

25TH DECEMBER 2018 – 27TH MARCH 2019

BOOK SCHMO – A GREAT MIND BEGINS WITH A GREAT BOOK | PROJECT REPORT

PRESENTED BY: RISHI MALHOTRA

CONTENTS

Analysis	6
Introduction	6
Project background	6
A couple of notes	7
Communication with third-party	9
Objectives	12
Database	12
Overall website functionality	12
Home page	12
Address-page	12
My cart page	13
My orders page	13
Checkout page	13
Trending page	13
View book page	13
Header	13
Creator's home page	14
creator's book listings page	14
List books page	14
Search bar and search page	14
Research.....	14
Potential programming solutions	14
Support used	16
A simple Model	11

Design	16
Summary.....	16
Systems Flowcharts And Human-Computer Interface (HCI).....	16
Overall system design	17
Login Page	20
Sign Up Page	Error! Bookmark not defined.
Trending page Design.....	21
Home Page and Customized Home Page design	22
My Cart page Design.....	24
My Orders page Design.....	25
Add Address Page Design.....	27
View all Address Page Design	28
My Book Listings Page.....	29
List a new book page	30
List an existing book page.....	31
Dataflow diagrams	31
Signup and Login Process.....	32
Ordering and Listing Processes.....	34
Displaying Sales, Order History and Trending Books Process	35
Algorithms.....	36
Login Flowchart.....	36
List a new book algorithm.....	37
List an existing book algorithm	38
Add to cart algorithm.....	39
Checkout algorithm.....	40

My orders page algorithm	41
My Book Listings Page algorithm	41
Trending page algorithm	41
Index page algorithm	42
Database design	43
Database Schema	43
Database Table: Accounts	44
Database Table: Address	44
Database Table: Books	45
Database Table: Listings	46
Database Table: Orderitems	46
Database Table: orders	47
How the database works	47
File Structure and Organization	49
Sample Database Queries	54
Index Page sql	54
Trending Page SQL Query	55
Checkout page SQL query	56
Storing the password and encryption	56
How it works	57
Unique book image names	58
Some notes on the design section	58
Technical solution	59
Utility files	59
Login System	64

Login.....	64
Signup	67
Logout	70
Address pages	70
Add Address	70
View Address	76
Index page.....	77
Trending page	83
Search Page.....	84
View book	86
Ordering The book	89
Add to cart/ remove from cart.....	89
Checkout	92
Order Confirmation.....	98
My orders.....	99
Utility files For creator's page	102
Creator's header	102
Footer.....	105
Connection file.....	106
Creator's index page	107
List a book	108
List a book menu.....	108
List a new book	109
List an existing book.....	114
My book listings	120

Testing	124
Testing the project	124
Evaluation.....	142
How well the objectives were met	142
Database	142
Overall website functionality.....	142
Home page	143
Address-page	143
My cart page	143
My orders page.....	144
Checkout page	144
Trending page	144
View book page.....	144
Header	145
Creator's home page.....	145
creator's book listings page	146
List books page.....	146
Search bar and search page.....	146
Potential Improvements	146
Feedback from my client.....	148
The learning curve	148

ANALYSIS

INTRODUCTION

BookSchmo is an e-commerce website dedicated to the sale of books online. Over the past decade, businesses have increasingly switched from brick-and-mortar stores to online retail warehouses. Large businesses such as Amazon and eBay have developed their businesses around online retail. Even more so, small businesses choose to build their own online retail platforms to promote sale of their products.

More generally, with the widespread use of the Internet, web-development has become a very desirable for businesses. By working on this project, I aim to improve my front-end and back-end web-development skills and to produce an end product that is representative of my capabilities.

PROJECT BACKGROUND

After learning HTML and CSS, I taught it at an after-school club in Year 12. Ever since, my aim in web-development has shifted to building a large-scale project. After my client, CEO of QuantM Ltd., requested an ecommerce website specially targeted towards the sale of books, I decided to start off the project by learning about back-end web-development.

BookSchmo is an online interface where users can order and sellers can sell books. The primary aim behind the website is to ensure a user-friendly interface with complete back-end functionality to provide customers with a satisfying experience.

A list of technical terms used in the report are displayed below along with their meanings

#	Term	Meaning
1	Webpage	A hypertext document displayed as a part of the website
2	HTML	Hyper-Text Markup language. This is the standard markup language for creating webpages and web applications
3	CSS	Cascading Style Sheets. These are the standard style sheets language used for describing the presentation of a document written in a markup language like HTML.
4	JavaScript	A high-level interpreted programming language that is used to program the behavior of web pages.
5	PHP	Hypertext Preprocessor Language. It is a server scripting language that is used to dynamically render pages from the database

6	SQL	A domain specific language designed for managing data held in a relational database management system.
7	PHP Superglobal	PHP superglobals are accessible from any level of scope within program code. This means that they can be accessed from any function, class or file.
8	PHP \$_SESSION	PHP Session is a superglobal. It is an associative array containing session variables available to the current script.
9	PHP \$_GET	PHP \$_GET is a superglobal. An associative array of variables passed to the current script by url parameters.
10	PHP \$_POST	PHP \$_POST is not a superglobal. An array of variables names and values sent by a form input.
11	PHP \$_FILES	PHP Files is a superglobal. An associative array of items uploaded to the current script via the HTTP POST method.
12	HTML navbar	In this report, the navbar refers to the horizontal menu at the top of every webpage

A COUPLE OF NOTES

A list of all account usernames and passwords are presented below. Please note that the passwords cannot be seen in the database because the passwords have been encrypted by a hashing algorithm. (More on this in the design section)

All accounts and passwords

Password for all accounts - strong

|
Creator's account

1. santapaws@creator.com // strong
2. chewbarka@creator.com // strong
3. marypuppins@creator.com //strong
4. sherlockbones@creator.com // strong
5. waldopuck@creator.com // strong
6. dodgerjones@creator.com // strong

User account

1. johndoe@user.com // strong
2. janedoe@user.com // strong
3. spideydoe@user.com // strong
4. winniedoe@user.com // strong
5. ozzydoe@user.com // strong
6. kobedoe@user.com // strong

The password for all these accounts is strong. These were the accounts used during the testing and development phase of the website. These accounts may be used for marking purposes. Alternatively, the examiner could create a new account by visiting the sign up page. To visit the sign up page, click the login button on the navbar, and then click the signup button at the bottom of the form.

COMMUNICATION WITH THIRD-PARTY

Meeting today

Inbox x



Rishi Malhotra <rishim@student.thebsbh.com>
to khurranapawan ▾

Wed, Aug 22, 2018, 1:07 PM



Hi Mr Pawan,

It was a pleasure meeting you today. It was really nice to get to know your company Quantm Ltd. and I hope I can be of assistance to your company in the future. In regards to the project that you mentioned today, I think it would be a great opportunity for me to build my skill set around it. Could you tell me more about it?

Regards,

Pawan Khurrana
to cspg2001, me ▾

Wed, Aug 22, 2018, 3:10 PM



Images are not displayed. Display images below - Always display images from khurranapawan@gmail.com

Hello Rishi. Thanks for your interest in the project. I have attached my head of HR, Ritu in the CC and she should be able to talk to you today. Can you send phone number and the time of comfort for you.

...

Rishi Malhotra <rishim@student.thebsbh.com>
to Pawan ▾

Wed, Aug 22, 2018, 3:15 PM



Here is my phone number - +973 39187925. 5pm should be fine

...



Pawan Khurrana

to me ▾

Thu, Aug 23, 2018, 7:02 AM



Images are not displayed. Display images below - Always display images from khurranapawan@gmail.com

Hi Rishi,

I heard Ritu allowed you to hard work on the project.

As far as I am aware you will be running a two-way e-commerce website.

- sellers can list book son the site

- buyers can buy books on the site

For this project to be of use to our research, model it to popular e-commerce sites like Flipkart or amazon.

Regards,



Rishi Malhotra <rishim@student.thebsbh.com>

to Pawan ▾

Thu, Aug 23, 2018, 9:04 AM



Thanks a lot for the opportunity.

I won't let you down. Also, this will be great for my A-Level Computer Science Project. So do you mind if I use this project for my report?

Best Regards,

Rishi



Pawan Khurrana

to me ▾

Thu, Aug 23, 2018, 9:07 AM



Images are not displayed. Display images below - Always display images from khurranapawan@gmail.com

Sure, we don't mind

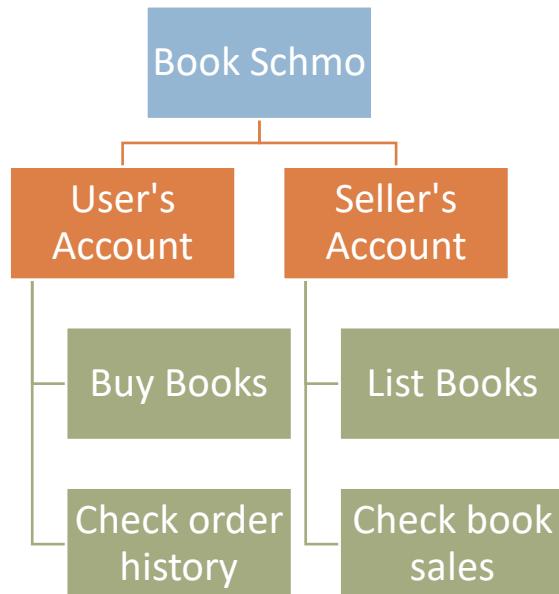
As can be seen in the email screenshots presented above, I was able to get in touch with my father's friend, Mr Pawan Khurrana. As CEO of QuantM Ltd, he asked me to work on a prototype e-commerce website, as he had plans on introducing an e-commerce section for selling books. Therefore, Ms Ritu asked me to work on an e-commerce website for books over the phone call.

A summary of the phone conversation led to the development of the simple model below.

- The project would have to be developed as a website.
- 2 accounts – A user's account and a seller's account have to be created.
- The user's account allows users to buy books
- The seller's account allows sellers to sell books.
- The user's account allows users to buy books by adding them to a cart and making the order.
- Sellers could list as many books as they could
- Users could order multiple books at once by adding them to a cart.
- An option for payment should not be added as the books don't exist.

A SIMPLE MODEL

It is to be noted that this model will be expanded upon in the design stage



OBJECTIVES

DATABASE

1. The database will have to store information on
 - a. Customers and Sellers' first name, last name, email, password and address to which shipped.
 - b. Every customers' order details, and the details on the address to which it is shipped
 - c. Each item in the order including details about the book, details on the seller, and the quantity of the item bought.
 - d. Each book will need details on the book's title, author, description, genre, ISBN and an image.
 - e. Each book may be listed at different prices by different sellers. The database will have to store information on these prices.

OVERALL WEBSITE FUNCTIONALITY

1. An option for users to sign up, log in and log out of their accounts of the website.
2. Two account types: User's Account, Creator's Account
 - a. The user's account allows users to
 - i. Add addresses
 - ii. Add books to a shopping cart and place the order
 - iii. Check their history of orders.
 - b. The Creator's Account allows creators to
 - i. View existing book listings
 - ii. List books on the website
 - iii. View sales

HOME PAGE

1. A list showing some of the most bought books on the website
2. If the user is logged in, some book recommendations should be made
 - a. The book recommendations will be made based on the user's order history.

ADDRESS-PAGE

1. Gives the user option to add addresses
2. Allows the user to view all existing addresses.

MY CART PAGE

1. A list of all the books the user added to his cart
 - a. The cart should display the book image, the title, the price and the quantity
2. An option to remove the book from the cart.

MY ORDERS PAGE

1. A display of all the orders the user has made including details on
 - a. The book name, author name, quantity, price and order total.

CHECKOUT PAGE

1. Displays the books in the cart including information on
 - a. Book Name, Book Price and Quantity
2. A select address option allows users to select an address to ship the book to.
3. A button to confirm purchase.

TRENDING PAGE

1. Displays the most sold books on the website
 - a. The book image, book title, author and description should be displayed for each book on the trending page

VIEW BOOK PAGE

1. The view book page is a page that displays information about each book.
 - a. This includes the book image, book title, book author, book description and price of the book as listed by all sellers.
2. Allows users to select the quantity of books to buy
3. Contains an add-to-cart button
 - a. The button successfully adds the book, along with the right quantity and at the right price to the cart.

HEADER

1. A navigation bar spanning across the width of the page containing details on
 - a. The business name (BookSchmo), link to home page, my cart page, my orders page.
 - b. A search bar
 - c. It should also allow the user to log out
2. The creator's header should span across the width of the page and contains links to

- a. Links to the creator's book listings, the list a book page, the user site.
- b. It should also allow the creator to log out

CREATOR'S HOME PAGE

- 1. The creator's home page allows all creators to
 - a. Access the book listings page
 - b. Access the list a book page.
 - c. Access the user site.

CREATOR'S BOOK LISTINGS PAGE

- 1. Allows creators to view all the books that they have listed.
 - a. This includes information on the book isbn, title, author, price it was listed at, the number of books sold at that price
- 2. Allows sellers to sort the books by price or sales

LIST BOOKS PAGE

- 1. Allows sellers to list books
 - a. A form should give sellers option to list the book by ISBN, book title, author, genre, description, price and an option to upload the book image
 - b. If the book already exists on the website, the seller is asked to enter the book ISBN and price to list the book

SEARCH BAR AND SEARCH PAGE

- 1. The search bar dynamically renders a page containing books which link with keywords in the search bar.

RESEARCH

POTENTIAL PROGRAMMING SOLUTIONS

There are multiple methods for building websites and I would have to choose the one I was the most comfortable with. From background knowledge, I would need **HTML** for the structure and presentation of the website, **CSS** for styling the html pages and **JavaScript** for adding animations to the website as well as making it more responsive and user friendly. I will also use **MySQL** to produce the sql queries designed for handling my database.

To produce my website's backend, there were a range of programming options available.

Python and Python Django

I have used Python and Python Django before for basic web development. Having used YouTube-based tutorials to learn the framework, I found the initial learning curve to be very steep, making it hard for me to learn the most basic functions of a website. However, a major advantage was that my familiarity with the Python programming language meant that I would not have to learn the syntax of another programming language. However, I would have to spend a large amount of time understanding the framework.

PHP

I have never used PHP before but I am aware of its popularity. According to my research online, it was the easiest and most widely used programming language for back end programming with Python's Django and Flask frameworks coming in second. Due to its popularity, it also provides a lot of online support in programming. Moreover, this programming solution was recommended to me by Ms Ritu over the phone call. Most of all, PHP provides a quick and easy method of incorporating cross-parametrized sql queries.

Ruby on Rails

Ruby on Rails provides developers with a relatively modest online support. It also requires a strict adherence to programming standards. I found out that Ruby used several established paradigm conventions for performing and support online. The online support for Ruby on Rails was large, although a bit less than the same for PHP. However, like Python Django, I would have to spend a lot of time learning Ruby on Rails before I could begin programming the project.

Front-end techniques

As the main focus of the project was on building the back-end functionality of the project, I used the bootstrap framework to allow me to design website with greater simplicity. Once again, this framework was recommended to me by my client, Ms Ritu from QuantM Ltd.

Moreover, I will use the **Bootstrap framework** for CSS and JavaScript, to allow me to incorporate polished designs into my website.

My solution

Adhering to my client's initial proposal, I used PHP to build the backend of my application.

I started off the project by consulting the documentations of the languages and frameworks to understand the language.

Tools I used

The project was programmed on the Windows 10 Operating System.

I used Notepad ++ initially and Sublime Text at the end for programming in HTML, CSS, JavaScript and PHP for its elegance and simplicity. I used the XAMPP Control Panel, ran the Apache and SQL services on it and hosted my website locally on my computer during the time of development.

Most of the SQL queries were programmed directly onto scripts to create cross-table parametrized SQL queries. However, initial SQL to create the database and tables was programmed into the MySQL administration tool, phpMyAdmin, provided by the XAMPP control panel.

SUPPORT USED

- To refresh my knowledge on HTML and CSS, I used <https://www.w3schools.com>.
- To learn PHP, I used [*mmtuts Youtube Tutorials*](#).
- I learnt SQL from my wonderful teacher, Ms. Tapscott and the AQA A-Level Computer Science book by Bob Reeves
- Access to Bootstrap JavaScript and CSS modules from - <https://getbootstrap.com/>
- Finally, I often looked at <https://stackoverflow.com/> for support with malfunctioning SQL queries and PHP code that provided no response.

DESIGN

SUMMARY

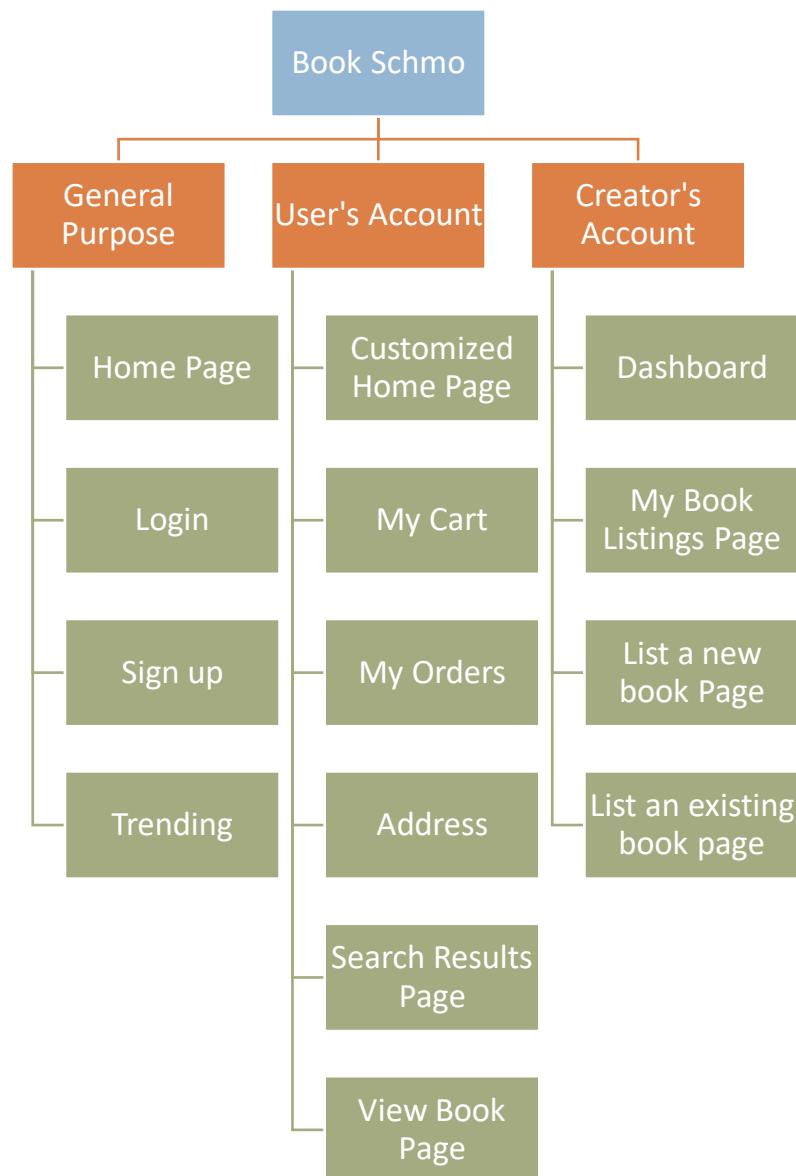
My project is an e-commerce platform that allows sellers to sell books to buyers. My project can be broken down into two parts – The user's account and the creator's account.

- The creator's account allows creators to sell books as well as view sales by book
- The user's account allows users to search for and buy books.

SYSTEMS FLOWCHARTS AND HUMAN-COMPUTER INTERFACE (HCI)

The flowcharts for individual pages are accompanied with human computer interfaces for the same pages. The HCI for each page are screenshots of what the website actually looks like.

OVERALL SYSTEM DESIGN



Module	Purpose
General Purpose	General purpose pages are pages that can be seen when the user is not logged in as user or creator.
Home Page	The home page is an attractive page that gives an overview of the books sold on the website.
Login Page	The login page allows unidentified users to login as users or creators.
Sign up page	The sign up page allows people to sign up as users or creators.
Trending page	The trending page displays some of the most sold books on the website

Module	Purpose
User's Account	All the pages displayed when a user is logged in.
Customized Home Page	The customized home page is an attractive page that gives an overview of the books sold on the website. It is different from the standard home page in that it displays book recommendations based on the user's order history.
My cart	Allows users to add multiple books to a 'shopping cart' and then order them all in one go.
My orders	The sign up page allows people to sign up as users or creators.
Address	Allows users to create and view their addresses.
Search Results	Allows users to search for the books they need. This page then displays all the books that match the search criteria.
View Book Page	When the user clicks a book to get more information about it, he is directed to the view book page. The view book page displays all information relating to that book.

Module	Purpose
Creator's Account	All the pages displayed when a creator is logged in.
Dashboard	The dashboard displays a welcome message and links to all the pages on the website.
My Book Listings Page	The My Book Listings Page displays all the books the seller has listed on the website, the prices they are selling at, and the sales made at each price the book was sold for.
List a new book page	Gives creators the option to list new books on the website.
List an existing book page	A page that allows creators to list books that were already listed on their website.

Notes:

- The creator has access to the home page and the trending page under the general purpose pages section.

[BookSchmo](#) Home Trending Login Check Page Check Page #2 Display Session Session Destroy [Search](#)

LOGIN

To BookSchmo

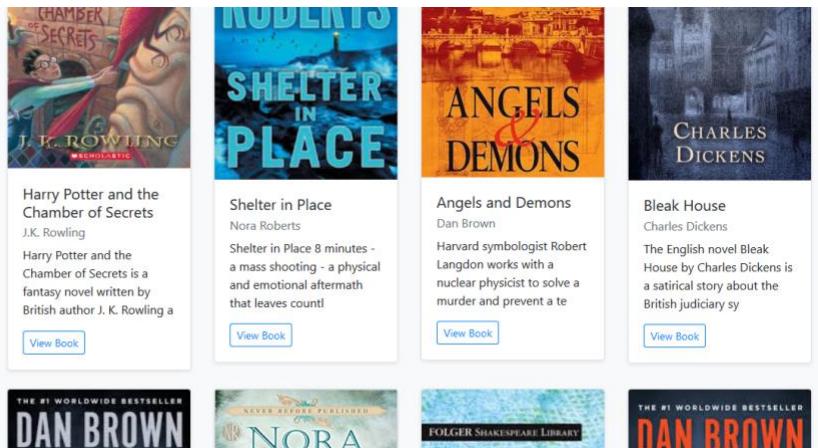
Email address

Password

[Sign in](#)

New to BookSchmo? [Create an account](#)

[BookSchmo](#) My Account Be BookSchmo About [Search](#)



TRENDING

Check what everyone's reading today!

[BookSchmo](#) Home Trending Login Check Page Check Page #2 Display Session Session Destroy [Search](#)

MY CART

Everything in one place

Book Image	Book Name	Price	Quantity	
	Inferno	\$4.99	10	Remove from cart
	Little Dorrit	\$4.99	10	Remove from cart

Subtotal: \$99.8 [Proceed to checkout](#)

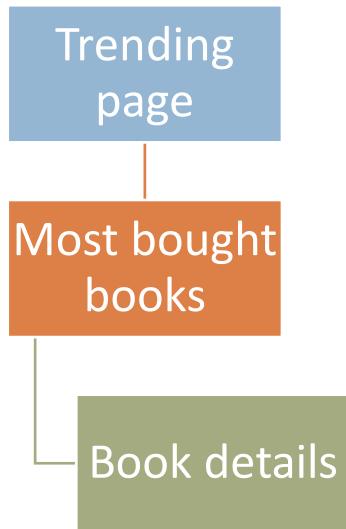
[BookSchmo](#) Home Trending Login Check Page Check Page #2 Display Session Session Destroy [Search](#)

SIGN UP

To BookSchmo

First Name	<input type="text"/>
Last Name	<input type="text"/>
Email address	<input type="text"/>
Date of Birth	<input type="text"/> mm/dd/yyyy
Password	<input type="password"/>
Re-enter Password	<input type="password"/>

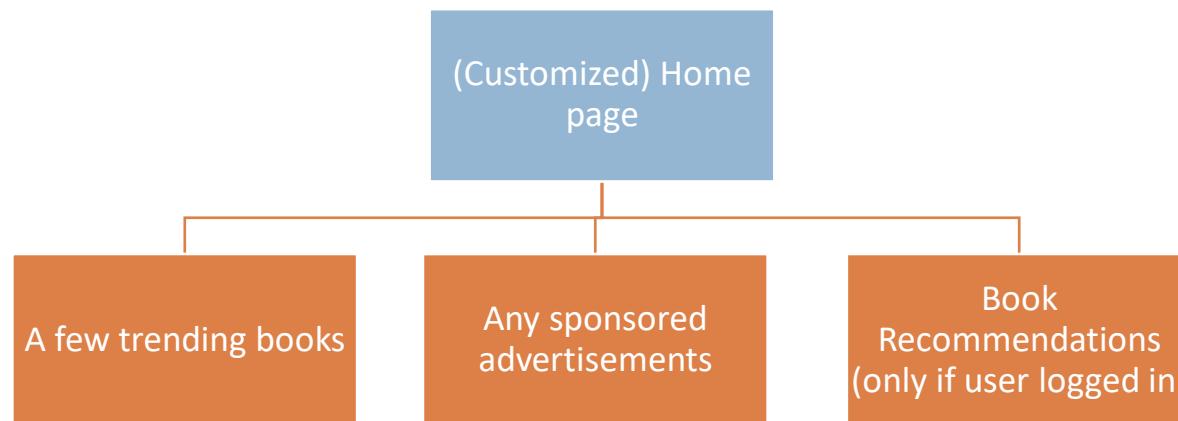
TRENDING PAGE DESIGN

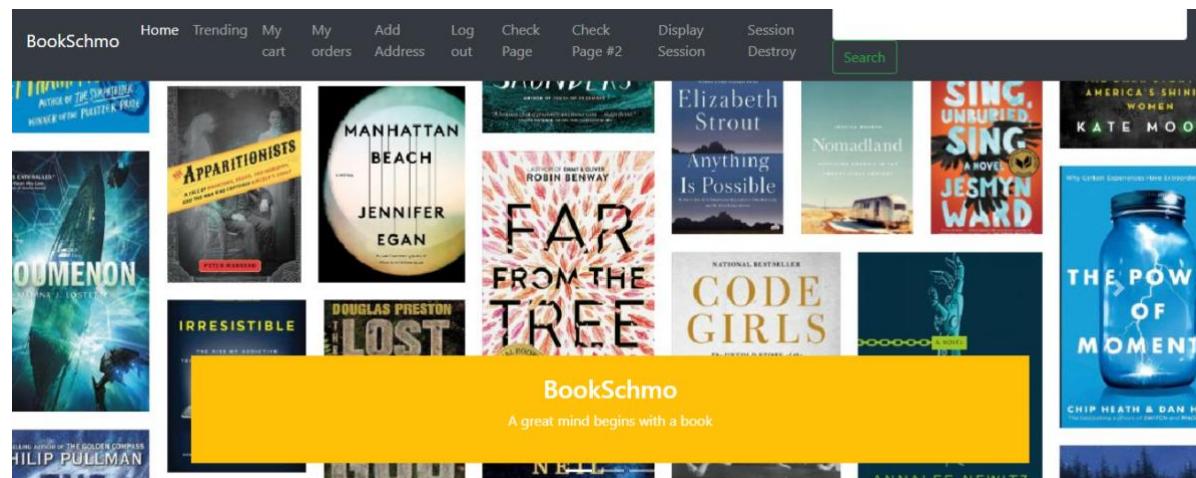


Notes:

- The trending book page is simple: it displays a list of all the trending books as blocks.
- The information about each trending book should include the book name, the author, a small book description, and a link to view the book.

