



WEB DEVELOPMENT GROUP ASSINGMENT TECHNOLOGY PARK MALAYSIA

082021-DMV

Web Development

Group Members:

Maxim Hew Mun Sum - TP060705

Teo Chen Ming - TP060668

Teo Kai Yii - TP058618

INTAKE CODE: UCDF2005(1) ICT(SE)

HAND OUT DATE: 26 August 2021

HAND IN DATE: 14 November 2021

LECTURER:

MR DANIEL MAGO VISTRO

MR MOHAMAD FIRDAUS CHE ABDUL RANI

Table Of Content

1.0 Gantt Chart	3
2.0 Introduction (Introduction to company and website, objectives, reflection on website)	4
3.0 System design	7
3.1 Wireframes	7
3.2 Navigational Structure.....	25
3.3 Flowchart.....	26
3.3.1 Admin Flowchart.....	26
3.3.2 Customer Flowchart	28
3.3.3 Guest Flowchart.....	33
3.4 Entity Relationship Diagram (ERD)	36
4.0 Implementation	37
4.1 Server-Side Scripting (PHP)	37
4.1.1 Create - HTML Form, PHP	37
4.1.2 Read - HTML Form, PHP	43
4.1.3 Update - HTML Form, PHP	64
4.1.4 Delete - HTML Form, PHP	68
4.1.5 Login - HTML Form, PHP	70
4.1.6 Signup – HTML Form, PHP.....	72
4.1.7 Self-created CSS and JavaScript	74
5.0 Main Section (User Guidance & Sample Screen).....	88
5.1 Guest Accessible Page	88
5.2 Customer Accessible Page	95
5.3 Admin Accessible Page.....	109
6.0 Conclusion	117
7.0 Workload Matrix.....	118
8.0 References.....	119

1.0 Gantt Chart

Activities	\	Week	3	4	5	6	7	8	9	10	11	12	13	14
Briefing and Grouping			Red											
Identify topic and concepts of website			Yellow	Yellow										
Create wireframe + erd				Yellow										
Create Login page + register, and other html layouts					Green	Green	Green							
Upgrade html pages with PHP and JavaScript						Blue	Blue	Blue	Blue	Blue				
Preparing Final Report			Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue				
Final Report Presentation											Dark Blue	Dark Blue		
Submission of Final Report													Purple	

Figure 1: Gantt Chart

2.0 Introduction (Introduction to company and website, objectives, reflection on website)

The objective of creating the Story Glory Online Bookstore is to provide users with the best experience when visiting the website as if they were in the physical bookstore, but online! It provides convenience to readers who still need physical books during the country lockdown. The website achieves this objective by providing users with an amazing experience while surfing the website, which consists of user-friendly interfaces. Purchasing a book on the website is not a hassle as the purchased design follows the three clicks rule. Users who are interested in purchasing books from the website can simply select the desired book, add it to the cart, and proceed to complete their payment. Their books will arrive within a week once the order has been placed. The website provides users with easy-to-understand instructions so that users can explore desired books with ease by using a search and sort function on the website. Story Glory utilizes the style design of a modern and minimalistic feel to grab the attention of readers from all ages, from kids to senior citizens who can't get enough of reading a good physical book as our target audience.

Guests, customers, and admin can view the list of available books on the homepage of the website. However, to purchase and view more details of a book, guests must register as a customer and sign into their account. The admin dashboard will be available to admin accounts to manage books on the website and orders that the customer has done.

In the designed website, each web page will have an appropriate header and footer. As for the header, there are plenty of user-friendly features, such as a well-sorted navigation bar that consists of sections such as Home, Category, My Wishlist, Shopping Cart, My Purchase, and My Profile for customers. An extra Admin Dashboard button will only be visible when an admin has logged in to the website. While for the footer of the website consist of an "About Us" section that displays to the users the value, product, and services of Story Glory. If guests and customers require any assistance, help, and support can be obtained by clicking on the social media icons that currently consists of Story Glory's Facebook and Instagram platform.

Throughout the module, our group has gained a lot of knowledge regarding Hypertext Preprocessor (PHP) coding, basic HyperText Markup Language (HTML) coding, Cascading Style Sheets (CSS) coding, as well as JavaScript (JS) coding. We learned that both front-end and back-end design of the website is very important in order to maintain a good website design as well as its functionalities. PHP coding is mainly used for database connection as well as to insert data, retrieve data, and update data to and from the database; while HTML coding is to

display the information all in the webpage as well as accept input data from the end-users; CSS coding is crucial to make sure that the webpage looks attractive and allows end-users to navigate through the website without getting bored; JavaScript is also another very important element in the website as it can edit the display contents as well as the design of it according to the values collected or by end-users interacting with the webpage. While doing the assignment, our group has encountered a lot of obstacles along the way, most of them being tough ones as one mistake or error can delay the assignment's progress. Our group has learned a lot throughout the error handling process, most of the answers are gotten through online research, and more research must be done to fully understand the code. Besides, our group also learned that bootstrap could be used to make coding and to design the website easier, but our group has decided not to use it. Our group plan to type everything from scratch and learn them along the way.

Besides coding skills, each member of our group also learned that teamwork is very important for a project to work out perfectly. Since every member is assigned a different role to create the code for it, each member all have different type of codes, also the variable names as well as navigation bars. If everything is not consistent, then the final result will be a badly designed webpage and unhappy end-users. Everyone in the group should work together to combine all the PHP, HTML, CSS, and JavaScript codes.

Moreover, communication skills are also very crucial throughout the assignment. If there is a problem with group member's communication, there might be duplicate codes and pages. Hence, good communication is needed to increase the group's efficiency.

Furthermore, leadership is also a very important skill to have and learn throughout the project. At least one person must be the leader of the group in order to delegate tasks to all the group members, and also to make sure that everyone can complete their part accordingly on time.

As for the design choice for the website, the design chosen for Story Glory is rather simple but nice enough to suit the taste of people of all ages. All the web pages are displayed in a way that is easy to navigate and read. The font styles are easy to read and are white in color, which suits the website that has a darker shade background. The navigation bar is located on top of every page. This means that users will be greeted with a navigational bar every time the page is accessed. Below, the navigation bar consists of the main body of the page, which is the content. The reason why the content is right after the navigation bar is so that users can

immediately view the contents that are needed. Finally, below the homepage will include the footer part, which consists of some information about the company, as well as hyperlinks to our social media. This information is located at the bottom of the page so users can just scroll down to find details about the company. All these elements are easy to access on the webpage and are very important, especially for new users, as new users can easily find what is needed on the website.

3.0 System design

3.1 Wireframes

The wireframe shows a registration form titled 'Create An Account'. It includes fields for Name, Email, Phone Number, Shipping Address, Password, and Confirm Password. There are also checkboxes for newsletter subscription and terms and conditions, along with three buttons at the bottom: 'Create Account' (green), 'Reset' (red), and 'Cancel' (grey).

Create An Account		
Name:	<input type="text" value="Enter a cool username"/>	
Email:	<input type="text" value="Enter Email"/>	
Phone Number:	<input type="text" value="Enter Phone Number"/>	
Shipping Address:	<input type="text" value="Enter Shipping Address"/>	
Password:	<input type="text" value="Enter Password"/>	
Confirm Passowrd:	<input type="text" value="Confirm Your Password"/>	
<input type="checkbox"/> Subscribe to newsletter		
<input type="checkbox"/> I Agree to Terms and Conditions		
<input type="button" value="Create Account"/>	<input type="button" value="Reset"/>	<input type="button" value="Cancel"/>

Figure 2:Register Account Wireframe

Before users can sign in, users need to register for an account on the website. Users can do this by entering their name, email, phone number, shipping address, password, and confirmation password in the respective input box. The formatting for the input that the user enters needs to be the correct formatting. For example, the email needs to be “example@email.com”. The user can choose whether to subscribe to the newsletter by clicking on the checkbox. However, the terms and conditions must be agreed before creating an account. If the user does not check the agree to terms and conditions box, users will cannot create an account. The user can press on the create account once all their credentials are inserted. Users can also click on the reset button to erase all their credentials.

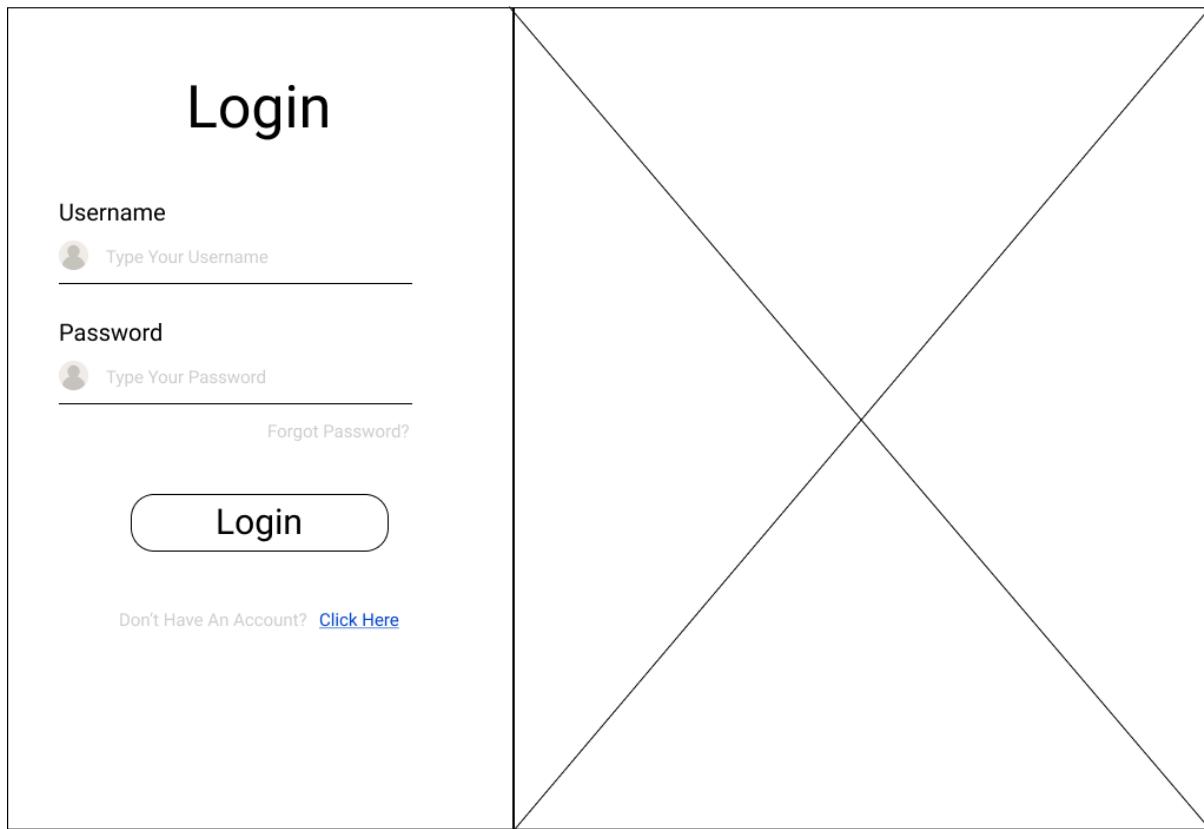


Figure 3: Login Wireframe

Users or admin can log in to their accounts on the login page. The user or admin needs to enter their username and password into the respective input box. If the username and password match an account in the database, the user will be logged in to their account and will be directed to the homepage of the website. If the user does not have an account, users can create one by clicking on the “Click Here” hyperlink.

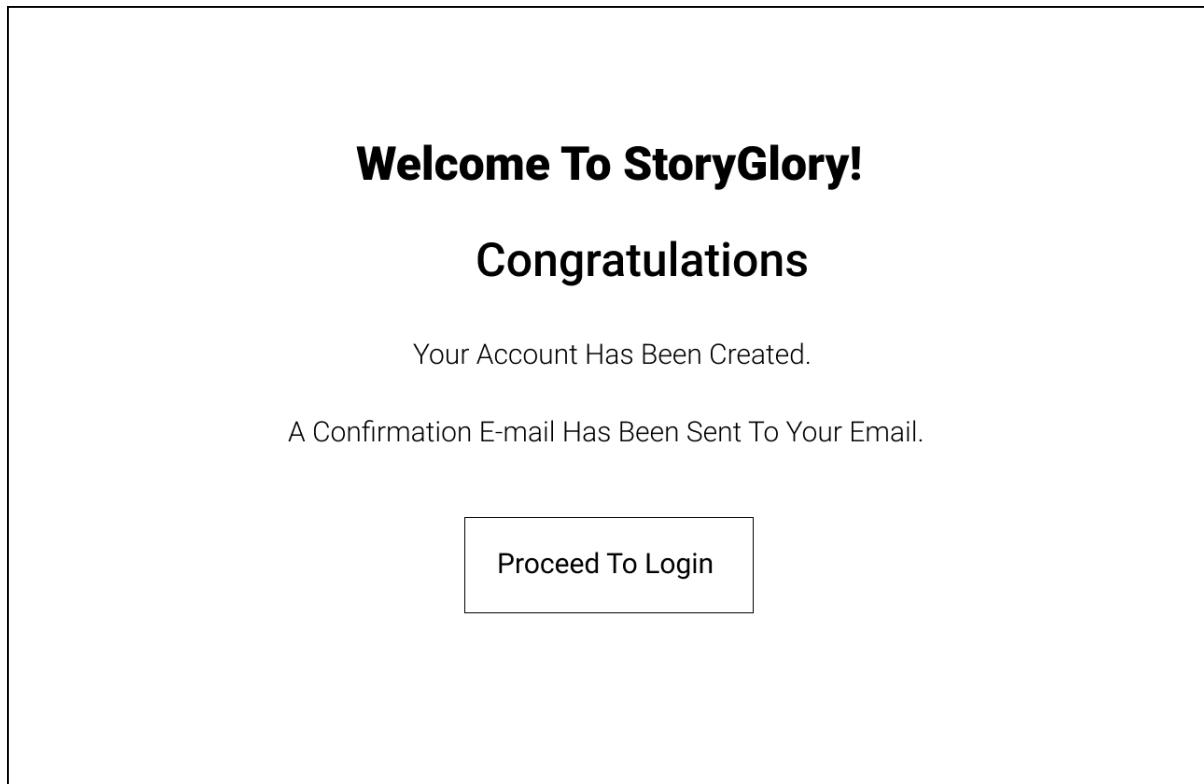


Figure 4: Create Account Successful Wireframe

Users will be greeted with the congratulations message prompting them that an account has been successfully created. Users can then log in to their newly created account by clicking on the proceed to the login button.

