast processing times that you are looking for in a book supplier. Our large inventory of high-margin bargain books is 75-90% off list price and easily browsable on our website — so you can stock the product your customers want, decrease your purchasing time, and improve your bottom line!



Useful Links

[Home](#) [Cart](#)
[New Product](#) [Featured Product](#)
[Accessories](#) [My Account](#)
[Order Tracking](#) [Wishlist](#)
[Wishlist](#) [Terms](#)

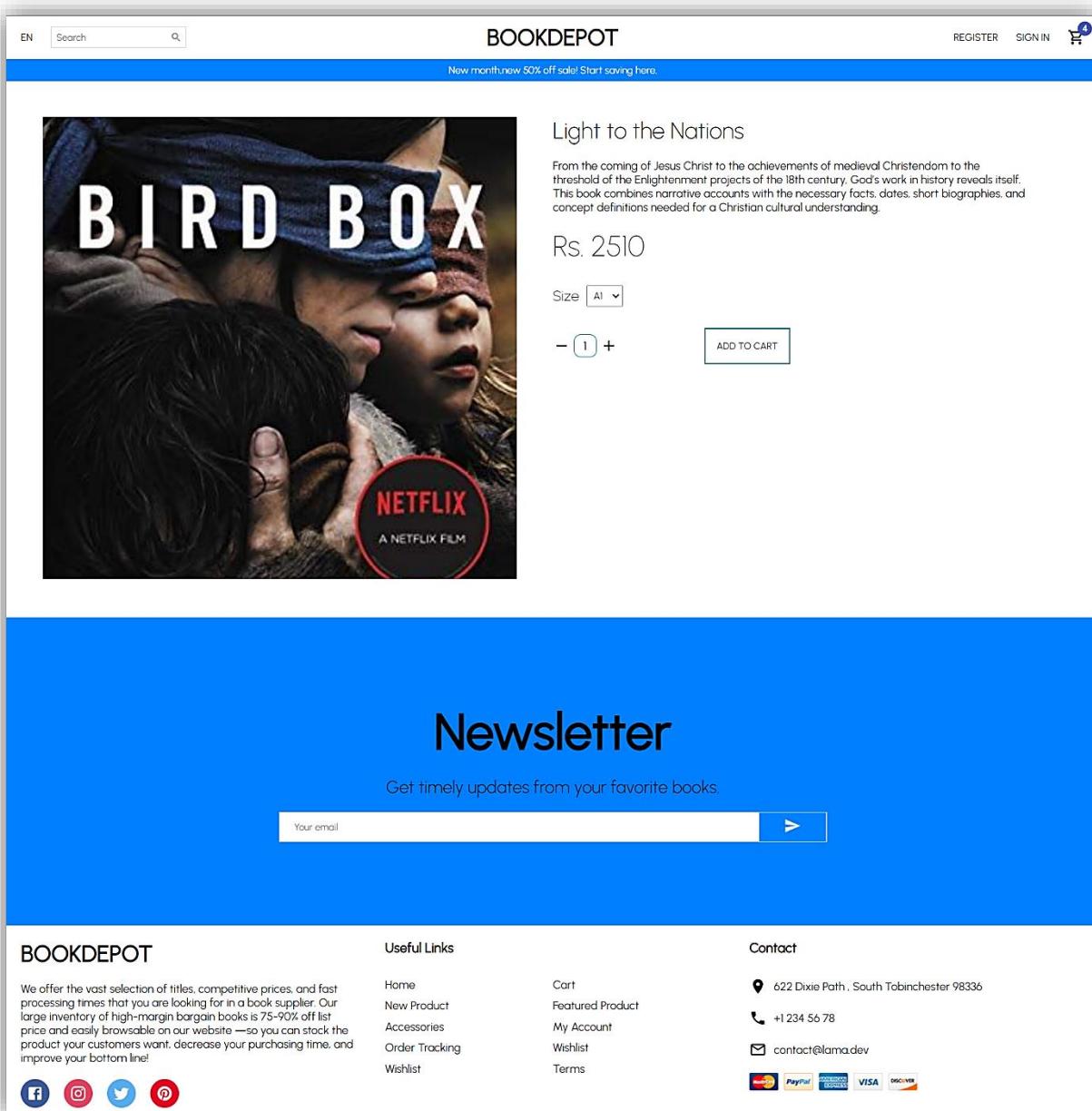
Contact

622 Dixie Path, South Tobinchester 98336
+1 234 56 78
contact@loma.dev



When we login the Online Book Shop Website first home page is displayed. The home page will appear as below. Any common user can use the search option in the home page to search for a book on his choice by entering a Title of the book. The bookshop announcement is available in the announcement bar. And also, home screen has three slides that has new products, featured products and collectible books. Home screen has category items that can access clicking “Show Now” button. This home screen already available in new books that can also user to get into their cart easily. If user have any issues related to the product, they can email the problem using newsletter and user can access bookshop details in home page footer.