HOME PAGE AND CUSTOMIZED HOME PAGE DESIGN





Trending Today

[Click to see more](#)

Harry Potter and the Chamber of Secrets
J.K. Rowling

Nora Roberts
SHELTER IN PLACE

DAN BROWN
THE LOST SYMBOL

CHARLES DICKENS
BLEAK HOUSE

Kobo E-Reader Sponsored

For a convenient and customized reading experience

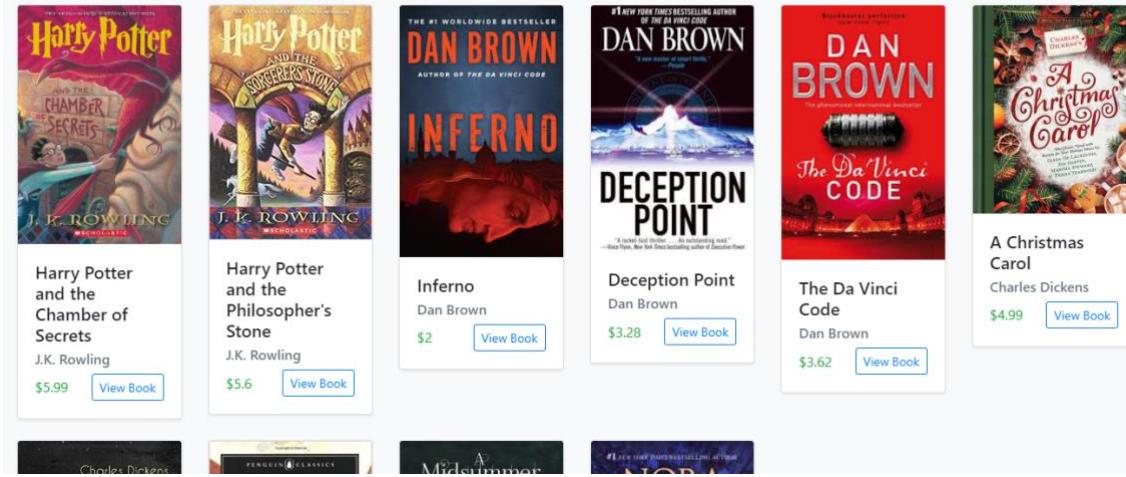
The Fall of Gondolin

There came the flood to Tuor and guided him along dark passages amid the mountains until he came out in the light once more, and saw that the river flowed swiftly in a ravine of great depth with sides unscalable. Now Tuor desired no more to return but went ever forward, and the river led him always toward the west.

The same lay behind his back and set before his face, and when the river had run through the boulders or fell over falls there were at times rainbows woven across the ravine, but at evening its smooth sides would glint in the setting sun, and for these reasons Tuor called it Golden Cleft or the Gully of the Rainbow Roof, which is in the speech of the Gnomes Glóðafær or Cris Íbrantloð.

Now Tuor journeyed here for three days, drinking the water that gushed from the broken sides of the falls; and these were gold and blue and silver and of many wondrous shapes. At length the ravine widened, and ever as it opened its sides became lower and more rough, and the bed of the river more impeded with boulders against which the waters foamed and spouted. Long times would Tuor sit and gaze at the splashing water

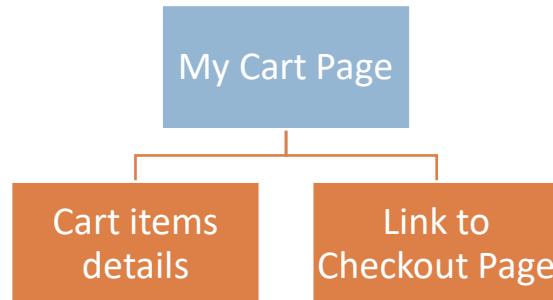
Recommended for you



Notes:

- The customized home page is different from the regular home page in that the customized home page shows book recommendations based on what the user has bought.
- Following on from the overall website design flowchart, the customized home page will only be shown if the person is logged in as a user. If the user is not logged in / or is logged in as a creator, then the regular home page will be displayed

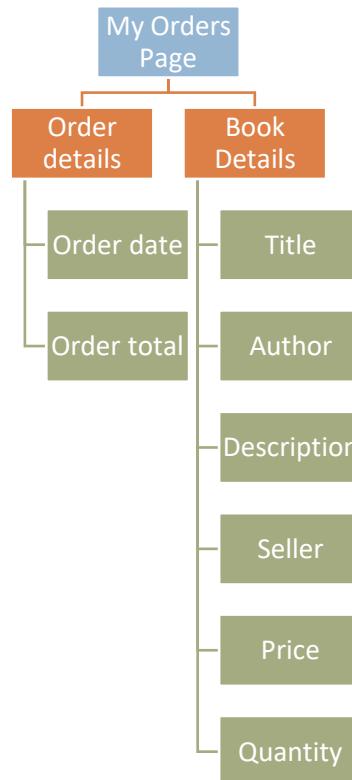
MY CART PAGE DESIGN



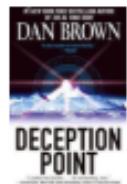
Notes:

- The cart item details include the book image, book name, price and quantity of books ordered.

MY ORDERS PAGE DESIGN



ORDER PLACED	ORDER TOTAL
2019-03-17 13:49:21	\$81.86

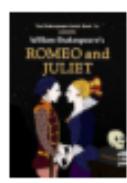


Deception Point
Sold by: Santa Paws

Deception Point is Dan Brown's third novel, released in 2001.

Price: \$3.28

Quantity: 14



Romeo and Juliet
Sold by: Santa Paws

Romeo and Juliet is a tragedy written by William Shakespeare early in his career about two young star-crossed lovers whose deaths ultimately reconcile their feuding families.

Price: \$5.99

Quantity: 6

ORDER PLACED	ORDER TOTAL
2019-03-17 13:38:06	\$5.99



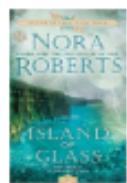
A Christmas Carol
Sold by: Mary Poppins

A Christmas Carol captured the zeitgeist of the mid-Victorian revival of the Christmas holiday. Dickens had acknowledged the influence of the modern Western observance of Christmas and later inspired several aspects of Christmas, including family gatherings.

Price: \$5.99

Quantity: 1

ORDER PLACED	ORDER TOTAL
2019-03-13 09:11:07	\$4.09



Island of Glass
Sold by: Santa Paws

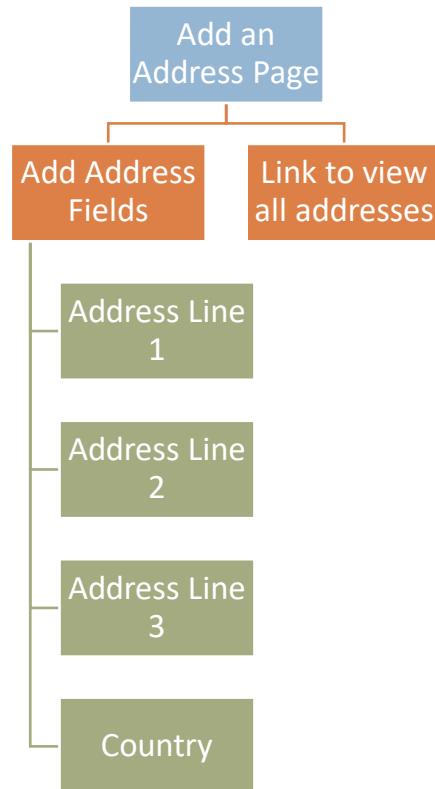
The final Guardians Trilogy novel from the #1 New York Times bestselling author of Bay of Sighs and Stars of Fortune.

Price: \$10.99

Quantity: 70

Notes:

- An order may have multiple books, an order heading must be displayed, and all the books ordered will be shown under that order heading.

ADD ADDRESS PAGE DESIGN

ADD AN ADDRESS

To BookSchmo

The form consists of five input fields and a submit button. The fields are labeled: 'Address Line 1 * Necessary field', 'Address Line 2', 'Address Line 3', and 'Country * Necessary field'. The 'Address Line 1' and 'Country' fields are marked as required. Below the fields is a blue 'Add' button.

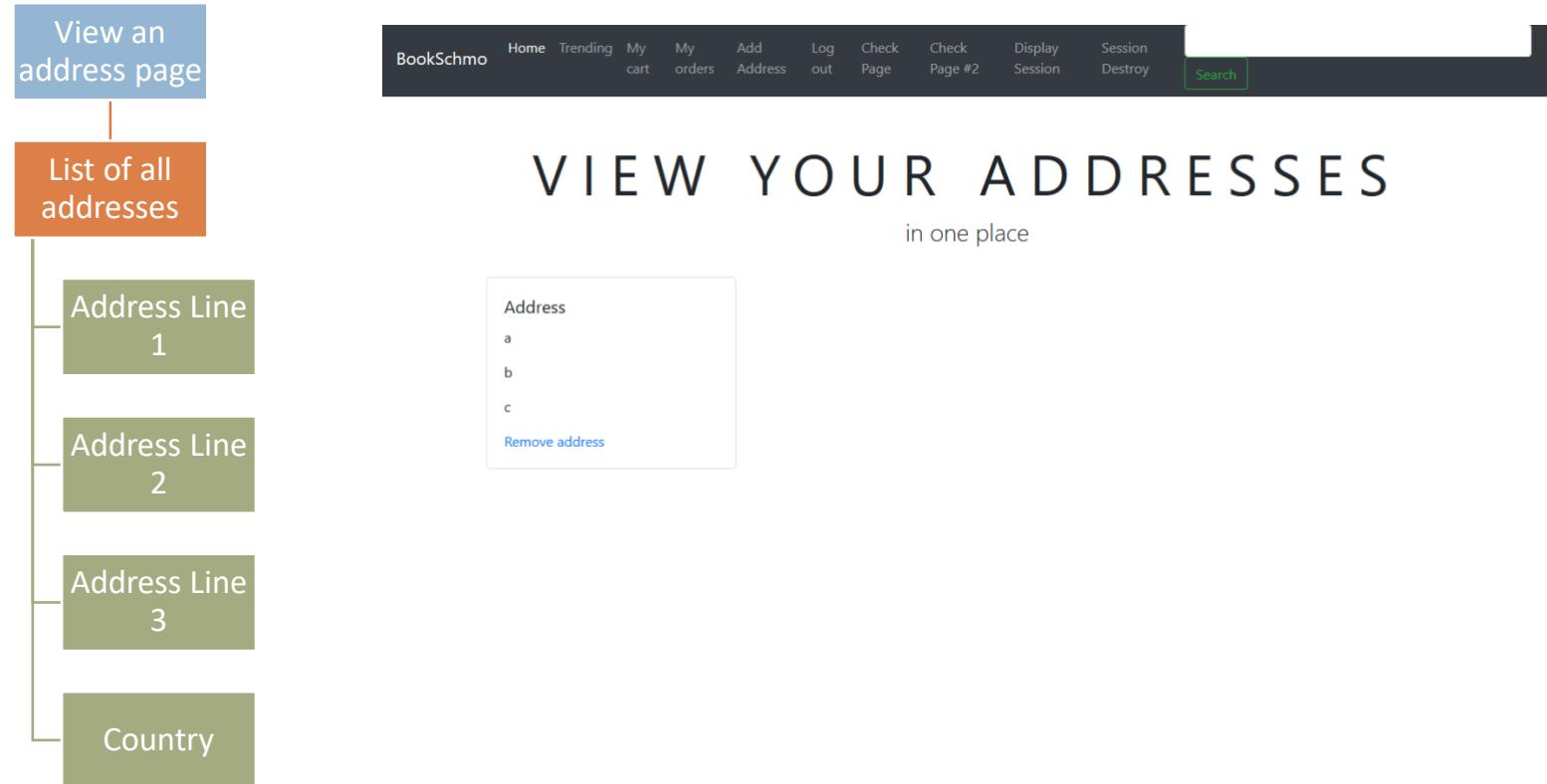
Address Line 1 * Necessary field
Address Line 2
Address Line 3
Country * Necessary field
Add

Notes:

- Address Line 1 and Country will be necessary fields

- Address Line 2 and Address Line 3 will be optional.

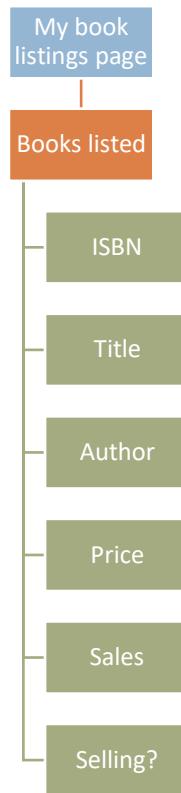
VIEW ALL ADDRESS PAGE DESIGN



Notes:

- Will show a list of all addresses the user has entered.
- A button to remove addresses will remove the address from the database

MY BOOK LISTINGS PAGE



MY BOOK LISTINGS

All your listings in one place

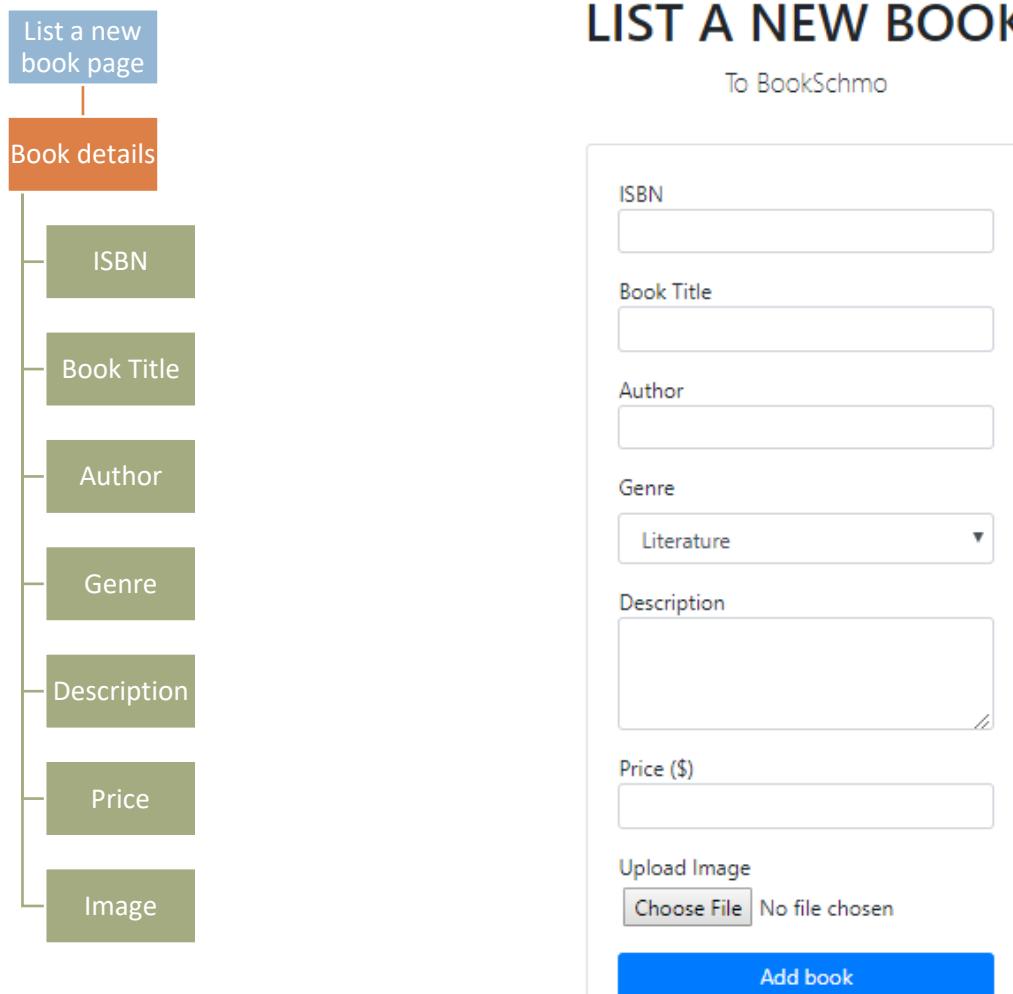
Sort By

ISBN	Title	Author	Price	Sales	Selling?
1234267891232	Harry Potter and the Chamber of Secrets	J.K. Rowling	\$7.9	200	Yes
2131131109874	Inferno	Dan Brown	\$4.99	13	No
2131131109874	Inferno	Dan Brown	\$5.99	15	Yes
8831131109874	Deception Point	Dan Brown	\$13.99	0	Yes
8831131109874	Deception Point	Dan Brown	\$6.99	0	No
8831131109874	Deception Point	Dan Brown	\$10.99	0	No
9038441109874	Secret Star	Nora Roberts	\$5.99	0	Yes

Notes:

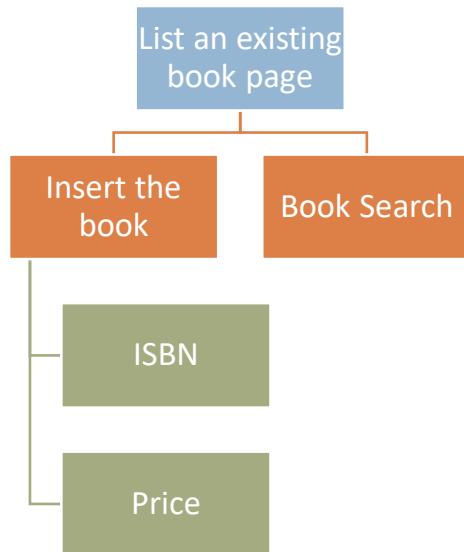
- Shows a list of all the books the seller has listed.
- The details of the book include ISBN, Title, Author, Price, Sales, and whether that listing is currently listing

LIST A NEW BOOK PAGE

**Notes:**

- Shows a list of all the books the seller has listed.
- The details of the book include ISBN, Title, Author, Price, Sales, and whether that listing is currently listing

LIST AN EXISTING BOOK PAGE



LIST AN EXISTING BOOK

To BookSchmo

Book Search

Insert the book

ISBN

Price (\$)

Notes:

- The book search allows creators to search for the isbn of the book that they are interested in listing
- The insert the book allows creators to list the book for any isbn.

DATAFLOW DIAGRAMS

The following screenshot underneath displays the notation I used for my dataflow diagrams

Yourdon and Coad

External Entity



Process



Data Store

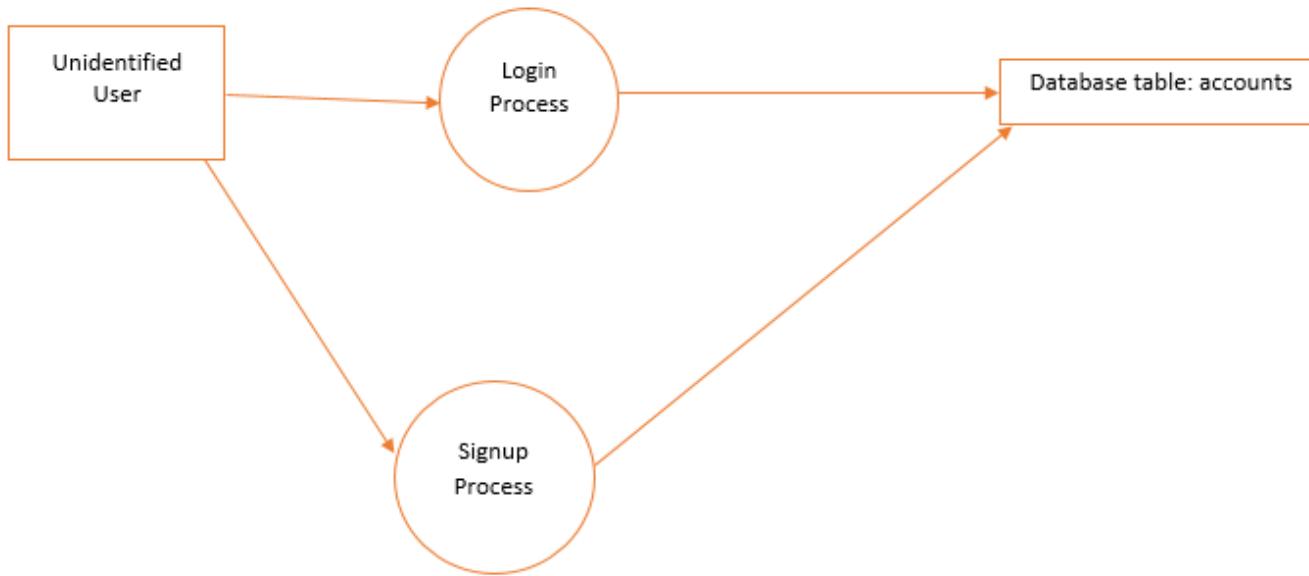


Data Flow



Dataflow diagrams represent the flow of data between entities and data stores.

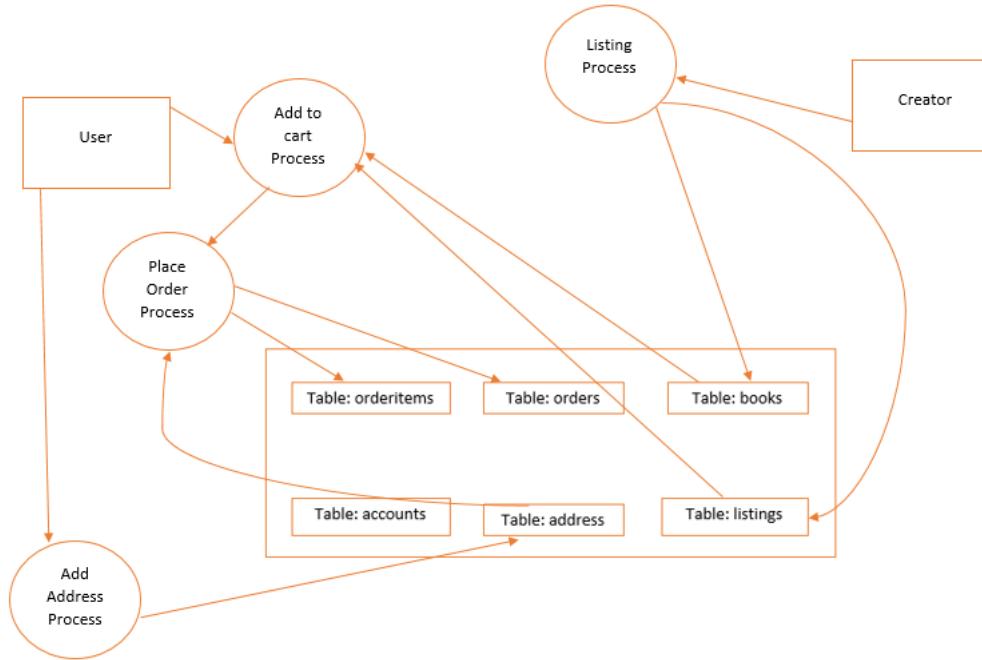
SIGNUP AND LOGIN PROCESS



Notes:

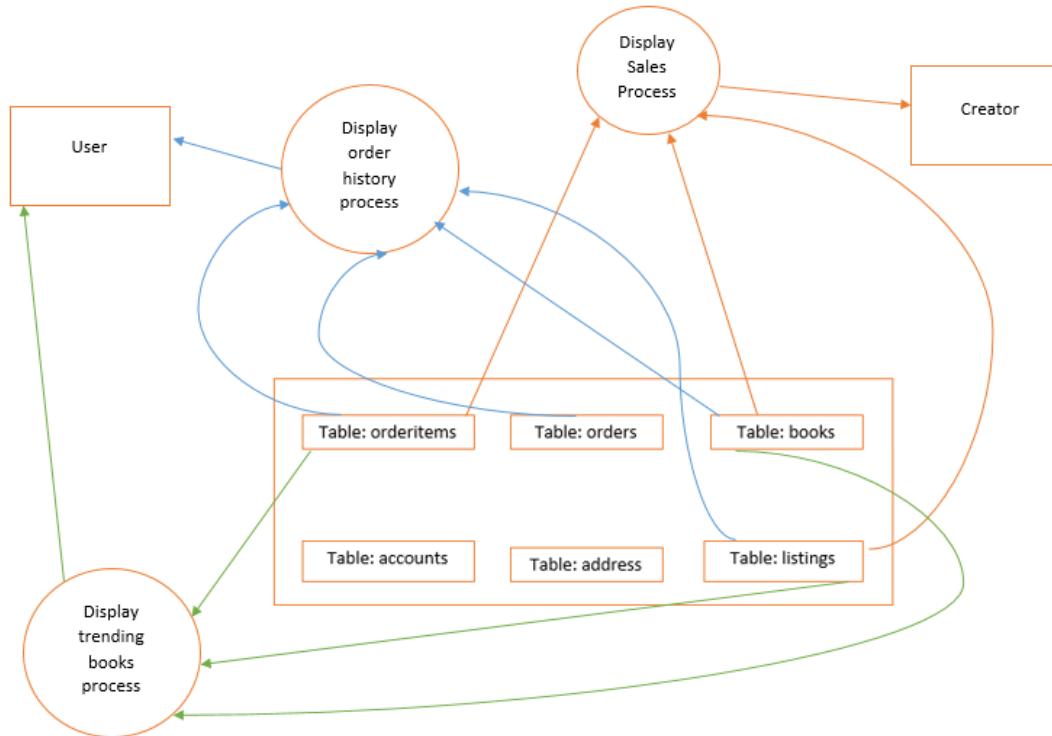
- When the unidentified user signs up or logs in using the signup or login process, the data is passed from the unidentified user to the processes to the database table accounts.

ORDERING AND LISTING PROCESSES

**Notes:**

- When the user adds an address using the add an address process, the data is passed into db table address.
- When the user adds an item to the cart, data from the books and listings tables is passed into the add to cart process.
- When the user checks out the data from the add to cart process is passed to the place order process. The place order process takes information from db table address. It then passes information to db tables orderitems and orders.
- When creator lists a book from the listing process, data is passed from the creator to the listing process to db table books.

DISPLAYING SALES, ORDER HISTORY AND TRENDING BOOKS PROCESS

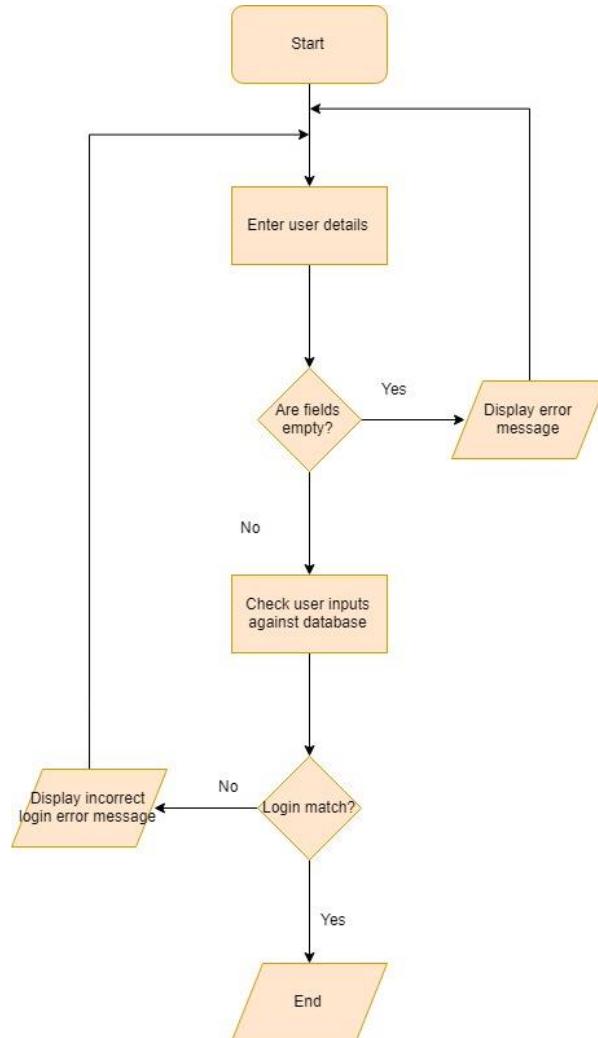
**Notes:**

- Three different chains of data flow are represented by different colours in the data flow diagram above.
- To display the trending books to the user, data is passed from db tables orderitems, listings and books to the display trending books process to the user.
- To display the user order history, data is passed from db tables orderitems, orders, listings and books into the order history process and then to the user
- To display the sales to the creator, data is passed from db tables orderitems, books, and listings to the display sales process and then to the creator.

ALGORITHMS

LOGIN FLOWCHART

The login algorithm is perhaps best visualized as a flowchart. The flowchart below shows the validations and checks that are taken before a user or creator is logged in.



Notes:

- The algorithm asks the user to enter user details
- If the fields are empty, an error message is displayed and the user is asked to enter details again
- The user inputs are checked against the database and if the user inputs do not match the login details in the database, then an error message is displayed and the user is asked to re-enter his details
- If the login details and the user inputs match, then the user is logged in.

LIST A NEW BOOK ALGORITHM

```
upload = True

GET userinputs isbn,title,author,genre,description,price,image
    IF (userinputs are empty OR isbn number is in incorrect format
        OR isbn is not 13 characters OR files uploaded are of incorrect
        extension OR fileSize is greater than 500kb)

        upload = False

    ELSE

        upload = True

TILL upload = True

ADD isbn,title,author,genre,description,image TO db table books
ADD isbn AND price AND SESSION[sellerID] TO db table listings
//The book is now uploaded to the database
```

Notes:

- The following pseudocode above explains the method of adding a new book to the BookSchmo database.