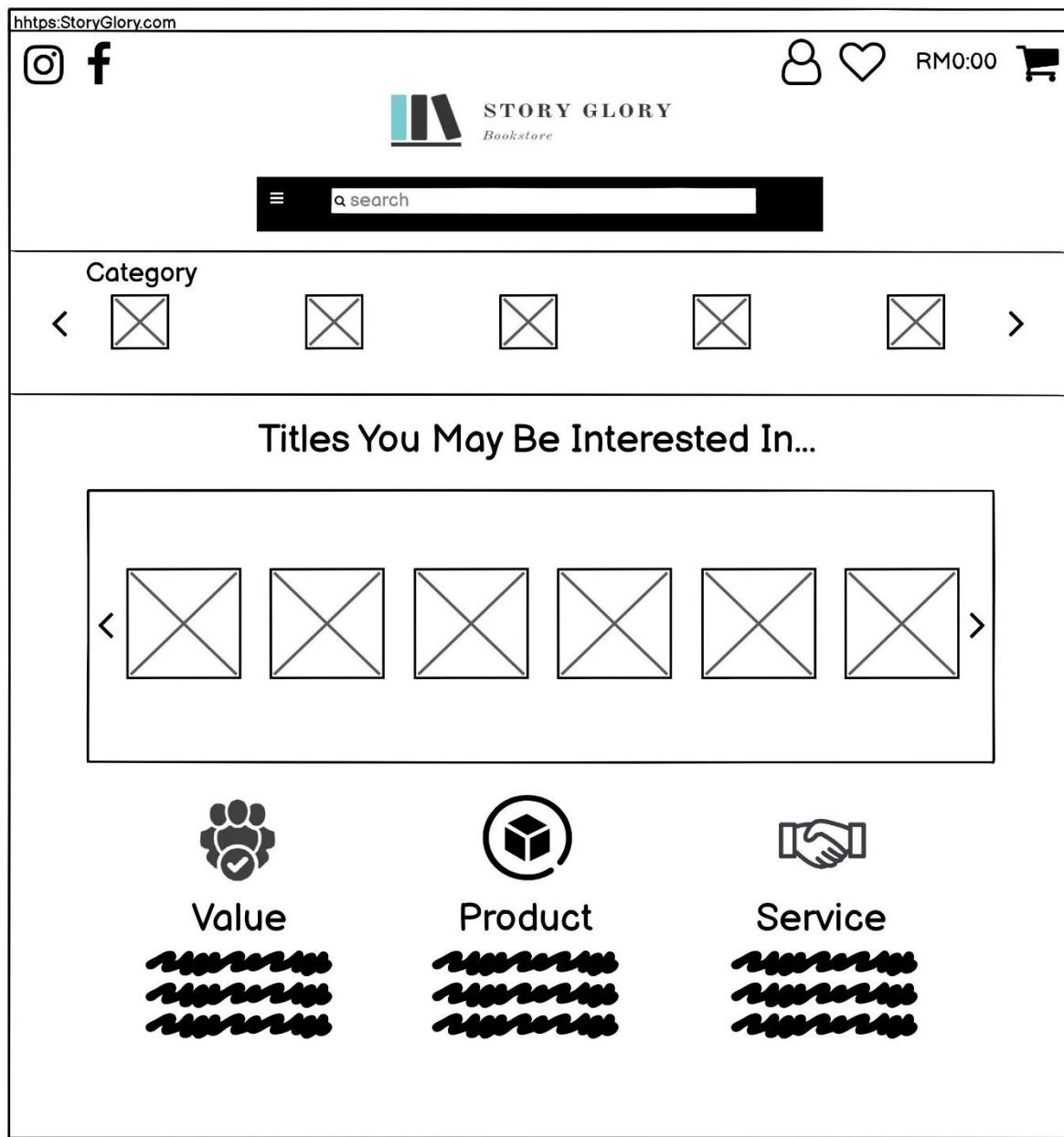


Figure 5: Homepage Wireframe

If users or admin has signed into their account successfully, the homepage of the website will display. At the top left, the users can click on the Instagram or Facebook icons to visit Story Glory's Instagram and Facebook page. The user icon at the right top corner is the user profile of the user, the heart icon is the Wishlist of the user, and the cart icon will direct the user to the shopping cart page, which includes the items that the user has added into their shopping cart. A search bar will be available for the users to search for a specific book that the user is interested in along with categories at the bottom such as motivational, kids, and academic. Titles that users may be interested in will be shown below. The value, product, and service of the company will also be shown along with an interactive icon.

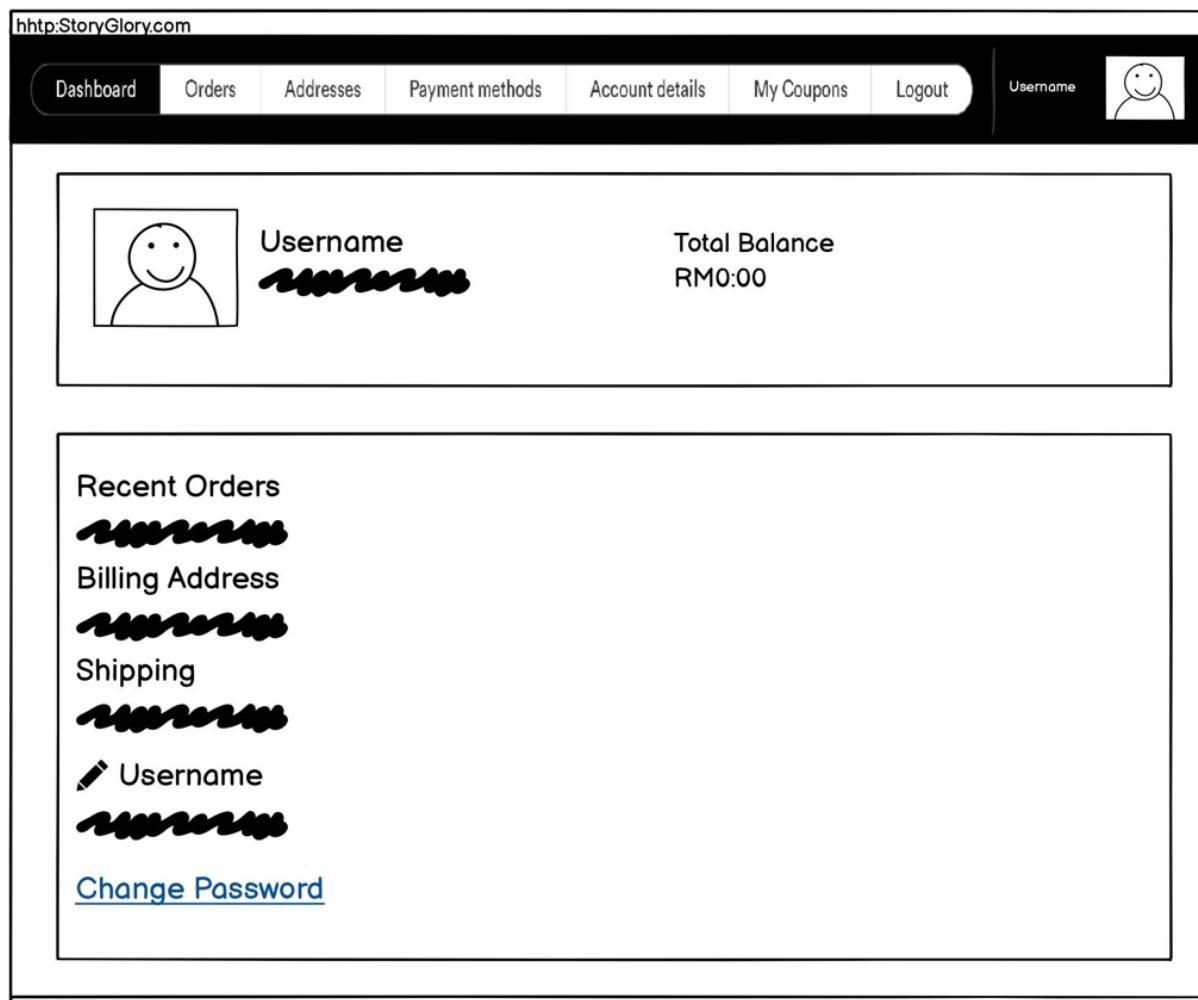


Figure 6: User Profile Wireframe

The user profile page will display all the relevant information about the user, such as their username, total balance, recent orders, billing address, shipping, and allow them to change their password. Users can edit their user profiles by accessing the user profile page. The user's profile picture can be seen on this user profile page as well.

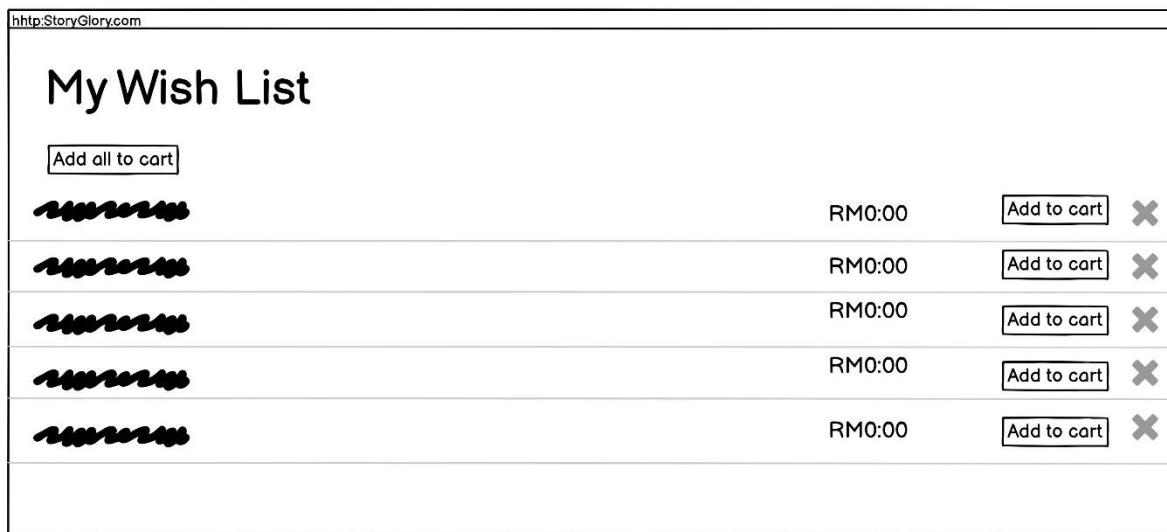


Figure 7: Wishlist Wireframe

The Wishlist page is where users can view their Wishlist items. If a user were to find a book that piques their interest, the book can be added to the user's Wishlist and purchase it some other time. On the Wishlist page, the user can view the book's price and add the Wishlist books to their cart. If users want to remove a book from their Wishlist, users can remove the book by clicking on the "x" icon in their Wishlist.

https://StoryGlory.com		
1.Shopping Cart	2.Shipping Details	3.Payment Options
My Shopping Cart		
Item	Unit Price	Summary
	RM0:00	Subtotal RM0:00
ISBN: Quantity: 1 Edit Remove		Shipping Free
	RM0:00	Taxes RM0:00
ISBN: Quantity: 1 Edit Remove		Total RM0:00
<input type="button" value="Next"/> <input type="button" value="Cancel"/>		

Figure 8: Shopping Cart Wireframe

The shopping cart page is a page where all the items that the user has added to their shopping cart will display. The details of the item, such as the item name, the ISBN of the book, quantity, unit price, subtotal, total price as well as the cover page of the book, will be displayed. The subtotal will be calculated from the quantity of the book and the unit price of the book, while the total price is the addition of all the subtotals along with the shipping fees. The next button will direct the users to the shipping detail page, while the cancel button will direct the users back to the homepage.

The wireframe shows a header with three tabs: 1.Shopping Cart, 2.Shipping Details (selected), and 3.Payment Options. Below the tabs is a section titled "Shipping Method". It contains four input fields: "Full Name", "Phone Number", "Address", and "Zip/Postal Code". To the right of these fields is a summary table:

Subtotal	RM0:00
Shipping	Free
Taxes	RM0:00
Total	RM0:00

Below the summary table are two radio button options for shipping: "Standard Shipping - RM5 Between 2 - 5 working days" (selected) and "Next Day Delivery - RM10 24 hours from checkout". At the bottom are "Next" and "Cancel" buttons.

Figure 9:Shipping Details Wireframe

The shipping details page will gather the user information for the shipment of the product. Users will need to input their full name, phone number, address, zip/postal code, city, and choose the type of shipment for the delivery of the product. If users choose standard shipping, it will cost them RM 5. However, if users were to choose a Next Day Delivery shipping, it would cost them RM10. Once all user details have been filled up, users can press next and proceed to the payment page.

The wireframe shows a payment method page with the following components:

- Header:** https://StoryGlory.com
- Navigation:** 1.Shopping Cart, 2.Shipping Details, 3.Payment Options
- Title:** Payment Method
- Credit Card Section:**
 - Method:** Credit Card (radio button selected)
 - Bank Options:** Maybank, CIMB Bank, UOB Bank, RHB Bank, Public Bank, Hong Leong Bank
 - Input Fields:** Card Number (0000 0000 0000 0000), MM/YY, CW, Card Holder Name
- Online Banking Section:**
 - Method:** Online Banking (radio button selected)
 - Bank Options:** Maybank, CIMB Bank, UOB Bank, RHB Bank, Public Bank, Hong Leong Bank
- Summary Table:**

Subtotal	RM0:00
Shipping	Free
Taxes	RM0:00
Total	RM0:00
- Buttons:** Next, Cancel

Figure 10:Payment Method Wireframe

The payment method page is a page where users can make their payment and choose the method of their payment. The users are required to input their credit card particulars and choose the type of payment. After the particulars and the payment method have been settled, the users can press next to proceed to checkout.