4) Product Screen



The screenshot shows a product page for the book "Bird Box". The main image is a movie poster for the Netflix film, featuring a woman with her eyes covered by a cloth. The title "BIRD BOX" is prominently displayed in white letters across the top of the poster. A small circular logo in the bottom right corner reads "NETFLIX A NETFLIX FILM". To the right of the image, the book's title "Light to the Nations" is listed, followed by a brief description: "From the coming of Jesus Christ to the achievements of medieval Christendom to the threshold of the Enlightenment projects of the 18th century, God's work in history reveals itself. This book combines narrative accounts with the necessary facts, dates, short biographies, and concept definitions needed for a Christian cultural understanding." Below the description is the price "Rs. 2510". A dropdown menu for "Size" is set to "A1". Below the price are quantity controls ("-", "+") and an "ADD TO CART" button. At the bottom of the page, there is a blue newsletter section with the heading "Newsletter" and a form for entering an email address.

BOOKDEPOT

We offer the vast selection of titles, competitive prices, and fast processing times that you are looking for in a book supplier. Our large inventory of high-margin bargain books is 75-90% off list price and easily browsable on our website —so you can stack the product your customers want, decrease your purchasing time, and improve your bottom line!

Useful Links

[Home](#)
[New Product](#)
[Accessories](#)
[Order Tracking](#)
[Wishlist](#)

[Cart](#)
[Featured Product](#)
[My Account](#)
[Wishlist](#)
[Terms](#)

Contact

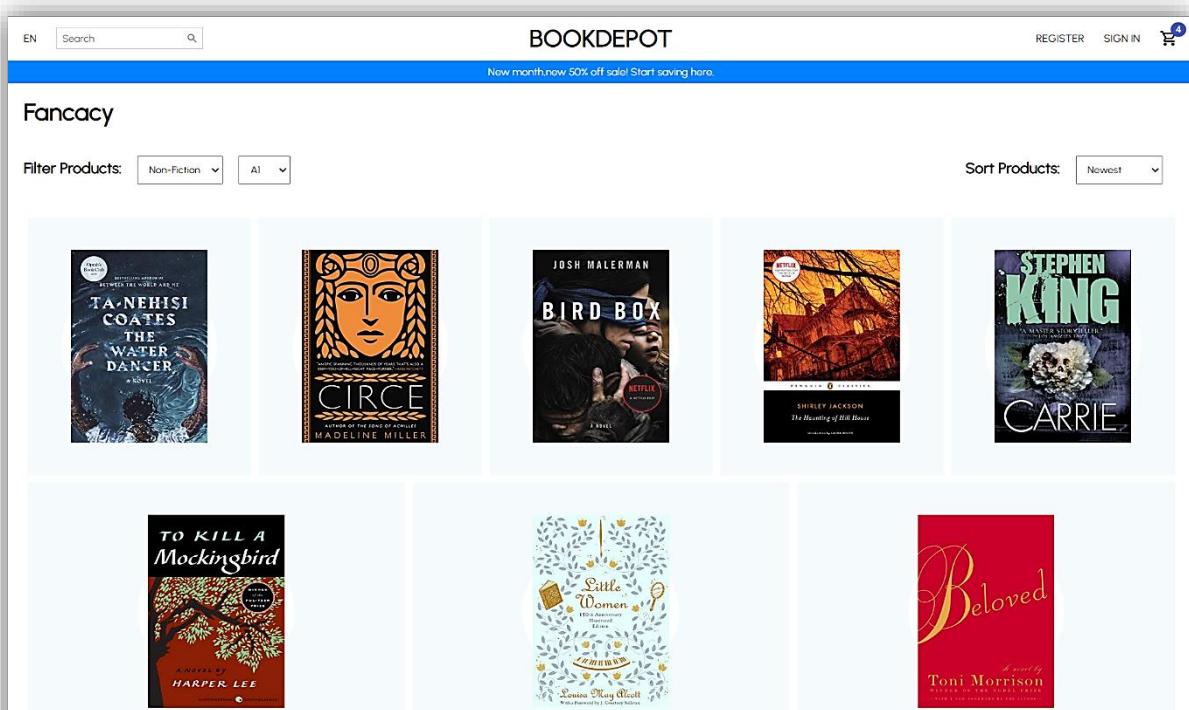
[622 Dixie Path, South Tobinchester 98336](#)
[+1 234 56 78](#)
contact@lama.dev

If the user is interested in any particular book, he can click the book image to get the book details. If he wants to search more precisely, he can use the search option on the product page. If user need to buy this product, user can add the product to the cart using “Add to Cart” Button.

The book details page will appear as below.

1. Book Name
2. Description
3. Product Price
4. Book Size
5. Book Quantity
6. Add to cart

5) Products Screen



The screenshot shows the BookDepot website's products screen. At the top, there is a navigation bar with 'EN', a search bar, and links for 'REGISTER' and 'SIGN IN'. A shopping cart icon with a count of 4 is also present. Below the navigation, a banner says 'New month,new 50% off sale! Start saving here.' The main content area is titled 'Fancacy' and features a grid of ten book covers. The books include 'The Water Dancer' by Ta-Nehisi Coates, 'Circe' by Madeline Miller, 'Bird Box' by Josh Malerman, 'Carrie' by Stephen King, 'To Kill a Mockingbird' by Harper Lee, 'Little Women' by Louisa May Alcott, and 'Beloved' by Toni Morrison. Below the grid, a large blue banner reads 'Newsletter' and encourages users to 'Get timely updates from your favorite books.' There is a text input field for an email address and a '►' button. At the bottom, there are sections for 'BOOKDEPOT' (with a brief description), 'Useful Links' (including Home, New Product, Accessories, Order Tracking, Wishlist, Cart, Featured Product, My Account, Wishlist, Terms), and 'Contact' information (address: 622 Dixie Path, South Tobinchester 98336, phone: +1 234 56 78, email: contact@lama.dev, payment methods: Credit Card, PayPal, American Express, VISA, DISCOVER).