LIST AN EXISTING BOOK ALGORITHM

```
searchInput = GET user input

//Informs the creators which books and ISBNs are selling
QUERY the books table in the database for searchInput

RETURN (ISBN, title, author, genre)

REPEAT

    ISBNInput = GET user input for ISBN to upload

    priceInput = GET user input for price to list book at

    upload = True

    IF (ISBNInput is empty OR priceInput is empty OR ISBN is incorrect format OR ISBN is not
        registered in the database)

        upload = False

    ELSE

        upload = True

    TILL upload = True

ADD ISBNInput AND priceInput AND SESSION[sellerID] TO db table listings
```

Notes:

- The following pseudocode above explains the method of listing an existing book to the BookSchmo database.

ADD TO CART ALGORITHM

```
checkout = False

REPEAT

    bookInput = GET user input ('Enter book to order')

    quantityInput = GET user input ('Enter quantity of book to order')

    RETRIEVE listing.id, booksID for bookInput FROM db table listings

    RETRIEVE title, author, genre, description, price, image FROM db table books

    ADD listing.id,title, author, genre, description, price, image,quantity to SESSION

    checkout = GET user input ('Do you want to checkout?')

    orderTotal = orderTotal + price

UNTIL checkout = True

ADD orderTotal to SESSION

FOR listings.id IN SESSION
    DISPLAY title, author, genre, description, price, image, quantity in table format
DISPLAY orderTotal
```

Notes:

- The following pseudocode above explains the method by which a user adds multiple books to his cart.

CHECKOUT ALGORITHM

```
FOR listings.id IN SESSION
    DISPLAY title, price in table format
DISPLAY orderTotal

FOR addressID in db table address
    DISPLAY addressLine1, addressLine2, addressLine3, country

addressID = GET button user input ('Select address')

submit = GET button user input

IF (submit button is clicked)
    IF(address is selected)
        INSERT SESSION[userID], addressID, SESSION[orderTotal] INTO db table orders
        ordersID = RETURN LAST INSERTED ID

        FOR listings.id in SESSION
            INSERT ordersID, listings.id, SESSION[quantity] INTO db table orderitems

    ELSE
        RETURN error message - "Select an address please"
```

Notes:

- The following pseudocode above explains the method of placing an order consisting of all the items in the cart.

MY ORDERS PAGE ALGORITHM

```

QUERY db table orders WHERE userID = SESSION[userID]

ordersArray = RETURN Fields [orders.id,orderTotal,createdAt] as a two-dimensional array

//ordersArray is a two-dimensional array. order is a one-dimensional array
FOR order in ordersArray

    DISPLAY order[orderTotal, createdAt]

    QUERY (db table orderitems JOIN db table books JOIN db table accounts) WHERE ordersID = order[id]

    orderitemsArray = RETURN query results into an array

    DISPLAY orderitemsArray[title, author, genre, description, price, image, quantity, firstName, lastName]

```

Notes:

- The following pseudocode above explains the method by which a user can check all the orders he has placed.

MY BOOK LISTINGS PAGE ALGORITHM

```

QUERY (db table listings JOIN db table orderitems JOIN db table orderitems)
WHERE sellerID = SESSION[sellerID] AND onSale = 1
GROUP BY listings.id

bookListingsArray = RETURN Fields [isbn,SUM(quantity), author, title, description, genre, sellerID, price,onSale] as a 2 dimensional array

FOR subArray in bookListingsArray
    //SUM[quantity] is the total number of books bought at that price
    DISPLAY subArray[title,author,genre,price,SUM[quantity]] in table format

```

Notes:

- The following pseudocode above explains the method by which a seller can check all the books he has listed on the website.

TRENDING PAGE ALGORITHM

```

QUERY (db table orderitems JOIN db table listings JOIN db table books)
GROUP BY isbn
ORDER BY SUM[quantity] DESCENDING

trendingArray = RETURN Fields[isbn, author, title, description, genre, imageName] as a 2D array

FOR subArray in trendingArray
    DISPLAY subArray[title,author,genre,description,imageName]

```

Notes:

- The following pseudocode above displays the method that displays the most sold books on the website.

INDEX PAGE ALGORITHM

```

QUERY (db tables books JOIN listings JOIN orderitems JOIN orders)
GROUP BY author
ORDER BY quantityByAuthor

recommendArray = RETURN Fields[author,SUM(quantity) as quantityByAuthor] As two-dimensional array

totalQuantity = 0
authorArray = []
FOR subArray IN recommendArray
    totalQuantity = totalQuantity + quantity
    APPEND subArray(author,quantityByAuthor) to authorArray

numToDisplay = 20
FOR subArray in authorArray
    quotaByAuthor = subArray[quantityByAuthor]/totalQuantity
    numToDisplayByAuthor = quotaByAuthor * numToDisplay

    QUERY db table books
    WHERE author = subArray[author]
    LIMIT numToDisplayByAuthor

    isbnByAuthor = RETURN [isbn] as a one-dimensional array

    FOR subArray IN isbnByAuthor
        QUERY (db table books JOIN listings)
        GROUP BY isbn
        ORDER BY isbnByAuthor

        booksArray = RETURN Fields[isbn,author,title,description,genre,imageName, MIN(price) as minPrice]

    FOR subArray in booksArray
        DISPLAY subArray[title,author,genre,description,imageName,minPrice] in grid-like format

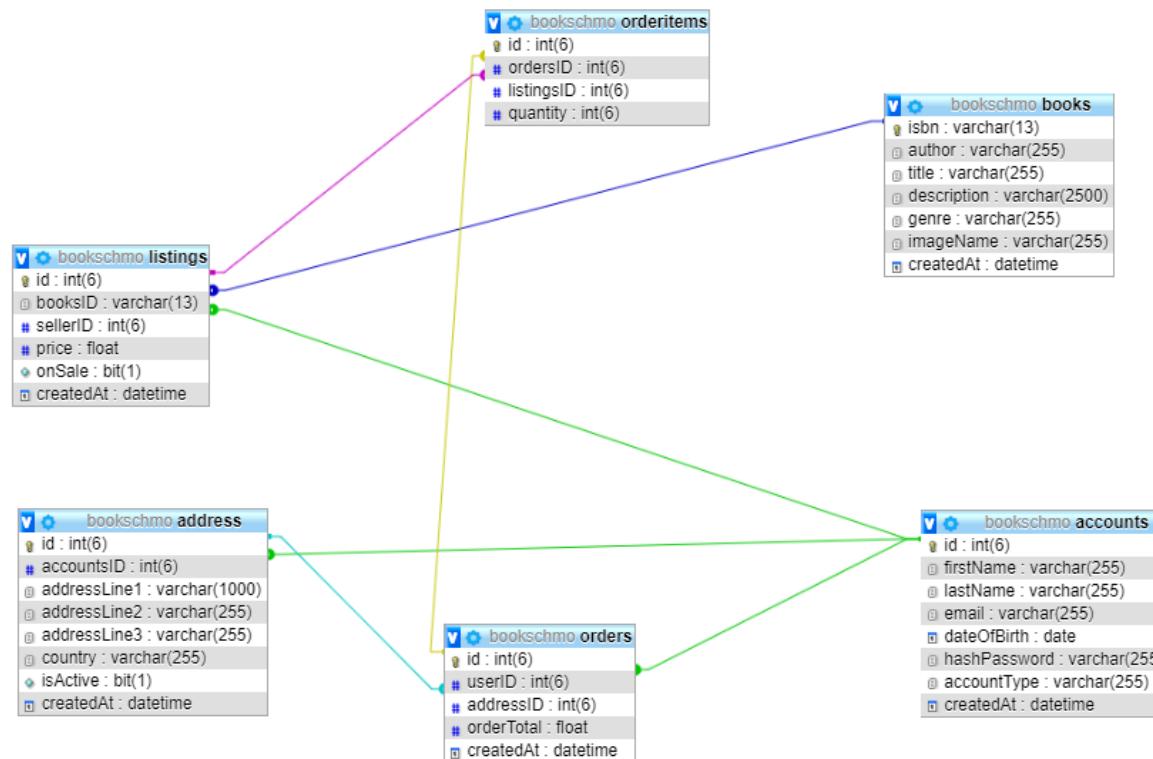
```

Notes:

- The following pseudocode above displays the method by which users are made book recommendations based on the books they have previously bought.

DATABASE DESIGN

DATABASE SCHEMA



DATABASE TABLE: ACCOUNTS

Name	Type	Size	Default	Purpose	Example
id	Integer	6	None	Auto-increment Primary Key	1
firstName	String	255	None	Stores the account's first name	Rishi
lastName	String	255	None	Stores the account's last name	Malhotra
email	String	255	None	Stores the account's email	rishimalhotra@gmail.com
dateOfBirth	Date	-	None	Stores the user's date of birth.	2001-12-20
hashPassword	Varchar	255	None	Stores the user's encrypted password.	\$2y\$10\$e.WO7olyicuCLj4xJXG5re9a/v.kz Y.kUKmYzdGSG1Ftz9QAhGVCK
accountType	Varchar	255	None	Stores the user's account type: Creator/User	creator
createdAt	Datetime	-	Current Timestamp	Date and time when the record was entered into the database.	2019-03-18 11:41:39

DATABASE TABLE: ADDRESS

Name	Type	Size	Default	Purpose	Example
id	Integer	6	None	Auto-increment Primary Key	1
accountsID	Integer	6	None	Foreign Key referencing accounts.id. Stores the id of the user who has entered the address.	1

addressLine1	Varchar	1000	None	Stores the user's first line of address as entered	Road Number 356, West Lane
addressLine2	Varchar	255	None	Stores the user's second line of address as entered	New York City
addressLine3	Varchar	255	None	Stores the user's third line of address as entered	New York
Country	Varchar	255	None	Stores the user's country as entered	USA
isActive	Bit	1	1	A flag that indicates if the user's address is in use or not	1
createdAt	Datetime	-	Current Timestamp	Date and time when the record was entered into the database.	2019-03-18 11:41:39

DATABASE TABLE: BOOKS

Name	Type	Size	Default	Purpose	Example
isbn	Varchar	13	None	Auto-increment Primary Key	9876543210123
Author	Varchar	255	None	Stores the book's author	Dan Brown
Title	Varchar	255	None	Stores the book's title	Inferno
Description	Varchar	255	None	Stores the book's description	Inferno is one of Dan Brown's best selling novels.
Genre	Varchar	255	None	Stores the book's genre	Literature
imageName	Varchar	255	None	Stores the book's image name	Inferno.png
createdAt	Datetime	-	Current Timestamp	Date and time when the record was entered into the database.	2019-03-18 11:41:39

DATABASE TABLE: LISTINGS

Name	Type	Size	Default	Purpose	Example
id	Integer	6	None	Auto-increment Primary Key	1
booksID	Varchar	13	None	Foreign key referencing books.id. Stores the id of the book being listed.	9876543210123
sellerID	Integer	6	None	Foreign key referencing accounts.id. Stores the id of the seller who has listed the book.	1
Price	Float	-	None	Stores the price of the listing	8.99
onSale	Bit	1	None	A flag that determines if the listing is active	1
createdAt	Datetime	-	Current Timestamp	Date and time when the record was entered into the database.	2019-03-18 11:41:39

DATABASE TABLE: ORDERITEMS

Name	Type	Size	Default	Purpose	Example
id	Integer	6	None	Auto-increment Primary Key	1
ordersID	Integer	6	None	Foreign key referencing orders.id	11
listingsID	Integer	6	None	Foreign key referencing accounts.id. Stores the listing at which the book was purchased.	12
Price	Integer	6	None	Stores the quantity of books being bought at the listing.	3

DATABASE TABLE: ORDERS

Name	Type	Size	Default	Purpose	Example
id	Integer	6	None	Auto-increment Primary Key	1
userID	Integer	6	None	Foreign key referencing accounts.id. Stores the id of the user who has made the order.	14
addressID	Integer	6	None	Foreign key referencing address.id. Stores the address to which the order is shipped to.	12
orderTotal	Float	-	None	Stores the subtotal of the order.	56.99
createdAt	Datetime	-	Current Timestamp	Date and time when the record was entered into the database.	2019-03-18 11:41:39

HOW THE DATABASE WORKS

1. The **accounts table** stores the users' personal details. This includes information about their first and last name, email, password, date of birth. The account type field stores information on whether the account registered is a user or creator.
2. The **address table** stores details on the user's address. Using a foreign key referencing to accounts.id, the user can add multiple addresses.
3. The **books table** uses the book isbn as the primary key and contains information on the title, author, description and genre of the book.
4. The **listings table** uses the foreign key booksID referencing to books.id. The foreign key relationship links the listings table with information about the book. The listings table contains information on the price the book is being sold at as well as the seller who is selling the book at that price.
5. A **listings table** stores all the prices that a book is selling / has been previously sold at. This is done to ensure that prices of books in previous orders is not changed if the book's price is changed in the future.

The onSale attribute takes a value of 1 or 0. A 1 denotes that the price of the item in the record is the current price of the item. A 0 indicates that the listing of the item is an old listing.

For example: If seller with sellerID X updates the price of book with bookID Y from price A to price Z, the record which denotes the current listing, will have its onSale attribute changed to a 0.

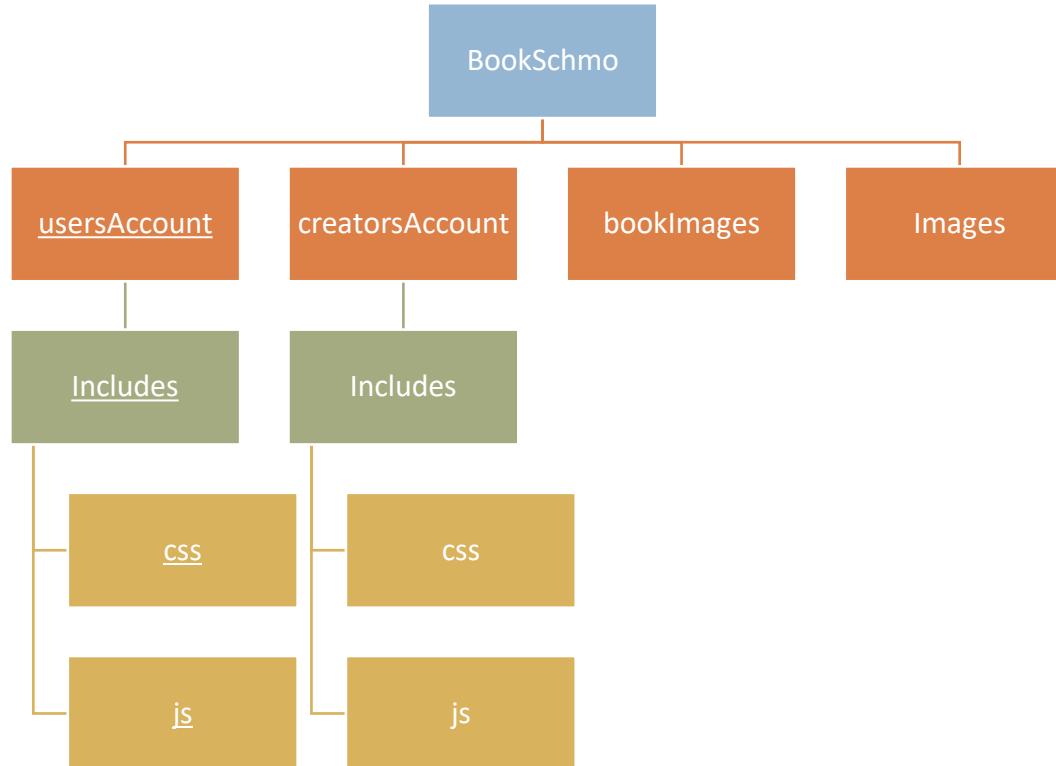
A new record with sellerID X, bookID Y, price Z and an onSale value of 1 and will then be inserted.

6. When a user makes an order, he may add multiple items to his shopping cart. The **orders table** stores information on the order as a whole, including the order total, the address to which the order is shipped and the date when the order is made.
7. The **orderitems table** is a way to contain information on all the books that belonged to a particular order. It contains a foreign keys listingsID referencing listings.id and ordersID referencing orders.id.

The orderitems table stores information on all listings ordered as a part of one order.

The orderitems table stores information on which listings comprised which order. If listing X and listing Y were bought as a part of order Z. One record with ordersID Z and listingsID X and another record with ordersID Z and listingsID Y will be inserted into the orderitems table. The quantities of listingID X and listingID Y will be inserted into the respective records.

FILE STRUCTURE AND ORGANIZATION



- All pages relating to the user's account are stored in the usersAccount folder.
- The scripts running behind the user's account pages are stored in the includes folder in the usersAccount folder
- All pages relating to the creator's account are stored in the creatorsAccount folder
- The scripts running behind the creator's account pages are stored in the includes folder in the creatorsAccount folder
- All pages that must be displayed when user/creator is not logged in are also stored in the usersAccount folder as these pages are largely interlinked with the user's account pages and code repetition is avoided.
- When a book image is uploaded, it is saved in the bookImages folder.
- Other images on the website, for instance the images on the creator's account dashboard and the slideshow images on the user home page are stored in the images folder.

Screenshots of the file hierarchy are displayed below

The BookSchmo folder-

This PC > Windows (C:) > xampp > htdocs > BookSchmo >				
Name	Date modified	Type	Size	
bookImages	3/24/2019 9:19 PM	File folder		
creatorsAccount	3/25/2019 10:49 AM	File folder		
images	3/28/2019 9:52 AM	File folder		
usersAccount	3/26/2019 9:20 AM	File folder		
accountsInformation.txt	3/28/2019 7:33 AM	Text Document	1 KB	
 index.php	3/2/2019 9:57 AM	PHP File	1 KB	
 logout.php	3/6/2019 1:42 PM	PHP File	1 KB	

The usersAccount folder -

This PC > Windows (C:) > xampp > htdocs > BookSchmo > usersAccount



Name	Date modified	Type	Size
includes	3/9/2019 11:59 AM	File folder	
addAddress.php	3/25/2019 8:44 AM	PHP File	4 KB
check.php	3/20/2019 1:30 PM	PHP File	1 KB
check2.php	3/1/2019 5:09 PM	PHP File	1 KB
checkout.php	3/25/2019 9:02 AM	PHP File	4 KB
displaysession.php	3/13/2019 11:15 AM	PHP File	1 KB
index.php	3/25/2019 9:30 AM	PHP File	7 KB
login.php	3/26/2019 8:47 AM	PHP File	2 KB
main.css	3/6/2019 6:51 PM	CSS File	1 KB
mycart.php	3/25/2019 8:43 AM	PHP File	3 KB
myorders.php	3/26/2019 9:09 AM	PHP File	5 KB
orderplaced.php	3/25/2019 7:47 AM	PHP File	1 KB
pageError.php	3/25/2019 7:56 AM	PHP File	1 KB
search.php	3/28/2019 7:43 AM	PHP File	2 KB
sessiondestroy1.php	3/5/2019 4:15 AM	PHP File	1 KB
sessiondestroy2.php	3/5/2019 4:16 AM	PHP File	1 KB
signup.php	3/25/2019 8:26 AM	PHP File	4 KB
trending.php	3/26/2019 9:04 AM	PHP File	3 KB
viewAddress.php	3/26/2019 9:33 AM	PHP File	2 KB
viewbook.php	3/26/2019 9:36 AM	PHP File	4 KB

The usersAccount/includes folder -

Name	Date modified	Type	Size
css	1/29/2019 9:17 PM	File folder	
js	1/29/2019 9:17 PM	File folder	
addAddressScript.php	3/25/2019 9:04 AM	PHP File	2 KB
addtocartScript.php	3/25/2019 9:16 AM	PHP File	1 KB
checkoutAddressScript.php	3/25/2019 9:17 AM	PHP File	1 KB
checkoutScript.php	3/25/2019 9:23 AM	PHP File	2 KB
connection.php	3/6/2019 12:17 PM	PHP File	1 KB
footer.php	3/25/2019 9:25 AM	PHP File	1 KB
header.php	3/25/2019 9:28 AM	PHP File	4 KB
loginScript.php	3/24/2019 10:02 PM	PHP File	3 KB
recommendScript.php	3/25/2019 9:36 AM	PHP File	3 KB
removeAddressScript.php	3/25/2019 9:37 AM	PHP File	1 KB
removefromcartScript.php	3/25/2019 9:38 AM	PHP File	1 KB
signupScript.php	3/25/2019 9:43 AM	PHP File	4 KB

The creatorsAccount folder

Name	Date modified	Type	Size
includes	3/8/2019 6:09 PM	File folder	
bookListings.php	3/26/2019 10:56 AM	PHP File	4 KB
creatorsIndex.php	3/26/2019 10:56 AM	PHP File	2 KB
displaysession.php	3/13/2019 11:15 AM	PHP File	1 KB
listbookmenu.php	3/6/2019 12:12 PM	PHP File	2 KB
listexistingbook.php	3/6/2019 1:14 PM	PHP File	4 KB
listnewbook.php	3/26/2019 10:48 AM	PHP File	5 KB

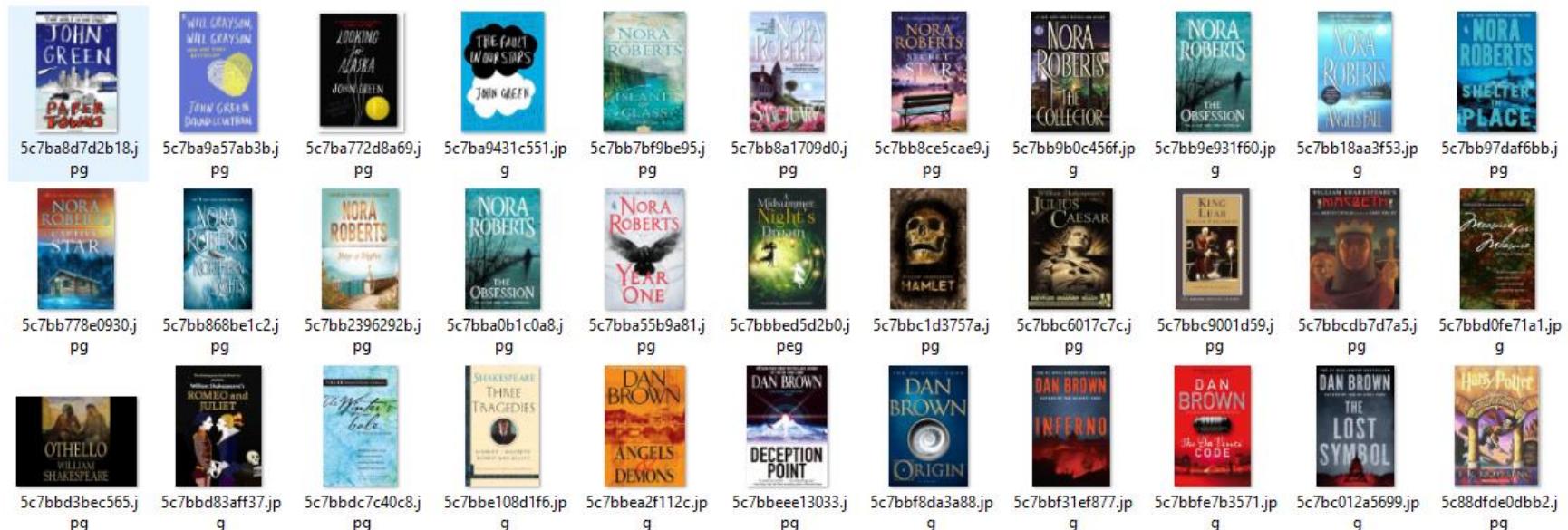
The creatorsAccount/includes folder

This PC > Windows (C:) > xampp > htdocs > BookSchmo > creatorsAccount > includes >

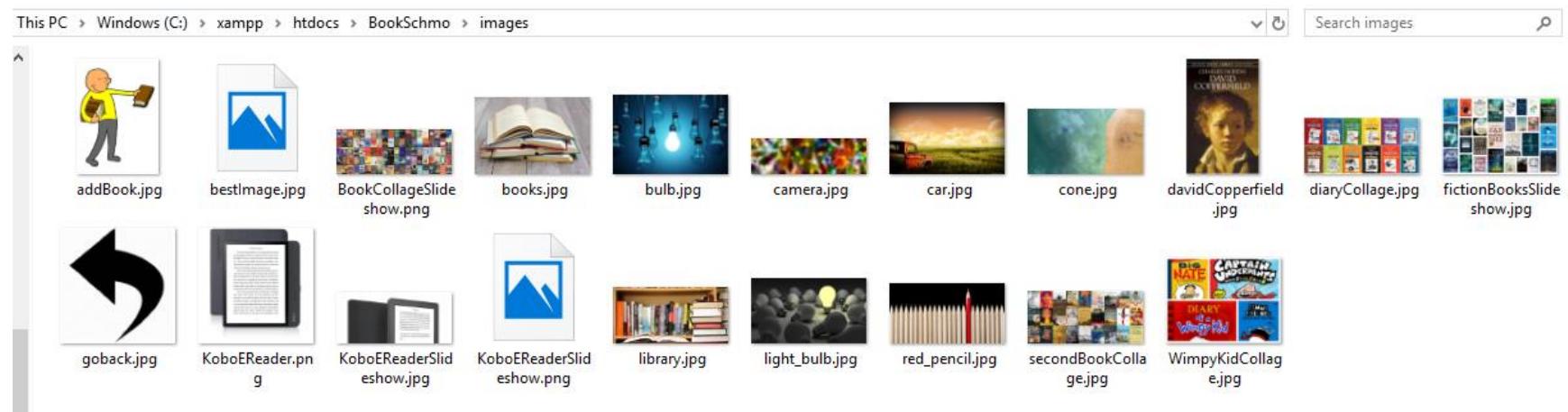
Name	Date modified	Type	Size
css	2/11/2019 9:55 PM	File folder	
js	2/11/2019 9:55 PM	File folder	
bookListingsScript.php	3/25/2019 10:50 AM	PHP File	1 KB
booksearchScript.php	3/25/2019 10:53 AM	PHP File	1 KB
connection.php	3/6/2019 12:13 PM	PHP File	1 KB
creatorsHeader.php	3/18/2019 4:06 PM	PHP File	3 KB
footer.php	3/26/2019 10:55 AM	PHP File	1 KB
listexistingbookScript.php	3/25/2019 10:55 AM	PHP File	4 KB
listnewbookScript.php	3/25/2019 10:56 AM	PHP File	5 KB

The bookImages folder-

PC > Windows (C:) > xampp > htdocs > BookSchmo > bookimages



The images folder



SAMPLE DATABASE QUERIES

Notes for this section: Whenever a dollar sign is seen in the queries below- the query is parametrized as the dollar sign in php is used to identifies variable, array or superglobal.

INDEX PAGE SQL

```
$sql = "
SELECT author,SUM(quantity) AS quantityByAuthor
FROM books
LEFT JOIN listings ON listings.booksID = books.isbn
LEFT JOIN orderitems ON orderitems.listingsID = listings.id
LEFT JOIN orders ON orderitems.ordersID = orders.id
WHERE orders.userID = $userID
GROUP BY author
ORDER BY quantityByAuthor DESC
";
```

The query above is a **cross-table parametrized SQL query**. It selects each book that the person has bought and groups the books by the author. It then sums up the quantity of books bought by each author and returns this as quantityByAuthor. The results are then ordered in descending order. The userID is passed as the parameter.

Later in the code, the results from the SQL query are used to produce the array isbnByAuthor

```
$sql="SELECT isbn,author,title,description,genre,imageName, MIN(price) as minPrice
      FROM books
      LEFT JOIN listings
      ON books.isbn = listings.booksID
      WHERE isbn in ($isbnArrayByAuthor) AND onSale = 1
      GROUP BY isbn
      ORDER BY FIELD(books.isbn,$isbnArrayByAuthor) DESC;";

//echo $sql;
$result = mysqli_query($conn,$sql);
$numRows = mysqli_num_rows($result);
```

The query above is a **cross-table parametrized SQL query**. It selects the book details and the Minimum price of the book by grouping the isbn numbers. The books table is Left Joined with the listings table to allow the minimum price calculation. Finally the results from the query are ordered by the array isbnArrayByAuthor in descending order. NB: The array isbnArrayByAuthor was produced using the first query and is passed as the parameter.

TRENDING PAGE SQL QUERY

```
$sql = "
  SELECT isbn,SUM(quantity) as sales, author, title, SUBSTRING(description,1,100) as description, genre, imageName
  FROM orderitems
  LEFT JOIN listings ON orderitems.listingsID= listings.id
  LEFT JOIN books ON books.isbn = listings.booksID
  WHERE onSale = 1
  GROUP by isbn
  ORDER BY sales DESC
  LIMIT 10;
";

$result = mysqli_query($conn,$sql);
$numRows = mysqli_num_rows($result);
```

The purpose of the trending page is to display the most sold books on the website and the query above selects these most sold books.