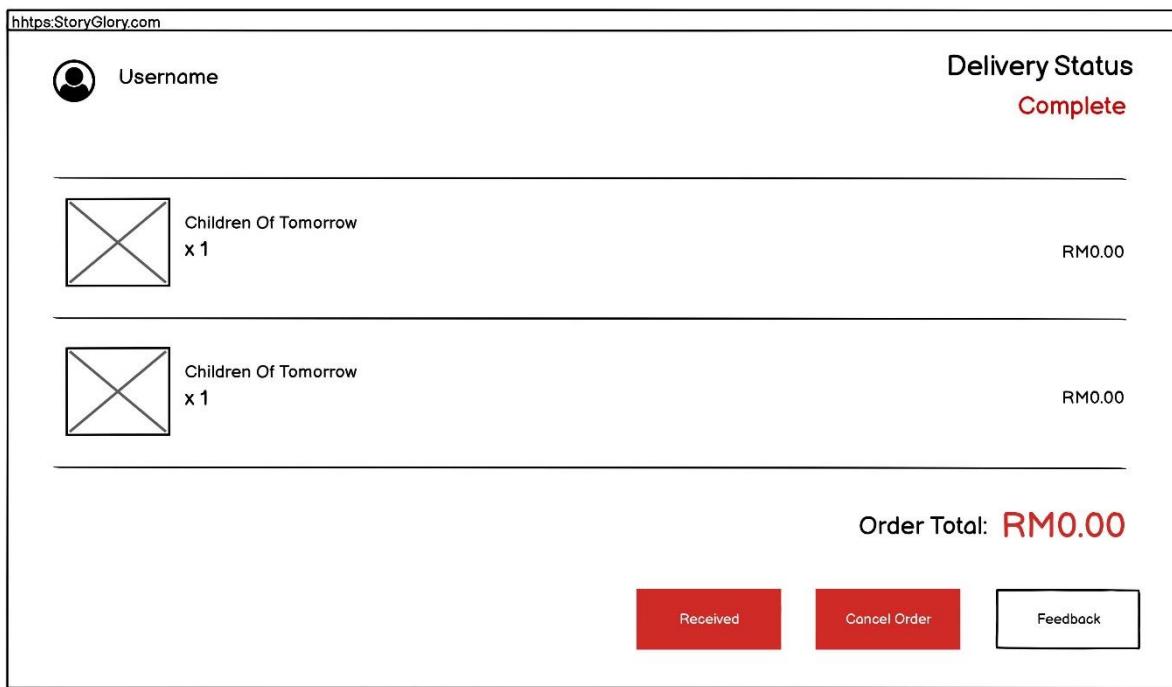
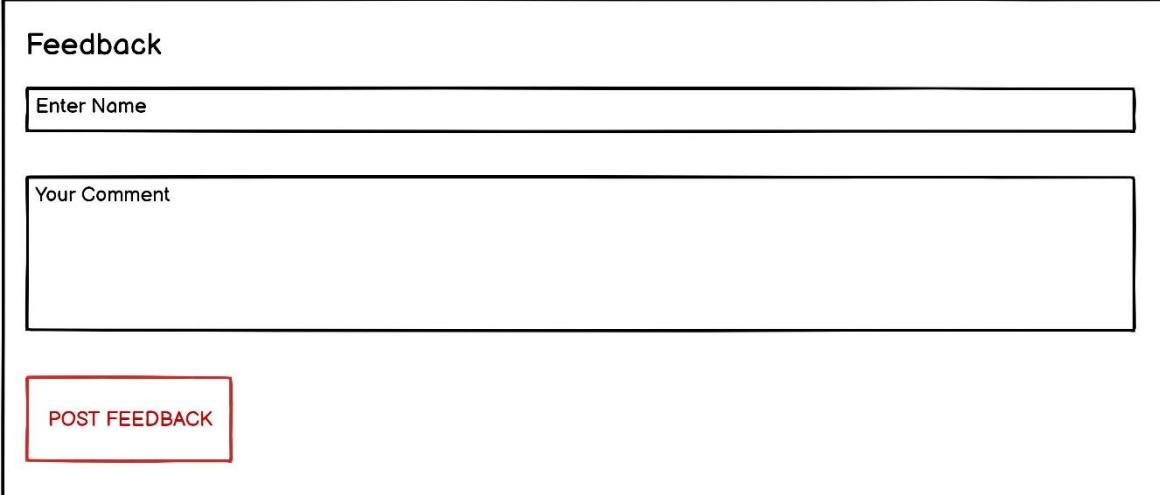


Figure 11: User Purchase Wireframe

The user purchase page will display after the user has made the purchase on the purchase method page. This page will show the purchases that the user has made and the status of delivery of their purchase. The price of each item, along with the quantity, will be shown. The total price after adding the type of shipment and all the subtotals will be displayed to the user as well. Users can click on the receive button to notify that user have received their item once the item has been shipped successfully to them. Users can cancel the order if the shipment has not been made or return the item if it is not up to the user's standard. Finally, users can provide their feedback about the service or their book purchase by clicking on the feedback button that will bring them to a feedback page where users can post their feedback on the website.



The wireframe for the feedback page is contained within a large rectangular frame. At the top left, the word "Feedback" is centered. Below it is a horizontal input field with a thin black border and the placeholder text "Enter Name". Underneath this is a larger vertical input field with a thin black border and the placeholder text "Your Comment". At the bottom left of the frame is a red-bordered rectangular button with the text "POST FEEDBACK" in white capital letters.

Figure 12:Feedback Wireframe

The feedback page is a page where users can give their feedback on the product that has been purchased. Users need to enter their name and the comment on the respective input boxes. Once the input has been made, the users need to click on the post feedback to complete their feedback and post it on the website.

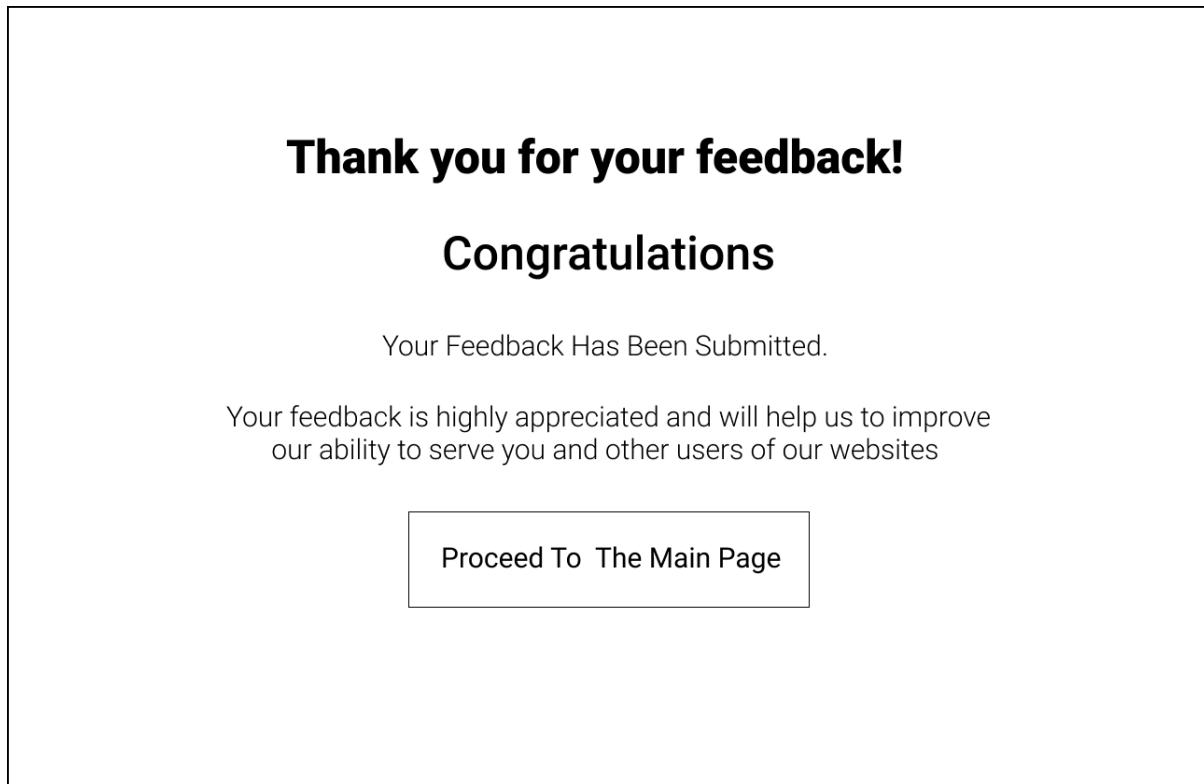


Figure 13:Successful Feedback Wireframe

The successful feedback page will display once the user's feedback has been accepted. Users will be prompted with the successful feedback interface and can click on the proceed to main page button to return to the homepage of the website.

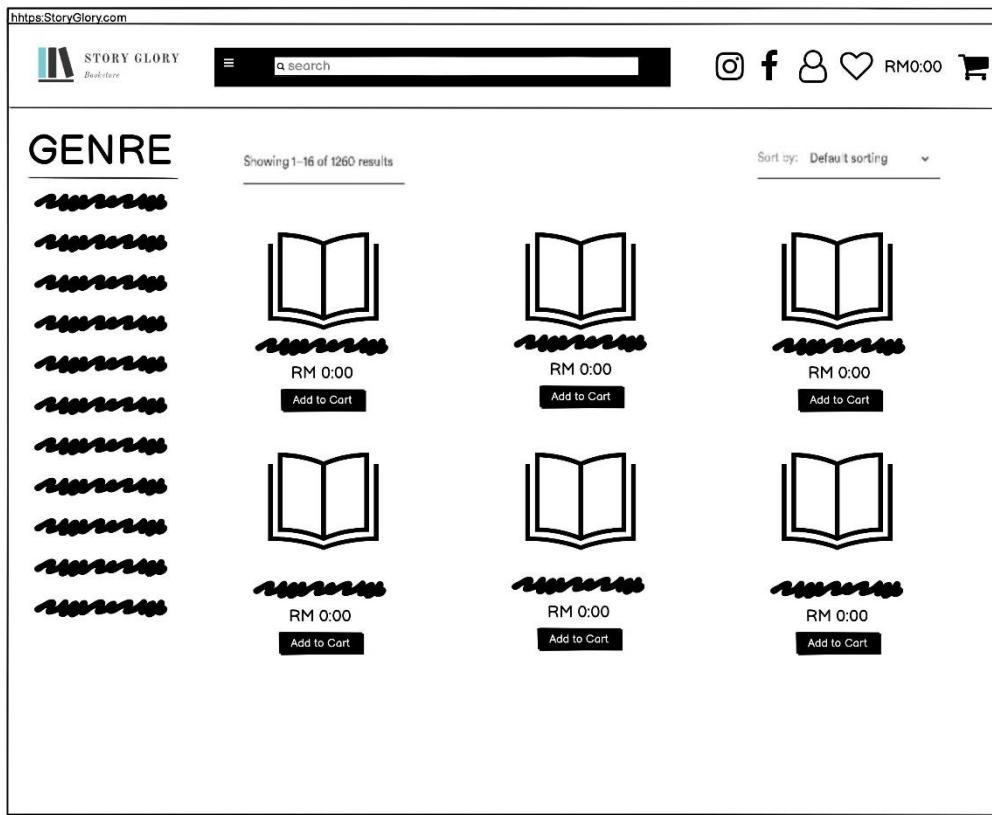


Figure 14: Category Wireframe

The category page will show the genre of books that the user is interested in. If the users were to pick kids' books in the category section, only kids' books would be displayed out to the users. The book cover, along with the price of the book, will be displayed. Users can add the book into their cart on the category page for purchase. Users can sort their search as well through the sorting icon at the top right of the display.

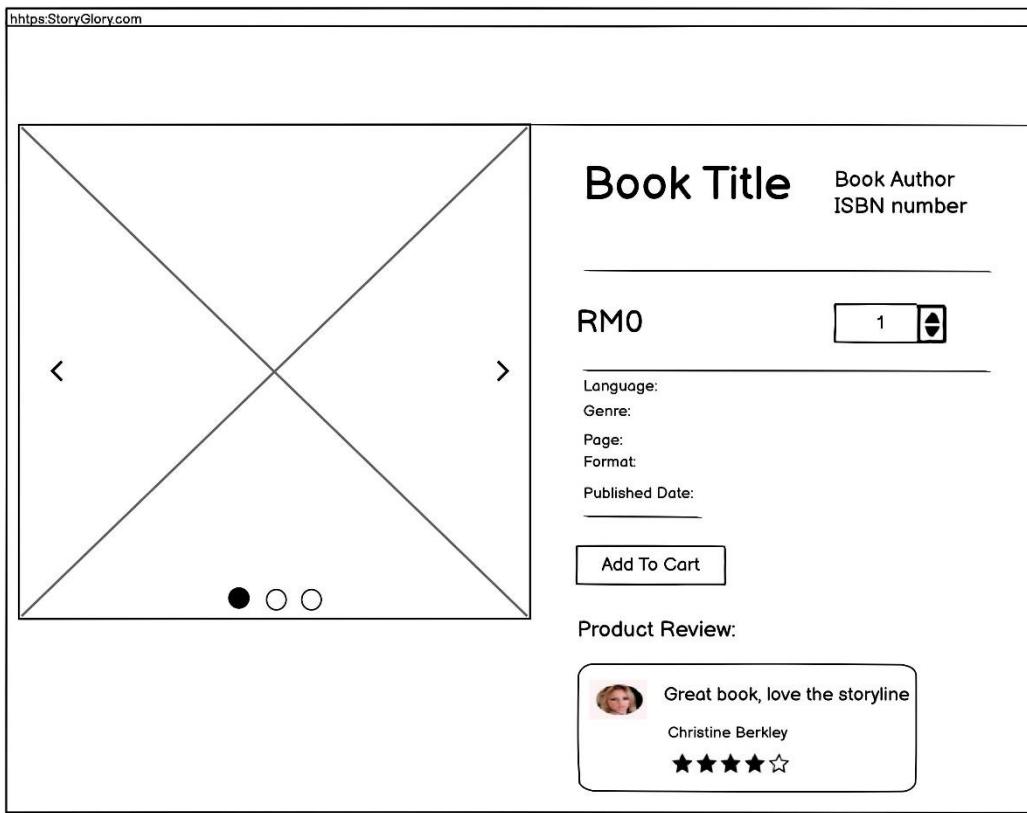


Figure 15: Product Page Wireframe

The product page will show the look of the book in a wider display and show the product reviews on that book. The price will be shown along with the book title. More information about that book can be seen below, such as the language of the book, book genre, number of pages, book format, and published date. Users can choose the number of books and add that book to their cart by clicking on the add to cart button.

The wireframe depicts a web application interface for an administrator. At the top, there is a header bar with standard browser controls (back, forward, search, etc.) and a URL field containing "http://AdminStoryGlory.com". To the right of the URL is a user profile icon labeled "Admin".

Below the header, two summary boxes are displayed: "Total Sales" with a bar chart icon and the value "21324", and "Total Income" with a hand holding a dollar sign icon and the value "21324".

The main content area is titled "Books". It includes a search bar labeled "Search ISBN" and three buttons: "New", "Delete", and "Update". Below these are six input fields for filtering book records: "ISBN", "Genre", "Price", "Pages", "Title", "Format", "Author", "Date Published", "Language", and "Quantity Available".

A section titled "Recent Customers" lists four customer entries, each represented by a redacted name and a phone number (e.g., "0123456789"). Next to each phone number is a "Update Delivery Status" dropdown menu.

Figure 16: Admin Homepage Wireframe

If the user happens to be an admin and has logged in to an admin account, the admin will have all the same functionalities as the customer with an additional admin homepage. Admin can access the admin homepage by clicking on the admin button. Here the admin can perform admin functionalities such as the search for a book by using the book ISBN, adding a new book by entering the book information such as the ISBN, price, title, author, language, genre, page, format, date published, and quantity available. Admin can delete books from the system as well by searching for the book and deleting it by clicking on the delete button icon. Admin can update book details as well by searching and updating specific information about the book. Admin can view the recent customers and view some basic customer information in the view recent customer tab. Lastly, the admin can track the progress of their sales by viewing the total sales at the top of their homepage as well as the total income generated.