To search for a book the user can use the search using book title. The user can search for a book based on price range, book title and Category. This option is provided in the products page. User can available every product related to the category and book sizes. And also, user can sort the products related to the newest and Price.

6) Cart Screen

The screenshot shows the BookDepot website's shopping cart page. At the top, there is a navigation bar with 'EN', a search bar, and links for 'REGISTER' and 'SIGN IN'. The main header 'BOOKDEPOT' is centered above a promotional banner 'Now month,new 50% off sale! Start saving here.' Below the banner, the heading 'YOUR BAG' is displayed. To the right of the heading are links for 'Shopping Bag(2)' and 'Your Wishlist (0)'. A large blue 'CHECKOUT NOW' button is positioned on the right side of the cart summary.

Product: JESSIE
ID: 93813718293
Qty Avail: 647

Product: HAKURA
ID: 93813718293
Qty Avail: 647

ORDER SUMMARY

Subtotal	\$ 80	
Estimated Shipping	\$ 5.90	
Shipping Discount	\$ -5.90	
Total	\$ 80	

CHECKOUT NOW

BOOKDEPOT

We offer the vast selection of titles, competitive prices, and fast processing times that you are looking for in a book supplier. Our large inventory of high-margin bargain books is 75-90% off list price and easily browsable on our website —so you can stock the product your customers want, decrease your purchasing time, and improve your bottom line!

Useful Links

- Home
- New Product
- Accessories
- Order Tracking
- Wishlist
- Cart
- Featured Product
- My Account
- Wishlist
- Terms

Contact

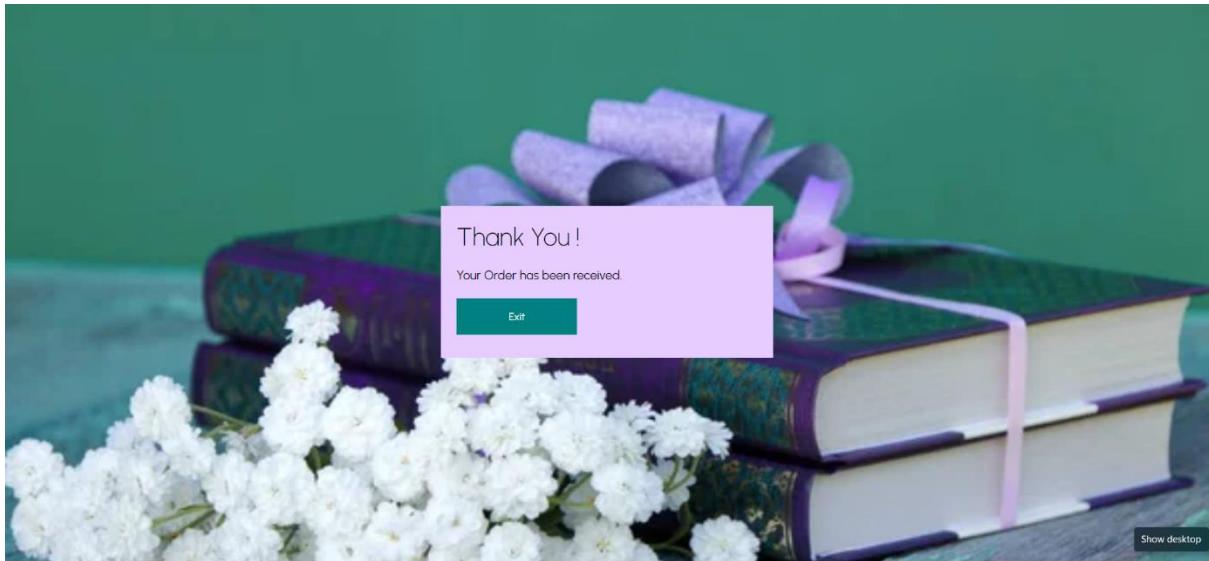
- 622 Dixie Path, South Tobinchester 98336
- +1 234 56 78
- contact@lama.dev

PAYMENT METHODS

Card, PayPal, American Express, VISA, Discover

User add books to the cart which will show the account and shopping cart details. The user can add the book to a shopping cart by entering the quantity and clicking the add to shopping cart button. The quantity value should always be a positive value. After adding to the shopping cart if the user wants to continue shopping, he should click the “checkout now” button. The user can also edit his shopping cart details. If he wants to change the quantity or cancel his order, he can use this page.

7) Order Successful Screen



The user add product to the shopping cart. If the user wants to continue shopping, he should click the “checkout now” button. After appear this massage to conform the order.

6) DATABASE STRUCTURE

MongoDB is an open source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB Atlas runs on AWS, Microsoft Azure, and Google Cloud Platform.

In this Bookshop web application, I create four databases to store data related to the bookshop using MongoDB Atlas.