The orderitems table is joined with the listings and books table on the constraints mentioned in the query. All the records are grouped by the book isbn and the SUM of quantities are selected as sales. These books are then selected in descending order of sales. The number of records returned is limited to 10, so 10 of the most sold books are displayed on the trending page.

CHECKOUT PAGE SQL QUERY

```

$sql = "INSERT INTO orders (userID,addressID,orderTotal)
VALUES ($userID,$addressID,$orderTotal);";
mysqli_query($conn,$sql);

//Selects the last inserted id from the orders table
$sql = "SELECT LAST_INSERT_ID() as ordersID FROM orders;";
$result = mysqli_query($conn,$sql);
$row = mysqli_fetch_assoc($result);
$ordersID = $row[ 'ordersID' ];

//Inserts the ordersID and listingsID and quantity into the orderitems table
foreach ($_SESSION['mycart'] as $subarray){
    $listingsID = $subarray['listingsID'];
    $quantity = $subarray['quantity'];
    $sql = "INSERT INTO orderitems (ordersID, listingsID, quantity)
VALUES ($ordersID,$listingsID,$quantity);";
    echo $sql;
    mysqli_query($conn,$sql);
}

```

Three queries each stored in the variable \$sql can be seen in the screenshot above.

The above queries are **paramterized sql insert queries**.

- The first query inserts the variables \$userID,\$addressID,\$orderTotal into the orders table.
- The second query returns the last inserted id from ordersID.
- The third query inserts the variables \$ordersID, \$listingsID and \$Quantity into the orderitems table

Throughout my project, there are multiple examples of cross-parametrized Insert Select queries using JOINS, ORDER BY, GROUP BY AND WHERE clauses. However, these will be displayed in the technical solution section of my report. The parametrized queries shown above are representative of the SQL queries that are seen in my project.

STORING THE PASSWORD AND ENCRYPTION

As can be seen in the database section of the analysis section of the document, the user's passwords are encrypted and stored in the database.

The following screenshot is taken from the accounts table of the database.

hashPassword	ac
\$2y\$10\$e.WO7olyicuCLj4xJXG5re9a/v.kzY.kUKmYzdGSG1F...	cr
\$2y\$10\$c4.h9TKu2oCmELUZCWv1uedmzjn7pZy6zadv/5qiMCo...	cr
\$2y\$10\$cH7nrRQwNKJ4zDNQnpSEo.cZ1NCwew7HTWGLLSMXKed...	cr
\$2y\$10\$cWHVaFTSp1bOUYlf05hHi.W2s/6UDVXlbXeyWAxs.2f...	cr
\$2y\$10\$tpKA6iHdAPYnCff5q2q3E.xWDVjNQKeKAPluHKtPmpT...	cr
\$2y\$10\$SkewoPDDVgRPBTAQLJ5CKO9em7D79um4p7UKegLEWJx...	cr
\$2y\$10\$en3ATZXKITxlqgXuVHoxLe9OblmNyPRfw2zdfLj/vSx...	us
\$2y\$10\$ipJEHtQvBRzbml7o96tl.eLNmmNamr9VIPrEAyz/8pd...	us
\$2y\$10\$BM1K02IkdpKvAosddP6tQeteyE1kN/GwRqsBgUrkJ13...	us
\$2y\$10\$5kUZEKc6koVSV2BC71Tn2urDhMnebHd4TnTrdw2zmsl...	us
\$2y\$10\$dM7sworNSd8f1PJynkGQJ.Ruz7SNSxWZe3lyeYgn2wd...	us
\$2y\$10\$ysPauG9KFJjwa.iQku9zmujKs4ShDKJyz6ny8q.GCPL...	us

The default function password_hash is used to hash the password before storing it in the database.

```
25 //Hashing the password
26 $hashPassword = password_hash($password, PASSWORD_DEFAULT);
27
```

The password_verify function is used to verify the password against the database hashed value

```
$row = mysqli_fetch_assoc($result);
$passwordVerified = password_verify($passwordInput,$row['hashPassword']);
```

HOW IT WORKS

- The password hash function generates a random salt and then hashes the user's password with the salt.
- It then concatenates the salt and the hash and the password_hash function returns this and this is stored in the database.
- The password_verify function takes the hashPassword field from the database and retrieves the salt from it. It then hashes the user password input on the login page with the salt. It then concatenates the result generated with the salt. This is then compared with the value in the database.
- If the values are the same, Boolean value True is returned. Otherwise, Boolean value False is returned.

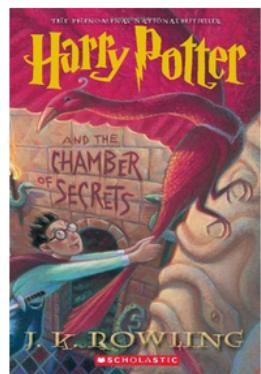
UNIQUE BOOK IMAGE NAMES

```
100 //Generates a unique number
101 $imageName = uniqid().".".getFileActualExt;
```

There is an issue with allowing users to upload book images to the website. If two sellers decide to upload images of the same name and extension type, then storing the two images in the same folder will not be possible as files must be uniquely identified in a folder. As a result, when the user uploads a file, a unique id is generated from the uniqid function. The unique id is based on the microtime of the execution of the script. As the script cannot be executed twice at the same time, the id generated is unique. This unique id is then concatenated with the file extension and the same file is stored with a unique name in the book images folder. The same unique id + file extension is inserted into the imageName field in the accounts table.

SOME NOTES ON THE DESIGN SECTION

The screenshot below shows that the Harry Potter book below is listed at three different prices \$7.9, \$6.9 and \$5.99. It may not make sense initially to buy a book at any listing other than at \$5.99. However, this website is a prototype. If this prototype was to be launched as a real project, book delivery times and shipping charges would be added. In the all sellers section, this information would also be displayed and the user would use all this information and not just the price before considering which listing to buy. Eg: It may be more worthwhile to buy a higher priced book if it is shipped in a lesser time. The display of all sellers listing their books at different prices is a step closer to modelling the project towards a real-world solution.



Harry Potter and the Chamber of Secrets

By J.K. Rowling

Harry Potter and the Chamber of Secrets is a fantasy novel written by British author J. K. Rowling and the second novel in the Harry Potter series

All sellers

Seller Name	Price	
Mary Puppins	\$7.9	<input type="button" value="1"/> Add to cart
Waldo Puck	\$6.9	<input type="button" value="1"/> Add to cart
Dodger Jones	\$5.99	<input type="button" value="1"/> Add to cart

TECHNICAL SOLUTION

UTILITY FILES

File displayed below – connection.php

```
1 <?php
2     session_start();
3     $servername="localhost";
4     $username="root";
5     $password="";
6     $dbname="bookschmo";
7
8     $conn=mysqli_connect($servername, $username, $password, $dbname);
9
10    if (!$conn) {
11        die("Connection failed: " . mysqli_connect_error());
12    }
13
14 ?>
15
```

The connection file is used to make a connection to the database in phpmyadmin.

File displayed below – main.css

```
1  /*Centering images on the front page slideshow*/
2  #slideshowImage1 {
3      object-fit: none; /* Do not scale the image */
4      object-position: center; /* Center the image within the element */
5      width: 100%;
6      max-height: 450px;
7      margin-bottom: 1rem;
8  }
9  .my-cart-product-image {
10     object-position: center; /* Center the image within the element */
11     width: 100%;
12     max-height: 450px;
13     margin-bottom: 1rem;
14 }
15 .my-cart-secondary-text{
16     font-size:1.3em;
17 }
18 .my-cart-price-quantity-headings{
19     font-size:1.2em;
20 }
21 .page-heading{
22     letter-spacing:0.25em;
23     font-size:65px;
24 }
25 .page-sub-heading{
26     font-size:28px;
27 }
28 .category-bar-heading{
29     font-size:20px;
30 }
31 .category-bar-list{
32     padding:inherit;
33     list-style-type:none;
34 }
```

The file main.css is used to style multiple html element on the website.

File displayed below – header.php

```
1  <!DOCTYPE html>
2  <?php
3      #includes the connection file
4      include "includes/connection.php";
5
6  ?>
7
8  <html>
9  <head>
10     <!-- Content delivery networks provides AJAX and Bootstrap files over the internet--&gt;
11
12     &lt;link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css"&gt;
13     &lt;script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"&gt;&lt;/script&gt;
14     &lt;script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.0/dist/umd/popper.min.js"&gt;&lt;/script&gt;
15     &lt;script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js"&gt;&lt;/script&gt;
16
17     &lt;link rel="stylesheet" href="includes/css/bootstrap.css"&gt;
18     &lt;link rel="stylesheet" href="includes/css/bootstrap.min.css"&gt;
19     &lt;link rel="stylesheet" href="includes/css/bootstrap-grid.min.css"&gt;
20     &lt;link rel="stylesheet" href="includes/css/bootstrap-reboot.css"&gt;
21     &lt;link rel="stylesheet" href="includes/css/bootstrap-reboot.min.css"&gt;
22     &lt;link rel="stylesheet" href="includes/css/bootstrap-grid.min.css"&gt;
23     &lt;link rel="stylesheet" href="main.css"&gt;!--Your own stylesheets at the last --&gt;
24 &lt;/head&gt;
25 &lt;body&gt;
26     <!--The main menu at the top of the screen--&gt;
27     &lt;nav class="navbar navbar-expand-lg navbar-dark bg-dark"&gt;
28         &lt;a class="navbar-brand" href="#"&gt;Bookschmo&lt;/a&gt;
29         &lt;button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarText"&gt;
30             &lt;span class="navbar-toggler-icon"&gt;&lt;/span&gt;
31         &lt;/button&gt;
32
33         <!--Navbar elements--&gt;
34         &lt;div class="collapse navbar-collapse" id="navbarText"&gt;
35             &lt;ul class="navbar-nav mr-auto"&gt;
36                 &lt;li class="nav-item active"&gt;
37                     &lt;a class="nav-link" href="index.php"&gt;Home &lt;span class="sr-only"&gt;(current)&lt;/span&gt;&lt;/a&gt;
38                 &lt;/li&gt;
39                 &lt;li class="nav-item"&gt;
40                     &lt;a class="nav-link" href="trending.php"&gt;Trending&lt;/a&gt;
41                 &lt;/li&gt;
42
43             &lt;/ul&gt;</pre>
```

```
44      <?php
45          //Display the following header navs below only if the person is logged in as a user
46          if(isset($_SESSION['accountDetails']['accountType'])){
47              if($_SESSION['accountDetails']['accountType'] == "user"){
48
49                  ?>
50                      <li class="nav-item">
51                          <a class="nav-link" href="mycart.php">My cart</a>
52                      </li>
53                      <li class="nav-item">
54                          <a class="nav-link" href="../usersAccount/myorders.php">My orders</a>
55                      </li>
56                      <li class="nav-item">
57                          <a class="nav-link" href="addAddress.php">Add Address</a>
58                      </li>
59
60                  <?php
61                      }
62                  ?>
63
64
65                  <?php
66                      //Display the nav below if the person is logged in as creator.
67                      if(isset($_SESSION['accountDetails']['accountType'])){
68                          if($_SESSION['accountDetails']['accountType'] == "creator"){
69
70                              ?>
71                                  <li class="nav-item">
72                                      <a class="nav-link" href="../creatorsAccount/creatorsIndex.php">Creator's Site</a>
73                                  </li>
74
75                  <?php
76                      }
77                  ?>
78
79
80      <!-- If the user is logged in, display the logged out button, if logged out, show the login button -->
81      <?php
82          if(isset($_SESSION['accountDetails']['id'])){
83              if($_SESSION['accountDetails']['id'] != ""){
84
85                  ?>
86                      <li class="nav-item">
87                          <a class="nav-link" href="../logout.php">Log out</a>
88                      </li>
89      
```

```
90      }
91      }
92  }else{
93 ▼
94     <li class="nav-item">
95       <a class="nav-link" href="login.php">Login</a>
96     </li>
97   <?php
98   }
99 ?>
100
101    <li class="nav-item">
102      <a class="nav-link" href="check.php">Check Page</a>
103    </li>
104    <li class="nav-item">
105      <a class="nav-link" href="check2.php">Check Page #2</a>
106    </li>
107    <li class="nav-item">
108      <a class="nav-link" href="displaysession.php">Display Session</a>
109    </li>
110    <li class="nav-item">
111      <a class="nav-link" href="sessiondestroy1.php">Session Destroy</a>
112    </li>
113  </ul>
114
115  <!--The search button-->
116  <form class="form-inline" action="search.php" method="POST">
117    <input class="form-control mr-sm-2" style="width:400px" name="search" type="Search">
118    <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search </button>
119  </form>
120
121  </div>
122 </nav>
123
124
```

The file header.php provides the horizontal navbar header to each user page on the website.

File displayed below – footer.php

```
1 <br><br>
2 <footer class="container py-5">
3   <div class="row">
4     <!-- The footer -->
5     <div class="col-12 col-md">
6       <h2 class="display-6">BookSchmo</h2>
7       <small class="d-block mb-3 text-muted">A great mind begins with a great book</small>
8     </div>
9     <div class="col-6 col-md">
10      <h5>My Account</h5>
11      <ul class="list-unstyled text-small">
12        <li><a class="text-muted" href="#">My orders</a></li>
13        <li><a class="text-muted" href="#">My cart</a></li>
14      </ul>
15    </div>
16    <div class="col-6 col-md">
17
18    </div>
19    <div class="col-6 col-md">
20
21    </div>
22  </div>
23 </footer>
24 </body>
25
26 </html>
```

The file footer.php is used to display a footer to all user pages on the website.

LOGIN SYSTEM

LOGIN

Files displayed below - login.php

```
1  <?php
2      include "includes/header.php";
3      //Person should not be able to access the login page if he is already logged in
4      if(isset($_SESSION['accountDetails']['id'])){
5          if($_SESSION['accountDetails']['id'] != ""){
6              header("Location: pageError.php");
7          }
8      }
9
10 ?>
11     <!-- Page Heading and Sub-Heading-->
12     <div class="pt-5 pb-3 text-center mt-100">
13         <h1 class="text-uppercase page-heading">Login</h1>
14         <p class="lead page-sub-heading">To BookSchmo</p>
15     </div>
16     <div class="container">
17         <div class="p-4 mx-auto border rounded" style="width:30%">
18             <?php
19                 // Displays any login errors such as incorrect email.
20                 if(isset($_SESSION["loginErrors"])){
21                     foreach ($_SESSION["loginErrors"] as $element){
22                         echo "<p class='text-danger'>$element</p>";
23                     }
24                 }
25             ?>
26             <!-- The login form below displays the email address and password input fields -->
27             <form action="includes/loginScript.php" method="POST">
28
29                 <div class="form-group">
30                     <label class="mb-0">Email address</label>
31                     <input type="email" name="email" class="form-control p-1" id="loginEmail" placeholder="">
32                 </div>
33                 <div class="form-group">
34                     <label class="mb-0">Password</label>
35                     <input type="password" name="password" class="form-control p-1" id="loginPassword" placeholder="">
36                 </div>
37
38                     <button type="submit" class="btn btn-primary d-block mx-auto w-100 pt-1 pb-1">Sign in</button>
39             </form>
40             <hr>
41             <p class="text-center mb-2" style="font-size:1.1em"> New to BookSchmo? </p>
42             <a type="submit" class="btn btn-primary d-block mx-auto w-100 pt-1 pb-1" href="signup.php">Create an account</a>
43         </div>
44
45     </div>
46     <?php
47         include "includes/footer.php"
48 ?>
```

The file login.php displays a simple login page. It sends data to the file below – loginScript.php

File displayed below - usersAccount/includes/loginScript.php

```
1 <?php
2     include "connection.php";
3     //Setting variables in the POST superglobal to variables in the script for later use
4     $email = mysqli_real_escape_string($conn,$_POST["email"]);
5     $passwordInput = mysqli_real_escape_string($conn,$_POST["password"]);
6
7
8     //SQL Query returns the hash password in the database for the email with
9     //which the user logs in.
10    $sql = "SELECT id,firstName,lastName,email,dateOfBirth,hashPassword,accountType FROM accounts WHERE email = '$email'";
11    $result = mysqli_query($conn,$sql);
12    $resultCheck = mysqli_num_rows($result);
13    $row = mysqli_fetch_assoc($result);
14    $passwordVerified = password_verify($passwordInput,$row['hashPassword']);
15
16    //Empty errors array stores all the errors within the login
17    $_SESSION["loginErrors"] = array();
18    $login = True;
19
20    //Validating empty email
21    if (empty($email)){
22        array_push($_SESSION["loginErrors"],"Please enter an email");
23        $login = False;
24    }
25
26    //Validating invalid email. If no results are returned, then an account is not registered with the email.
27    if($resultCheck == 0){
28        array_push($_SESSION["loginErrors"],"An account with this email does not exist");
29        $login = False;
30    }
31
32    //Validating passwords that do not match
33    if (!$passwordVerified){
34        array_push($_SESSION["loginErrors"],"Incorrect password");
35        $login = False;
36    }
37
38 }
```

```

40     if($login==False){
41         header("Location: ../login.php?login=failure");
42         exit();
43     }else{//If the login is a success
44         unset($_SESSION["loginErrors"]);
45         $_SESSION["accountDetails"] = [];
46         $_SESSION["accountDetails"]['id'] = $row['id'];
47         $_SESSION["accountDetails"]['firstName'] = $row['firstName'];
48         $_SESSION["accountDetails"]['lastName'] = $row['lastName'];
49         $_SESSION["accountDetails"]['email'] = $row['email'];
50         $_SESSION["accountDetails"]['dateOfBirth'] = $row['dateOfBirth'];
51         $_SESSION["accountDetails"]['accountType'] = $row['accountType'];
52         $_SESSION['mycart'] = [];
53
54         if($row['accountType'] == "user"){
55             header("Location: ../index.php?login=success");
56             exit();
57         }elseif ($row['accountType'] == "creator") {
58             header("Location: ../../creatorsAccount/creatorsIndex.php?login=success");
59             exit();
60         }
61     }
62 }
63
64
65
66
67 // ?>

```

The login algorithm shown above produced below in the [Design section](#) of the report and is explained in detail there.

SIGNUP

File displayed below – signup.php

The file below displays the signup webpage. It passes data into the file – signupScript

```

1 <?php
2     include "includes/header.php";
3
4     if (isset($_SESSION["signupDetails"])){
5         print_r($_SESSION["signupDetails"]);
6
7     }
8     print_r($_SESSION);
9 ?>
10
11
12     <!-- Page Heading and Sub-Heading-->
13     <div class="pt-5 pb-3 text-center mt-100">
14         <h1 class="text-uppercase page-heading">Sign up</h1>
15         <p class="lead page-sub-heading">To BookSchmo</p>
16     </div>
17
18     <!--Form elements-->
19     <!--Each form element has a value attribute attached to it. If the user submits form incorrectly-->
20     <!--and is redirected to the signup page, then his old submissions will be stored in the form input field-->
21     <div class="container">
22         <div class="p-4 mx-auto border rounded" style="width:30%">
23             <?php
24             if(isset($_SESSION["signupErrors"])){
25                 foreach ($_SESSION["signupErrors"] as $element){
26                     echo "<p class='text-danger'>$element</p>";
27                 }
28             }
29         ?>
30         <!-- Signup form displays the fields - First Name, Last name, Email Address, Date of Birth, Password, Re-enter Password, User/Creator -->
31         <form action="includes/signupScript.php" method="POST">
32             <div class="form-group">
33                 <label class="mb-0">First Name</label>
34                 <input type="text" name="firstName" class="form-control p-1" id="signupFirstName" placeholder="" value="<?php
35                     if(isset($_SESSION["signupDetails"])[0]){
36                         echo $_SESSION["signupDetails"][0];
37                     }
38                 ?>">
39             </div>
40             <div class="form-group">
41                 <label class="mb-0">Last Name</label>
42                 <input type="text" name="lastName" class="form-control p-1" id="signupLastName" placeholder="" value="<?php
43                     if(isset($_SESSION["signupDetails"])[1]){
44                         echo $_SESSION["signupDetails"][1];
45                     }
46                 ?>">

```

File displayed below – signupScript.php

The file signupScript displays the signup algorithm displayed in the Design Section.

```
49         </div>
50     <div class="form-group">
51         <label class="mb-0">Email address</label>
52         <input type="email" name="email" class="form-control p-1" id="signupEmail" placeholder="" value="<?php
53             if(isset($_SESSION["signupDetails"])[2])){
54                 echo $_SESSION["signupDetails"]的文化[2];
55             }
56         ?>">
57     </div>
58     <div class="form-group">
59         <label class="mb-0">Date of Birth</label>
60         <input type="date" name="dateOfBirth" class="form-control p-1" id="signupEmail" placeholder="" value="<?php
61             if(isset($_SESSION['signupDetails'][3])){
62                 echo $_SESSION['signupDetails'][3];
63             }
64         ?>">
65     </div>
66     <div class="form-group">
67         <label class="mb-0">Password</label>
68         <input type="password" name="password" class="form-control p-1" id="signupPassword" placeholder="">
69     </div>
70     <div class="form-group">
71         <label class="mb-0">Re-enter Password</label>
72         <input type="password" name="confirmPassword" class="form-control p-1" id="signupConfirmPassword" placeholder="">
73     </div>
74
75     <div class="form-check">
76         <input class="form-check-input" type="radio" name="accountType" id="radioUser" value="user" checked>
77         <label class="form-check-label">User</label>
78     </div>
79     <div class="form-check mb-3">
80         <input class="form-check-input" type="radio" name="accountType" id="radioCreator" value="creator">
81         <label class="form-check-label">Creator</label>
82     </div>
83
84         <button type="submit" class="btn btn-primary d-block mx-auto w-100 pt-1 pb-1">Sign up</button>
85     </form>
86     <hr>
87     <!-- Give users/creators an option to go back to the login page if they already have an account -->
88     <p class="text-center mb-2" style="font-size:1.1em"> Already have an account? </p>
89     <a type="submit" class="btn btn-primary d-block mx-auto w-100 pt-1 pb-1" href="login.php">Login</a>
90
91     </div>
92 </div>
```

```
97  <?php
98  |    include "includes/footer.php"
99  ?>
100
```

LOGOUT

```
1  <?php
2  |    session_start();
3  |    session_unset();
4  |    header("Location: usersAccount/index.php");
5  |    exit();
6  ?>
```

ADDRESS PAGES

ADD ADDRESS

File displayed below – addAddress.php

```
1 <?php
2     include "includes/header.php";
3     //If the person has logged in with a user account, only then can he access this page
4     if(!isset($_SESSION["accountDetails"]["accountType"])){
5         header("Location: ../usersAccount/login.php");
6     }else{
7         if($_SESSION["accountDetails"]["accountType"] != "user"){
8             header("Location: ../usersAccount/login.php");
9         }
10    }
11    // exists($_SESSION['addressDetails'],0);
12    if (isset($_SESSION["addressDetails"])){
13        print_r($_SESSION["addressDetails"]);
14    }
15 }
16 ?>
17
18
19 <!-- Page Heading and Sub-Heading-->
20 <div class="pt-5 pb-3 text-center mt-100">
21     <h1 class="text-uppercase page-heading">Add an Address</h1>
22     <p class="lead page-sub-heading">To BookSchmo</p>
23 </div>
24
25 <!--Form elements-->
26 <!--Each form element has a value attribute attached to it. If the user submits form incorrectly-->
27 <!--and is redirected to the signup page, then his old submissions will be stored in the form input field-->
```

```
28     <div class="container">
29         <div class="p-4 mx-auto border rounded" style="width:30%">
30             <?php
31                 if(isset($_SESSION["addressErrors"])){
32                     foreach ($_SESSION["addressErrors"] as $element){
33                         echo "<p class='text-danger'>$element</p>";
34                     }
35                 }
36                 if(isset($_GET["insertaddress"])){
37                     if($_GET["insertaddress"]=="success"){
38                         echo "<p class=text-success>Your address has been added</p>";
39                     }
40                 }
41             ?>
42             <!-- 4 inout fields are provided - Address Line 1, Address Line 2, Address Line 3, Country -->
43             <form action="includes/addAddressScript.php" method="POST">
44                 <div class="form-group">
45                     <label class="mb-0">Address Line 1</label>
46                     <span class="text-secondary" style="font-size: 0.9rem"> * Necessary field </span>
47                     <input type="text" name="addressLine1" class="form-control p-1" id="addersLine1" placeholder="" value="<?php
48                         if(isset($_SESSION["addressDetails"])[0]){
49                             echo $_SESSION["addressDetails"][0];
50                         }
51                     ?>">
52                 </div>
53                 <div class="form-group">
54                     <label class="mb-0">Address Line 2</label>
55                     <input type="text" name="addressLine2" class="form-control p-1" id="addersLine2" placeholder="" value="<?php
56                         if(isset($_SESSION["addressDetails"])[1]){
57                             echo $_SESSION["addressDetails"][1];
58                         }
59                     ?>">
60                 </div>
61                 <div class="form-group">
62                     <label class="mb-0">Address Line 3</label>
63                     <input type="text" name="addressLine3" class="form-control p-1" id="addersLine3" placeholder="" value="<?php
64                         if(isset($_SESSION["addressDetails"])[2]){
65                             echo $_SESSION["addressDetails"][2];
66                         }
67                     ?>">
68                 </div>
69             <!-- -->
```

```
75      <div class="form-group">
76          <label class="mb-0">Country</label>
77          <span class="text-secondary" style="font-size: 0.9rem"> * Necessary field </span>
78          <input type="text" name="country" class="form-control p-1" id="addressLine4" placeholder=""
79          value=<?php
80              if(isset($_SESSION["addressDetails"])[3])){
81                  echo $_SESSION["addressDetails"]的文化[3];
82              }
83          ?>">
84      </div>
85
86      <button type="submit" class="btn btn-primary d-block mx-auto w-100 pt-1 pb-1">Add</button>
87  </form>
88  <hr>
89  <p class="text-center mb-2" style="font-size:1.1em"> Already have an address? </p>
90  <a type="submit" class="btn btn-primary d-block mx-auto w-100 pt-1 pb-1" href="viewAddress.php">Check them here</a>
91 </div>
92 </div>
93
94
95 <?php
96     include "includes/footer.php"
97 ?>
```

The file below displays the add address web page. It passes data to the addAddress script

File displayed below – addAddressScript.php

```

1 <?php
2     include "connection.php";
3     //retrieves the accountsID from the session
4     $accountsID = $_SESSION["accountDetails"]['id'];
5     //retrieves the variables below from the form submitted on addAddress.php
6     $addressLine1 = $_POST['addressLine1'];
7     $addressLine2 = $_POST['addressLine2'];
8     $addressLine3 = $_POST['addressLine3'];
9     $country = $_POST['country'];
10
11     $_SESSION["addressErrors"] = array();
12     $addAddress = True;
13     //Validating empty form elements
14     if (empty($addressLine1) || empty($country)){
15         array_push($_SESSION["addressErrors"], "Missing Fields");
16         $addAddress = False;
17     }
18     if($addAddress == False){//If addAddress is False return errors
19         $_SESSION["addressDetails"] = array();
20         array_push($_SESSION["addressDetails"],$addressLine1,$addressLine2,$addressLine3,$country);
21         header("Location: ../addAddress.php?insertaddress=failure");
22         exit();
23     }else{//If addAddress is True, then add the address to the database
24         $sql = "INSERT INTO address(accountsID,addressLine1,addressLine2,addressLine3,country)
25             VALUES('$accountsID','$addressLine1','$addressLine2','$addressLine3','$country')";
26         mysqli_query($conn,$sql);
27         //Redirects to the addAddress Page with confirmation in the url
28         header("Location: ../addAddress.php?insertaddress=succcess");
29     }
30
31
32
33
34 ?>
```

The addAddress script allows users to add addresses as presented in the design section.

File displayed below – removeAddressScript.php

```
1 <?php
2     include "connection.php";
3
4     //Gets the addressID from the url
5     $id = $_GET["id"];
6
7     //Updates the address and sets the flag isActive in the database to 0 to show that the address is no longer in use.
8     $sql = "UPDATE address
9             SET isActive = 0
10            WHERE id = $id;";
11
12     mysqli_query($conn,$sql);
13     //Redirects to the location viewAddress.php
14     header("Location: ../viewAddress.php");
15     exit();
16
17 ?>
```

The remove address algorithm allows the user to remove addresses.