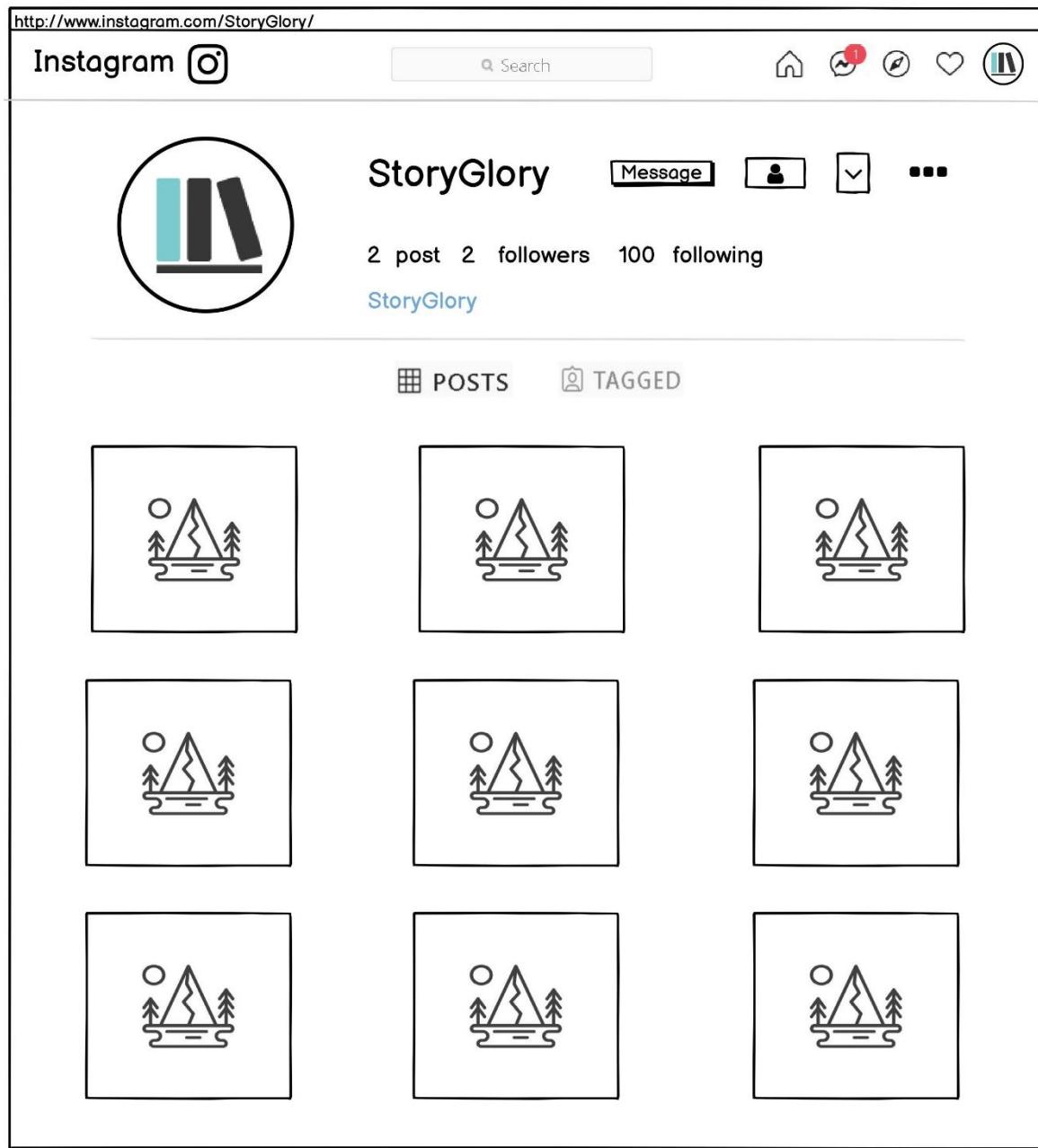


Figure 17:Instagram Page Wireframe

Users can visit StoryGlory's Instagram page to get assistance and keep up to date with StoryGlory's events.



Figure 18;Facebook Page Wireframe

Users can visit StoryGlory's Facebook page to get assistance and keep up to date with StoryGlory's events.

3.2 Navigational Structure

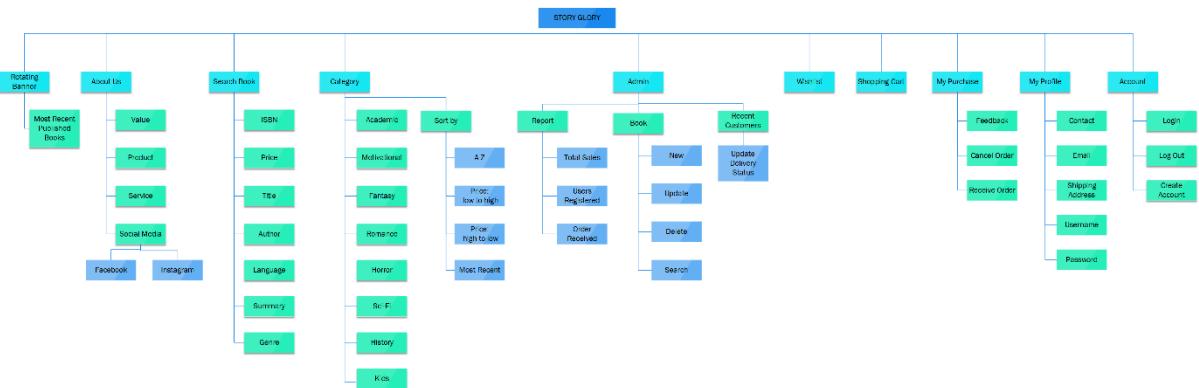


Figure 19: Navigational Structure of Story Glory

3.3 Flowchart

3.3.1 Admin Flowchart

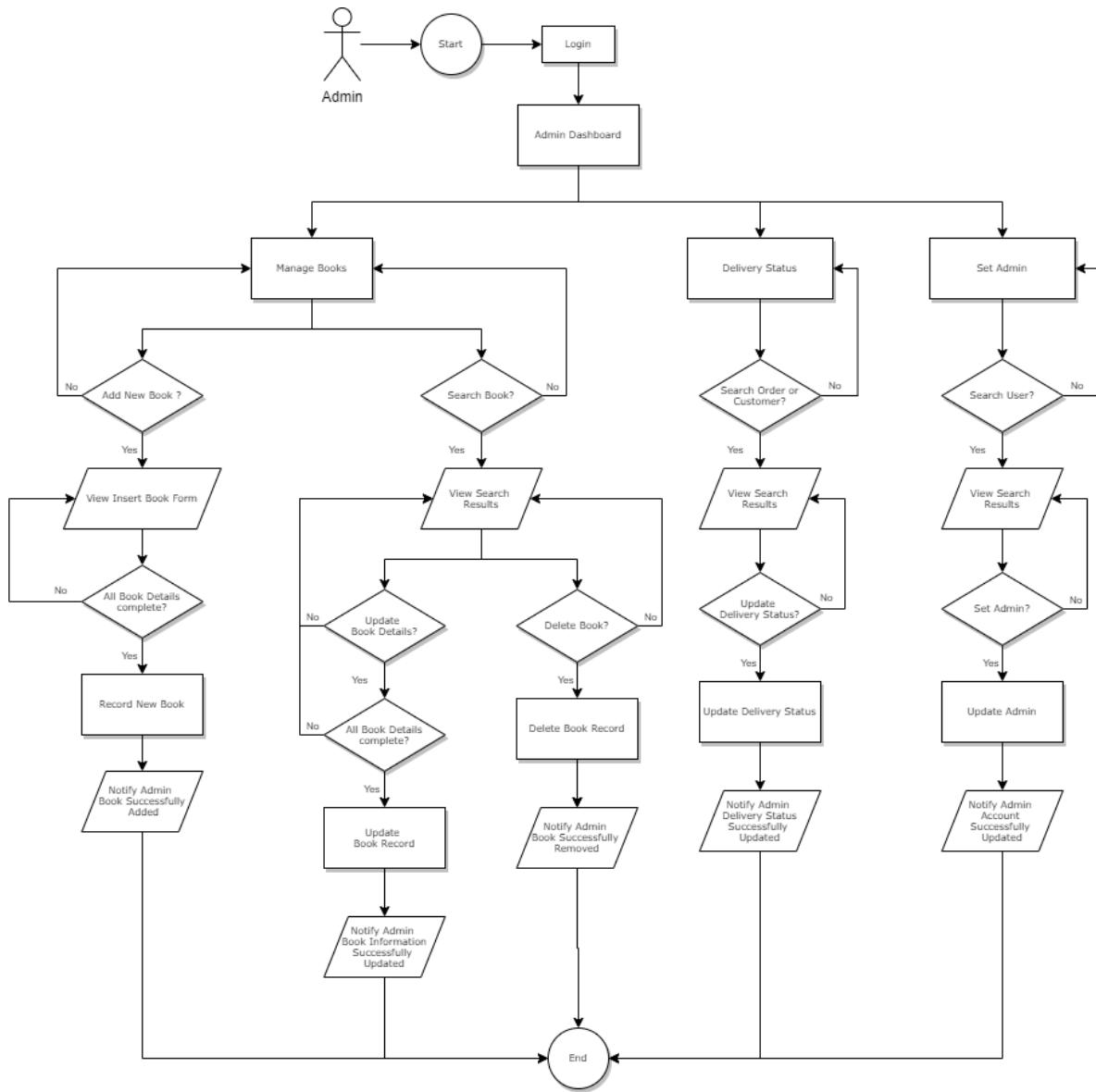


Figure 20: Admin flowchart

The admin functionalities include all the customer functionalities that will be described later. The only difference between admin and normal customer is that admins have more functionalities, including manage books, updating delivery status as well as setting a user as admin.

After the admin successfully logs in, the admin will be able to access the admin dashboard. Admin can manage books, update delivery status, and set admin in the admin dashboard.

Manage Books

Admin must click on the “Add New Book” button to view the insert book form. After filling in all the book details, the admin can press the “Add New Book” button again to submit the form. The book details submitted will be recorded immediately, and a message box will notify the admin that the book has been added. To update book details or delete a book, the admin must first search a book. To update book details, the admin can edit the book details in the search results and click on the “Update” button. The book details will be updated, and a notification will be sent to the admin. The book details will only be updated when the book details are completed. To delete a book, the admin can click on the “Delete” button, the book record will be removed from the database, and a message box will notify the admin the book is deleted.

Delivery Status

The update delivery status function in the admin dashboard requires an admin to search for the order by the Order_ID or Customer. In the search result, the admin can change the delivery status and click on the “Update” button to update the delivery status. A message box will notify the admin the delivery status has been successfully updated.

Set Admin

The set admin function in the admin dashboard allows the admin to set a user to become an admin. To set an admin account, the admin must search for the user and click on the “Set Admin” button. The user will become an admin and will be able to view the admin dashboard. A message box will notify the admin when the admin account has been updated.

3.3.2 Customer Flowchart

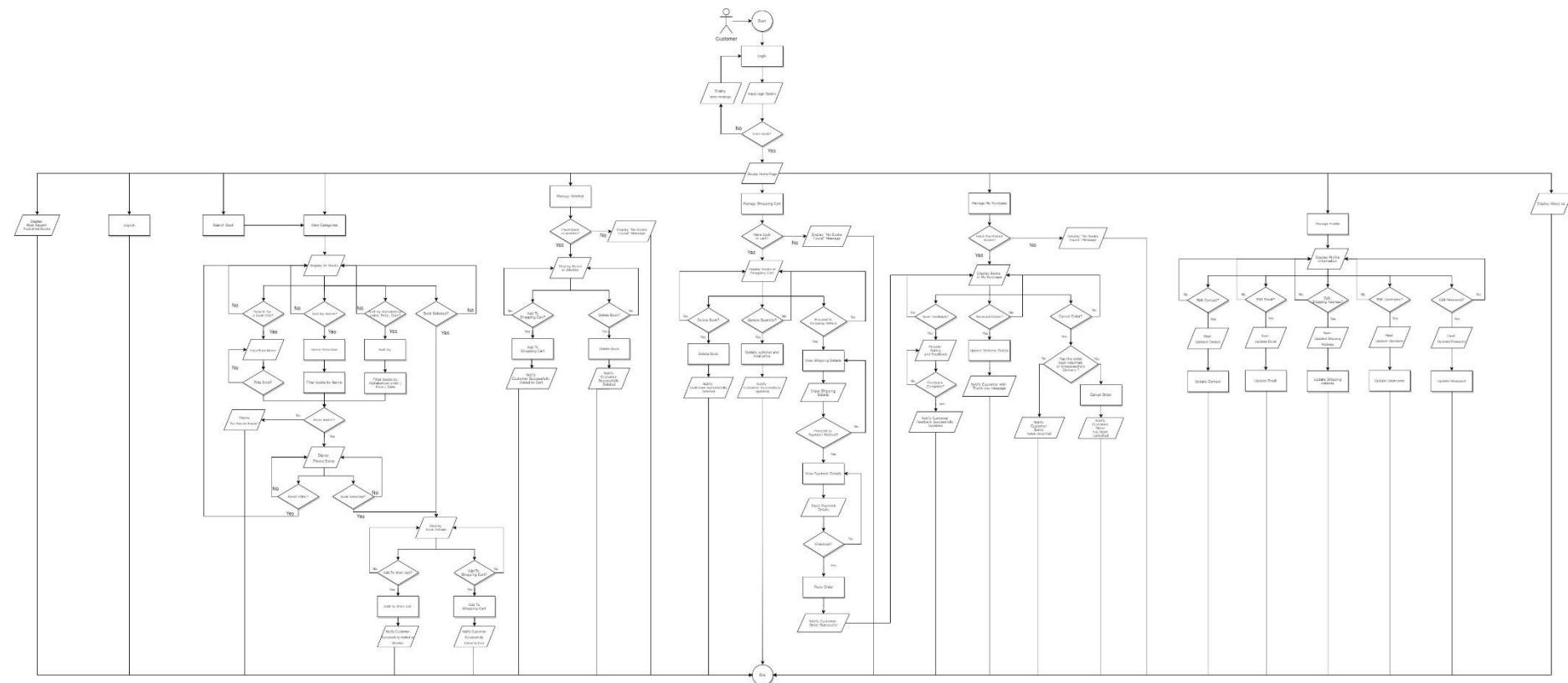


Figure 21:Customer Flowchart

The customer is one of the end-users that will use the website, and there will be a few basic functions for the customer to easily navigate through the website. Before accessing the main page or as known as the home page of the website, a customer who is registered already will have to log in via the login page.

Login

In the login page, the customer would have to enter their login details such as username and password. If their username or password is incorrect, it will display an error message and requires the user to enter their username and password once again. If the customer has successfully logged in, then it will redirect them to the homepage.