The screenshot shows the MongoDB Atlas interface for the 'bookshopweb' database. The 'orders' collection is selected. A document is displayed with the following fields:

```
_id: ObjectId("62cf0c12ed9d016b12ecd3166")
userId: "52ce49d9ab87f925a6e3b1f1"
products: [
  {
    > 0: Object
    > 1: Object
    amount: 50
    address: "USA"
    status: "pending"
    createdAt: 2022-08-13T18:16:50.938+00:00
    updatedAt: 2022-07-13T18:16:50.938+00:00
    __v: 0
  }
]
id: ObjectId("62cf0c12ed9d016b12ecd316a")
```

The screenshot shows the MongoDB Atlas interface for the 'bookshopweb' database. The 'orders' collection is selected. Two documents are displayed with the following fields:

```
_id: ObjectId("62cf0c12ed9d016b12ecd316a")
userId: "52ce49d9ab87f925a6e3b1f1"
products: [
  {
    > 0: Object
    > 1: Object
    amount: 100
    address: "USA"
    status: "pending"
    createdAt: 2022-07-13T18:17:01.838+00:00
    updatedAt: 2022-07-13T18:17:01.838+00:00
    __v: 0
  }
]
_id: ObjectId("62d55372bf2d08aeef7f3a5a6")
userId: "52ce49d9ab87f925a6e3b1f1"
```

The above Database Interfaces shows the data related to the bookshop Orders. This Database includes the order details of the customer or user. User Id, Products, Amount, Address and ordered dates are the details that include in the Order database.

This screenshot shows the MongoDB Atlas interface for the 'bookshopweb' database. The 'products' collection is selected. A specific document is highlighted, showing its internal structure:

```

_id: ObjectId("62d3cad89e08fbcc4ad20a650")
title: "Bird Box: A Novel Kindle Edition"
desc: "Written with the narrative tension of The Road and the exquisite terror...""
img: "https://hips.hearstapps.com/vader-prod.s3.amazonaws.com/1572295284-41s...)"
categories: Array
  > categories: Array
  > size: Array
  > type: Array
  > price: 850
  > instock: true
  > createdAt: 2022-07-17T04:12:22.068+00:00
  > updatedAt: 2022-07-17T04:12:22.068+00:00
  > v: 8

```

This screenshot shows the MongoDB Atlas interface for the 'bookshopweb' database. The 'products' collection is selected. The results of a query are displayed:

QUERY RESULTS: 1-8 OF 8

```

_id: ObjectId("62d38a5c9e08fbcd20a64c")
title: "The Water Dancer"
desc: "Young Hiram Walker was born into bondage. When his mother was sold away...""
img: "https://hips.hearstapps.com/vader-prod.s3.amazonaws.com/1572293716-511...)"
categories: Array
  > categories: Array
  > size: Array
  > type: Array
  > price: 750
  > instock: true
  > createdAt: 2022-07-17T04:04:44.842+00:00
  > updatedAt: 2022-07-17T04:04:44.842+00:00
  > v: 8

```

The above Database Interfaces shows the data related to the bookshop Products. This Database includes the products details of the Bookshop and available products in the shop. The admin can add the product details and products into the website. Book Title, Description, Image, Category, Page Size, Book Type, Prize and In Stock details are include in the Product database.

The screenshot shows the MongoDB Atlas interface for the 'bookshopweb' database. The left sidebar includes sections for Deployment, Data Services, Security, and Atlas Milestones. The main area displays a search bar and navigation tabs for Find, Indexes, Schema Anti-Patterns, Aggregation, and Search Indexes. A 'FILTER' button is present. The results section shows two documents:

```
_id: ObjectId("62ce448138cb9d681c5c4ccc")
username: "admin"
email: "admin@gmail.com"
password: "U2FsdGVkXBSAxlJ3mBttaVKN6o4cEsFMcdnT3pcZs="
isAdmin: true
createdAt: 2022-07-11T04:05:21.538+00:00
updatedAt: 2022-07-13T04:05:21.538+00:00
__v: 0

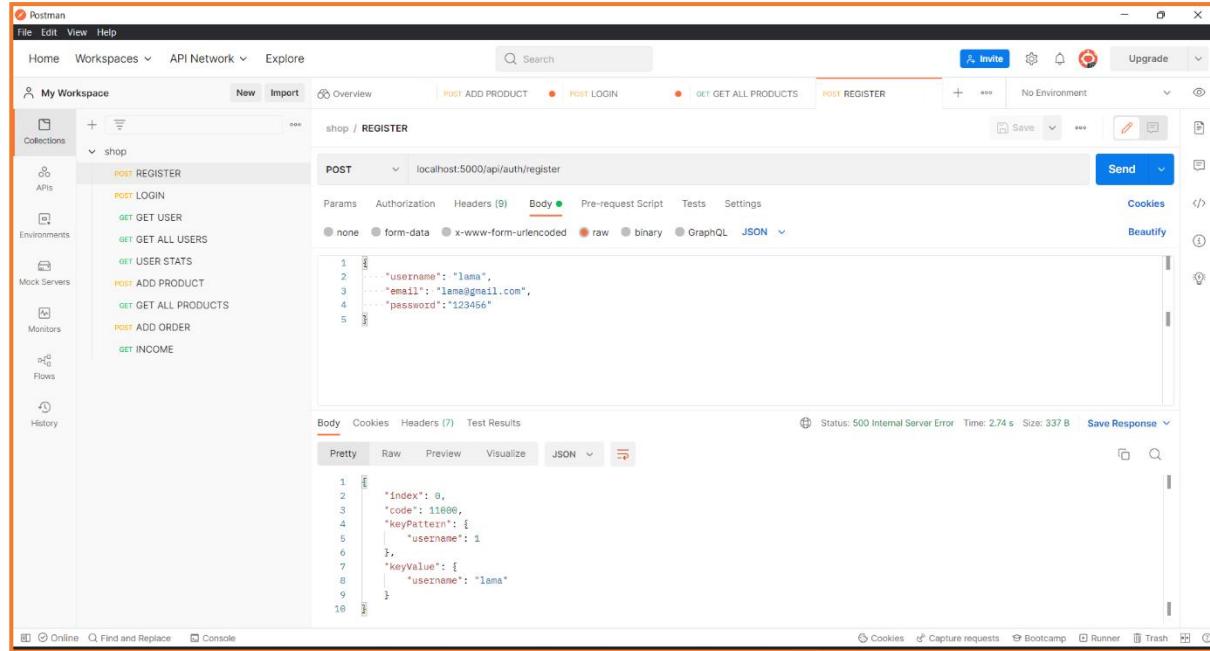
_id: ObjectId("62ce450038cb9d681c5c4cd0")
username: "lana"
email: "lana@gmail.com"
password: "U2FsdGVkXBSA4enV7eBLPhjpeA7VjnCaHa75qV3dyC9="
isAdmin: false
createdAt: 2022-07-13T04:07:28.120+00:00
updatedAt: 2022-07-13T04:07:28.120+00:00
__v: 0
```