VIEW ADDRESS

```

1  <?php
2      include "includes/header.php";
3      $accountsID = $_SESSION["accountDetails"]['id'];
4
5      //Retrieves all the addresses submitted by a particular user
6      $sql = "SELECT id, addressLine1, addressLine2, addressLine3, country
7          FROM address
8          WHERE accountsID = $accountsID AND isActive = 1";
9      $result = mysqli_query($conn,$sql);
10     $resultCheck = mysqli_num_rows($result);
11
12 ?>
13     <!-- Page Heading and Sub-Heading-->
14     <div class="pt-5 pb-3 text-center mt-100">
15         <h1 class="text-uppercase page-heading">View your addresses</h1>
16         <p class="lead page-sub-heading">in one place</p>
17     </div>
18     <div class="container">
19         <!-- Displays all the addresses submitted by a particular user -->
20         <div class="row">
21             <?php
22                 if($resultCheck > 0){
23                     while ($row = mysqli_fetch_assoc($result)){
24                         // The while loop loops through all the records returned by the sql query and displays them
25                     ?>
26
27                         <div class="col-3 mr-4 mb-4">
28                             <div class="card" style="width: 18rem;">
29                                 <div class="card-body">
30                                     <h5 class="card-title">Address</h5>
31                                     <p class="card-text"><?php echo $row['addressLine1'];?></p>
32                                     <p class="card-text"><?php echo $row['addressLine2'];?></p>
33                                     <p class="card-text"><?php echo $row['addressLine3'];?></p>
34                                     <a href="includes/removeAddressScript.php?id=<?php echo $row['id'];?>" class="card-link">Remove address</a>
35                                 </div>
36                             </div>
37                         </div>
38                     </div>
39
40                 <?php
41                     }
42                 ?>
43             </div>
44         </div>
45     </div>
46
47     <?php
48         include "includes/footer.php"
49     ?>
50
51

```

The view address page allows users to view all the addresses they have submitted. It implements the view address algorithm

INDEX PAGE

File displayed below – index.php

```

1  <?php
2      include "includes/header.php";
3      include "includes/recommendScript.php";
4
5      //The trending sql below displays the 4 most bought books on the website
6      $trendingSql = "
7          SELECT isbn,SUM(quantity) as sales, author, title, SUBSTRING(description,1,100) as description, genre, imageName, price
8          FROM orderitems
9          LEFT JOIN listings ON orderitems.listingsID= listings.id
10         LEFT JOIN books ON books.isbn = listings.booksID
11         GROUP by isbn
12         ORDER BY sales DESC
13         LIMIT 4;
14     ";
15     $trendingResult = mysqli_query($conn,$trendingSql);
16     $trendingNumRows = mysqli_num_rows($trendingResult);
17
18
19     <!--Slideshow-->
20     <div id="homePageSlideshow" class="carousel slide" data-ride="carousel">
21         <!-- Slideshow bottom of the frame indicators -->
22         <ol class="carousel-indicators">
23             <li data-target="#homePageSlideshow" data-slide-to="0" class="active"></li>
24             <li data-target="#homePageSlideshow" data-slide-to="1"></li>
25             <li data-target="#homePageSlideshow" data-slide-to="2"></li>
26         </ol>
27         <!-- Slideshow frames -->
28         <div class="carousel-inner">
29             <div class="carousel-item active">
30                 
31                 <div class="carousel-caption d-none d-md-block bg-warning">
32                     <h3 class="">BookSchmo</h3>
33                     <p class="">A great mind begins with a book</p>
34                 </div>
35             </div>
36             <div class="carousel-item">
37                 
38                 <div class="carousel-caption d-none d-md-block bg-warning">
39                     <h3 class="">Kobo E-Reader</h3>
40                 </div>
41             </div>
42             <div class="carousel-item">
43                 
44                 <div class="carousel-caption d-none d-md-block bg-warning">
45                     <h3 class="">Books for Kids</h3>
46                 </div>
47             </div>
48         </div>

```

```
49      <!--slideshow next and previous buttons -->
50      <a class="carousel-control-prev" href="#homePageSlideshow" role="button" data-slide="prev">
51          <span class="carousel-control-prev-icon"></span>
52          <span class="sr-only">Previous</span>
53      </a>
54      <a class="carousel-control-next" href="#homePageSlideshow" role="button" data-slide="next">
55          <span class="carousel-control-next-icon"></span>
56          <span class="sr-only">Next</span>
57      </a>
58  </div>
59
60  <div class="container-fluid">
61      <div class="row">
62          <!--The row is split into two sections of equal width -->
63          <div class="col-6">
64              <div class="text-center overflow-hidden">
65                  <!--A section that displays some of the trending books -->
66                  <div class="my-3 py-3">
67                      <h2 class="display-5">Trending Today</h2>
68                      <a class="lead" href="trending.php">Click to see more</a>
69                  </div>
70                  <div class="row">
71
72          <?php
73              if($trendingNumRows > 0){
74                  while ($trendingRow = mysqli_fetch_assoc($trendingResult)){
75                      $isbn = $trendingRow['isbn'];
76                      $author = $trendingRow['author'];
77                      $title = $trendingRow['title'];
78                      $description = $trendingRow['description'];
79                      $genre = $trendingRow['genre'];
80                      $imageName = $trendingRow['imageName'];
81                      $price = $trendingRow['price'];
82
83                      <div class="col-6 pb-3">
84                          <a href="viewbook.php?isbn=<?php echo $isbn?>"></a>
85                      </div>
86
87
88              <?php
89          }
90
91      </?>
```

```
92         </div>
93     </div>
94   <!--A section devoted to the sponsored Kobo E-Reader-->
95   <div class="col-6">
96     <div class="bg-light text-center overflow-hidden">
97       <div class="my-3 p-3">
98         <h2 class="display-5 d-inline">Kobo E-Reader</h2><a href="https://us.kobobooks.com/" class="ml-2"><small>Sponsored</small></a>
99         <p class="lead mb-0">For a convenient and customized reading experience</p>
100    </div>
101    <a href="https://us.kobobooks.com/">
102      
103    </a>
104  </div>
105 </div>
106 </div>
107 </div>
108 </div>
109
110 <?php
111   if(isset($_SESSION["accountDetails"]['id']) AND !empty($isbnArrayByAuthor)){
112     //Implodes the array isbnArrayByAuthor and splits it into strings separated by commas
113     $isbnArrayByAuthor = implode(",",$_SESSION['isbnArrayByAuthor']);
114
115     $sql="SELECT isbn,author,title,description,genre,imageName, MIN(price) as minPrice
116           FROM books
117           LEFT JOIN listings
118             ON books.isbn = listings.booksID
119           WHERE isbn in ($isbnArrayByAuthor) AND onSale = 1
120           GROUP BY isbn
121           ORDER BY FIELD(books.isbn,$isbnArrayByAuthor) DESC";
122
123     $result = mysqli_query($conn,$sql);
124     $numRows = mysqli_num_rows($result);
125   ?>
```

```

127      <!-- Shows the user books according to his order history-->
128      <div class="album py-5 bg-light p-3">
129          <h3 class="mb-4"> Recommended for you </h3>
130          <div class="row">
131              <?php
132                  if($numRows > 0){
133                      //Loops through the rows returned from the sql
134                      while ($row = mysqli_fetch_assoc($result)){
135                          $isbn = $row['isbn'];
136                          $author = $row['author'];
137                          $title = $row['title'];
138                          $description = $row['description'];
139                          $genre = $row['genre'];
140                          $imageName = $row['imageName'];
141                          $minPrice = $row['minPrice'];
142                  }
143                  <!-- The cards below display all the book recommendations according to the user's order history -->
144                  <div class="col-md-2">
145                      <div class="card mb-4 shadow-sm">
146                          
147                          <div class="card-body">
148                              <h5 class="card-title"><?php echo $title;?></h5>
149                              <h6 class="card-subtitle mb-2 text-muted"><?php echo $author;?></h6>
150                              <div class="d-flex justify-content-between align-items-center">
151                                  <span class="text-success">$<?php echo $minPrice ?></span>
152                                  <div class="btn-group">
153                                      <a href="viewbook.php?isbn=<?php echo $isbn;?>" class="btn btn-sm btn-outline-primary">View Book</a>
154                                  </div>
155                              </div>
156                          </div>
157                      </div>
158                  </div>
159                  <?php
160                      }
161                  }
162                  ?>
163              </div>
164          </div>
165          <?php
166              }
167          ?>
168
169      <?php
170          include "includes/footer.php"
171      ?>

```

The index page displays a slideshow, the top 4 trending books and a recommended books section if the user is logged in.

File displayed below – recommendScript.php

```
1 <?php
2 if(isset($_SESSION["accountDetails"]["accountType"])){
3     if($_SESSION["accountDetails"]["accountType"] == "user"){
4
5         $userID = $_SESSION["accountDetails"]["id"];
6         //The sql below selects the number of books that the user bought by author
7         $sql = "
8             SELECT author,SUM(quantity) AS quantityByAuthor
9                 FROM books
10                LEFT JOIN listings ON listings.booksID = books.isbn
11                LEFT JOIN orderitems ON orderitems.listingsID = listings.id
12                LEFT JOIN orders ON orderitems.ordersID = orders.id
13                WHERE orders.userID = $userID
14                GROUP BY author
15                ORDER BY quantityByAuthor DESC
16            ";
17         $authorArray = [];
18         $isbnArrayByAuthor = [];
19         $totalQuantity = 0;
20         $result = mysqli_query($conn,$sql);
21         $resultCheck = mysqli_num_rows($result);
22
23         if($resultCheck > 0){
24             //loops for each row returned from the sql query
25             while ($row = mysqli_fetch_assoc($result)){
26                 //print_r($row);
27                 $author = $row['author'];
28                 $quantityByAuthor = $row['quantityByAuthor'];
29                 $totalQuantity += $quantityByAuthor;
30
31                 //Adds author quantity association to authorArray
32                 $authorArray[] = [$author, $quantityByAuthor];
33             }
34         }
35     }
36 }
```

```
35 //This array calculates what percentage of total sales are books of a particular author and appends them
36 //as the third element to each subArray
37
38
39 $index = -1; //Initializes an index for the for loop
40 $numToDisplay = 12;
41
42
43 foreach($authorArray as $subArray){
44     $index++;
45     $author = $subArray[0];
46     $quantityByAuthor = $subArray[1];
47
48     //Calculates the number of books bought by an author as a percentage of all the books the user has bought
49     $quotaByAuthor = $quantityByAuthor/$totalQuantity;
50
51     //Calculates the total number of books to display by author
52     $numToDisplayByAuthor = $quotaByAuthor * $numToDisplay;
53     $roundednumToDisplayByAuthor = ROUND($numToDisplayByAuthor);
54
55     //Selects the isbn of the book for all books by that author. Number of results are limited by the
56     //variable roundedNumToDisplayByAuthor
57     $sql = "
58         SELECT isbn
59         FROM books
60         WHERE author = '$author'
61         LIMIT $roundednumToDisplayByAuthor
62     ";
63     $result = mysqli_query($conn,$sql);
64     $resultCheck = mysqli_num_rows($result);
65
66     if($resultCheck > 0){
67         //loops through the rows produced by the sql query
68         while ($row = mysqli_fetch_assoc($result)){
69             $isbn = $row['isbn'];
70             //adds the isbn of each book to the isbnArrayByAuthor
71             array_push($isbnArrayByAuthor,$isbn);
72         }
73     }
74
75
76     //Reverse the array so that the results are seen in the correct order on the index page
77     $isbnArrayByAuthor = array_reverse($isbnArrayByAuthor);
78     //the isbnArrayByAuthor is appended to the session, so that it can be accessed on index.php
79     $_SESSION['isbnArrayByAuthor'] = $isbnArrayByAuthor;
80 }
```

```

80
81
82    }
83
84 ?>

```

The recommendScripts sends data to the index.php page. It implements the index page algorithm shown in the algorithms section in the design section

TRENDING PAGE

File displayed below – trending.php

```

1  <?php
2      include "includes/header.php";
3
4      //The sql returns the book details of the most sold books on the website.
5      //The limit clause at the end restricts the number of rows returned to 10
6      $sql = "
7          SELECT isbn,SUM(quantity) as sales, author, title, SUBSTRING(description,1,100) as description, genre, imageName
8          FROM orderitems
9          LEFT JOIN listings ON orderitems.listingsID= listings.id
10         LEFT JOIN books ON books.isbn = listings.booksID
11         WHERE onSale = 1
12         GROUP by isbn
13         ORDER BY sales DESC
14         LIMIT 10;
15     ";
16
17     $result = mysqli_query($conn,$sql);
18     $numRows = mysqli_num_rows($result);
19
20
21 <div class="container-fluid">
22
23     <!-- Page Heading and Sub-Heading--|
24     <div class="pt-5 pb-5 text-center mt-100">
25         <h1 class="text-uppercase page-heading">Trending</h1>
26         <p class="lead page-sub-heading"> Check what everyone's reading today! </p>
27     </div>
28     <!--Starts a grid structure-->
29     <div class="row bg-light">
30         <div class="col-2">
31             <div class="mr-md-3 pt-1 px-1 pt-md-1 px-md-1 overflow-hidden">
32                 </div>
33             </div>
34
35         <div class="col-10">
36
37             <div class="album py-5 bg-light p-3">
38                 <div class="row">
39                     <!-- Displays all the trending books returned form the sql query above in a grid like structure -->

```

```

40
41         <?php
42         if($numRows > 0){
43             while ($row = mysqli_fetch_assoc($result)){
44                 $isbn = $row['isbn'];
45                 $author = $row['author'];
46                 $title = $row['title'];
47                 $description = $row['description'];
48                 $genre = $row['genre'];
49                 $imageName = $row['imageName'];
50             ?>
51
52             <!-- Individual cards for each book below display the book details -->
53             <div class="col-md-3">
54                 <div class="card mb-4 shadow-sm">
55                     
56                     <div class="card-body">
57                         <h5 class="card-title"><?php echo $title;?></h5>
58                         <h6 class="card-subtitle mb-2 text-muted"><?php echo $author;?></h6>
59                         <p class="card-text"><?php echo $description;?></p>
60                         <div class="d-flex justify-content-between align-items-center">
61                             <div class="btn-group">
62                                 <a href="viewbook.php?isbn=<?php echo $isbn;?>" class="btn btn-sm btn-outline-primary">View Book</a>
63                             </div>
64                         </div>
65                     </div>
66                 </div>
67
68             <?php
69             }
70         }
71     ?>
72
73         </div>
74     </div>
75   </div>
76 </div>
77 </div>
78 <?php
79     include "includes/footer.php"
80 ?>
81

```

Trending.php displays the most bought books on the website. It implements the trending algorithm displayed in the algorithms section of the design stage.

SEARCH PAGE

File displayed below – search.php

```
1 <?php
2     include "includes/header.php";
3     if (!isset($_POST['search'])){
4         header("Location: index.php");
5     }
6     $searchInput = $_POST["search"];
7
8     //Joins books table and listings table on isbn = booksID and returns the book details and the minimum price
9     $sql = "SELECT isbn,author,title,description,genre,imageName, MIN(price) as minPrice
10        FROM books
11        LEFT JOIN listings
12          ON books.isbn = listings.booksID
13          WHERE (title LIKE '%$searchInput%' OR author LIKE '%$searchInput%' OR genre LIKE '%$searchInput%') AND onSale = 1
14          GROUP BY booksID;";
15     $result = mysqli_query($conn,$sql);
16     $resultCheck = mysqli_num_rows($result);
17
18 ?>
19 <!--Page structure split into left-hand categories column and right hand search results column-->
20 <div class="row mt-5">
21     <div class="col-2">
22
23     </div>
24     <div class="col-10">
25         <p> Returning <?php echo $resultCheck ?> search results</p>
26         <hr>
27         <!--Each book returned from the search query-->
28
29 <?php
30     if($resultCheck > 0){
31         while ($row = mysqli_fetch_assoc($result)){
32             $isbn = $row['isbn'];
33             $author = $row['author'];
34             $title = $row['title'];
35             $description = $row['description'];
36             $genre = $row['genre'];
37             $imageName = $row['imageName'];
38             $minPrice = $row['minPrice'];
39     }
40
41     <div class="row pt-3">
42         <div class="col-2">
43             
44         </div>
```

```
48         <a class="btn btn-primary pt-1 pb-1" href="viewbook.php?isbn=<?php echo $isbn?>">View Book</a>
49     </div>
50
51
52     </div>
53     <hr>
54
55     <?php
56     {
57     }
58     ?>
59
60
61
62     </div>
63   </div>
64 </div>
65
66 <?php
67   include "includes/footer.php"
68 ?>
69
```

Search.php displays the results from the user's search. It allows the user to search for books

VIEW BOOK

File displayed below viewbook.php

```
1  <?php
2      include "includes/header.php";
3      //Gets the isbn from the page url
4      $isbn = $_GET['isbn'];
5
6      //The sql query below retuns the sellers that have listed the book in consideration
7      $sql = "SELECT listings.id as listingsID, isbn,author,title,description,genre,imageName, price, sellerID, firstName, lastName
8          FROM books
9          LEFT JOIN listings ON listings.booksID = books.isbn
10         LEFT JOIN accounts ON accounts.id = listings.sellerID
11         WHERE isbn = '$isbn' AND onSale = 1";
12
13     $result = mysqli_query($conn,$sql);
14     $resultCheck = mysqli_num_rows($result);
15     $row = mysqli_fetch_assoc($result);
16
17     $isbn = $row['isbn'];
18     $author = $row['author'];
19     $title = $row['title'];
20     $description = $row['description'];
21     $genre = $row['genre'];
22     $imageName = $row['imageName'];
23
24 ?>
25
26 <div class="row mt-5 ml-4">
27     <!-- The column below displays the book Image -->
28     <div class="col-2">
29         
30     </div>
31     <!-- The column below displays the book title, book author and book description -->
32     <div class="col-6">
33
34         <h1 class="mt-4"><?php echo $row['title']?></h1>
35         <p class="">By <?php echo $row['author']?></p>
36         <p class="lead"><?php echo $row['description']?></p>
37
38     </div>
```

```
38      <!-- The column below displays all the sellers who have listed the book -->
39      <div class="col-4">
40          <h4> All sellers </h4>
41
42          <?php
43              //Executing the sql query
44              $result = mysqli_query($conn,$sql);
45              $numRows = mysqli_num_rows($result);
46
47          ?>
48
49      <div class="container">
50          <!--Simple table containing all sellers-->
51          <table class="table table-hover">
52              <thead>
53                  <tr>
54                      <th scope="col">Seller Name</th>
55                      <th scope="col">Price</th>
56                  </tr>
57              </thead>
58              <tbody>
59                  <?php
60
61                      if($numRows > 0){
62                          while ($row = mysqli_fetch_assoc($result)){
63                              $listingsID = $row['listingsID'];
64                              $sellerID = $row['sellerID'];
65                              $firstName = $row['firstName'];
66                              $lastName = $row['lastName'];
67                              $price = $row['price'];
68
69                              //Loop the html code below for every row returned by the sql query
70                          ?>
71
72                          <tr>
73                              <td scope="row"><?php echo $firstName." ".$lastName;?></td>
74                              <td>$<?php echo $price;?></td>
75                              <td>
76                                  <form action="includes/addtocartscript.php?sellerid=<?php echo $sellerID?>" method="POST">
77
78                                      <!-- Sending all these items to the addtocartScript as hidden input elements where they are added to the session -->
79                                      <input type="hidden" name="listingsID" value="<?php echo $listingsID?>">
80                                      <input type="hidden" name="author" value="<?php echo $author?>">
81                                      <input type="hidden" name="title" value="<?php echo $title?>">
```

```
81          <input type="hidden" name="title" value="php echo $title?&gt;"
82          <input type="hidden" name="description" value="php echo $description?&gt;"
83          <input type="hidden" name="genre" value="php echo $genre?&gt;"
84          <input type="hidden" name="imageName" value="php echo $imageName?&gt;"
85          <input type="hidden" name="price" value="php echo $price?&gt;"
86          <input style="width:50px" type="number" name="quantity" value=1>
87          <button type="submit" class="btn btn-primary">Add to cart</button>
88      </form>
89  </td>
90 </tr>
91
92    <?php
93    :
94    ?
95  ?>
96
97
98      </tbody>
99  </table>
100 </div>
101 </div>
102 </div>
103 <?php
104   include "includes/footer.php"
105 ?>
106
```

The view book page algorithm displays all information about one particular book

ORDERING THE BOOK

ADD TO CART/ REMOVE FROM CART

File displayed below – mycart.php

```
1 <?php
2     include "includes/header.php";
3
4     //If the person has logged in with a user account, only then can he access this page
5     if(!isset($_SESSION["accountDetails"]["accountType"])){
6         header("Location: ../usersAccount/login.php");
7     }else{
8         if($_SESSION["accountDetails"]["accountType"] != "user"){
9             header("Location: ../usersAccount/login.php");
10        }
11    }
12
13 ?>
14
15 <div class="container-fluid">
16     <!--Page heading and subheading -->
17     <div class="pt-5 pb-3 text-center mt-100">
18         <h1 class="text-uppercase page-heading">My cart</h1>
19         <p class="lead page-sub-heading">Everything in one place</p>
20     </div>
21
22     <!--Page split between cart orders || Checkout button and other related items -->
23     <div class="row">
24         <div class="col-9">
25             <div class="my-cart-item-bar ml-5 mt-5 mb-0">
26                 <!--The price and Quantity headings -->
27                 <table class="table table-hover">
28                     <thead>
29                         <tr>
30                             <th scope="col-2">Book Image</th>
31                             <th scope="col-2">Book Name</th>
32                             <th scope="col-2">Price</th>
33                             <th scope="col-2">Quantity</th>
34                             <th scope="col-2"></th>
35                         </tr>
36                     </thead>
37                     <tbody>
```

```
38         <?php
39             $counter = -1;
40             $orderTotal = 0;
41             foreach($_SESSION['mycart'] as $subArray){
42                 $counter++;
43                 //Used to calculate the total price of the order
44                 $orderTotal += ($subArray['price'] * $subArray['quantity']);
45             ?>
46             <tr>
47                 <td>"></td>
48                 <td><?php echo $subArray["title"]; ?></td>
49                 <td><?php echo $subArray["price"]; ?></td>
50                 <td><?php echo $subArray["quantity"]; ?></td>
51                 <td><a href="includes/removefromcartScript.php?index=<?php echo $counter?>">Remove from cart</a></td>
52             </tr>
53
54         <?php
55             }
56             $_SESSION['mycartinfo']['orderTotal'] = $orderTotal;
57         ?>
58         </tbody>
59     </table>
60     </div>
61 </div>
62     <!-- Displays the order subtotal in the right-hand side column --&gt;
63     &lt;div class="col-3"&gt;
64         &lt;div class="bg-light pt-5 pb-5 h-100"&gt;
65             &lt;h5 class = "text-center"&gt; Subtotal: $&lt;?php echo $orderTotal; ?&gt; &lt;/h5&gt;
66             &lt;a href="checkout.php" type="button" class="btn btn-primary d-block mx-auto"&gt;Proceed to checkout&lt;/a&gt;
67         &lt;/div&gt;
68     &lt;/div&gt;
69 &lt;/div&gt;
70
71 &lt;?php
72     include "includes/footer.php"
73 ?&gt;
74</pre>
```

File displayed below – addToCartScript.php

```
1 <?php
2     include "connection.php";
3     //set up a shopping cart in the session
4
5     //when user clicks the add to cart button from the viewbook.php, a form is submitted
6     array_push($_SESSION['mycart'],$_POST);
7     header("Location: ../mycart.php");
8 ?>
```

The addToScript.php implements the add to cart algorithm

File displayed below removefromcartScript.php

```
1 ▼ <?php
2     include "connection.php";
3
4     //As values may be deleted from anywhere in the array the array_values function re-indexes the array
5     $_SESSION['mycart'] = array_values($_SESSION['mycart']);
6
7     //Gets the index from the url
8     $index = $_GET['index'];
9
10    //removes the book from the cart by unsetting the index in the array $_SESSION['mycart']
11    unset($_SESSION['mycart'][$index]);
12
13    //redirects to the page mycart.php
14    header("Location: ../mycart.php");
15
16 ?>
```

removefromcartScript.php removes elements from the cart.

CHECKOUT

File displayed below checkout.php

```
1 <?php
2     include "includes/header.php";
3
4     //If the person has logged in with a user account, only then can he access this page
5     if(!isset($_SESSION["accountDetails"]["accountType"])){
6         header("Location: ../usersAccount/login.php");
7     }else{
8         if($_SESSION["accountDetails"]["accountType"] != "user"){
9             header("Location: ../usersAccount/login.php");
10        }
11    }
12
13
14     //Retrieves the account id from the session
15     $accountsID = $_SESSION["accountDetails"]['id'];
16
17     //Retrieves all the addresses submitted by a particular user
18     $sql = "SELECT id, addressLine1, addressLine2, addressLine3, country
19             FROM address
20             WHERE accountsID = $accountsID";
21     $result = mysqli_query($conn,$sql);
22     $resultCheck = mysqli_num_rows($result);
23
24 ?>
```

```
25 ▼    <div class="container">
26 ▼        <div class="card text-center mt-5 mb-5">
27
28            <div class="card-header">
29                </div>
30
31 ▼        <div class="card-body">
32            <h2 class="card-title">Checkout</h2>
33
34            <p class="card-text">Place your order here</p>
35 ▼            <?php
36                //Displays any errors associated with placing the order
37 ▼                if(isset($_SESSION["checkoutErrors"])){
38                    foreach ($_SESSION["checkoutErrors"] as $element){
39                        echo "<p class='text-danger'>$element</p>";
40                    }
41                }
42            ?>
43            <!-- Displays a table below containing a list of all book names,prices and quantities bought by the user -->
44 ▼            <div class="my-cart-item-bar ml-5 mt-5 mb-0">
45                <!--The price and Quantity headings -->
46                <table class="table table-hover">
47                    <thead>
48                        <tr>
49                            <th scope="col-2">Book Name</th>
50                            <th scope="col-2">Price</th>
51                            <th scope="col-2">Quantity</th>
52                        </tr>
53                    </thead>
54                    <tbody>
55 ▼                    <?php
56                        $counter = -1;
57                        foreach($_SESSION['mycart'] as $subArray){
58                            $counter++;
59                        }
60                    ?>
61                        <tr>
62                            <td><?php echo $subArray["title"]; ?></td>
63                            <td><?php echo $subArray["price"]; ?></td>
64                            <td><?php echo $subArray["quantity"]; ?></td>
65                        </tr>
66
67                        <?php
68                            }
69
70                    </tbody>
71                </table>
72            </div>
```

```

73
74
75 ▼
76 ▼
77 ▼
78     <br>
79     <hr>
80     <div class="row">
81         <?php
82             if($resultCheck > 0){
83                 //For every row returned from the sql query, a list of addresses are displayed
84                 while ($row = mysqli_fetch_assoc($result)){
85                     ?>
86                     <div class="col-3 mr-4 mb-4">
87                         <div class="card" style="width: 18rem;">
88                             <div class="card-body">
89                                 <h5 class="card-title">Address</h5>
90                                 <?php
91                                     //Displays the confirmation message if an address has been selected.
92                                     if(isset($_SESSION['mycartinfo']['selectedAddressID'])){
93                                         if($_SESSION['mycartinfo']['selectedAddressID']==$row['id']){
94                                             //exit();
95                                             echo "<p class='text-success'> Selected </p>";
96                                         }
97                                     }
98                                 ?>
99                                 <p class="card-text"><?php echo $row['addressLine1'];?></p>
100                                <p class="card-text"><?php echo $row['addressLine2'];?></p>
101                                <p class="card-text"><?php echo $row['addressLine3'];?></p>
102                                <p class="card-text"><?php echo $row['country'];?></p>
103                                <a href="includes/checkoutAddressScript.php?id=<?php echo $row['id'];?>" class="card-link">Select address</a>
104                            </div>
105                        </div>
106
107                    <?php
108                }
109            ?>
110        </div>
111        <!-- Display the order total -->
112        <h5 class='text-success'> Total: $<?php echo $_SESSION['mycartinfo']['orderTotal'];?></h5>
113
114        <!-- The checkout button -->
115        <a href="includes/checkoutScript.php" class="btn btn-primary">Checkout</a>
116
117    </div>
118    <div class="card-footer text-muted">
119    </div>
120
121 <?php
122     include "includes/footer.php";
123 ?>

```

Checkout.php displays the cart once again. It passes data into the script – checkoutScript.php

File displayed below – checkoutScript.php

```

1 <?php
2     include "connection.php";
3
4     $_SESSION["checkoutErrors"] = [];
5     $checkout = True;
6     //Checks if the address is set
7     if(!isset($_SESSION['mycartinfo']['selectedAddressID'])){
8         $checkout = False;
9         array_push($_SESSION["checkoutErrors"],"Please select an address");
10    }
11
12    if($checkout == True){
13        $userID = $_SESSION["accountDetails"]['id'];
14        $orderTotal = $_SESSION['mycartinfo']['orderTotal'];
15        $addressID = $_SESSION['mycartinfo']['selectedAddressID'];
16
17        //Inserts the addressID and orderTotal into the orders table
18        $sql = "INSERT INTO orders (userID,addressID,orderTotal)
19            VALUES ($userID,$addressID,$orderTotal);";
20        mysqli_query($conn,$sql);
21
22        //Selects the last inserted id from the orders table
23        $sql = "SELECT LAST_INSERT_ID() as ordersID FROM orders;";
24        $result = mysqli_query($conn,$sql);
25        $row = mysqli_fetch_assoc($result);
26        $ordersID = $row['ordersID'];
27
28        //Inserts the ordersID and listingsID and quantity into the orderitems table
29        foreach ($_SESSION['mycart'] as $subarray){
30            $listingsID = $subarray['listingsID'];
31            $quantity = $subarray['quantity'];
32            $sql = "INSERT INTO orderitems (ordersID, listingsID, quantity)
33                VALUES ($ordersID,$listingsID,$quantity);";
34
35            mysqli_query($conn,$sql);
36        }
37        $_SESSION['mycart'] = [];
38        $_SESSION['mycartinfo'] = [];
39
40        header("Location: ../orderplaced.php");
41    }else{
42        header("Location: ../checkout.php?checkout=fail");
43    }
44
45 ?>
```

checkoutScript implements the checkout algorithm presented in the algorithms section on the design page.