Homepage

In the homepage, customers are greeted with a big welcome message. On top of the screen is a navigation bar that allows users to navigate through the website with ease. In the navigation bar, the admin button that links to the admin page is hidden by default. The navigation bar consists of the logo of our company, a search book button, a home button that links back to the homepage, a category to link to the category page, My Wishlist that goes to the customer's Wishlist, Shopping Cart that directs them to their own shopping cart, My Purchase to view the purchases customers have made, My Profile for them to view and edit their profile details as well as a Log Out button to log out from the website. When customers scroll down at the homepage, there will be a most recent published books section where customers can see what books are published recently.

View Category

On the category page, there will be a list of books shown to the user. By default, the genre is set to no specific genre, no specific sort by the filter, and no specific title search. It will show every available book in the e-bookstore. Customers can search for a specific book title in the search bar by entering the book's title or a part of the book's title. If a book title matches what is inputted by the customer, it will show the results. The same goes for genre and sort by filter. If the customer wishes to search for a book that is in a specific genre while sorted by either alphabetical order, ascending or descending price, and most recent, the customer may select it, and results will be shown accordingly. Besides that, the customer may also reset the filters, and it will display the remaining results with the remaining filters. If there are no results, then a message "No results found." will be displayed. If the customer found a book, by clicking on the book, it will redirect them to the book details page, which will show all info about the book, including the price, full name, ISBN, and so on. Customers could also choose to add the book to a Wishlist or shopping cart. If the customers choose to either add it to Wishlist or shopping cart, then it will be added, and an alert message will inform the user that the book is added.

My Wishlist

Another page that can be accessed would be the My Wishlist page. In this page, it will display all books that are added to the Wishlist by the customer. If there are no books in their Wishlist then it will display "No Books Found", else if there are books then it will display all the books in the Wishlist. Once all the books have been displayed out, the customer can choose to add the books to shopping cart or remove them from the Wishlist. If the customer chooses to add the book to shopping cart, there will be a button for it and the system will update the database, it will also notify the customer that the book has been successfully added to the cart; if the customer were to delete the book from the Wishlist, the customer can click on the button and the system will update the database, and also notify the customer that the book has been successfully deleted.

Manage Shopping Cart

The manage shopping cart functionalities will first check if the users have any books in the shopping cart. If the user does not have any books in their shopping cart, then a message will display telling the users that “no books are found”. If the user has already added books to their shopping cart, then the books that are added to the cart will be displayed. In the shopping cart page, users can do three functionalities which includes deleting book, updating book quantity, and proceeding to shipping details page. For the delete book function, if users click on the delete book, then the book will be removed from their shopping cart and the user will be notified that their book has been successfully deleted. The next functionality that the user can perform is to update the quantity of the book. If the user were to increase or decrease the quantity of the book, then the subtotal and the total price of the book will increase according to the quantity that the user had set, and the system will notify the user that the quantity has been updated. The last function that users can perform in the shopping cart page to click next on the page and proceed to the shipping details page to fill up their shipping information. Once the shipping information has been filled up, click on the next button to proceed to the payment method. The payment method page will require the users to fill in their payment information such as their bank details. Once filled up, the user can continue to checkout to confirm their payment. Once the user clicked checkout and their payment details are valid, an alert message will notify the user that their order was successful, and the user will be redirected to the My Purchase page to view their purchases.

Manage My Purchase

The My Purchase page will show the purchases that has been done by the user. If the user has not purchased any books, a message will display notifying the user that no books have been purchased. If books have been purchased by the user, the list of purchased books will be shown in the page. Users can do three functions in this page. Users can give their feedback for the books ordered, notify received order, or cancel order. If users click on the give feedback button, users will be directed to the feedback page where users can input their feedback and hit submit once users are done to submit their feedback to the system. Once the user clicks on the submit button upon entering their feedback, a message will appear telling the user that the feedback entered was successful. Users can click on the receive order button once the item has been received and the delivery status of the user will be updated. A notification will alert the user that to tell them that their delivery status has been updated. The cancel order function is available to the users in this page as well. If users click on the cancel order button, the system will check if the order has been returned by the customer or an unsuccessful delivery has been done. If the shipping details meet the stated requirements, users will be allowed to cancel their order and their order status will change while being notified if the cancelation of their order is successful.

Manage Profile

Users can manage their personal details in the manage profile page. When users click on the manage profile page, the profile information of that user will be displayed. Users can choose to edit their contact by clicking on the edit icon in their contact section. Users can do the same with their email, shipping address, username and password. If the new inputs given by the user is in the correct formatting and no issue arises, the information will be updated, and the user will be alerted on the changed that has been made.

3.3.3 Guest Flowchart

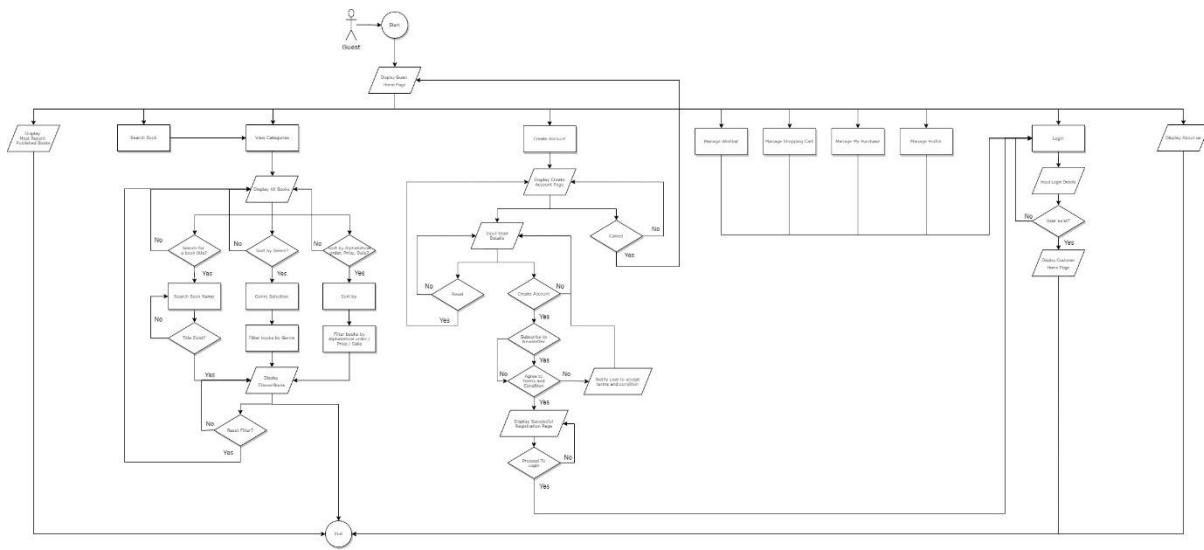


Figure 22: Guest Flowchart

Create Account

Guest can create an account in the create an account page by entering their particulars. When the create an account page is accessed, a display of all the information needed to register will be shown. The user can input their details such as their name, email, phone number, shipping address, password, and confirmation password. Users can choose whether to subscribe to a newsletter or not to get updated on future events and promotions. Users are required to check the agree to terms and condition checkbox. If users were to create an account without accepting the terms and conditions, the users account will not be created, and an error message will appear telling them to accept the terms and condition before creating an account. If user input the right data and checked the terms and conditions checkbox, users can click on the create account button to create their account. Once the button has been clicked, the user account will be created, and the user will be directed to a congratulation page notifying them that the account registration was successful. Users can reset their details in the register account page. This function will erase all the written information the user has keyed into the input boxes in the register account page. Cancel is another option that the users can do to cancel the registration and head back to the homepage as a guest. If the user is successful in their registration, the successful registration page will be displayed where users can click on the login button to login to their newly registered account on the website.

Login

The login page is a page that allows guest that has already created an account to sign into their registered account. If the input credentials of the user are valid, users will login and will be directed at the homepage of the website. Else, if there are no data in the database with the user's name and password, an error message will appear telling the user that their information is incorrect.

Homepage

In the homepage, guests are greeted with a big welcome message. On top of the screen is a navigation bar that allows guest to navigate through the website with ease. In the navigation bar, the admin button that links to the admin page is hidden by default. The navigation bar consists of the logo of Story Glory, a search book button, home button which links back to the homepage, category to link to the category page, My Wishlist that goes to the customer's Wishlist, Shopping Cart that directs them to their own shopping cart, My Purchase to view the purchases customers have made, My Profile for them to view and edit their profile details. Only the homepage and the view category page are accessible by the guests, meaning that guest can view and search for books to find but if guest wants to either view book details or other functionalities, guest will be redirected to the login page. When guest scroll down at the homepage, there will be a most recent published books section where customers can see what books are published recently.

View Category

In the category page, there will be a list of books shown to the guest. By default, the genre is set to no specific genre, no specific sort by filter, and no specific title search, it will show every available book in the e-bookstore. Guests can search for a specific book title in the search bar by entering the book's title or a part of the book's title. If a book title matches what is inputted by the guests, it will show the results. Same goes to genre and sort by filter, if the customer wishes to search for a book that is in a specific genre while sorted by either alphabetical order, ascending or descending price and most recent, the guest may select it and results will be shown accordingly. Besides that, guests may also reset the filters and it will display the remaining results with the remaining filters. If there are no results, then a message "No results found." will be displayed. If the guest found a book, by clicking on the book it will redirect them to the login page before continuing.

3.4 Entity Relationship Diagram (ERD)

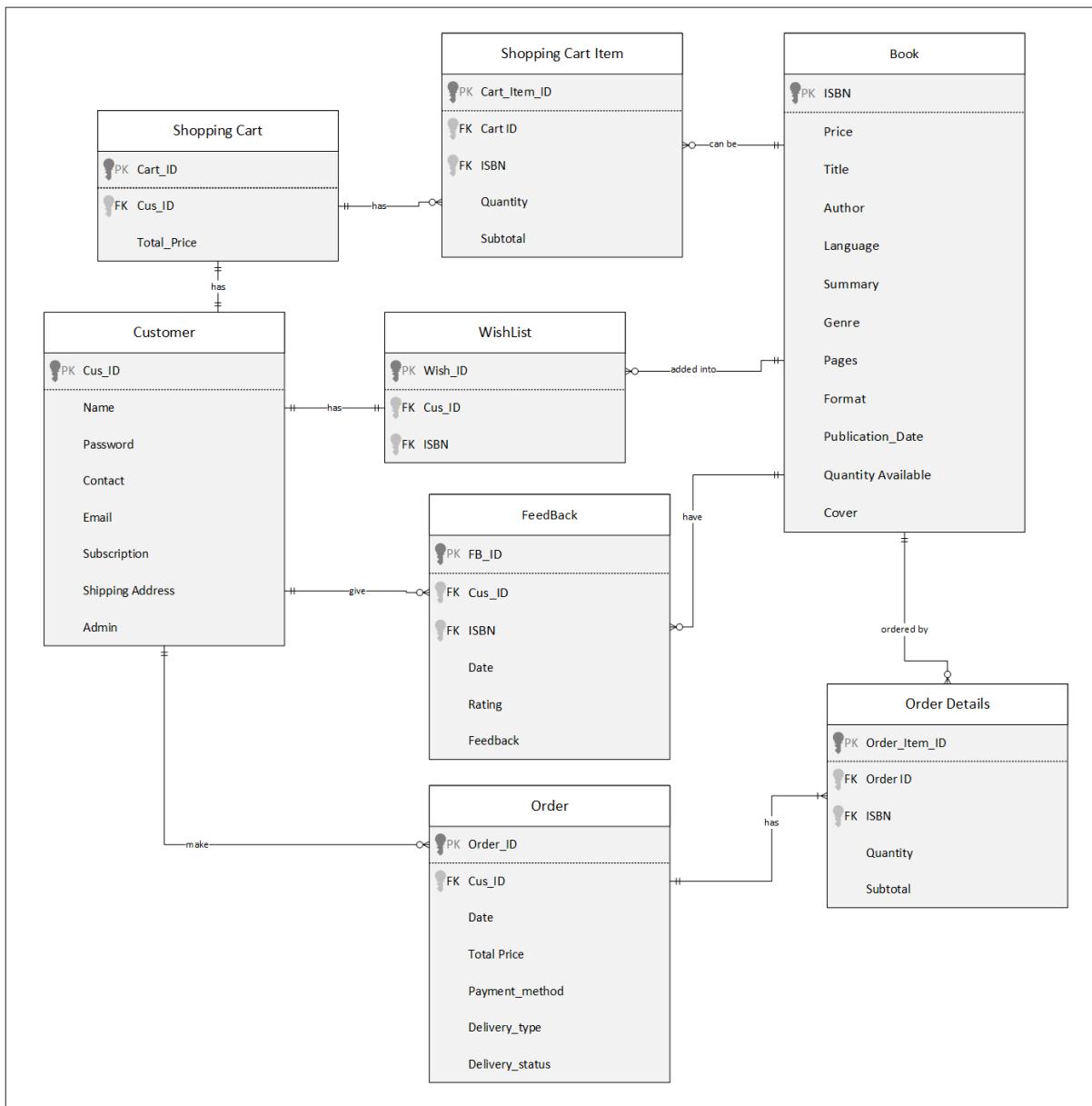


Figure 23: Entity Relationship Diagram (ERD) of Story Glory

4.0 Implementation

4.1 Server-Side Scripting (PHP)

4.1.1 Create - HTML Form, PHP

Create Customer Table

This is the SQL CREATE TABLE statement creates a new table in our “story_glory” database. Each column name is followed with a datatype parameter which specifies what data can be entered in that particular column. VARCHAR datatype allows letters and numbers within the specified size. TINYINT allows small integers to be entered. DEFAULT is used to set the number “0” in the admin column when a new customer is added. PRIMARY KEY allows the database to uniquely recognize each customer.

```
CREATE TABLE `customer` (
  `Cus_ID` int NOT NULL AUTO_INCREMENT,
  `Name` varchar(18) NOT NULL,
  `Password` varchar(12) NOT NULL,
  `Contact` varchar(12) NOT NULL,
  `Email` varchar(255) NOT NULL,
  `Subscription` tinyint(1) NOT NULL,
  `Shipping Address` varchar(255) NOT NULL,
  `Admin` tinyint(1) NOT NULL DEFAULT '0',
  PRIMARY KEY (`Cus_ID`)
)
```