This screenshot is identical to the one above, showing the MongoDB Atlas interface for the 'bookshopweb' database. It displays the same two user documents with their respective details.

The above Database Interfaces shows the data related to the bookshop Users. This Database includes every Users Details of the Bookshop. The admin can access all data and Users only by the products in the bookshop. This Users Database use to add the all users how register the website using Web Application Registration.

7) A DETAILED EXPLANATION OF EACH API IMPLEMENTATIONS AND RESPECTIVE ENDPOINTS

Postman supports API-first development with the API Builder. I design my API directly in Postman using the API Builder. After creating API and can test the API that I created to the Web Application. So, I created Nine APIs to built up the web application.



This API used to register all users how need to access the web application and use it. Register all users using this API and this registered data store on Users Database. It can check using this postman and used to test it. In this API used to get the “UserName”, “Email” and “Password” of the users to register them.

Postman API Test Result:

```

{
  "_id": "62ce448138cb9d681c5c4ccc",
  "username": "admin",
  "email": "admin@gmail.com",
  "isAdmin": true,
  "createdAt": "2022-07-13T04:05:21.530Z",
  "updatedAt": "2022-07-13T04:05:21.530Z",
  "__v": 0
}

```

This API used to log in all users how need to log into the web application. Register all users using API and this registered data store on Users Database. The resisted all users can login though this API. It can check using this postman and used to test it. In this API used to get the “UserName”, and “Password” of the users to log into the web application. Postman used to check the login user and their details.

Postman API Test Result:

```

{
  "_id": "62ced40038cb9d681c5c4cd0",
  "username": "lama",
  "email": "lama@gmail.com",
  "isAdmin": false,
  "createdAt": "2022-07-13T04:07:28.120Z",
  "updatedAt": "2022-07-13T04:07:28.120Z",
  "__v": 0
}

```

Postman API Test Result for GET /GET USER:

```

1 _id: "62ce450038cb9d681c5c4cd0",
2   "username": "lana",
3   "email": "lana@gmail.com",
4   "isAdmin": false,
5   "createdAt": "2022-07-13T04:09:28.126Z",
6   "updatedAt": "2022-07-13T04:09:28.126Z",
7   "__v": 0
8
9

```

This API used to get user how are registered in the system. We can register user using API and this registered data store on Users Database. The registered user details can access only the admin in the web application. Admin's Access Token used to get the that user details in the User Database and admin can only access the user's details. It can check using this postman and used to test it.

Postman API Test Result for GET /GET ALL USERS:

```

5   "email": "admin@gmail.com",
6   "password": "U2FsdGVkX18Ax1wL3omB7taVKN6o4oEsFNCdnT3pczs=",
7   "isAdmin": true,
8   "createdAt": "2022-07-13T04:05:21.538Z",
9   "updatedAt": "2022-07-13T04:05:21.538Z",
10  "__v": 0
11 }
12 {
13   "_id": "62ce450038cb9d681c5c4cd0",
14   "username": "lana",
15   "email": "lana@gmail.com"

```

The screenshot shows the Postman interface with a collection named 'shop'. Under the 'shop' collection, there is a 'GET ALL USERS' endpoint. The request URL is set to 'localhost:5000/api/users'. In the 'Headers' tab, there is a header named 'token' with the value 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6...'. The response status is 200 OK, and the response body is displayed in JSON format:

```

4   "username": "admin",
5   "email": "admin@gmail.com",
6   "password": "U2FsdGVkX18ax1w3om8w1aVN6o4oEsFHcdnT3pczs=",
7   "isAdmin": true,
8   "createdAt": "2022-07-13T04:05:21.538Z",
9   "updatedAt": "2022-07-13T04:05:21.538Z",
10  "__v": 0
11 },
12 {
13   "_id": "62ce459038cb9d681c5c4cd9",
14   "username": "lama",
15   "email": "lama@gmail.com",
16   "password": "U2FsdGVkX1/4enV7e8LPTlspEa7VjnCaHs7SqV3dyC0=",
17   "isAdmin": false,

```

This API used to get all users how are registered in the web application. We can Register all users using API and this registered data store on Users Database. The resisted all users' details can access only the admin in the web application. Admin's Access Token used to get the all-users details in the User Database.