File displayed below – checkoutAddressScript.php

```
1 <?php
2     include "connection.php";
3     $_SESSION["mycartinfo"]['selectedAddressID'] = $_GET['id'];
4     //The if block checks if the select an address error exists and removes the error from the checkout errors
5     //so that it is not displayed.
6     if(isset($_SESSION['checkoutErrors'])){
7         $index = array_search("Please select an address",$_SESSION['checkoutErrors']);
8         unset($_SESSION['checkoutErrors'][$index]);
9     }
10    print_r($_SESSION['checkoutErrors']);
11    header("Location: ../checkout.php");
12
13 ?>
```

checkoutAddressScript allows a user to select an address on the checkout page. It implements the checkout algorithm

ORDER CONFIRMATION

```
1  <?php
2      include "includes/header.php"
3  ?>
4      <!-- This page is the order confirmation page --&gt;
5  &lt;div class="container"&gt;
6      <!-- The card underneath displays the order confirmation text --&gt;
7  &lt;div class="card text-center mt-5 mb-5"&gt;
8      &lt;div class="card-header"&gt;
9          &lt;/div&gt;
10
11     &lt;div class="card-body"&gt;
12         &lt;hr&gt;
13         &lt;br&gt;
14         &lt;h2 class="card-title"&gt;Your order has been placed&lt;/h2&gt;
15         &lt;p class="card-text"&gt;Thank you for shopping with us&lt;/p&gt;
16         &lt;a href="index.php" class="card-link"&gt;Back to home&lt;/a&gt;
17         &lt;br&gt;
18         &lt;hr&gt;
19
20     &lt;/div&gt;
21
22     &lt;div class="card-footer text-muted"&gt;
23         &lt;/div&gt;
24     &lt;/div&gt;
25 &lt;/div&gt;
26
27
28
29 &lt;?php
30     include "includes/footer.php"
31 ?&gt;</pre>
```

The order confirmation page displays the order success message once the order has been confirmed.

MY ORDERS

```
1 <?php
2     include "includes/header.php";
3
4     //If the person has logged in with a user account, only then can he access this page
5     if(!isset($_SESSION["accountDetails"]["accountType"])){
6         header("Location: ../usersAccount/login.php");
7     }else{
8         if($_SESSION["accountDetails"]["accountType"] != "user"){
9             header("Location: ../usersAccount/login.php");
10        }
11    }
12
13    //This query returns all the orders that were made by the user
14    $userID = $_SESSION["accountDetails"]['id'];
15    $ordersSql = "SELECT id, userID, addressID, orderTotal, createdAt
16        FROM orders
17        WHERE userID = $userID
18        ORDER BY createdAt DESC";
19    $ordersResult = mysqli_query($conn,$ordersSql);
20    $ordersResultCheck = mysqli_num_rows($ordersResult);
21
22 ?>
23     <!-- Page Heading and Sub-Heading-->
24     <div class="pt-5 pb-3 text-center mt-100">
25         <h1 class="text-uppercase page-heading">My orders</h1>
26         <p class="lead page-sub-heading">Everything in one place</p>
27     </div>
28
```

```
29 ▼    <div class="container">
30 ▼    <?php
31
32 ▼        if($ordersResultCheck > 0){
33 ▼            while ($ordersRow = mysqli_fetch_assoc($ordersResult)){
34                $ordersID = $ordersRow['id'];
35                $userID = $ordersRow['userID'];
36                $addressID = $ordersRow['addressID'];
37                $orderTotal = $ordersRow['orderTotal'];
38                $createdAt = $ordersRow['createdAt'];
39
40                    //Selects all information about the book from multiple tables
41                    $orderitemssql = "
42                        SELECT orderitems.id, listingsID, quantity, isbn, title, author, description, imageName, price, firstName, lastName
43                        FROM orderitems
44                        LEFT JOIN listings ON orderitems.listingsID = listings.id
45                        LEFT JOIN books ON listings.booksID = books.isbn
46                        LEFT JOIN accounts ON listings.sellerID = accounts.id
47                        WHERE orderitems.ordersID = $ordersID;";
48
49
50                    $orderitemsResult = mysqli_query($conn,$orderitemssql);
51                    $orderitemsResultCheck = mysqli_num_rows($orderitemsResult);
52                    //Generates a new section for each order placed
53 ▼    ?>
54
55        <!--All the information about order including header and actual order is contained in order-listing block -->
56 ▼    <div class="order-listing">
57        <!--Price, Date and Order Details headings for orders -->
58        <div class="row bg-secondary pt-2">
59
60            <div class="col-2">
61                <p class="text-uppercase mb-0 text-white"> Order placed </p>
62                <p class="text-white"> <?php echo $createdAt?></p>
63            </div>
64
65            <div class="col-2">
66            </div>
67
68            <div class="col-8">
69                <p class="text-uppercase text-right text-white mb-0"> Order Total </p>
70                <p class="text-right text-white"> $<?php echo $orderTotal?> </p>
71                <!-- <p class="text-right text-uppercase mb-0"> Order details </p> -->
72            </div>
73        </div>
74
```

```
73
74
75    <?php
76    if($orderitemsResultCheck > 0){
77        while ($orderitemsRow = mysqli_fetch_assoc($orderitemsResult)){
78            $quantity = $orderitemsRow['quantity'];
79            $isbn = $orderitemsRow['isbn'];
80            $title = $orderitemsRow['title'];
81            $author = $orderitemsRow['author'];
82            $description = $orderitemsRow['description'];
83            $imageName = $orderitemsRow['imageName'];
84            $price = $orderitemsRow['price'];
85            $firstName = $orderitemsRow['firstName'];
86            $lastName = $orderitemsRow['lastName'];
87        ?>
88        <!--All the individual product data about the order-->
89        <div class="row pt-3">
90            <div class="col-2">
91                
92            </div>
93            <div class="col-6">
94                <h4 class="mb-0"> <?php echo $title?> </h4> <!-- Book Title -->
95                <small class="text-secondary"> Sold by: <?php echo $firstName." ".$lastName?> </small> <!--Book Seller-->
96                <p class="lead mt-3"> <?php echo $description?> </p> <!-- Book description -->
97                <p class="text-success"> Price: <?php echo $price?> </p> <!-- Book price -->
98                <p class=""> Quantity: <?php echo $quantity?> </p> <!-- Book quantity -->
99
100            </div>
101        </div>
102
103        <?php
104            }
105        }
106    ?>
107    </div>
108    <hr>
109
110    <?php
111        }
112    ?>
113
114    </div>
115
116    <?php
117        include "includes/footer.php"
118    ?>
119
```

UTILITY FILES FOR CREATOR'S PAGE

CREATOR'S HEADER

```
1 <!DOCTYPE html>
2 <?php
3     #includes the connection file
4     include "includes/connection.php";
5 ?>
6
7 <html>
8 <head>
9     <!-- Content delivery networks provides AJAX and Bootstrap files over the internet--&gt;
10
11     &lt;link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css"&gt;
12     &lt;script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"&gt;&lt;/script&gt;
13     &lt;script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.0/dist/umd/popper.min.js"&gt;&lt;/script&gt;
14     &lt;script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js"&gt;&lt;/script&gt;
15
16     &lt;link rel="stylesheet" href="includes/css/bootstrap.css"&gt;
17     &lt;link rel="stylesheet" href="includes/css/bootstrap.min.css"&gt;
18     &lt;link rel="stylesheet" href="includes/css/bootstrap-grid.min.css"&gt;
19     &lt;link rel="stylesheet" href="includes/css/bootstrap-reboot.css"&gt;
20     &lt;link rel="stylesheet" href="includes/css/bootstrap-reboot.min.css"&gt;
21     &lt;link rel="stylesheet" href="includes/css/bootstrap-grid.min.css"&gt;
22     &lt;link rel="stylesheet" href="main.css"&gt;!--Your own stylesheets at the last --&gt;
23 &lt;/head&gt;
24 &lt;body&gt;
25     <!--The main menu at the top of the screen--&gt;
26     &lt;nav class="navbar navbar-expand-lg navbar-dark bg-dark"&gt;
27         &lt;a class="navbar-brand" href="#"&gt;BookSchmo&lt;/a&gt;
28         &lt;button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarText"&gt;
29             &lt;span class="navbar-toggler-icon"&gt;&lt;/span&gt;
30         &lt;/button&gt;
31
32         <!--Navbar elements--&gt;
33         &lt;div class="collapse navbar-collapse" id="navbarText"&gt;
34             &lt;ul class="navbar-nav mr-auto"&gt;
35                 &lt;li class="nav-item active"&gt;
36                     &lt;a class="nav-link" href="creatorsIndex.php"&gt;Home &lt;span class="sr-only"&gt;(current)&lt;/span&gt;&lt;/a&gt;
37                 &lt;/li&gt;
38                 &lt;li class="nav-item"&gt;
39                     &lt;a class="nav-link" href="bookListings.php"&gt;My Book Listings&lt;/a&gt;
40                 &lt;/li&gt;
41             &lt;!--
42                 &lt;li class="nav-item"&gt;
43                     &lt;a class="nav-link" href="sales.php"&gt;Sales&lt;/a&gt;
44                 &lt;/li&gt; --&gt;
45                 &lt;li class="nav-item"&gt;
46                     &lt;a class="nav-link" href="listbookmenu.php"&gt;List a book&lt;/a&gt;
47                 &lt;/li&gt;</pre>
```

```
38      <li class="nav-item">
39          <a class="nav-link" href="bookListings.php">My Book Listings</a>
40      </li>
41      <!--
42          <li class="nav-item">
43              <a class="nav-link" href="sales.php">Sales</a>
44          </li> -->
45          <li class="nav-item">
46              <a class="nav-link" href="listbookmenu.php">List a book</a>
47          </li>
48          <li class="nav-item">
49              <a class="nav-link" href="../index.php">User site</a>
50          </li>
51          <li class="nav-item">
52              <a class="nav-link" href="../logout.php">Log out</a>
53          </li>
54          <li class="nav-item">
55              <a class="nav-link" href="displaySession.php">Display Session</a>
56          </li>
57      </ul>
58
59  </div>
60 </nav>
61
```

The creator's header displays links to all creator pages. It is included at the top of every creator page.

FOOTER

```
1   <br><br>
2   <!-- Displays the footer -->
3 ▼ <footer class="container py-5">
4 ▼   <div class="row">
5     <div class="col-12 col-md">
6       <h2 class="display-6">BookSchmo</h2>
7       <small class="d-block mb-3 text-muted">A great mind begins with a book</small>
8     </div>
9     <div class="col-6 col-md">
10    </div>
11   </div>
12   <div class="col-6 col-md">
13     </div>
14   </div>
15   <div class="col-6 col-md">
16     </div>
17   </div>
18 </div>
19 </footer>
20 </body>
21
22 </html>
```

The footer is included at the bottom of all creator pages.

CONNECTION FILE

```
<?php
    session_start();
    $servername="localhost";
    $username="root";
    $password="";
    $dbname="bookschmo";

    $conn=mysqli_connect($servername, $username, $password, $dbname);

    if (!$conn) {
        die("Connection failed: " . mysqli_connect_error());
    }

    //If the person has logged in with a creator account, only then he will be allowed to access the creator pages
    //This is placed in the connection file, because all the creator pages and the creators scripts pages include the connection file.
    //So none of these pages can be accessed without logging in with a creator's account.
    if(!isset($_SESSION["accountDetails"]["accountType"])){
        header("Location: ../usersAccount/login.php");
    }else{
        if($_SESSION["accountDetails"]["accountType"] != "creator"){
            header("Location: ../usersAccount/index.php");
        }
    }
}

?>
```

CREATOR'S INDEX PAGE

```

1  <?php
2      include "includes/creatorsHeader.php";
3
4  ?>
5      <!-- Page Heading and Sub-Heading--&gt;
6  &lt;div class="container"&gt;
7      &lt;div class="pt-5 pb-3 text-center mt-100"&gt;
8          &lt;h1 class="text-uppercase page-heading"&gt;Creator's Dashboard&lt;/h1&gt;
9          &lt;p class="lead page-sub-heading"&gt;Good Morning&lt;/p&gt;
10     &lt;/div&gt;
11     <!-- Displays all the options available for the seller in a card format --&gt;
12     &lt;div class="row"&gt;
13         &lt;div class="col"&gt;
14             &lt;div class="card" style="width: 18rem;"&gt;
15                 &lt;img class="card-img-top" src="../images/books.jpg"&gt;
16                 &lt;div class="card-body"&gt;
17                     &lt;h5 class="card-title"&gt;My Book Listings&lt;/h5&gt;
18                     &lt;p class="card-text"&gt;Some information on the books you have listed on BookSchmo&lt;/p&gt;
19                     &lt;a href="bookListings.php" class="btn btn-primary"&gt;Take me there&lt;/a&gt;
20                 &lt;/div&gt;
21             &lt;/div&gt;
22         &lt;/div&gt;
23         &lt;div class="col"&gt;
24             &lt;div class="card" style="width: 18rem;"&gt;
25                 &lt;img class="card-img-top" src="../images/addBook.jpg"&gt;
26                 &lt;div class="card-body"&gt;
27                     &lt;h5 class="card-title"&gt;List a book&lt;/h5&gt;
28                     &lt;p class="card-text"&gt;Some information on the books you have listed on BookSchmo&lt;/p&gt;
29                     &lt;a href="listbookmenu.php" class="btn btn-primary"&gt;Take me there&lt;/a&gt;
30                 &lt;/div&gt;
31             &lt;/div&gt;
32         &lt;/div&gt;
33         &lt;div class="col"&gt;
34             &lt;div class="card" style="width: 18rem;"&gt;
35                 &lt;img class="card-img-top" src="../images/goback.jpg"&gt;
36                 &lt;div class="card-body"&gt;
37                     &lt;h5 class="card-title"&gt;User site&lt;/h5&gt;
38                     &lt;p class="card-text"&gt;Go back to the user site&lt;/p&gt;
39                     &lt;a href="../index.php" class="btn btn-primary"&gt;Take me there&lt;/a&gt;
40                 &lt;/div&gt;
41             &lt;/div&gt;
42         &lt;/div&gt;
43
44     &lt;/div&gt;
45 &lt;/div&gt;
46 &lt;?php
47     include "includes/footer.php"
48 ?&gt;</pre>

```

Creator's index page displays links to all creator's pages. It is also the creator's dashboard.

LIST A BOOK

LIST A BOOK MENU

File displayed below – listbookmenu.php

```
1 <?php
2     include "includes/creatorsheader.php";
3
4     ?>
5     <!-- This page gives creators an option to list a new book/ list an existing book --&gt;
6     &lt;div class=container-fluid&gt;
7
8
9         &lt;div class="row"&gt;
10            &lt;div class="col-3"&gt;
11                &lt;/div&gt;
12
13            &lt;!-- Option to list a new book --&gt;
14            &lt;div class="col"&gt;
15                &lt;div class="card mt-5 mx-auto" style="width: 18rem;"&gt;
16                    &lt;div class="card-body"&gt;
17                        &lt;h5 class="card-title"&gt;&lt;a href = "listnewbook.php"&gt; List a new book &lt;/a&gt;&lt;/h5&gt;
18                        &lt;p class="card-text"&gt;List a book which does not exist on the website.&lt;/p&gt;
19                    &lt;/div&gt;
20                &lt;/div&gt;
21            &lt;/div&gt;
22
23            &lt;!-- Option to list an existing book --&gt;
24            &lt;div class="col"&gt;
25                &lt;div class="card mt-5 mx-auto" style="width: 18rem;"&gt;
26                    &lt;div class="card-body"&gt;
27                        &lt;h5 class="card-title"&gt;&lt;a href = "listexistingbook.php"&gt; List an existing book &lt;/a&gt;&lt;/h5&gt;
28                        &lt;p class="card-text"&gt;List a book which already exists on the website.&lt;/p&gt;
29                    &lt;/div&gt;
30                &lt;/div&gt;
31            &lt;/div&gt;
32
33            &lt;div class="col-3"&gt;
34                &lt;/div&gt;
35            &lt;/div&gt;
36        &lt;/div&gt;
37    &lt;?php
38
39    include "includes/footer.php";
40 ?&gt;</pre>
```

The listbookmenu is a page that contains links to two other pages – listnewbook.php and listexistingbook.php

LIST A NEW BOOK

File displayed below – listnewbook.php

```

1  <?php
2      include "includes/creatorsHeader.php";
3
4  ?>
5      <!--Page headings and sub-headings-->
6      <div class="pt-5 pb-3 text-center mt-100">
7          <h1 class="text-uppercase page-heading">List a new book</h1>
8          <p class="lead page-sub-heading">To BookSchmo</p>
9      </div>
10     <div class="container">
11         <div class="p-4 mx-auto border rounded" style="width:30%">
12             <?php
13                 //If any, list the errors associated with uploading a book at the top of the form.
14                 if(isset($_SESSION["listnewbookErrors"])){
15                     foreach ($_SESSION["listnewbookErrors"] as $element){
16                         echo "<p class='text-danger'>$element</p>";
17                     }
18                 }
19
20                 //Display the success message if the book was uploaded
21                 if(isset($_GET["listnewbook"])){
22                     if($_GET["listnewbook"] == "success"){
23                         echo "<p class='text-success'>Book uploaded</p>";
24                     }
25                 }
26             ?>
27             <!--Form takes in ISBN, title, author, genre, description, price inputs and a file upload as well.-->
28             <form action="includes/listnewbookScript.php" method="POST" enctype="multipart/form-data">
29
30                 <!--ISBN input-->
31                 <div class="form-group">
32                     <label class="mb-0">ISBN</label>
33                     <input type="text" name="isbn" class="form-control p-1" id="bookISBN" placeholder="" value="<?php
34                         if(isset($_SESSION["listnewbookDetails"])[0])){
35                             echo $_SESSION["listnewbookDetails"][0];
36                         }
37                         ?>">
38                 </div>
39
40                 <!--Title input-->
41                 <div class="form-group">
42                     <label class="mb-0">Book Title</label>
43                     <input type="text" name="title" class="form-control p-1" id="bookUploadTitle" placeholder="" value="<?php
44                         if(isset($_SESSION["listnewbookDetails"])[1])){
45                             echo $_SESSION["listnewbookDetails"][1];
46                         }
47                     ?>">
48             </form>

```

```
49         ?>">
50     </div>
51
52     <!--Author input--&gt;
53     &lt;div class="form-group"&gt;
54         &lt;label class="mb-0"&gt;Author&lt;/label&gt;
55         &lt;input type="text" name="author" class="form-control p-1" id="bookUploadAuthor" placeholder=""
56         value="<?php
57             if(isset($_SESSION["listnewbookDetails"])[2]){
58                 echo $_SESSION["listnewbookDetails"][2];
59             }
60         ?&gt;"&gt;
61     &lt;/div&gt;
62
63     <!--Genre input--&gt;
64     &lt;div class="form-group"&gt;
65         &lt;label&gt;Genre&lt;/label&gt;
66         &lt;select class="form-control" id="bookUploadGenre" name="genre" default=""&gt;
67             &lt;option &gt;?php
68                 if(isset($_SESSION['listnewbookDetails'][3])){
69                     if($_SESSION['listnewbookDetails'][3] == 'Literature'){
70                         echo "selected='selected'";
71                     }
72                 }
73
74             ?&gt;&gt;Literature&lt;/option&gt;
75             &lt;option &gt;?php
76                 if(isset($_SESSION['listnewbookDetails'][3])){
77                     if($_SESSION['listnewbookDetails'][3] == 'Fiction'){
78                         echo "selected='selected'";
79                     }
80                 }
81
82             ?&gt;&gt;Fiction&lt;/option&gt;
83             &lt;option &gt;?php
84                 if(isset($_SESSION['listnewbookDetails'][3])){
85                     if($_SESSION['listnewbookDetails'][3] == 'Romance'){
86                         echo "selected='selected'";
87                     }
88                 }
89
90             ?&gt;&gt;Romance&lt;/option&gt;
91         &lt;/select&gt;
92     &lt;/div&gt;
93 </pre>
```

```

55
94      <!--Description input-->
95      <div class="form-group">
96          <label class="mb-0">Description</label>
97          <textarea class="form-control" id="exampleFormControlTextarea1" rows="3" name="description"><?php
98              if(isset($_SESSION['listnewbookDetails'][4])){
99                  echo $_SESSION['listnewbookDetails'][4];
100             }
101
102         ?></textarea>
103     </div>
104
105     <!--Price input-->
106     <div class="form-group">
107         <label class="mb-0">Price ($)</label>
108         <input type="number" step="0.01" name="price" class="form-control p-1" id="bookUploadPrice" placeholder="" value="<?php
109             if(isset($_SESSION["listnewbookDetails"][5])){
110                 echo $_SESSION["listnewbookDetails"][5];
111             }
112         ?>">
113     </div>
114
115     <!--The image upload -->
116     <div class="form-group">
117         <label class="mb-0">Upload Image</label>
118         <input type="file" name="file" class="form-control-file p-1" id="signupConfirmPassword">
119     </div>
120
121         <button type="submit" name="submit" class="btn btn-primary d-block mx-auto w-100 pt-1 pb-1">Add book</button>
122     </form>
123     <hr>
124   </div>
125 </div>
126 </div>
127 <?php
128     include "includes/footer.php";
129
130     |>
131 ?>
```

The file listnewbook.php allows creators to list new books on the website. It sends data to listnewbookScript.php

File displayed below – listnewbookScript.php

```
1 <?php
2     include "connection.php";
3
4     //The sql commands take all the isbns and store them in an array.
5     $sql = "SELECT isbn FROM books";
6     $result = mysqli_query($conn,$sql);
7     $resultCheck = mysqli_num_rows($result);
8
9     $isbnArray = array();
10    if($resultCheck > 0){
11        while ($row = mysqli_fetch_assoc($result)){
12            $isbnArray[] = $row['isbn'];
13        }
14    }
15
16
17    if(isset($_POST["submit"])){
18        //Setting form inputs to variables for later use
19        $isbn = mysqli_real_escape_string($conn,$_POST["isbn"]);
20        $title = mysqli_real_escape_string($conn,$_POST["title"]);
21        $author = mysqli_real_escape_string($conn,$_POST["author"]);
22        $genre = mysqli_real_escape_string($conn,$_POST["genre"]);
23        $description = mysqli_real_escape_string($conn,$_POST["description"]);
24        $price = mysqli_real_escape_string($conn,$_POST["price"]);
25
26        //File variables set from the FILE superglobal
27        $userFileName = $_FILES["file"]['name'];
28        $fileTmpName = $_FILES["file"]['tmp_name'];
29        $fileSize = $_FILES["file"]['size'];
30        $fileError = $_FILES["file"]['error'];
31        $fileType = $_FILES["file"]['type'];
32
33        //Splits the file name into image name and extension
34        $fileExt = explode('.', $userFileName);
35
36        //Takes the second part of the file name (the extension) and converts to lowercase
37        $fileActualExt = strtolower(end($fileExt));
38
39        //The allowed image formats
40        $allowedFileFormats = array("jpg", "jpeg", "png");
41
42        //Form validation
43
44        //Empty errors array stores all the errors within the form submission
45        $_SESSION["listnewbookErrors"] = array();
46        $listnewbook = True;
47    }
```

```
46     $listnewbook = True;
47
48     //Validating empty form elements
49     if (empty($isbn) || empty($title) || empty($author) || empty($genre) || empty($description)|| empty($price) ){
50         array_push($_SESSION["listnewbookErrors"],"Missing Fields");
51         $listnewbook = False;
52     }
53     //Validating invalid characters
54     if(!preg_match("/^[\d-]*$/", $isbn)){
55         array_push($_SESSION["listnewbookErrors"],"Invalid characters in the ISBN");
56         $listnewbook = False;
57     }
58     //Validating isbn numbers that are not 13 characters.
59     if(strlen($isbn) != 13){
60         array_push($_SESSION["listnewbookErrors"],"ISBN has to be 13 characters");
61         $listnewbook = False;
62     }
63     //If user-defined isbn already exists in the database, then a new entry cannot be added.
64     if(in_array($isbn,$isbnArray)){
65         array_push($_SESSION["listnewbookErrors"],"A book with the same ISBN already exists on the website.");
66         $listnewbook = False;
67     }
68     //Validating invalid file formats
69     if(!in_array($fileActualExt, $allowedFileFormats)){
70         array_push($_SESSION["listnewbookErrors"],"Can only upload files of JPG, JPEG and PNG format");
71         $listnewbook= False;
72     }
73     //Validating file sizes greater than 500KB
74     if ($fileSize > 500000{
75         array_push($_SESSION["listnewbookErrors"],"Please upload a file less than 500kb!");
76         $listnewbook= False;
77     }
78     //Validating other errors
79     if($fileError != 0){
80         array_push($_SESSION["listnewbookErrors"],"There was an error uploading your file");
81         $listnewbook= False;
82     }
83
84 }
```

```

85      if($listnewbook==False){//If listnewbook is a failure
86          //Add all user details to the listnewbookDetails inside the Sessions superglobal
87          //This is done so that user input is not lost if something is entered wrong
88          $_SESSION["listnewbookDetails"] = array();
89          array_push($_SESSION["listnewbookDetails"],$isbn,$title,$author,$genre,$description,$price);
90          //exit();
91          //Redirecting the user to the listnewbook page if there are errors in the listnewbook form.
92          header("Location: ../listnewbook.php?listnewbook=failure");
93          exit();
94      }else{//If the listnewbook is a success
95
96          //Once the book is listnewbooked, the error array is re-instantiated
97          unset($_SESSION["listnewbookErrors"]);
98          //unset($_SESSION["listnewbookDetails"]);
99          //Generates a unique number
100         $imageName = uniqid().".". $fileActualExt;
101         $fileDestination = "../../../../bookImages/".$imageName;
102         //Moves the listnewbooked file from temporary location to permanent location
103         move_uploaded_file($fileTmpName,$fileDestination);
104
105         //Insert the user details into the database using SQL
106         $sql="INSERT INTO books(isbn,author,title,description,genre,imageName)
107             VALUES('$isbn','$author','$title','$description','$genre','$imageName')";
108         mysqli_query($conn, $sql);
109         //Retrieve the accountID from the session
110         $accountID = $_SESSION['accountDetails']['id'];
111
112         // Insert the isbn,accountID,price and onSale value 1 into db table listings
113         $sql = "INSERT INTO listings(booksID,sellerID,price,onSale) VALUES ('$isbn',$accountID,$price,1)";
114         mysqli_query($conn, $sql);
115         //Redirect users to the home page
116
117         header("Location: ../listnewbook.php?listnewbook=success");
118         exit();
119     }
120 }
121 }
122 ?>
123
124

```

The script listnewbookScript.php implements the algorithm List New book as presented in the design section.