Figure 24: Create Customer Table

Create Book Table

The SQL CREATE TABLE statement below creates a new table in the “story_glory” database. Each column name is followed with a datatype parameter which specifies what data can be entered in that column. The datatype used to create the book table consist of VARCHAR, FLOAT, INT, and DATE. VARCHAR datatype allows letters and numbers within the specified size while FLOAT only allows number with the display of decimal places. INT is like FLOAT but only accepts whole numbers while DATE accepts data in date format. Lastly a PRIMARY KEY is set to allows the database to uniquely recognize each book in the database.

```
CREATE TABLE `book` (
  `ISBN` int NOT NULL,
  `Price` float NOT NULL,
  `Title` varchar(110) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL,
  `Author` varchar(42) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL,
  `Language` varchar(8) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL,
  `Summary` varchar(1508) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL,
  `Genre` varchar(12) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL,
  `Pages` int NOT NULL,
  `Format` varchar(9) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL,
  `Publication_Date` date NOT NULL,
  `Quantity_Available` int NOT NULL,
  `Cover` varchar(88) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL,
  PRIMARY KEY (`ISBN`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8
```

Figure 25:Create Book Table

Create Shopping Cart Table

In order to store customer's to-purchase books, a Shopping Cart table is created. Every table should have a primary key and a column named "Cart_ID" is created. The Cart_ID Primary key is set to integer type input, and it is auto increment, meaning that it will automatically assign a new value after the previous value once a new book is entered into the cart. Another column is the "Cus_ID" column that records the customer's id and serves as an indicator of the owner of that cart, this column also is set to only integer values. The final column would be the "Total_Price" column that has a float value to store the total price that is calculated from the books inside the shopping cart of the customer. All these columns are set to NOT NULL to not allow NULL values as every single shopping cart should have a value in each column.

```
CREATE TABLE `shopping_cart` (
  `Cart_ID` int NOT NULL AUTO_INCREMENT,
  `Cus_ID` int NOT NULL,
  `Total_Price` float NOT NULL,
  PRIMARY KEY (`Cart_ID`),
  KEY `Cus_ID` (`Cus_ID`)
) ENGINE=MyISAM AUTO_INCREMENT=23 DEFAULT CHARSET=utf8
```

Figure 26:Create Shopping Cart Table

Admin Dashboard – New Book

This new book function allow admin to insert new book into the book database. The book information will be accepted by the new book form with the POST method. The book information will be retrieved by PHP POST method once the form is submitted. To insert the book, SQL INSERT INTO statement is used to specify columns names and values to be insert to the database. The PHP mysqli_query() function will execute the query. At last, a message box will notify the admin once the book is added to the database.

```
<div class="admin-book-col">
  <form method="post">
    <button type="submit" class="gold-btn" name="new-book" onclick ="add_book()">NEW</button>
```

Figure 27: New Book Form of admin.php

```
if(isset($_POST['new-book']))
{
  include("session.php");
  include("conn.php");
  $ISBN = $_POST['isbn'];
  $Price = $_POST['price'];
  $Pages = $_POST['pages'];
  $Title = $_POST['title'];
  $Format = $_POST['format'];
  $Author = $_POST['author'];
  $Publication_date = $_POST['date'];
  $Language = $_POST['language'];
  $Quantity = $_POST['qty'];
  $Summary = $_POST['summary'];
  $Cover = $_POST['cover'];
  $Genre = $_POST['genre'];

  $sql="INSERT INTO `book` (
  `ISBN`, `Price`, `Title`, `Author`, `Language`, `Summary`, `Genre`, `Pages`, `Format`, `Publication_Date`, `Quantity_Available`, `Cover`)
VALUES
('{$ISBN}', '{$Price}', '{$Title}', '{$Author}', '{$Language}', '{$Summary}', '{$Genre}', '{$Pages}', '{$Format}', '{$Publication_date}', '{$Quantity}', '{$Cover}');

  if(mysqli_query($con, $sql)) {
    mysqli_close($con);
    echo '<script>
      alert("Book Successfully Added!");
    </script>';
  }
}
```

Figure 28: New Book section of admin.php

Add book to Wishlist

If a customer is interested in a book, the customer may add the book to their Wishlist. When the button is clicked by the user, the website collects the customer id and the book id with the `$_POST` method that collects the data from the previously submitted form and stores it into variables `$Cus_ID` and `$Book_ID`. A query is then created to insert both the values into the `Cus_ID` and `Book_ID` column in the `Wishlist` table in the database. If the query is executed successfully, then the connection will be closed, and a message will pop up saying “Added to Wishlist!”.

```
include("conn.php");
$Cus_ID = $_POST['Cus_ID'];
$Book_ID = $_POST['Book_ID'];

$sql="INSERT INTO `wishlist` (
`Cus_ID`, `Book_ID`)
VALUES
('$Cus_ID', '$Book_ID');

if(mysqli_query($con, $sql)) {
    mysqli_close($con);
    echo '<script>
        alert("Added To Wishlist!");
    </script>';
}
```

Figure 29: Add Book to Wishlist

Add book to Shopping Cart

The figure below shows the php code to insert books into cart. When the add book to cart button is clicked, the customer id, book id, quantity of books, price, and subtotal are collected from the submitted form with the `$_POST` method by retrieving the values of the input id ‘Cus_ID’, ‘Book_ID’, ‘quantity’ as well as ‘Price’. The values collected are stored in variables `$Cus_ID`, `$Book_ID`, `$Quantity` and `$Price`. Before `$ quantity` and `$ price` are stored these two data are converted to integers to be used to calculate the subtotal and store this answer in a variable named `$ subtotal`. A query is created to select the customer id from shopping cart to check if a shopping cart exists for the customer. `Mysqli_num_rows` is used to find the number of rows of the result obtained by the query. If the result matches 0, a query will be created to insert the customer id into the shopping cart database and auto-increment a cart id for the customer. Another query is done to get the cart id of the customer, and the cart id is inserted into the shopping cart item database along with the book id, quantity, and subtotal that the user wants to insert into their shopping cart.

```

if(isset($_POST['add-book-to-cart']))
{
    include("conn.php");

    $Cus_ID = $_POST['Cus_ID'];
    $Book_ID = $_POST['Book_ID'];
    $Quantity = intval($_POST['quantity']);
    $Price = intval($_POST['Price']);
    $Subtotal = intval($Quantity * $Price);

    // check if cus already have shopping cart
    $cart_sql = "SELECT Cus_ID FROM shopping_cart WHERE Cus_ID = '$Cus_ID'";
    $cart_result = mysqli_query($con,$cart_sql);

    // if no shopping cart
    if(mysqli_num_rows($cart_result) == 0){

        // add shopping_cart
        $add_cart_sql = "INSERT INTO `shopping_cart`
        (Cus_ID) VALUES ($Cus_ID)";

        $add_cart = mysqli_query($con,$add_cart_sql);

    }

    // get cart id to insert into cart item table
    $get_cart_id = "SELECT Cart_ID FROM shopping_cart WHERE Cus_ID = '$Cus_ID'";
    $cart_id_result = mysqli_query($con, $get_cart_id);
    $cart_id = mysqli_fetch_array($cart_id_result);
    $Cart_ID = $cart_id['Cart_ID'];

    $insert_item_sql="INSERT INTO `shopping_cart_item` (
        `Cart_ID`, `Book_ID`, `Quantity` , `Subtotal`)
    VALUES
        ('$Cart_ID', '$Book_ID', '$Quantity', '$Subtotal')";

    $insert_item = mysqli_query($con,$insert_item_sql);
}

```

Figure 30:Insert Book to Cart

4.1.2 Read - HTML Form, PHP

Admin Dashboard – Search Book

This search book function allows the admin to search and manage the books in the online bookstore. If the admin wants to search, edit or delete books, admin will need to enter the search value in the input field of the HTML form as shown below. The search value would be accepted by the form using POST method after the form is submitted by one of the submit button which is “SEARCH”, “UPDATE” and “DELETE”. The admin will be directed to the Book Editing page to view search result by HTML form action attribute. The books will be search by the SQL select query with the WHERE clause, LIKE and OR operator. All books result will be display with a while loop that wraps a HTML form. If there are no search result, an error message will be displayed on the screen.

```
<form method="post" action="admin_search_book.php">
<div class="admin-book-col">
  <div><i class="fas fa-search-dollar"></i>Search &nbsp;&nbsp;</div>
  <input type="text" name="search-name" placeholder="Search Book">
  <button type="submit" class="gold-btn" name="search-book">SEARCH</button>
  <button type="submit" class="gold-btn">UPDATE</button>
  <button type="submit" class="gold-btn">DELETE</button>
</div>
</form>
```

Figure 31: Book Editing Form of admin.php

```
<?php
include("conn.php");
if(isset($_POST['search-name']))
{
    $search_name = $_POST['search-name'];

    $query = "SELECT *
    FROM `book`
    WHERE `ISBN` LIKE '%$search_name%' OR
    `Price` LIKE '%$search_name%' OR
    `Title` LIKE '%$search_name%' OR
    `Author` LIKE '%$search_name%' OR
    `Language` LIKE '%$search_name%' OR
    `Summary` LIKE '%$search_name%' OR
    `Genre` LIKE '%$search_name%' OR
    `Pages` LIKE '%$search_name%' OR
    `Format` LIKE '%$search_name%' OR
    `Publication_Date` LIKE '%$search_name%' OR
    ORDER BY `ISBN` DESC";

    $result = mysqli_query($con, $query);

    // Return the number of rows in result set
    $rowcount=mysqli_num_rows($result);

    if ( $rowcount==0 ) {
        echo " <div class=\"no-books-found\"> No Books Found </div>";
    }

    while($row = mysqli_fetch_array($result))
    {
?>
```

Figure 32: Search section of admin_search_book.php

Admin Dashboard - Total Sales

This Total Sales report allow admin to get the total sales of online bookstore. The total sales will be calculated with SQL query. SQL Select statement got the sum of the total price column from order database. The Round () function is used to round the total price to 2 decimal places. PHP script is used to execute SQL query and fetch the results.

```
<?php

include("conn.php");

$query = "SELECT ROUND(SUM(Total_Price),2) AS 'Total Sales' FROM `order`";
$result = mysqli_query($con, $query);
$total_sales = mysqli_fetch_array($result);
echo '<h6> RM ' . $total_sales['Total Sales'] . '</h6>';

mysqli_close($con);

?>
```

Figure 33: Total Price section of admin.php

Admin Dashboard - Users Registered

This Users Registered report allow the admin to discover the number of users registered in the online bookstore. The of users registered will be obtained by the PHP script using the mysqli_num_rows() function. number of rows will be counted from the SQL select query and mysqli_query() return result.

```
<?php  
  
include("conn.php");  
  
$query = "SELECT Cus_ID FROM customer";  
$result = mysqli_query($con, $query);  
  
$users_registered = mysqli_num_rows($result);  
echo '<h6>' . $users_registered . '</h6>';  
  
mysqli_close($con);  
  
?>
```

Figure 34: Users Registered section of admin.php

Admin Dashboard – Recent Customers

This Recent Customers section allow the admin to view all the recent customers with order ID, name, phone number, delivery status and a “UPDATE” button. The recent customers are read by the SQL SELECT statement, WHERE clause to connect customer ID and ORDER BY clause to sort list of recent customers in descending order. The PHP mysqli_query() function will execute the SQL query and the rows of result will be stored in a variable using mysqli_fetch_array() function. A while loop that wraps a HTML form will display every row of results in the webpage. The information is displayed by HTML form and PHP script as shown below.

```

<?php
include("conn.php");

$query = "SELECT *
          FROM `customer`, `order`
         WHERE `customer`.Cus_ID = `order`.Cus_ID
           ORDER BY `order`.Order_ID DESC";

$result = mysqli_query($con, $query);

while($row = mysqli_fetch_array($result))
{
?>



|                                                                      |                                                                  |                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------------------------------------------------|------------------------------------------------------------------|---------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <h3>&lt;?php echo \$row['Order_ID'] ?&gt;&lt;/h3&gt;&lt;/td&gt;</h3> | <h3>&lt;?php echo \$row['Name'] ?&gt;&lt;/h3&gt;&lt;/td&gt;</h3> | <h3>&lt;?php echo \$row['Contact'] ?&gt;&lt;/h3&gt;&lt;/td&gt;</h3> | <form method="post">           <input type="hidden" name="order_id" value="<?php echo \$row['Order_ID'] ?>">            <select name="Delivery_status" required>             <option value="">Please update delivery status</option>             <option value="Delivered">               <?php if(\$row['Delivery_status'] == "Delivered"){ ?>                 selected="selected"             </option>           </select>         </form> |
|----------------------------------------------------------------------|------------------------------------------------------------------|---------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|


```

Figure 35: Recent Customers section of admin.php

Feedback – Display Book Title

This display book title function displays the book title in the feedback form. SQL query with SELECT statement and WHERE clause will get the book title from database. PHP mysqli_query() function will execute the query and mysqli_fetch_array() function will get the query results. The Book Title were displayed within the form by PHP echo statement that retrieve book title from the database.

```
<form class="feedback-form" method="post">

<?php
    include("conn.php");

    $sql = "SELECT `Title` FROM `book` WHERE `ISBN`='$Book_ID'";
    $result = mysqli_query($con, $sql);
    $book_title = mysqli_fetch_array($result);
?>
<input type="hidden" name="Book_ID" value="<?php echo $Book_ID ?>">
<input type="hidden" name="Cus_ID" value="<?php echo $cus_id ?>">

<h4>Book</h4>
<h3><?php echo $book_title['Title'] ?></h3>
```

Figure 36: Feedback Form of feedback.php

Product – Display Book Rating

This display book rating function calculate the average book rating and display the rating result with star icons. An SQL query will get the average rating of a single book from the feedback database and round the average rating to one decimal place. PHP mysqli_query() function and mysqli_fetch_array() function is used to retrieve the book rating in float value using the floatval() function. PHP if else statement will also display the number of star icons according to the average book rating.

```
<?php
    $book_rating_sql = "SELECT ROUND(AVG(Rating), 1) As Rating
    FROM `feedback`
    WHERE Book_ID = '$Book_ID'";
    $book_rating_result = mysqli_query($con, $book_rating_sql);
    $book_rating_handle = mysqli_fetch_array($book_rating_result);
    $book_rating = floatval($book_rating_handle['Rating']);

?>
```

Figure 37: Book Rating section of product.php

```
<div class="show-rating">
<?php
    if($book_rating <= 1 ){
        echo '<i class="fas fa-star"></i>
        <i class="far fa-star"></i>
        <i class="far fa-star"></i>
        <i class="far fa-star"></i>
        <i class="far fa-star"></i>';
    }else
    if($book_rating > 1 ){
        echo '<i class="fas fa-star"></i>
        <i class="fas fa-star"></i>
        <i class="far fa-star"></i>
        <i class="far fa-star"></i>
        <i class="far fa-star"></i>';
    }else
    if($book_rating > 2 ){
        echo '<i class="fas fa-star"></i>
        <i class="fas fa-star"></i>
        <i class="fas fa-star"></i>
        <i class="far fa-star"></i>
        <i class="far fa-star"></i>';
    }else
    if($book_rating > 3 ){
        echo '<i class="fas fa-star"></i>
        <i class="fas fa-star"></i>
        <i class="fas fa-star"></i>
        <i class="far fa-star"></i>
        <i class="far fa-star"></i>';
    }else
    if($book_rating > 4 ){
        echo '<i class="fas fa-star"></i>
        <i class="fas fa-star"></i>
        <i class="fas fa-star"></i>
        <i class="fas fa-star"></i>
        <i class="fas fa-star"></i>';
    }
?>
<h6> <?php echo $book_rating; ?> </h6>
</div>
```