The screenshot shows the Postman interface with a collection named 'shop'. Under the 'shop' collection, there is a 'GET USER STATS' endpoint. The request URL is set to 'localhost:5000/api/users/stats'. In the 'Headers' tab, there is a header named 'token' with the value 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6...'. The response status is 200 OK, and the response body is displayed in JSON format:

```

1  [
2    {
3      "_id": 7,
4      "total": 2
5    }
6  ]

```

This API used to get number of users that they registered in the web application. We can calculate all users using this User Stats API using “Users” Database. The registered all users' details can access only the admin in the web application. Admin's Access Token used

to get the all-users details in the User Database. Therefore, this details only can access the admin in the web application.

The screenshot shows the Postman interface with a collection named 'shop'. A POST request titled 'shop / ADD PRODUCT' is selected. The URL is 'localhost:5000/api/products'. The 'Body' tab is active, showing a JSON payload:

```

1 "title": "Beloved",
2   "desc": "Sethe, its protagonist, was born a slave and escaped to Ohio, but eighteen years later she is still not free. She has too many
3     memories of Sweet Home, the beautiful farm where so many hideous things happened. And Sethe's new home is haunted by the ghost of
4       her baby, who died nameless and whose tombstone is engraved with a single word: Beloved.",
5         "img": "https://hips.hearstapps.com/vader-prod.s3.amazonaws.com/1572292769-41Rdzbihq7L.jpg?crop=1.00xw:0.974xh;0,0.016xh&resize=980:*",
6           "categories": ["Classics", "Edited"],
7             "size": "A3",
8               "type": ["Edited", "Non-Fiction"],
9                 "price": 1500,
10                   "inStock": true ...

```

The response status is 200 OK with a time of 159 ms and a size of 976 B. The response body is identical to the request body.

This API used to add Products in the web application. We can add all Product into the Product Database using this “Add Product” API. The registered all users can appear after add this all details of the product to the database. Only admin can add product data to the database and Admin’s Access Token used to include data to the Product Database. Therefore, the admin only can access and change these details in the web application.

The screenshot shows the Postman interface with a collection named 'shop'. A GET request titled 'shop / GET ALL PRODUCTS' is selected. The URL is 'localhost:5000/api/products/'. The 'Headers' tab is active, showing a 'token' header with the value 'Bearer eyJhbGciOiJIUzI1NilsInRScIi6lkpxVCJ9eyJpZCI6...'. The 'Body' tab is active, showing the response body:

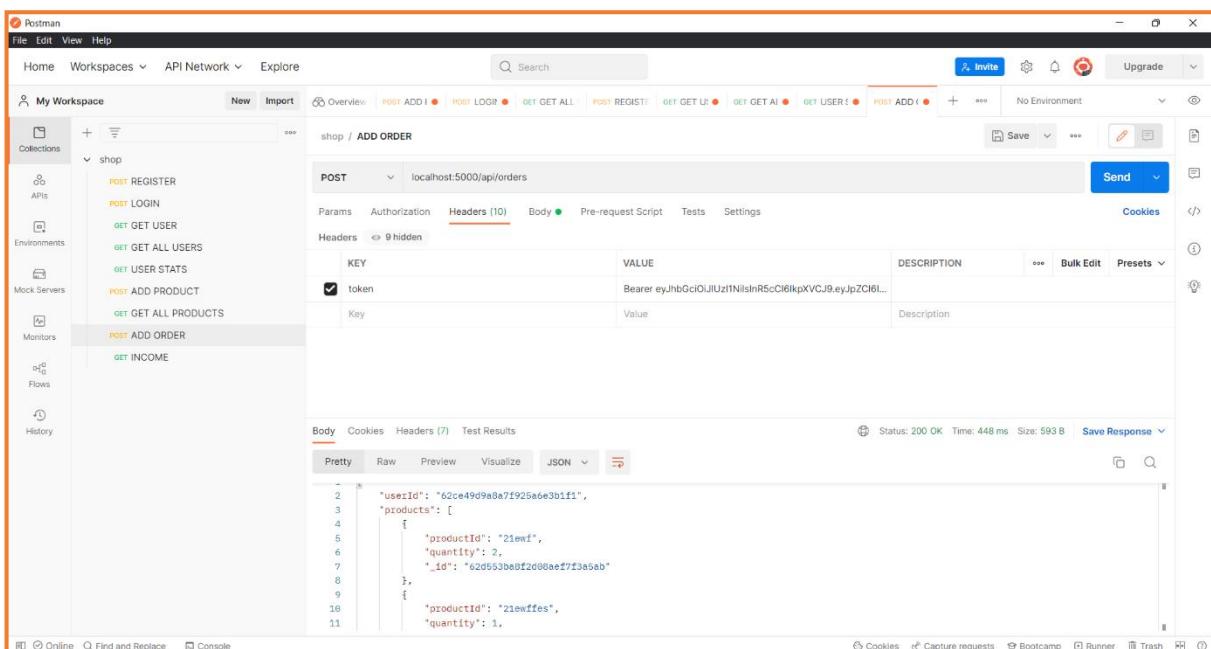
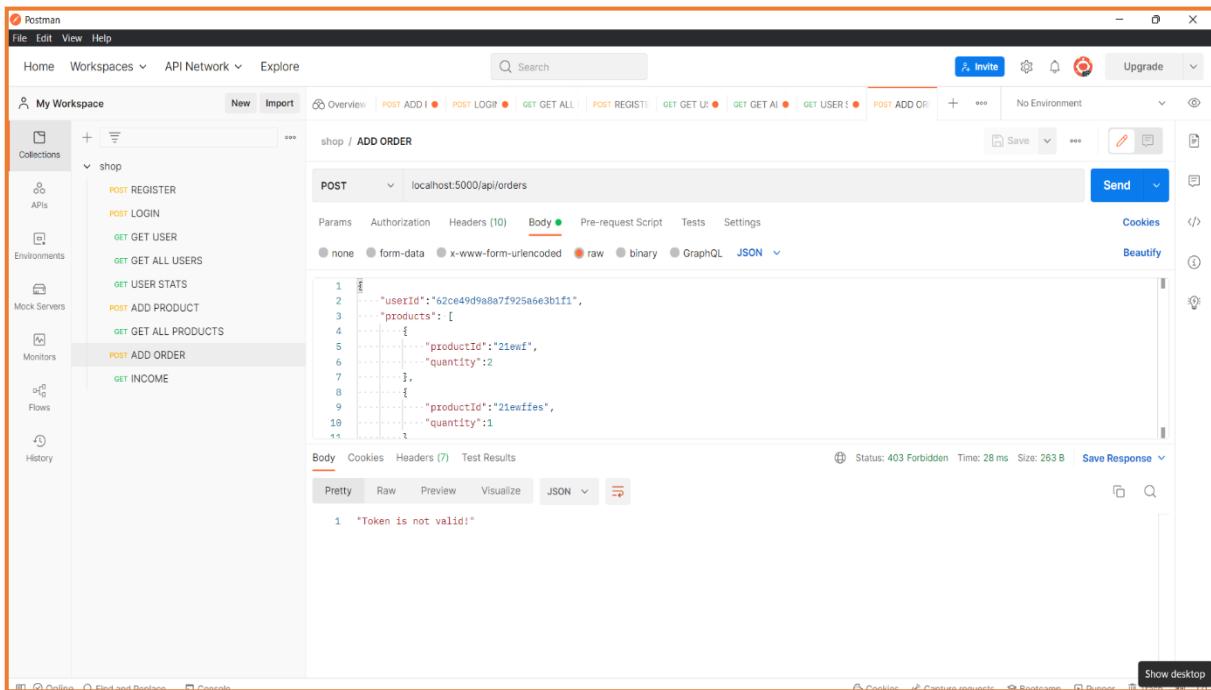
```

1 [
2   {
3     "_id": "62d38a5c9e08fbca4ad20aa64c",
4       "title": "The Water Dancer",
5         "desc": "Young Hiram Walker was born into bondage. When his mother was sold away, Hiram was robbed of all memory of her—but was
6           gifted with a mysterious power. Years later, when Hiram almost drowns in a river, that same power saves his life. This brush
7             with death births an urgency in Hiram and a daring scheme: to escape from the only home he's ever known.",
8               "img": "https://hips.hearstapps.com/vader-prod.s3.amazonaws.com/1572293716-51tgizW5dL.jpg?crop=1xw:0.987xh;center,top&resize=980:*",
9                 "categories": [
10                   "Fantasy",
11                     "Non-Fiction"
12                       ],
13                         "size": [
14                           "A3"
15                           ...

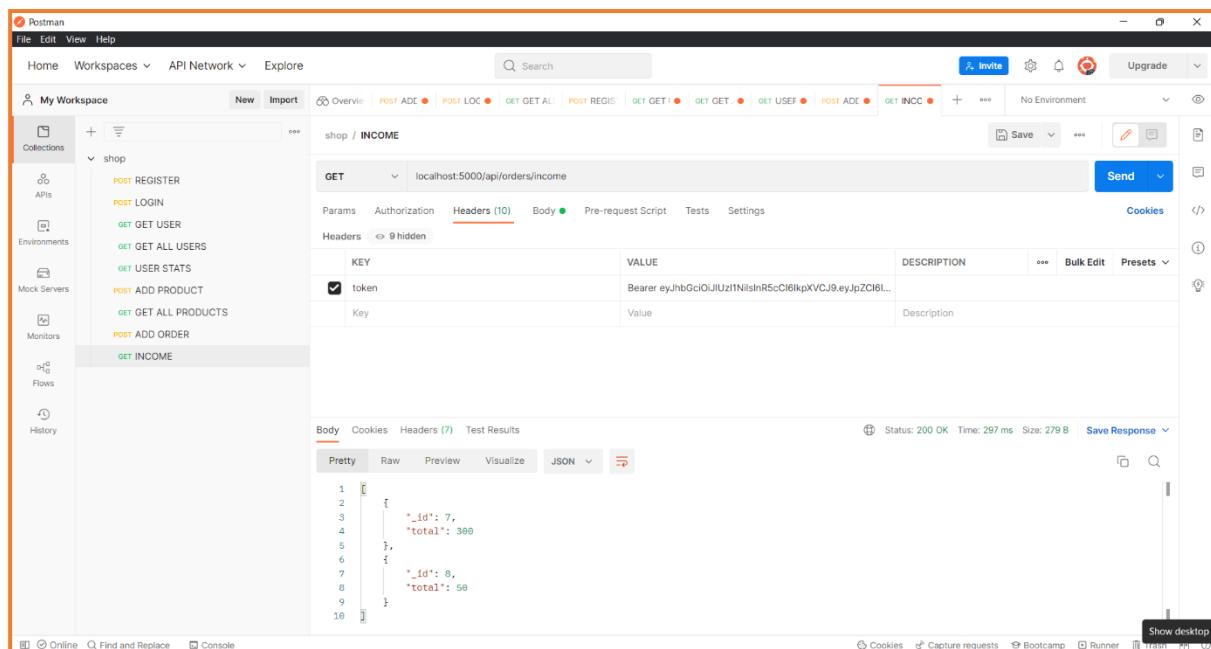
```