LIST AN EXISTING BOOK

File displayed below – listexistingbook.php

```
1 <?php
2     include "includes/creatorsHeader.php";
3
4 ?>
5     <div class="container-fluid">
6         <!-- Page heading and sub-heading -->
7         <div class="pt-5 pb-3 text-center mt-100">
8             <h1 class="text-uppercase page-heading">List an existing book</h1>
9             <p class="lead page-sub-heading">To BookSchmo</p>
10        </div>
11
12
13     <div class="row">
14         <div class="col">
15             <div class="p-4 mx-auto border rounded" style="width:50%">
16                 <?php
17                     // Displays any errors associated with listing existing books
18                     if(isset($_SESSION["listexistingbookErrors"])){
19                         foreach ($_SESSION["listexistingbookErrors"] as $element){
20                             echo "<p class='text-danger'>$element</p>";
21                         }
22                     }
23
24                     //Displays a success message if the book was listed successfully
25                     if(isset($_GET["listing"])){
26                         if($_GET["listing"] == "success"){
27                             echo "<p class='text-success'>Book uploaded</p>";
28                         }
29                     }
30
31         <!-- Form takes in ISBN and price inputs -->
32         <form action="includes/listexistingbookScript.php" method="POST" enctype="multipart/form-data">
33
34             <!-- ISBN input -->
35             <div class="form-group">
36                 <h3> Insert the book </h3>
37                 <br>
38                 <label class="mb-0">ISBN</label>
39                 <input type="text" name="isbn" class="form-control p-1" id="bookIsbn" placeholder="" value="<?php
40                     if(isset($_SESSION["listabookDetails"])[0]){
41                         echo $_SESSION["listabookDetails"][0];
42                     }
43                     ?>">
44             </div>
45
46
```

```
47          <!-- Price input -->
48  ▼        <div class="form-group">
49            <label class="mb-0">Price ($)</label>
50            <input type="number" step="0.01" min="0.00" name="price" class="form-control p-1" id="bookUploadPrice" placeholder="" value="<?php
51                if(isset($_SESSION["listabookDetails"])[1]){
52                    echo $_SESSION["listabookDetails"][1];
53                }
54            ?>">
55        </div>
56
57        <button type="submit" name="submit" class="btn btn-primary d-block mx-auto w-100 pt-1 pb-1">Add book</button>
58    </form>
59
60    <hr>
61  </div>
62</div>
63
64  <!-- Search for book within the existing page -->
65  ▼        <div class="col">
66    <h4 class="mt-4"> Book Search </h4>
67    <br>
68
69    <!-- Form takes in a search field input -->
70  ▼        <form class="form-inline mb-3" action="includes/booksearchScript.php" method="POST">
71    <div class="form-group">
72      <input class="form-control form-control-lg mr-2" name="searchInput" type="text" placeholder="Search a book...">
73      <button type="submit" class="btn btn-primary btn-lg">Search</button>
74    </div>
75</form>
```

```
76
77      <!-- Table displays the results from the search query --&gt;
78      &lt;?php
79          if(isset($_SESSION['searchResults'])){
80      ?&gt;          &lt;table class="table table-hover"&gt;
81              &lt;thead&gt;
82                  &lt;tr&gt;
83                      &lt;th scope="col"&gt;ISBN&lt;/th&gt;
84                      &lt;th scope="col"&gt;Title&lt;/th&gt;
85                      &lt;th scope="col"&gt;Author&lt;/th&gt;
86                  &lt;/tr&gt;
87              &lt;/thead&gt;
88              &lt;tbody&gt;
89          &lt;?php
90          //Loop the html code below for every row returned by the sql query
91          $numRows = sizeof($_SESSION['searchResults']);
92          if($numRows &gt; 0){
93              foreach ($_SESSION['searchResults'] as $counter){
94          ?&gt;
95                  &lt;tr&gt;
96                      &lt;td scope="row"&gt;&lt;?php echo $counter['isbn']?&gt;&lt;/td&gt;
97                      &lt;td&gt;&lt;?php echo $counter['title'];?&gt;&lt;/td&gt;
98                      &lt;td&gt;&lt;?php echo $counter['author'];?&gt;&lt;/td&gt;
99                  &lt;/tr&gt;
100
101
102                  &lt;?php
103                      }
104                  }
105          ?&gt;
106                  &lt;/tbody&gt;
107          &lt;/table&gt;
108      &lt;?php
109      }
110
111          &lt;/div&gt;
112      &lt;/div&gt;
113  &lt;/div&gt;
114
115  &lt;?php
116      include "includes/footer.php";
117  ?&gt;</pre>
```

The file listexistingbook.php displays the form for listing existing books on the website.

File displayed below – listexistingbookScript.php

```

1 ▼ <?php
2     include "connection.php";
3     //The sql commands take all the ISBNs and store them in an array.
4     $sql = "SELECT ISBN FROM books";
5     $result = mysqli_query($conn,$sql);
6     $resultCheck = mysqli_num_rows($result);
7
8     $ISBNArray = array();
9     if($resultCheck > 0){
10         while ($row = mysqli_fetch_assoc($result)){
11             $ISBNArray[] = $row['ISBN'];
12         }
13     }
14     $accountID = $_SESSION['accountDetails']['id'];
15
16
17     if(isset($_POST["submit"])){
18
19
20         //Setting form inputs to variables for later use
21         $ISBN = mysqli_real_escape_string($conn,$_POST["ISBN"]);
22         $price = mysqli_real_escape_string($conn,$_POST["price"]);
23
24         //Form validation
25         //Empty errors array stores all the errors within the form submission
26         $_SESSION["listexistingbookErrors"] = array();
27         $listexistingbook = True;
28
29         //Validating empty form elements
30         if (empty($ISBN) || empty($price)){
31             array_push($_SESSION["listexistingbookErrors"],"Missing Fields");
32             $listexistingbook = False;
33         }
34         //Validating invalid characters
35         if(!preg_match("/^[\d-]*$/",$ISBN)){
36             array_push($_SESSION["listexistingbookErrors"],"Invalid characters");
37             $listexistingbook = False;
38         }
39         //Validating ISBN numbers that are not 13 characters.
40         if(strlen($ISBN) != 13){
41             array_push($_SESSION["listexistingbookErrors"],"ISBN has to be 13 characters");
42             $listexistingbook = False;
43         }
44
45         //If user-defined ISBN already exists in the database, then a new entry cannot be added.
46         if(!in_array($ISBN,$ISBNArray)){
47             array_push($_SESSION["listexistingbookErrors"],"The book does not exist");
48             $listexistingbook = False;

```

```

49
50
51
52 } //If the user has entered a ISBN and seller ID
53
54 if($listexistingbook==False){//If listexistingbook is a failure
55     //Add all user details to the listexistingbookDetails inside the Sessions superglobal
56     //This is done so that user input is not lost if something is entered wrong
57     $_SESSION["listexistingbookDetails"] = array();
58     array_push($_SESSION["listexistingbookDetails"],$isbn,$price);
59
60     //Redirecting the user to the listexistingbook page if there are errors in the listexistingbook form.
61     header("Location: ../listexistingbook.php?listing=failure");
62     exit();
63
64 }else{//If the listexistingbook is a success
65
66     $sql = "SELECT id
67         FROM listings
68         WHERE sellerID = $accountID AND booksID = $isbn AND onSale = 1";
69     $result = mysqli_query($conn,$sql);
70     $resultCheck = mysqli_num_rows($result);
71     $row = mysqli_fetch_assoc($result);
72     $listingIDChange = $row['id'];
73
74     //Add the ids of the rows where the same book has already been listed by the seller and are on sale
75     if($resultCheck > 0){
76         $sql = "UPDATE listings
77             SET onSale = 0
78             WHERE id = $listingIDChange";
79         mysqli_query($conn,$sql);
80     }
81     //accountID is retrieved from the SESSION
82     $accountID = $_SESSION['accountDetails']['id'];
83
84     //Insert the isbn, accountID and price and onSale value 1 into the db table listings
85     $sql = "INSERT INTO listings(booksID,sellerID,price,onSale)
86             VALUES ('$isbn',$accountID,$price,1);";
87     mysqli_query($conn, $sql);
88
89     //Once the book is listexistingbooked, the error array is re-instantiated
90     unset($_SESSION["listexistingbookErrors"]);
91
92     //Redirect users to the home page
93     header("Location: ../listexistingbook.php?listing=success");
94     exit();
95
96 }
97 ?>
98

```

The file listexistingbookScript.php implements the list existing books algorithm presented in the design section of the document.

File displayed below – bookSearch Script

```
1 <?php
2     include "connection.php";
3
4     //If the user is logged in as a creator, only then he will have access to the page
5
6     //The variable searchInput is set to the the form input from bookSearch.php
7     $searchInput = $_POST['searchInput'];
8
9     //The sql below returns all rows which have the author, title or genre match the search criteria.
10    $sql = "SELECT isbn,author,title,description,genre, imageName FROM books
11        WHERE title LIKE '%$searchInput%'
12        OR author LIKE '%$searchInput%'
13        OR genre LIKE '%$searchInput%'";
14    $result = mysqli_query($conn,$sql);
15    $resultCheck = mysqli_num_rows($result);
16    $_SESSION['searchResults'] = array();
17    if($resultCheck>0){
18        //Loop for every row returned by the sql query
19        while($row = mysqli_fetch_assoc($result)){
20            //Add the inputs to the array searchResults
21            $_SESSION['searchResults'][] = $row;
22        }
23    }
24    header("Location: ../listexistingbook.php");
25
26
27 ?>
```

The file bookSearchScript allows creators to search for books and their isbn numbers.

MY BOOK LISTINGS

File displayed below – bookListings.php

```
1 <?php
2     include "includes/creatorsHeader.php";
3
4     // Retrieve the sellerID from the session
5     $sellerID = $_SESSION['accountDetails']['id'];
6
7
8     //If the orderby parameters are set in the url, then make a string called orderbySql
9     $orderbySql = "";
10    $sorted=False;
11    if(isset($_GET['sortby']) AND isset($_GET['order'])){
12        $sortby = $_GET['sortby'];
13        $order = $_GET['order'];
14        $orderbySql = "$sortby $order,";
15        $sorted=True;
16    }
17
18    //append the orderbySql at the end to sort the results.
19    //The sql below shows all the books the seller has listed along with the total sales for that book
20    $sql = "
21        SELECT isbn,SUM(quantity) as sales,author, title, description, genre, sellerID, price, onSale
22        FROM books
23        LEFT JOIN listings ON books.isbn = listings.booksID
24        LEFT JOIN orderitems ON orderitems.listingsID = listings.id
25        WHERE sellerID = $sellerID
26        GROUP BY listings.id
27        ORDER BY $orderbySql isbn";
28
29
30    $result = mysqli_query($conn,$sql);
31    $numRows = mysqli_num_rows($result);
32
33
34 ?>
```

```
35      <!-- Page Heading and Sub-Heading-->
36  ▼  <div class="pt-5 pb-3 text-center mt-100">
37      <h1 class="text-uppercase page-heading">My book listings</h1>
38      <p class="lead page-sub-heading">All your listings in one place</p>
39  </div>
40
41  ▼  <div class="container">
42  ▼  <div class="form-group">
43  ▼  <?php
44      if($sorted){
45          echo "<p class='text-success float-left'>Sorted</p>";
46      }
47  ?>
48
49      <!-- Redirects to the bookListingsscript page which inserts a get request in the url -->
50  ▼  <form class="float-right mb-3" action ="includes/bookListingsScript.php" method="POST">
51      <label> Sort By </label>
52      <input list="sortbyItems" name="sortby" autocomplete="off">
53  ▼  <datalist id="sortbyItems">
54      <option value="Sales"></option>
55      <option value="Price"></option>
56  </datalist>
57
58      <input list="order" name="order">
59  ▼  <datalist id="order">
60      <option value="ASC">Ascending</option>
61      <option value="DESC">Descending</option>
62  </datalist>
63      <input type="submit">
64  </form>
65  </div>
66      <!--Simple table containing all books-->
67  ▼  <table class="table table-hover">
68  ▼  <thead>
69  ▼  <tr>
70      <th scope="col">ISBN</th>
71      <th scope="col">Title</th>
72      <th scope="col">Author</th>
73      <th scope="col">Price</th>
74      <th scope="col">Sales</th>
75      <th scope="col">Selling?</th>
76  </tr>
77  </thead>
78  <tbody>
```

```
79      <?php
80      //Loop the html code below for every row returned by the sql query
81      if($numRows > 0){
82          while ($row = mysqli_fetch_assoc($result)){
83
84      ?>
85
86      <tr>
87          <td scope="row"><?php echo $row['isbn'];?></td>
88          <td><?php echo $row['title'];?></td>
89          <td><?php echo $row['author'];?></td>
90          <td>$<?php echo $row['price']; ?></td>
91          <td><?php
92              if($row['sales'] == 0){
93                  echo 0;
94              }else{
95                  echo $row['sales'];
96              }?></td>
97          <td><?php //If the book is onSale, display Yes. If not on sale display No
98              if($row['onSale'] == 0){
99                  echo "<p class='text-danger'>No</p>";
100             }else{
101                 echo "<p class='text-success'>Yes</p>";
102             }?></td>
103      </tr>
104      <?php
105          }
106      }
107      ?>
108      </tbody>
109  </table>
110 </div>
111 <?php
112     include "includes/footer.php"
113 ?>
```

The file bookListings.php allows users to look at existing listings of books made by the creator. It sends data to bookListingsScript.

File displayed below – bookListingsScript.php

```

1 <?php
2     include "connection.php";
3     //set the variables sortby and order to form inputs
4     $sortby = $_POST['sortby'];
5     $order = $_POST['order'];
6     //redirect to the booklistings page and insert the sort conditions in the url
7     header("Location: ../bookListings.php?sortby=$sortby&order=$order");
8     exit();
9 ?>

```

The book listings script runs the scripts for sorting the books.

TESTING

TESTING THE PROJECT

No.	Item tested	Test Description	Test Result/Code Provided	Pass or Fail
1	Login Page. Logout button	When a user logs in as a creator – he should be directed to the creator's dashboard. When the user logs in as user, redirected to the user's website. If wrong password and empty username is entered, correct errors are displayed. When the logout button is clicked, the user is logged out.	<p>Please refer to 1.0Login.mp4</p> <p>The login page and logout button both work successfully.</p> <p>The page was tested under normal conditions – allowing users to login with the correct details.</p> <p>The page was tested under erroneous conditions – checking if user could login if he entered the wrong details.</p>	Pass
2	View Address Page Add Address Page	A user should be allowed to view all his current addresses on the view address page and add new addresses on the add address page. On the add address page, not filling in fields - address line 1 and the country will raise an error.	<p>Please refer to 2Address.mp4</p> <pre> <hr> <p class="text-center mb-2" style="font-size:1.1em"> Already have an address? </p> Check them here </pre> <p>The href tag redirects to login.php instead of viewAddress.php. As the user is already logged in, this redirects the user to pageError.php</p> <p>Simple fix by changing the href tag</p>	Fail

		Filling in address lines 2 and 3 is optional. Remove address button, removes the address		
2.1		<p>Refer to 2.1Address.mp4</p> <p>Code fixes-</p> <pre><hr> <p class="text-center mb-2" style="font-size:1.1em"> Already have an address? </p> Check them here </div> ...</div></pre> <p>Addresses are successfully added. The fields address line 1 and country are mandatory. The rest are not.</p> <p>Some addresses cannot be removed. Some can be removed.</p> <p>After echoing the sql query that failed produced by the code below and running it directly on the database,</p> <pre><?php include "connection.php"; \$id = \$_GET["id"]; \$sql = "DELETE FROM address WHERE id = \$id;"; mysqli_query(\$conn,\$sql); echo \$sql; exit(); header("Location: ../viewAddress.php"); exit(); ?></pre> <p>Error</p> <p>SQL query:</p> <pre>DELETE FROM address WHERE id = 12</pre> <p>MySQL said: </p> <pre>#1451 - Cannot delete or update a parent row: a foreign key constraint fails ('bookschmo`.`orders', CONSTRAINT `orders_ibfk_2` FOREIGN KEY (`addressID`) REFERENCES `address` (`id`))</pre>		Fail

		<p>The SQL query failed because it had been used as a foreign key in the orders table, when an order was made to the address.</p> <p>As these addresses cannot be deleted, another column called isActive needs to be created and a 0 will be set to the address if it has been ‘deleted’.</p> <p>Current address table structure –</p> <table border="1"> <thead> <tr> <th>#</th><th>Name</th><th>Type</th><th>Collation</th><th>Attributes</th><th>Null</th><th>Default</th><th>Comments</th><th>Extra</th></tr> </thead> <tbody> <tr> <td>1</td><td>id</td><td>int(6)</td><td></td><td></td><td>No</td><td>None</td><td></td><td>AUTO_INCREMENT</td></tr> <tr> <td>2</td><td>accountsID</td><td>int(6)</td><td></td><td></td><td>Yes</td><td>None</td><td></td><td></td></tr> <tr> <td>3</td><td>addressLine1</td><td>varchar(1000)</td><td>latin1_swedish_ci</td><td></td><td>No</td><td>None</td><td></td><td></td></tr> <tr> <td>4</td><td>addressLine2</td><td>varchar(255)</td><td>latin1_swedish_ci</td><td></td><td>No</td><td>None</td><td></td><td></td></tr> <tr> <td>5</td><td>addressLine3</td><td>varchar(255)</td><td>latin1_swedish_ci</td><td></td><td>No</td><td>None</td><td></td><td></td></tr> <tr> <td>6</td><td>country</td><td>varchar(255)</td><td>latin1_swedish_ci</td><td></td><td>No</td><td>None</td><td></td><td></td></tr> <tr> <td>7</td><td>createdAt</td><td>datetime</td><td></td><td></td><td>No</td><td>CURRENT_TIMESTAMP</td><td></td><td></td></tr> </tbody> </table> <p>The page was tested under normal conditions – allowing users to add addresses correctly.</p> <p>The page was tested under erroneous conditions – checking if it was possible to add addresses if they did not supply all the required fields.</p>	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	1	id	int(6)			No	None		AUTO_INCREMENT	2	accountsID	int(6)			Yes	None			3	addressLine1	varchar(1000)	latin1_swedish_ci		No	None			4	addressLine2	varchar(255)	latin1_swedish_ci		No	None			5	addressLine3	varchar(255)	latin1_swedish_ci		No	None			6	country	varchar(255)	latin1_swedish_ci		No	None			7	createdAt	datetime			No	CURRENT_TIMESTAMP			
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra																																																																			
1	id	int(6)			No	None		AUTO_INCREMENT																																																																			
2	accountsID	int(6)			Yes	None																																																																					
3	addressLine1	varchar(1000)	latin1_swedish_ci		No	None																																																																					
4	addressLine2	varchar(255)	latin1_swedish_ci		No	None																																																																					
5	addressLine3	varchar(255)	latin1_swedish_ci		No	None																																																																					
6	country	varchar(255)	latin1_swedish_ci		No	None																																																																					
7	createdAt	datetime			No	CURRENT_TIMESTAMP																																																																					
2.2		<p>The following sql query was ran on the database</p> <pre>ALTER TABLE address ADD COLUMN `isActive` bit(1) NOT NULL DEFAULT b'1' AFTER country</pre> <p>New address table structure</p>	Pass																																																																								

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	int(6)			No	None		AUTO_INCREMENT
2	accountsID 	int(6)			Yes	None		
3	addressLine1	varchar(1000)	latin1_swedish_ci		No	None		
4	addressLine2	varchar(255)	latin1_swedish_ci		No	None		
5	addressLine3	varchar(255)	latin1_swedish_ci		No	None		
6	country	varchar(255)	latin1_swedish_ci		No	None		
7	isActive	bit(1)			No	'1'		
8	createdAt	datetime			No	CURRENT_TIMESTAMP		

Display only the addresses which have a 1 as value in the isActive field

Previous sql query to view all the addresses

```
//Retrieves all the addresses submitted by a particular user
$sql = "SELECT id, addressLine1, addressLine2, addressLine3, country
       FROM address
      WHERE accountsID = $accountsID";
$result = mysqli_query($conn,$sql);
$resultCheck = mysqli_num_rows($result);
```

Updated sql query –

```
//Retrieves all the addresses submitted by a particular user
$sql = "SELECT id, addressLine1, addressLine2, addressLine3, country
       FROM address
      WHERE accountsID = $accountsID AND isActive = 1";
$result = mysqli_query($conn,$sql);
$resultCheck = mysqli_num_rows($result);
```

Previous sql query to delete an address from the address table

```
$sql = "DELETE FROM address
        WHERE id = $id;";
mysqli_query($conn,$sql);
echo $sql;
die();
```

		<p>New sql query updates the current row by setting isActive for that row to 0</p> <pre>\$sql = "UPDATE address SET isActive = 0 WHERE id = \$id;"; mysqli_query(\$conn,\$sql);</pre> <p>Please refer to 2.2Address.mp4</p> <p>NB: The isActive for these addresses were set back to 1, so that the orders can be shipped to correct locations and so that future testing can be done on the orders using these addresses.</p>		
3.0	Search Page	<p>The search page should query the search in the books title, books author and books genre fields in the database and display the search results on the page.</p>	<p>Please refer to 3.0Search.mp4</p> <p>The search works perfectly fine. The sql query runs properly across the title, author and genre fields in the books table of the database. The sql query also uses LIKE command and matches the keywords in the search with values in the database. For example it matches the strings 'Dan' or 'Brown' with the author Dan Brown and displays all the books that match the criteria.</p>	Pass
4.0	List a new book page	<p>The List a new book page should allow creators to list a new book successfully. Any errors encountered – Invalid Characters in the ISBN, an ISBN not 13 characters long, a repeated ISBN, a wrong file format, a file too big will have appropriate error messages displayed.</p>	<p>Please refer to 4.0ListNewBook.mp4</p> <p>The Invalid Characters in ISBN, incorrect length of ISBN, repeated ISBN, wrong file format, large file size are validated and the correct error messages are displayed. The resulting book is not added to the database. If none of these errors exist, then the book is added to the database and the book can be seen in the search query.</p>	Pass
5.0	List an existing book	<p>Creators should be able to list books which already exist on the website. The Search on</p>	<p>Please refer to 5.0ListExistingBook.mp4</p>	Pass

		<p>the List an existing book should work as well.</p> <p>Errors should be handled with, and the book should not be uploaded</p>	<p>When creators list a book which they have not listed before, it creates a new record in the listings table with onSale value 1</p> <p>When creators list a book which they have already listed before, the onSale field in the previous record is set to 0 and the onSale value of the inserted record is set to 1.</p> <p>Errors such as non-existent ISBN in the book table or invalid characters in ISBN or an ISBN not 13 characters long or missing fields are met with appropriate error messages and the book is not uploaded to the database.</p>	
6.0	View Book Page	<p>Test that the correct details about the book are shown on the view book page. Test if books listed by different sellers can be added to the cart from the view book page.</p>	<p>Please refer to 6.0ViewBook.mp4</p> <p>In the video testing, the correct details about each book are displayed on the view book page. Moreover, when the seller Chew Barka decided to list the books – A Christmas Carol by Charles Dickens and Shelter in Place by Nora Roberts and the pages were refreshed, their listings showed up on the view Book Page.</p> <p>Please refer to 6.0.1ViewBook.mp4</p> <p>The second part of the test involved a seller re-listing a book at a different price. This new price needs to be displayed on the view book page.</p> <p>Seller Santa Paws re-price his book Inferno to a price of \$2.00.</p> <p>At timestamp 55 seconds the page was refreshed with the keyboard shortcut Ctrl+R, and the price information was updated.</p>	Pass
7.0	Cart Page	<p>Test the cart page for all the items and the quantities added and the cart subtotal and the remove from cart button</p>	<p>Please refer to 7.0 MyCart.mp4</p> <p>The item was added to the cart but when the quantity was set to 11, the MyCart page did not calculate the subtotal of the cart correctly. The code that produced the error is shown below</p>	Fail

		<pre><?php \$counter = -1; \$orderTotal = 0; foreach(\$_SESSION['mycart'] as \$subArray){ \$counter++; //Used to calculate the total of the order \$orderTotal += \$subArray['price']; } <tr> <td><img class="w-25" src="../bookImages/<?php echo \$subArray["imageName"]; ?>"></td> <td><?php echo \$subArray["title"] ; ?></td> <td><?php echo \$subArray["price"] ; ?></td> <td><?php echo \$subArray["quantity"] ; ?></td> <td><a href="includes/removefromcartScript.php?index=<?php echo \$counter?>">Remove from cart</td> </tr> <?php } \$_SESSION['mycartinfo']['orderTotal'] = \$orderTotal; ?></pre>	
		The line containing the error is highlighted. The orderTotal variable calculates the cart total by adding all the prices only. It does not take into account quantity. This is why the order total is displayed incorrectly.	
7.1		<p>The amendments were made to the code below. The price of each book was multiplied by the quantity of that book bought.</p> <pre><?php \$counter = -1; \$orderTotal = 0; foreach(\$_SESSION['mycart'] as \$subArray){ \$counter++; //Used to calculate the total of the order \$orderTotal += (\$subArray['price'] * \$subArray['quantity']); } <tr> <td><img class="w-25" src="../bookImages/<?php echo \$subArray["imageName"]; ?>"></td> <td><?php echo \$subArray["title"] ; ?></td> <td><?php echo \$subArray["price"] ; ?></td> <td><?php echo \$subArray["quantity"] ; ?></td> <td><a href="includes/removefromcartScript.php?index=<?php echo \$counter?>">Remove from cart</td> </tr> <?php } \$_SESSION['mycartinfo']['orderTotal'] = \$orderTotal; ?></pre> <p>Please refer to 7.1ViewBook.mp4</p>	Pass

			As the video shows, items once added to the cart will show the correct order total as the quantity is taken into account. In the video, several items were added to the cart and several removed, and the state of the cart and the order total was observed to change correctly.	
8.0	Checkout page	<p>Test the checkout page for all the items and the quantities added to the cart, the subtotal, the address, select address to ship to.</p>	<p>Please refer to 8.0Checkout.mp4</p> <p>In the video testing, 3 books of Romeo and Juliet, 6 books of Angels and Demons and 6 books of Harry Potter and the Philosopher's Stone were added to the cart. Their prices and quantities were both displayed correctly on the checkout page.</p> <p>When the user attempted to checkout without selecting an address, an error message was displayed. When the address was selected, the error message was removed.</p> <p>When the address was selected and the user successfully checked out, the user was taken to a thank you page and the data was inserted into the orders table.</p> <p>The following sql query run below shows that the last order inserted had ordersid 19</p>  <pre> 1 SELECT * FROM `orders` ORDER BY `createdAt` DESC </pre> <pre> 1 SELECT isbn, title, author, firstName, lastName, quantity, price, addressLine1, addressLine2, address 2 FROM orders 3 LEFT JOIN address ON address.id = orders.addressID 4 LEFT JOIN orderitems ON orderitems.ordersID = orders.id 5 LEFT JOIN listings ON listings.id = orderitems.listingsID 6 LEFT JOIN books ON books.isbn = listings.booksID 7 LEFT JOIN accounts ON accounts.id = listings.sellerID 8 WHERE orders.id = 19 </pre>	Pass

		<p>The result yielded shows that the correct books were ordered from the correct sellers and were shipped to the correct address.</p> <table border="1"> <thead> <tr> <th>isbn</th><th>title</th><th>author</th><th>firstName</th><th>lastName</th><th>quantity</th><th>price</th><th>addressLine1</th><th>addressLine2</th></tr> </thead> <tbody> <tr> <td>9331441109874</td><td>Romeo and Juliet</td><td>William Shakespeare</td><td>Mary</td><td>Puppins</td><td>3</td><td>5.99</td><td>123</td><td>NY</td></tr> <tr style="background-color: #cccccc;"> <td>9831431109874</td><td>Angels and Demons</td><td>Dan Brown</td><td>Sherlock</td><td>Bones</td><td>6</td><td>5.99</td><td>123</td><td>NY</td></tr> <tr> <td>9731131509874</td><td>Harry Potter and the Philosopher's Stone</td><td>J.K. Rowling</td><td>Mary</td><td>Puppins</td><td>6</td><td>5.6</td><td>123</td><td>NY</td></tr> </tbody> </table>	isbn	title	author	firstName	lastName	quantity	price	addressLine1	addressLine2	9331441109874	Romeo and Juliet	William Shakespeare	Mary	Puppins	3	5.99	123	NY	9831431109874	Angels and Demons	Dan Brown	Sherlock	Bones	6	5.99	123	NY	9731131509874	Harry Potter and the Philosopher's Stone	J.K. Rowling	Mary	Puppins	6	5.6	123	NY	
isbn	title	author	firstName	lastName	quantity	price	addressLine1	addressLine2																															
9331441109874	Romeo and Juliet	William Shakespeare	Mary	Puppins	3	5.99	123	NY																															
9831431109874	Angels and Demons	Dan Brown	Sherlock	Bones	6	5.99	123	NY																															
9731131509874	Harry Potter and the Philosopher's Stone	J.K. Rowling	Mary	Puppins	6	5.6	123	NY																															
9.0	My Orders	Tests if the orders placed by the user are displayed correctly in the my orders page.	In the previous test, an order for 3 books of Romeo and Juliet, 6 books of Angels and Demons and 6 books of Harry Potter and the Philosopher's Stone was made. The result is checked on the My Orders Page	Pass																																			