Figure 38: Book Rating section of product.php

Product – Book Reviews

The book reviews section will show all the customer's name, rating, and comment on the book. The book ID of the book will be retrieved from the URL with the PHP GET method. An SQL query will get the customer's name and feedback details using the INNER JOIN command and WHERE clause. PHP mysqli_query() function will execute the query and fetch the result with mysqli_fetch_array() function. List of book reviews will be display with the while loop. If there are no book reviews from customer, a message will be displayed on the screen.

```
<?php
    include("conn.php");

    $Book_ID = $_GET["isbn"];

    if(isset($Book_ID))
    {
        $sql = "SELECT customer.Name, feedback.*"
        FROM feedback
        INNER JOIN customer ON customer.Cus_ID = feedback.Cus_ID
        WHERE feedback.Book_ID = '$Book_ID';

        $result = mysqli_query($con, $sql);

        $rowcount=mysqli_num_rows($result);

        if ( $rowcount==0) {
            echo " <div class=\"no-books-found\"> No Feedback Provided </div>";
        }

        while($row = mysqli_fetch_array($result))
        {
    ?>
```

Figure 39: Book Reviews section of the product.php

Shopping Cart – Books details in shopping cart

The \$sql code select all the details of the shopping cart database while having an inner join with the shopping cart item, customer, and book database based on the \$cus_id variable which is taken from the login session of the user previously. From the \$sql query, a \$result variable is created to connect the query with the database that has been created which is using a \$con variable. The \$rowcount is a variable that will count that number of rows the query has gotten. If \$rowcount has no queries found, this means that the customer has no books in their shopping cart and a no books added text will appear. However, if there is a \$rowcount number, this means that the customer has books in their shopping cart and the cover page, book title, book price, book quantity, and subtotal of the book will be displayed.

```
<?php

include("conn.php");

if(isset($cus_id))
{
    $sql = "SELECT *
    FROM `shopping_cart`
    INNER JOIN `shopping_cart_item` ON shopping_cart.Cart_ID = shopping_cart_item.Cart_ID
    INNER JOIN `customer` ON shopping_cart.Cus_ID = customer.Cus_ID
    INNER JOIN book ON shopping_cart_item.Book_ID = book.ISBN
    WHERE shopping_cart.Cus_ID = '$cus_id'";

    $result = mysqli_query($con, $sql);

    // Return the number of rows in result set
    $rowcount=mysqli_num_rows($result);

    if ( $rowcount==0) {
        echo " <div class='no-books-found'> No Books Added </div>";
        echo "<input type='hidden' id='book_count' value='$rowcount' >";
    }

    while($row = mysqli_fetch_array($result))
    {
    }
}

?>
```

Figure 40:Book Details query in shopping cart

Shopping Cart – Display books details in shopping cart

The display book details figure below will display out the book title, book price, book quantity and book subtotal from the previous query done.

```
<td></td>
<td>
    <h3 id = "bookdetails" > <?php echo $row['Title']?> </h3>
    <h3>Price: RM <?php echo $row['Price']?> </h3>
    <br>
    <button type="submit" class="cart-special" name="delete-cart">Remove <i class="fas fa-trash"></i></button>
</td>

<td>
    <input type="number" name="Quantity" value="<?php echo $row['Quantity']?>">
    <button type="submit" class="cart-special" name="update-qty"><i class="fas fa-check-circle"></i></button>
</td>
<td class="shopping-subtotal">RM <?php echo $row['Subtotal']?></td>
```

Figure 41:Display Book Details in shopping cart

Shopping Cart – Display total book price

The display total book price figure shows an \$sql variable that select all the attributes form the shopping cart database where the Cus_ID = \$cus_id. \$cus_id is taken from a session variable which is the cus_id when customers were to log in to their account. A \$result variable is created to connect the \$sql query to our database using the \$con variable. The number of rows is calculated using the mysqli_num_rows function. A \$row function is created to fetch the array from \$result variable. The total price is then displayed out in the Total Price section.

```
<?php

include("conn.php");

if(isset($cus_id))
{
    $sql = "SELECT *
    FROM `shopping_cart`
    WHERE Cus_ID = '$cus_id'";

    $result = mysqli_query($con, $sql);

    // Return the number of rows in result set
    $rowcount=mysqli_num_rows($result);

    $row = mysqli_fetch_array($result);
?>
<div class="total-price">
    <table>
        <tr>
            <td>Total Price</td>
            <td>RM <?php echo $row['Total_Price']?> </td>
        </tr>
    </div>
```

Figure 42:Display Total Book Price

Shipping Details – User details in shipping details

An \$sql variable is created to select all elements of the database table shopping_cart along with the inner join of shopping_cart_item, customer, and book database tables. The query is filtered out with the shopping_cart customer id based on the \$cus_id variable that is taken from session when the user log in. A \$result variable is created to make a connection with the query and the database. The \$result variable array is fetched and stored in a \$row variable.

```
<?php
include('session.php');
include("conn.php");

$sql = "SELECT *
FROM `shopping_cart`
INNER JOIN `shopping_cart_item` ON shopping_cart.Cart_ID = shopping_cart_item.Cart_ID
INNER JOIN `customer` ON shopping_cart.Cus_ID = customer.Cus_ID
INNER JOIN book ON shopping_cart_item.Book_ID = book.ISBN
WHERE shopping_cart.Cus_ID = '$cus_id';

$result = mysqli_query($con, $sql);

$row = mysqli_fetch_array($result);

?>
```

Figure 43: User details query in shipping details

Shipping Details – Display user details in shipping details

The display book shipping details figure shows the display username, user contact, user shipping address, total price of book, and total price after choosing a shipment method. The displays are done by using the \$row variable and displaying the attributes into the website.

```
<div class="input-row">
    <div class="input-group">
        <input type="text" placeholder="Full Name" required value="<?php echo $row['Name'] ?>">
    </div>

    <div class="input-group">
        <input type="tel" placeholder="Phone Number" required value="<?php echo $row['contact'] ?>">
    </div>
</div>

<textarea rows="5" placeholder="Address" required><?php echo $row['Shipping_Address'] ?></textarea>
<input type="hidden" name="cus_ID" value="<?php echo $row['cus_ID'] ?>">

<section class="shipping-right">
    <div class="container">
        <h4>Cart <span style="color:black"><i class="fas fa-shopping-cart"></i></span></h4>
        <hr>
        <input type="hidden" id="subtotal_hidden" value="<?php echo $row['Total_Price'] ?>">
        <p id="subtotal">Subtotal<span>RM<span id="subtotal_amount"><?php echo $row['Total_Price'] ?></span></span></p>
        <p>Shipping<span id="Shipping">RM10.00</span></p>
        <hr>
        <input type="hidden" name="total_price_hidden" id="total_price_hidden" value="<?php echo $row['Total_Price'] + 10; ?>">
        <p>Total <span>RM<span class="price" style="color:black" id="total_amount"><b><?php echo $row['Total_Price'] + 10; ?></b></span></span></p>
    </div>
</section>
```

Figure 44: Display user details in shipping details

Payment Method – Book title and subtotal in payment method

\$cart_query variable is created to query for the book title and subtotal of the book. The query selects from book. An inner join of shopping_cart_item and shopping cart is done in the query based on the customer id which is taken from the previous login session of the customer. The connection of the database to the query is done and stored to \$cart_result variable.

```
//get book title and Subtotal
$cart_query = "SELECT shopping_cart_item.subtotal, book.Title FROM book
INNER JOIN shopping_cart_item ON book.ISBN = shopping_cart_item.Book_ID
INNER JOIN shopping_cart ON shopping_cart_item.Cart_ID = shopping_cart.Cart_ID
WHERE Cus_ID = $cus_id";

$cart_result = mysqli_query($con, $cart_query);
```

Figure 45: Book title and subtotal in payment method

Payment Method – Display Book title and subtotal in payment method

A while loop is done, and all the data is stored into \$row variable. The \$row variable is used to get the attribute of book title and attribute of the book subtotal. With the while loop, if there are more than 1 book title and book subtotal, the loop will display out all the results till there are no more results for existing book title or book subtotal.

```
<?php
while ($row = mysqli_fetch_array($cart_result))
{
?>
<br>
<p class="title"><span ><?php echo $row['Title']; ?></span></p><span class="payment"><?php echo $row['subtotal']; ?></span>
<br>
<?php
}
?>
```

Figure 46: Display Book title and subtotal in payment method

Payment Method – Total unique book order query and display

\$book_quantity_sql variable is a sql query that use the count function to count the number of unique book_id in the database of shopping_cart_item table based on the customers unique Cart_ID and display it out. The book_quantity_sql is then connected to the database using the mysqli_query function with the \$con variable and stored in \$book_quantity_result. Mysqli_fetch_array is then used to fetch the arrays and stored into \$book_quantity_row variable. The \$book_quantity_row variable is then echo out in the html area to display out the total unique order done by the customer next to a cart icon.

```
//Book Quantity
$book_quantity_sql = "SELECT count(Book_ID) AS Total_unique_ordered
FROM shopping_cart_item WHERE Cart_ID = $cus_id";

$book_quantity_result = mysqli_query($con, $book_quantity_sql);
$book_quantity_row = mysqli_fetch_array($book_quantity_result);

<h4>Cart <span class="payment" style="color:black"><i class="fa fa-shopping-cart"></i> <b><?php echo $book_quantity_row['Total_unique_ordered']; ?></b></span></h4>
```

Figure 47:Query and display total unique customer book order

Payment Method – Query to display customer information

In the figure below, a query is done to select all the attributes from the customer table based on the customers unique customer id and store inside the \$sql variable. A connection is made, and a while loop is done to while fetching the array and storing it into the \$row variable. The customers data will be displayed automatically when the customer was to make their purchase would include their full name, email, and address. All these mentioned information was taken from the customer database that the system has taken during the registrations of the customer.

```

if(isset($cus_id))
{
    $sql = "SELECT *
    FROM `customer`
    WHERE Cus_ID = '$cus_id'";

    $result = mysqli_query($con, $sql);

    while($row = mysqli_fetch_array($result))
    {
        <?php
    }
}
?>

<input name="deliver_type" type="hidden" value=" <?php echo $deliver_type; ?>">
<label for="fname"><i class="fa fa-user"></i> Full Name</label>
<input type="text" id="fname" name="fullname" placeholder="Full Name" value="<?php echo $row['Name']; ?>" required>
<label for="email"><i class="fa fa-envelope"></i> Email</label>
<input type="text" id="email" name="email" placeholder="Email" value="<?php echo $row['Email']; ?>" required>
<label for="adr"><i class="fa fa-address-card-o"></i> Address</label>
<input type="text" id="adr" name="address" placeholder="Address" value="<?php echo $row['Shipping_Address']; ?>" required>
<label for="city"><i class="fa fa-institution"></i> City</label>
<input type="text" id="city" name="city" placeholder="Kuala Lumpur" required>

<?php
}
}
?>

```

Figure 48:Display Customer Info in Payment Section

My Purchase – Query to check and display user order

The figure below shows a query that select all attributes from order database while having an inner join with the customer database based on the customer unique id. A connection is made between the query and the database of the system and stored into \$result variable. The number of rows is calculated using the mysqli_num_rows function and stored into \$rowcount variable. If the \$rowcount variable is 0, means there are no books ordered and a display message telling the user that no books are found will be displayed. If \$rowcount variable has at least 1 book, it will not display anything but run a while loop getting all the order information of the customer and displaying it to the customer such as the delivery status of the customer's order.

```

<form method="POST">
<div class='purchase-box'>
    <div class="purchase-header">
        <div class="Purchase-User">
            <i class="fas fa-user-circle"></i>
            <p><?php echo $login_session; ?></p>
        </div>
    </div>
    <?php

    $sql = "SELECT * FROM `order`
INNER JOIN customer ON `order`.Cus_ID = customer.Cus_ID
WHERE `order`.Cus_ID = '$cus_id'";

$result = mysqli_query($con, $sql);

// Return the number of rows in result set
$rowcount=mysqli_num_rows($result);

if ( $rowcount==0) {
    echo "<div class='purchase-box'>";
    echo "<div class='purchase-header'>";
    echo "<div class='Purchase-status'>";
    echo "<p>Delivery Status</p>";
    echo "</div>";
    echo "</div>";
    echo "<br>";
    echo "<br>";
    echo "<div class=\"no-books-found\"> No Purchase </div>";
}

}else{
    echo "";
}

//get order based on customer id
while($order_row = mysqli_fetch_array($result))
{
    ?>

    <div class="purchase-header">
        <div class="Purchase-status">
            <p>Delivery Status</p>
            <p style="color:red; font-size: 20px;"><?php echo $order_row['Delivery_Status']; ?></p>
        </div>
    </div>
    <br>
    <br>
}

```

Figure 49:Query to check and display user order

My Purchase – Query to check and display user order

The figure below is connected to the previous figure where it takes the \$order_row variable created and gets the order_ID and store into a variable. A query is made to select all details from order details where the order_ID is equal to the \$order_id variable. A while loop is done to get the order details based on the order id. A query is made inside that while loop to get all attribute from order details databased while inner join with book database based on the order_item_id. A final while loop is done to get all the book details based on the order item id and display out to the customer such as the book cover, book title, quantity, subtotal, and book id.

```
<?php
$order_id = $order_row['Order_ID'];
$get_order_item_sql = "SELECT * FROM `order_details` WHERE Order_ID = '$order_id'";
$get_order_item_result = mysqli_query($con, $get_order_item_sql);
//get order details based on order id
while($item_row = mysqli_fetch_array($get_order_item_result))
{
    $order_item_id = $item_row['Order_Item_ID'];
    $item_details_sql = "SELECT * FROM `order_details`"
        . " INNER JOIN book ON order_details.Book_ID = book.ISBN"
        . " WHERE Order_Item_ID = '$order_item_id'";
    $item_details_result = mysqli_query($con, $item_details_sql);
    //get book details based on order item id
    while($item_details_row = mysqli_fetch_array($item_details_result))
    {
        $item_result =
            '<div class="purchase-item">
                
                <div class="purchase-text">
                    <p class="purchase_title">'. $item_details_row['Title'] .'
                    <div class="purchase-price">
                        <p>x '. $item_details_row['Quantity'] .'
                        <p>'. $item_details_row['Subtotal'] .'
                    </div>
                    <div class="purchase-feedback-btn">
                        <a href="feedback.php?isbn='.$item_details_row['Book_ID'].'">Feedback</a>
                    </div>
                </div>
                <br>
                <br>
            ';
        echo $item_result;
    }
}
?>
```

Figure 50: Check display user order

Category.php – display list of books

In the category page, it consists of a search bar, a genre section, and a sort by section, alongside with a book display section with 3 rows. Each row has 5 books and users can display the next 15 books by clicking on the next button.