The response status is 200 OK with a time of 140 ms and a size of 5.73 KB. The response body is identical to the request body.

API used to get all Products in the web application. We can get all Product in the Product Database using this “Get All Product” API. The registered all users can appear after add this all details of the product. Only admin can get the product data through the database and Admin’s Access Token used to access data in the Product Database. Therefore, the admin only can access and change these details in the web application.



This API used to add Orders in the web application. We can get all Orders in the Order Database using this “Add Order” API. The registered all users can appear after add this all details of the product. Only admin can get the product data through the database and Admin’s Access Token used to access data in the Product Database. The admin only can access and change these details in the web application.



The screenshot shows the Postman application interface. On the left, the sidebar displays 'My Workspace' with a collection named 'shop' containing various API endpoints. The main workspace shows a 'GET' request to 'localhost:5000/api/orders/income'. The 'Headers' tab is selected, showing a 'token' header with a value of 'Bearer eyJhbGciOiJIUzI1Ni...'. The 'Body' tab shows a JSON response with two items:

```
1  [
2    {
3      "_id": 7,
4      "total": 300
5  },
6  {
7    "_id": 8,
8    "total": 50
9  }
10 ]
```

This API used to get in the income of the all ordered products in the web application. We can get all income using the Order Database using this “Income” API. Only admin can get the income of the products through the database and Admin’s Access Token used to access data in the Order Database. The admin only can access and change these details in the web application.

SUMMARY

The main objective of the project is to create an online book shop that allows users to search and purchase a book online based on title, author and subject. I develop a website using Mern Stack with MongoDB, Express, React, NodeJS. The books are divided into many categories based on subject like Biography, Children and Fiction etc. The Administrator will have additional functionalities when compared to the common user. He can add, delete and update the book details, book categories, member information and also confirm a placed order.

Book, Order, Order Status, Stock, Online Book Store also manage the Book Store Details online for Order Status details, Stock details, Books, Manage the information of Customer, Shows the information and description of the Books, Order, to increase efficiency of managing the Books, Customer and It deals with monitoring the information and transaction of Order Status are major parts of this web application.

The GUI Builder makes it possible to build professional-looking GUIs without an intimate understanding of layout managers. I used CSS and HTML to design interfaces of the app in VS code and java script used for do a functionality of the web application. Seven interfaces have included in this app. Home Screen, Login Screen, Register Screen, Product Screen, Product Description Screen and Cart Screen are the Interfaces of this Bookshop web application.

In this Bookshop web application, I create four databases to store data related to the bookshop using MongoDB Atlas. Postman supports API-first development with the API Builder. I design my API directly in Postman using the API Builder. After creating API and can test the API that I created to the Web Application. So, I created Nine APIs to build up the web application.

REFERENCE

- [1] M. Jean-Philippe, “21 of the Most Popular Book Genres, Explained,” *Oprah Daily*, Oct. 29, 2019. <https://www.oprahdaily.com/entertainment/books/a29576863/types-of-book-genres/>
- [2] “API Development Overview,” *Postman Learning Center*. <https://learning.postman.com/docs/designing-and-developing-your-api/the-api-workflow/> (accessed Jul. 19, 2022).
- [3] Wikipedia Contributors, “MongoDB,” *Wikipedia*, Oct. 31, 2019. <https://en.wikipedia.org/wiki/MongoDB>
- [4] S. Malik, “Online Bookstore - A Web App for Book Readers,” *AttainU*, Apr. 15, 2022. <https://attainu.com/blog/online-bookstore-a-web-app-for-book-readers/> (accessed Jul. 19, 2022).
- [5] “What is the MERN Stack? Introduction & Examples | Simplilearn,” *Simplilearn.com*, Dec. 27, 2021. <https://www.simplilearn.com/tutorials/mongodb-tutorial/what-is-mern-stack-introduction-and-examples#:~:text=MERN%20stack%20is%20a%20collection> (accessed Jul. 19, 2022).