		<p>ORDER PLACED 2019-03-17 14:45:52</p> <p>ORDER TOTAL 87.51</p> <p>Romeo and Juliet Sold by: Mary Puppins</p> <p>Romeo and Juliet is a tragedy written by William Shakespeare early in his career about two young star-crossed lovers whose deaths ultimately reconcile their feuding</p> <p>Price: \$5.99</p> <p>Quantity: 3</p> <p>Angels and Demons Sold by: Sherlock Bones</p> <p>Harvard symbologist Robert Langdon works with a nuclear physicist to solve a murder and prevent a terrorist act against the Vatican</p> <p>Price: \$5.99</p> <p>Quantity: 6</p> <p>Harry Potter and the Philosopher's Stone Sold by: Mary Puppins</p> <p>Harry Potter and the Philosopher's Stone is a fantasy novel written by British author J. K. Rowling. The first novel in the Harry Potter series and Rowling's debut novel.</p> <p>Price: \$5.6</p> <p>Quantity: 6</p>	
<p>The details for each book placed in the order, along with the price of the book, the quantity, the order total, the date and time of the order and the seller name are displayed correctly.</p> <p>Please refer to 9.0MyOrders.mp4</p>			

			In the video testing, 10 books of Three Tragedies and 4 books of Island of Glass were purchased. At 23 seconds, when the orders page is visited, all this information along with seller information, order total and date and time of order is displayed.	
10.0	Sign up page	Tests if the signup page functions properly. The signup page should not allow signup if any of the fields are empty, if the password and re-enter password do not match, invalid emails, or if an account with the email already exists. The signup page should successfully allow sign up of users and creators.	Please refer to 10.0Signup.mp4 The testing was a success as it prevented all the errors mentioned in the test description and returned correct error messages. In the video testing, Dodger Jones successfully registered as a creator and Kobe Doe successfully registered as a user.	Pass
11.0	My Book Listings – Creator page	Check if the books listed by the creator and their correct details are present on the book listings page for that creator. Check that if the price is updated, the price is updated on the book listings page. Check that sorting by sales and price ascending and descending sorts the results in the correct order.	Please refer to 11.0MyBookListings.mp4 When the price of the book was updated from \$7.99 to \$5.99, the price was no updated on the book listings page.	Fail

11. 1	My Book Listings – Creator Page	<p>The code that produced the error is shown below.</p> <pre>//append the orderbySql at the end to sort the results. \$sql = " SELECT isbn,SUM(quantity) as sales,author, title, description, genre, sellerID, price FROM books LEFT JOIN listings ON books.isbn = listings.booksID LEFT JOIN orderitems ON orderitems.listingsID = listings.id WHERE sellerID = \$sellerID GROUP by isbn ". " " . \$orderbySql; //echo \$sql; echo \$sql; \$result = mysqli_query(\$conn,\$sql); \$numRows = mysqli_num_rows(\$result);</pre> <p>The code does not check if the onSale flag is set to 1 in the listings table and therefore displays the first listing of that book – displaying a price of \$7.99</p> <table border="1" data-bbox="840 768 1917 861"> <thead> <tr> <th>ISBN</th><th>Title</th><th>Author</th><th>Price</th><th>Sales</th></tr> </thead> <tbody> <tr> <td>1234267891232</td><td>Harry Potter and the Chamber of Secrets</td><td>J.K. Rowling</td><td>\$7.99</td><td>0</td></tr> </tbody> </table> <p>The code is amended and the onSale is set to 1 in the sql query – and the correct price of \$5.99 is displayed.</p> <pre>//append the orderbySql at the end to sort the results. \$sql = " SELECT isbn,SUM(quantity) as sales,author, title, description, genre, sellerID, price FROM books LEFT JOIN listings ON books.isbn = listings.booksID LEFT JOIN orderitems ON orderitems.listingsID = listings.id WHERE sellerID = \$sellerID AND onSale = 1 GROUP by isbn ". " " . \$orderbySql;</pre> <table border="1" data-bbox="840 1274 1917 1367"> <thead> <tr> <th>ISBN</th><th>Title</th><th>Author</th><th>Price</th><th>Sales</th></tr> </thead> <tbody> <tr> <td>1234267891232</td><td>Harry Potter and the Chamber of Secrets</td><td>J.K. Rowling</td><td>\$5.99</td><td>0</td></tr> </tbody> </table>	ISBN	Title	Author	Price	Sales	1234267891232	Harry Potter and the Chamber of Secrets	J.K. Rowling	\$7.99	0	ISBN	Title	Author	Price	Sales	1234267891232	Harry Potter and the Chamber of Secrets	J.K. Rowling	\$5.99	0
ISBN	Title	Author	Price	Sales																		
1234267891232	Harry Potter and the Chamber of Secrets	J.K. Rowling	\$7.99	0																		
ISBN	Title	Author	Price	Sales																		
1234267891232	Harry Potter and the Chamber of Secrets	J.K. Rowling	\$5.99	0																		

11. 2	My Book Listings – Creator Page		Refer to 11.1MyBookListings.mp4 When Dodger Jones uploaded the book Deception Point for price of 39.99, it registered correctly on the My Book Listings Page. When Dodger Jones changed the price of Deception Point to \$5.99, the price was updated correctly. When Deception Point's price was updated to \$10.99 and \$13.99, the change in price was registered correctly on the My Book Listings Page.	Pass
12. 0	Sales on My Book Listings – Creator Page	Tests if the sales for each book by seller is registered correctly on the My Book Listings Page	Please refer to 12.0Sales.mp4 This test failed. When John Doe bought 13 books of A Christmas Carol from Mary Puppins, the number of books sold successfully incremented by 13. However, when price was changed to 15.99, the sales was set back to 0. <pre>//append the orderbySql at the end to sort the results. \$sql = " SELECT isbn,SUM(quantity) as sales,author, title, description, genre, sellerID, price FROM books LEFT JOIN listings ON books.isbn = listings.booksID LEFT JOIN orderitems ON orderitems.listingsID = listings.id WHERE sellerID = \$sellerID AND onSale = 1 GROUP by isbn ". " " . \$orderbySql;</pre> In test 11.1, the following code 'onSale = 1' added only displays quantity sold for the latest listing. This is why the sales were set back to 0 as no new books were bought at the new listing.	Fail
12. 1	-		It would not make sense for sellers to have their sales updated when that book's price was raised or lowered. It would give the sellers a misleading impression of the number of books they sold at the higher price. So a new column called Selling? was added	Pass

```
<table class="table table-hover">
<thead>
  <tr>
    <th scope="col">ISBN</th>
    <th scope="col">Title</th>
    <th scope="col">Author</th>
    <th scope="col">Price</th>
    <th scope="col">Sales</th>
    <th scope="col">Selling?</th>
  </tr>
</thead>
<tbody>
<?php
//Loop the html code below for every row returned by the sql query
if($numRows > 0){
  while ($row = mysqli_fetch_assoc($result)){
    ?>

    <tr>
      <td scope="row"><?php echo $row['isbn'];?></td>
      <td><?php echo $row['title'];?></td>
      <td><?php echo $row['author'];?></td>
      <td>$<?php echo $row['price']; ?></td>
      <td><?php
        if($row['sales'] == 0){
          echo 0;
        }else{
          echo $row['sales'];
        }
      ?></td>
      <td><?php
        if($row['onSale'] == 0){
          echo "<p class='text-danger'>No</p>";
        }else{
          echo "<p class='text-success'>Yes</p>";
        }
      ?></td>
    </tr>
  <?php
    }
  ?>
```

The onSale = 1 was removed again and the onSale column was selected and it was ordered by the orderBySql variable and the isbn number.

```
//If the orderby parameters are set in the url, then make a string called orderbysql
$orderbySql = "";
$sorted=False;
if(isset($_GET['sortby']) AND isset($_GET['order'])){
    $sortby = $_GET['sortby'];
    $order = $_GET['order'];
    $orderbySql = "$sortby $order";
    $sorted=True;
}

//append the orderbySql at the end to sort the results.
$sql = "
    SELECT isbn,SUM(quantity) as sales,author, title, description, genre, sellerID, price, onSale
    FROM books
    LEFT JOIN listings ON books.isbn = listings.booksID
    LEFT JOIN orderitems ON orderitems.listingsID = listings.id
    WHERE sellerID = $sellerID
    GROUP BY listings.id
    ORDER BY $orderbySql isbn";
```

ISBN	Title	Author	Price	Sales	Selling?
9388809701326	A Christmas Carol	Charles Dickens	\$5.99	53	No
9388809701326	A Christmas Carol	Charles Dickens	\$15.99	0	Yes

This gives an accurate representation of the quantity of books sold at each price.

Please refer to 12.1Sales.mp4

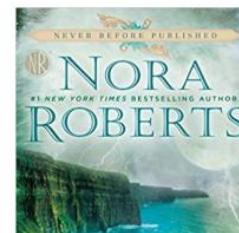
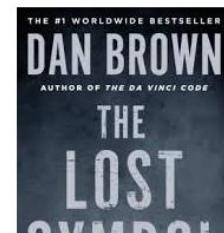
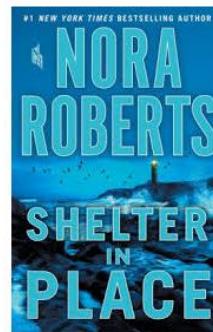
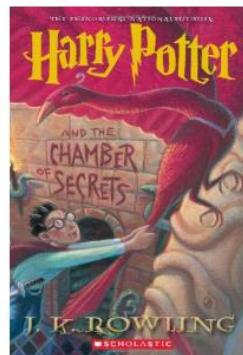
When the price was updated to \$5.99, a new row in the table was added and the “selling?” for the old row was set to No. The selling? for the new row was set to Yes. Moreover, the sales for the previous price were not deleted and the sales for the new row were set to 0. When 15 of those books were bought at the new price, the sales for the new listing were updated.

13.0	Trending page – users account	Tests if the trending page shows the most bought books.	The sql code below produced the following result.	
------	-------------------------------	---	---	--

		<pre> 1 SELECT isbn,SUM(quantity) as sales, author, title, SUBSTRING(description,1,100) as description, genre, imageName 2 FROM orderitems 3 LEFT JOIN listings ON orderitems.listingsID= listings.id 4 LEFT JOIN books ON books.isbn = listings.booksID 5 WHERE onSale = 1 6 GROUP by isbn 7 ORDER BY sales DESC 8 LIMIT 10 </pre> <table border="1"> <thead> <tr> <th>isbn</th><th>sales</th><th>author</th><th>title</th><th>description</th><th>genre</th><th>imageName</th></tr> </thead> <tbody> <tr><td>9038441119874</td><td>375</td><td>Nora Roberts</td><td>Shelter in Place</td><td>Shelter in Place 8 minutes - a mass shooting - a p...</td><td>Romance</td><td>5c7bb97daf6bb.jpg</td></tr> <tr><td>9831130109874</td><td>122</td><td>Dan Brown</td><td>The Lost Symbol</td><td>In this stunning follow-up to the global phenomeno...</td><td>Fiction</td><td>5c7bc012a5699.jpg</td></tr> <tr><td>9138432109874</td><td>74</td><td>Nora Roberts</td><td>Island of Glass</td><td>The final Guardians Trilogy novel from the #1 New ...</td><td>Romance</td><td>5c7bb7bf9be95.jpg</td></tr> <tr><td>92384411109874</td><td>67</td><td>William Shakespeare</td><td>The Winter's Tale</td><td>The Winter's Tale is a play by William Shakespear...</td><td>Literature</td><td>5c7bbdc7e40c8.jpg</td></tr> <tr><td>2131131109874</td><td>35</td><td>Dan Brown</td><td>Inferno</td><td>Inferno is a 2013 mystery thriller novel by Americ...</td><td>Fiction</td><td>5c7bbf31ef877.jpg</td></tr> <tr><td>92184411109874</td><td>33</td><td>William Shakespeare</td><td>A Midsummer Night's Dream</td><td>A Midsummer Night's Dream is a comedy written by ...</td><td>Literature</td><td>5c7bbbed5d2b0.jpeg</td></tr> <tr><td>8831131109874</td><td>26</td><td>Dan Brown</td><td>Deception Point</td><td>Deception Point is Dan Brown's third novel, rel...</td><td>Fiction</td><td>5c7bbeee13033.jpg</td></tr> <tr><td>90384411109874</td><td>20</td><td>Nora Roberts</td><td>Secret Star</td><td>This is the final story in the Stars of Mithra tri...</td><td>Romance</td><td>5c7bb8ce5cae9.jpg</td></tr> <tr><td>9876543210987</td><td>17</td><td>John Green</td><td>Looking for Alaska</td><td>Looking for Alaska is John Green's first novel, pu...</td><td>Romance</td><td>5c7ba772d8a69.jpg</td></tr> <tr><td>93314411109874</td><td>15</td><td>William Shakespeare</td><td>Romeo and Juliet</td><td>Romeo and Juliet is a tragedy written by William S...</td><td>Literature</td><td>5c7bbd83aff37.jpg</td></tr> </tbody> </table> <p>This shows that the most bought book was Nora Roberts' book – Shelter in Place.</p> <p>Please refer to 13.0Trending.mp4</p> <p>When 400 copies of Harry Potter and the Chamber of Secrets were bought, the top book in the trending changed to Harry Potter and the Chamber of Secrets as the previously top sold book – Shelter in Place only sold 375 copies.</p>	isbn	sales	author	title	description	genre	imageName	9038441119874	375	Nora Roberts	Shelter in Place	Shelter in Place 8 minutes - a mass shooting - a p...	Romance	5c7bb97daf6bb.jpg	9831130109874	122	Dan Brown	The Lost Symbol	In this stunning follow-up to the global phenomeno...	Fiction	5c7bc012a5699.jpg	9138432109874	74	Nora Roberts	Island of Glass	The final Guardians Trilogy novel from the #1 New ...	Romance	5c7bb7bf9be95.jpg	92384411109874	67	William Shakespeare	The Winter's Tale	The Winter's Tale is a play by William Shakespear...	Literature	5c7bbdc7e40c8.jpg	2131131109874	35	Dan Brown	Inferno	Inferno is a 2013 mystery thriller novel by Americ...	Fiction	5c7bbf31ef877.jpg	92184411109874	33	William Shakespeare	A Midsummer Night's Dream	A Midsummer Night's Dream is a comedy written by ...	Literature	5c7bbbed5d2b0.jpeg	8831131109874	26	Dan Brown	Deception Point	Deception Point is Dan Brown's third novel, rel...	Fiction	5c7bbeee13033.jpg	90384411109874	20	Nora Roberts	Secret Star	This is the final story in the Stars of Mithra tri...	Romance	5c7bb8ce5cae9.jpg	9876543210987	17	John Green	Looking for Alaska	Looking for Alaska is John Green's first novel, pu...	Romance	5c7ba772d8a69.jpg	93314411109874	15	William Shakespeare	Romeo and Juliet	Romeo and Juliet is a tragedy written by William S...	Literature	5c7bbd83aff37.jpg	
isbn	sales	author	title	description	genre	imageName																																																																										
9038441119874	375	Nora Roberts	Shelter in Place	Shelter in Place 8 minutes - a mass shooting - a p...	Romance	5c7bb97daf6bb.jpg																																																																										
9831130109874	122	Dan Brown	The Lost Symbol	In this stunning follow-up to the global phenomeno...	Fiction	5c7bc012a5699.jpg																																																																										
9138432109874	74	Nora Roberts	Island of Glass	The final Guardians Trilogy novel from the #1 New ...	Romance	5c7bb7bf9be95.jpg																																																																										
92384411109874	67	William Shakespeare	The Winter's Tale	The Winter's Tale is a play by William Shakespear...	Literature	5c7bbdc7e40c8.jpg																																																																										
2131131109874	35	Dan Brown	Inferno	Inferno is a 2013 mystery thriller novel by Americ...	Fiction	5c7bbf31ef877.jpg																																																																										
92184411109874	33	William Shakespeare	A Midsummer Night's Dream	A Midsummer Night's Dream is a comedy written by ...	Literature	5c7bbbed5d2b0.jpeg																																																																										
8831131109874	26	Dan Brown	Deception Point	Deception Point is Dan Brown's third novel, rel...	Fiction	5c7bbeee13033.jpg																																																																										
90384411109874	20	Nora Roberts	Secret Star	This is the final story in the Stars of Mithra tri...	Romance	5c7bb8ce5cae9.jpg																																																																										
9876543210987	17	John Green	Looking for Alaska	Looking for Alaska is John Green's first novel, pu...	Romance	5c7ba772d8a69.jpg																																																																										
93314411109874	15	William Shakespeare	Romeo and Juliet	Romeo and Juliet is a tragedy written by William S...	Literature	5c7bbd83aff37.jpg																																																																										
14.0	Index.php	Checks if the trending books on index.php are the correct books. Checks that the books are recommended in the correct order based on which books were bought the most by the user.	In test 13.0, 400 copies of Harry Potter and the Chamber of Secrets were bought.	Pass																																																																												

Trending Today

[Click to see more](#)



On the index page, this is correctly shown in the top left corner of the trending today section.

On the index page, in the recommended for you section, books by J.K. Rowling are displayed because the user recently bought 400 copies of Harry Potter and the Chamber

		<p>of Secrets.</p> <p>Recommended for you</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Harry Potter and the Chamber of Secrets J.K. Rowling \$6.99 View Book</p> </div> <div style="text-align: center;"> <p>Harry Potter and the Philosopher's Stone J.K. Rowling \$6.99 View Book</p> </div> <div style="text-align: center;"> <p>A Midsummer Night's Dream PLAY BY WILLIAM SHAKESPEARE \$6.99 View Book</p> </div> <div style="text-align: center;"> <p>The Winter's Tale William Shakespeare \$6.99 View Book</p> </div> <div style="text-align: center;"> <p>NORA ROBERTS SECRET STAR \$6.99 View Book</p> </div> <div style="text-align: center;"> <p>NORA ROBERTS SHELTER IN PLACE \$6.99 View Book</p> </div> </div> <div style="display: flex; justify-content: space-around; margin-top: 20px;"> <div style="text-align: center;"> <p>DAN BROWN</p> </div> <div style="text-align: center;"> <p>CHARLES DICKENS' BLEAK HOUSE</p> </div> </div>	
		<p>Please refer to 14.0.0Index.mp4</p> <p>In the video testing, 230 copies of Charles Dickens' Bleak House were bought and the ranking of Charles Dickens' Books was observed to increase. After that, 230 Dan Brown books were bought, and then Dan Brown book recommendations were observed to move up the rankings.</p> <p>Please refer to 14.0.1Index.mp4</p> <p>If user is not logged on or the user is logged on but has not bought any books yet, then the recommended section is not displayed on the index page. When the user logged in as Kobe Doe, no recommended books were displayed as Kobe did not buy any books. But when Jane Doe was logged on, recommended books were displayed as Jane Doe had made orders previously.</p> <p>In future tests, some books were bought from Kobe Doe's account, so that is why in future video testings, books may be shown under the recommended section for Kobe Doe's account.</p>	

EVALUATION

HOW WELL THE OBJECTIVES WERE MET

I have listed the objective headings below.

DATABASE

1. The database will have to store information on
 - a. Customers and Sellers' first name, last name, email, password and address to which shipped.
The database stores information on user's first name, last name, email address and date of birth, password and hashed password in the accounts table. The database table address stores information on the user's address.
 - b. Every customers' order details, and the details on the address to which it is shipped
Db table orders store all this information
 - c. Each item in the order including details about the book, details on the seller, and the quantity of the item bought.
Db table orderdetails stores this information.
 - d. Each book will need details on the book's title, author, description, genre, ISBN and an image.
Db table books stores this information
 - e. Each book may be listed at different prices by different sellers. The database will have to store information on these prices.
Db table listings stores this information.

All conditions under the database section are met

OVERALL WEBSITE FUNCTIONALITY

1. An option for users to sign up, log in and log out of their accounts of the website.
The login and signup pages allow the user to sign up or log in to the website. Moreover, the logout button on the header allows creators/users to logout easily. Moreover, the person can log in or sign up as a user or creator
2. Two account types: User's Account, Creator's Account
 - a. The user's account allows users to
 - i. Add addresses
The add address page allows the user/creator to add and view addresses.
 - ii. Add, books to a shopping cart and place the order
The view book page allows users to add books to the cart. On this page the user can see the book details and alter the quantity he wants to buy.
The my cart page allows the users to see the books in their cart. The checkout page allows users to make an order of all the books in the cart.

- iii. Check their history of orders.
The my orders page allows users to check their order history.
- b. The Creator's Account allows creators to
 - i. View existing book listings
The book listings page allows creators to view their existing book listings.
 - ii. List books on the website
The two page: list new book and list existing book allow creators to list new or existing books on the website.
 - iii. View sales
The book listings page allows creators to check the number of sales they have made per listing.

All conditions under the Overall Website Functionality Section are met

HOME PAGE

- 1. A list showing some of the most bought books on the website
The trending section consisting of 4 books displays the 4 most bought books on the website.
- 2. If the user is logged in, some book recommendations should be made
 - a. The book recommendations will be made based on the user's order history.
The user is made book recommendations under the recommended for you section on the home page.

All conditions under the Home Page Section are met

ADDRESS-PAGE

- 1. Gives the user option to add addresses
The add address page allows users to do this
- 2. Allows the user to view all existing addresses.
The view address page allows users view all existing addresses.

All conditions under the Address Page Section are met

MY CART PAGE

- 1. A list of all the books the user added to his cart
 - a. The cart should display the book image, the title, the price and the quantity
The my cart page displays the book image, title, price and quantity in a table format.
- 2. An option to remove the book from the cart.

The remove from cart link allows users to remove books from their cart.

All conditions under the my cart page section are met

MY ORDERS PAGE

1. A display of all the orders the user has made including details on
 - a. The book name, author name, quantity, price and order total.

The my orders page successfully displays all this information

All conditions under the my orders section are met

CHECKOUT PAGE

1. Displays the books in the cart including information on
 - a. Book Name, Book Price and Quantity

The checkout page displays this information in a tabular format.
2. A select address option allows users to select an address to ship the book to.

The select address option allows users to select from a range of addresses.
3. A button to confirm purchase.

The checkout page displays the order again and the checkout button allows users to place the order.

All conditions under the checkout page section are met

TRENDING PAGE

1. Displays the most sold books on the website
 - a. The book image, book title, author and description should be displayed for each book on the trending page

The trending page displays the 12/13 most bought books on the website. All book information is displayed in neat cards.

All conditions under the trending page section are met

VIEW BOOK PAGE

1. The view book page is a page that displays information about each book.

- a. This includes the book image, book title, book author, book description and price of the book as listed by all sellers.

The view book page shows all this information.

2. Allows users to select the quantity of books to buy

The user can alter the quantity of books bought from the option next to the seller names.

3. Contains an add-to-cart button

- a. The button successfully adds the book, along with the right quantity and at the right price to the cart.

The add to cart button allows users to add the book to the cart.

All conditions under the view book page section are met

HEADER

1. A navigation bar spanning across the width of the page containing details on
 - a. The business name (BookSchmo), link to home page, my cart page, my orders page.
 - b. A search bar
 - c. It should also allow the user to log out

The BookSchmo Header presents all these links

2. The creator's header should span across the width of the page and contains links to
 - a. Links to the creator's book listings, the list a book page, the user site.
 - b. It should also allow the creator to log out

The creator's header is different from the user's header and presents all these links as well.

All conditions under the header section are met

CREATOR'S HOME PAGE

1. The creator's home page allows all creators to
 - a. Access the book listings page
 - b. Access the list a book page.
 - c. Access the user site.

The creator's home page displays all these links in a card-like format

All conditions under the creator's home page section are met

CREATOR'S BOOK LISTINGS PAGE

1. Allows creators to view all the books that they have listed.
 - a. This includes information on the book isbn, title, author, price it was listed at, the number of books sold at that price
The book listings page displays all this information in a tabular format.
2. Allows sellers to sort the books by price or sales
The sorting can be done at the top of the page.

All conditions under the creator's book listings section are met

LIST BOOKS PAGE

1. Allows sellers to list books
 - a. A form gives sellers option to list the book by ISBN, book title, author, genre, description, price and an option to upload the book image
 - b. If the book already exists on the website, the seller is asked to enter the book ISBN and price to list the book
As explained in the overall functionality objectives, the list a new book page and list an existing book page allow the creator to do this.

All conditions under the list books page section are met

SEARCH BAR AND SEARCH PAGE

1. The search bar dynamically renders a page containing books which link with keywords in the search bar.
The user's search page is a core part of the website. The search bar in the website header allows users to search for books.

All conditions under the search bar and search page section are met

POTENTIAL IMPROVEMENTS

As with any project, there are many improvements that can be made. Although the project met all the requirements as set by my client, it must be noted that this project was only a prototype. The improvements to be made to the project, **if it were to be implemented as a real-life project** are listed below.

- Introduce a backend warehouse that stores books and produce a management system that links to the website database so that the book can be shipped once the order can be made.
- Add a payment option allowing users to pay by card before they place the order.
- Place a cap on the maximum number of books that can be bought by one user. The cap was not placed on the existing project to allow testing of features like the trending and book recommendations page by allowing the user to buy an extraordinarily large number of books to see visible changes on these pages to confirm that they function properly
- Store the user search history and user cart in the database. Dynamically render the cart from the database instead of the session. This way the cart will load even after the user logs out, ensuring that he does not lose the items he added to the cart. The search history, once stored in the database could be used for future search suggestions or even book recommendations
- Improve the book recommendations. Despite the complexity of the existing book recommendations algorithm, it is safe to say that book recommendation based on user search history, and based on user order history will prove as better recommendations.
Incorporating a certain level of **deep learning/machine learning** on book clicks and orders based on recommendations would improve the recommendation algorithm over time.
- Introduce seller quantity numbers as seller will have a limited stock of book. Introduce seller location details and the places where they can ship to.
- Introduce shipping costs. This will need to be linked to shipping companies shipping the books.
- Add more complex search sql queries as the current query simply looks for the search input in the fields in the book database. A more complex search page query would involve breaking up the words separated by spaces and searching for each individual word in the fields in the database. Concatenate all the rows received from the individual database searches and present these on the search page. However, simple words such as "a", "an", or "the" should not be searched in the database as they will produce undesirable books in the search results. To prevent this, a lookup array of these words would have to be made and each word in the search input would have to be checked against this array. This may introduce scope for PHP object-oriented programming. Allow sorting of results on the search page.
- An improvement to website security would go a long way in improving the reliability of the website. The website could protect against sql injection attacks on HTML forms by splitting the SQL result set into pages and making superusers using PostgreSQL.
- Another added level of security could include making the session more secure. Although the website was hosted locally, encrypting it with HTTPS when it is hosted on a www server would make the website more secure against eavesdropping attempts. Finally, if payment options are to be added to the website, any page redirects to MasterCard or Visa or payment website would need to be secured.

FEEDBACK FROM MY CLIENT

Once I had finished the project, I presented the project to my client at QuantM Ltd. They were extremely pleased by the progress I had made on my project and looking at my progress on the website, they even asked me to work for them on an internship/training this summer.

With regards to the project, some of the points they made are presented below.

- The project had gone above and over their expectations. When I had met my client, I had little knowledge about back-end programming. Although they had asked me to create a buyer's and seller's account, they had only expected me to be able to create the buyer's account at the maximum. It was quite pleasing for them to see that I had been able to produce both types of accounts. Moreover, they had been extremely pleased that I added the my orders page to the users account and my book listings page to the seller's account
- The project could adapt additional security measures before being implemented in the real world
- The HCI was excellent. However, the website could benefit from professional designers and front-end web-developers whose support they would provide me at the summer internship.
- The back-end code was fully functional and working and met all the objective that I had sent them (the analysis section of this report).
- The cross-table parametrized SQL relational tables and queries, they said, were quite above school-level expectations. However, to implement the project in real-life, multiple levels of complexity would need to be added.
- Finally, they noted that the project would be an extremely useful prototype by adapting the project to be hosted from PC to their company's linux server.

THE LEARNING CURVE

This A-Level NEA Project meant a great deal to me. I spent over 400 hours, learning, researching, programming, and producing the report. It truly was a great task to learn front-end and back-end web-development. I appreciate the fact that in industry, multiple programmers are involved in building a project. Building the project alone was very enjoyable, difficult, intriguing and tiring at the same time.

Some of the things I learnt from this project –

- Programming in PHP. Using SQL. Getting better at HTML and CSS.
- Producing a documented report
- How to use Microsoft Word to its full limits
- How to manage time when producing a large scale project
- How to use the internet effectively for search up doubts and queries

Thank you for taking the time to read this report. I hope you liked it.