The main body of the category.php is the display of all the books. This page is more php driven as it requires a lot of different queries to filter the books the customer is trying to find. When the customer first visits the category page, it will display all the books available.

```
$valueToSearch = $_GET['valueToSearch'];

$page = $_GET['page'];
$sortfilter = $_GET['sort'];
$genre = $_GET['genre'];

if ($page == "1") {
    $startingnumber = "0";
    $startingnumber2 = $startingnumber + 5;
    $startingnumber3 = $startingnumber2 + 5;
    $endingnumber = $startingnumber3 + 5;

} else{
    $newpage = $page - 1;
    $startingnumber = $newpage * 15;
    $startingnumber2 = $startingnumber + 5;
    $startingnumber3 = $startingnumber2 + 5;
    $endingnumber = $startingnumber3 + 5;
}

$genrequery = "WHERE `Genre` like '%$genre%'";
$secondsearchquery = "`Title` LIKE '%$valueToSearch%'";

$totalrow_result = mysqli_query($con,"SELECT count(*) FROM `book` $genrequery AND $secondsearchquery");
$totalrow = mysqli_fetch_array($totalrow_result);
$total = $totalrow[0];

if ($total == 0){
    $noresults = "No Results Found.";
}

$total = (int)($total);
$maximum_page = intdiv($total, 15);
$maximum_page = $maximum_page + 1;
```

Figure 51: Category.php

The valuetosearch, page, sortfilter and genre are all obtained from the URL using the \$_GET method. It also assigns the current page number into a variable and some extra calculations for startingnumber, startingnumber2, startingnumber3 and endingnumber for some queries that will be used later. A genre related query and valueToSearch query is preset including the stored variables above. There is also another total books query to obtain the number of total books that are in the database that matches the query. If the number of total books is 0 then an error message “No results found.” will be stored in a variable to be echoed out later.

```

if ($sortfilter == "alphabet") {
    $sortname = "A to Z";
    $searchquery1 = "SELECT * FROM `book` $genrequery AND $secondsearchquery ORDER BY `Title` ASC LIMIT $startingnumber,5";
    $searchquery2 = "SELECT * FROM `book` $genrequery AND $secondsearchquery ORDER BY `Title` ASC LIMIT $startingnumber2,5";
    $searchquery3 = "SELECT * FROM `book` $genrequery AND $secondsearchquery ORDER BY `Title` ASC LIMIT $startingnumber3,5";
} elseif ($sortfilter == "pricelth"){
    $sortname = "Price Low to High";
    $searchquery1 = "SELECT * FROM `book` $genrequery AND $secondsearchquery ORDER BY `Price` ASC LIMIT $startingnumber,5";
    $searchquery2 = "SELECT * FROM `book` $genrequery AND $secondsearchquery ORDER BY `Price` ASC LIMIT $startingnumber2,5";
    $searchquery3 = "SELECT * FROM `book` $genrequery AND $secondsearchquery ORDER BY `Price` ASC LIMIT $startingnumber3,5";
} elseif ($sortfilter == "pricehtl"){
    $sortname = "High to Low";
    $searchquery1 = "SELECT * FROM `book` $genrequery AND $secondsearchquery ORDER BY `Price` DESC LIMIT $startingnumber,5";
    $searchquery2 = "SELECT * FROM `book` $genrequery AND $secondsearchquery ORDER BY `Price` DESC LIMIT $startingnumber2,5";
    $searchquery3 = "SELECT * FROM `book` $genrequery AND $secondsearchquery ORDER BY `Price` DESC LIMIT $startingnumber3,5";
} elseif ($sortfilter == "recent"){
    $sortname = "Recently Published";
    $searchquery1 = "SELECT * FROM `book` $genrequery AND $secondsearchquery ORDER BY `Publication_Date` DESC LIMIT $startingnumber,5";
    $searchquery2 = "SELECT * FROM `book` $genrequery AND $secondsearchquery ORDER BY `Publication_Date` DESC LIMIT $startingnumber2,5";
    $searchquery3 = "SELECT * FROM `book` $genrequery AND $secondsearchquery ORDER BY `Publication_Date` DESC LIMIT $startingnumber3,5";
} else{
    $sortname = "";
    $searchquery1 = "SELECT * FROM `book` $genrequery AND $secondsearchquery LIMIT $startingnumber,5";
    $searchquery2 = "SELECT * FROM `book` $genrequery AND $secondsearchquery LIMIT $startingnumber2,5";
    $searchquery3 = "SELECT * FROM `book` $genrequery AND $secondsearchquery LIMIT $startingnumber3,5";
}
?>

```

Figure 52: category.php part 2

Figure below shows the if else statement if the sortfilter has different string stored in. Each of the different sortfilter values will result in a different query sentence. For instance, if sortfilter has the value of “alphabet” then sortname will be assigned the string “A to Z”. and three queries that sort the results based on book “Title” from A to Z will be generated.

```

<span class="genre">
    <input id="genreid" type="hidden" value=<?php echo $genre; ?>>
    <h2>Genres</h2>
    <ul class="genre-list">
        <li><a href="category.php?page=1&sort=<?php echo $sortfilter; ?>&genre=Academi&valueToSearch=<?php echo $valueToSearch?>">Academic</a></li>
        <li><a href="category.php?page=1&sort=<?php echo $sortfilter; ?>&genre=Motivational&valueToSearch=<?php echo $valueToSearch?>">Motivational</a></li>
        <li><a href="category.php?page=1&sort=<?php echo $sortfilter; ?>&genre=Fantasy&valueToSearch=<?php echo $valueToSearch?>">Fantasy</a></li>
        <li><a href="category.php?page=1&sort=<?php echo $sortfilter; ?>&genre=Romance&valueToSearch=<?php echo $valueToSearch?>">Romance</a></li>
        <li><a href="category.php?page=1&sort=<?php echo $sortfilter; ?>&genre=Horror&valueToSearch=<?php echo $valueToSearch?>">Horror</a></li>
        <li><a href="category.php?page=1&sort=<?php echo $sortfilter; ?>&genre=Sci-fi&valueToSearch=<?php echo $valueToSearch?>">Sci-fi</a></li>
        <li><a href="category.php?page=1&sort=<?php echo $sortfilter; ?>&genre=History&valueToSearch=<?php echo $valueToSearch?>">History</a></li>
        <li><a href="category.php?page=1&sort=<?php echo $sortfilter; ?>&genre=Kids&valueToSearch=<?php echo $valueToSearch?>">Kids</a></li>
        <li><a href="category.php?page=1&sort=<?php echo $sortfilter; ?>&genre=&valueToSearch=<?php echo $valueToSearch?>"><strong>Reset Genre</strong></a></li>
    </ul>
</span>

```

Figure 53: category.php genre filter

Figure below shows the genre filter list that is located at the left side of the website. Every element in the list is a genre, if clicked on a genre, it will redirect to the same page but with a different URL as the string that comes after the genre is different. For example, while clicked on “Academic” then the string after genre in the URL will be changed to “Academic”.

```
<span class="sortby">Sort by:&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<a href="category.php?page=1&sort=alphabet&genre=<?php echo $genre; ?>&valueToSearch=<?php echo $valueToSearch?>">A-Z</a>&nbsp;|&nbsp;
<a href="category.php?page=1&sort=pricelth&genre=<?php echo $genre; ?>&valueToSearch=<?php echo $valueToSearch?>">Price: low to high</a>&nbsp;|&nbsp;
<a href="category.php?page=1&sort=pricehtl&genre=<?php echo $genre; ?>&valueToSearch=<?php echo $valueToSearch?>">Price: high to low</a>&nbsp;|&nbsp;
<a href="category.php?page=1&sort=recent&genre=<?php echo $genre; ?>&valueToSearch=<?php echo $valueToSearch?>">Most recent</a>&nbsp;|&nbsp;
<a href="category.php?page=1&sort=&genre=<?php echo $genre; ?>&valueToSearch=<?php echo $valueToSearch?>">Reset Sorting</a>&nbsp;|&nbsp;
</span>
```

Figure 54: category.php sort by

Same goes to the sort by menu at the right. When clicked, the customer will be redirected to the same page but with a different URL to achieve different query results.

Below the search bar is where all the books are shown. In this section, a while loop is used to display all the books row by row with different row queries. All the books looped by the while loop are in a so that it all the 5 books will be in the same line.

```
<div class="content">
<?php
$sql_query = mysqli_query($con,$searchquery1) or die(mysqli_error($con));
while ($bookrow = mysqli_fetch_array($sql_query,MYSQLI_ASSOC))
{
?>
<span class="box"><a href='product.php?isbn=<?php echo $bookrow['ISBN'] ?>&'><img src='<?php echo $bookrow['Cover'] ?>'></a></span>
<?php
}
?>
<br>
</div>
```

Figure 55: category.php content part books

4.1.3 Update - HTML Form, PHP

Admin Dashboard – Update Book

This update book function will collect book information with the HTML form POST method. If the “UPDATE” button is clicked, all the book information will be retrieved by PHP POST method to get values of variables. All the book information will be updated by SQL query with a WHERE clause to update a specific book in the book database based on the old ISBN value given. A message box will appear once the book has been successfully updated.

```
<form method="post">
<table class="edit-book">
    <tr>
        <td>ISBN</td>
        <input type="hidden" name="old_isbn" value="php echo $row['ISBN'] ?&gt;"&gt;
        &lt;td&gt;&lt;/td&gt;
        &lt;td&gt;&lt;input type="text" name="new_isbn" placeholder="ISBN" minlength="9" maxlength="9" value="<?php echo $row['ISBN'] ?&gt;" required&gt;&lt;/td&gt;</pre

```

Figure 56: Book Editing form of admin_search_book.php

```
<div class="edit-btns">
    <button type="submit" class="gold-btn" name="update-book">UPDATE</button>
    <button type="submit" class="gold-btn" name="delete-book" >DELETE</button>
</div>
</form>
```

Figure 57: Book Editing form of admin_search_book.php

```
if(isset($_POST['update-book']))
{
    include("conn.php");
    $old_ISBN = $_POST['old_isbn'];
    $new_ISBN = $_POST['new_isbn'];
    $Price = $_POST['price'];
    $Pages = $_POST['pages'];
    $Title = $_POST['title'];
    $Format = $_POST['format'];
    $Author = $_POST['author'];
    $Publication_date = $_POST['date'];
    $Language = $_POST['language'];
    $Quantity = $_POST['qty'];
    $Summary = $_POST['summary'];
    $Cover = $_POST['cover'];
    $Genre = $_POST['genre'];

    $query=" UPDATE `book` SET
        `ISBN` = '$new_ISBN',
        `Price` = '$Price',
        `Title` = '$Title',
        `Author` = '$Author',
        `Language` = '$Language',
        `Summary` = '$Summary',
        `Genre` = '$Genre',
        `Pages` = '$Pages',
        `Format` = '$Format',
        `Publication_Date` = '$Publication_date',
        `Quantity_Available` = '$Quantity',
        `Cover` = '$Cover'
        WHERE `ISBN` = '$old_ISBN'";

    if(mysqli_query($con, $query)) {
        mysqli_close($con);
        echo '<script>
            alert("Book Information Successfully Updated!");
        </script>';
    }
}
```

Figure 58: Update Book PHP script of admin_search_book.php

Shopping Cart –Update shopping cart total price

The figure below shows a query to update the total price when a user was to delete books from their shopping cart. A query is made to calculate the total price of the items bought by the user and stored into a variable. The variable is stored into the \$Total_price variable and an update query is done. The update query will update the total price in the shopping_cart database to the \$Total_price variable. If the update is successful, an alert message will notify the user that their cart has been successfully updated.

```
// calc total price
$total_price_sql = "SELECT shopping_cart.Cus_ID, ROUND(SUM(shopping_cart_item.Subtotal),2) AS Total
FROM shopping_cart
INNER JOIN shopping_cart_item ON shopping_cart.cart_ID = shopping_cart_item.cart_ID
WHERE shopping_cart.Cus_ID = '$cus_id'
GROUP BY shopping_cart.Cart_ID";

$cal_total_price = mysqli_query($con,$total_price_sql);
$total_price = mysqli_fetch_array($cal_total_price);

$Total_Price = $total_price['Total'];

$update_sql = "UPDATE shopping_cart SET
Total_Price = '$Total_Price'
WHERE Cus_ID = '$cus_id'";

if(mysqli_query($con, $update_sql)) {
    mysqli_close($con);
    echo '<script>
        alert("Successfully Updated The Cart!");
    </script>';
}
```

Figure 59:Update shopping cart total price

Shopping Cart –Update shopping cart quantity

Figure below shows the storing of variable which has been taken by POST method from the input of customer. The variables used this query consist of book id, quantity, price, and subtotal which are all taken from the POST method. A query that selects the cart_id from shopping cart database based on the customer id is done and stored into the \$Cart_ID variable. Another query is then created to update the quantity and subtotal in the shopping_cart_item database with the variables of \$ quantity and \$Subtotal done in the previous query.

```

if(isset($_POST['update-qty'])){
    include("conn.php");

    $Book_ID = $_POST['Book_ID'];
    $Quantity = intval($_POST['Quantity']);
    $Price = floatval($_POST['Price']);
    $Subtotal = floatval($Quantity * $Price);

    // get cart id to update cart item table & cart
    $get_cart_id = "SELECT Cart_ID FROM shopping_cart WHERE Cus_ID = '$cus_id'";
    $cart_id_result = mysqli_query($con, $get_cart_id);
    $cart_id = mysqli_fetch_array($cart_id_result);
    $Cart_ID = $cart_id['Cart_ID'];

    //update qty n subtotal
    $update_item_sql = "UPDATE `shopping_cart_item` SET
        Quantity = '$Quantity',
        Subtotal = '$Subtotal'
        WHERE Book_ID = '$Book_ID' AND Cart_ID = '$Cart_ID'";

    mysqli_query($con, $update_item_sql);

    // calc total price
    $total_price_sql = "SELECT shopping_cart.Cus_ID,
        ROUND(SUM(shopping_cart_item.Subtotal),2) AS Total
        FROM shopping_cart
        INNER JOIN shopping_cart_item ON shopping_cart.Cart_ID =
            shopping_cart_item.Cart_ID
        WHERE shopping_cart.Cus_ID = '$cus_id'
        GROUP BY shopping_cart.Cart_ID";

    $cal_total_price = mysqli_query($con,$total_price_sql);
    $total_price = mysqli_fetch_array($cal_total_price);

    $Total_Price = $total_price['Total'];

    $update_sql = "UPDATE shopping_cart SET
        Total_Price = '$Total_Price'
        WHERE Cus_ID = '$cus_id'";

    if(mysqli_query($con, $update_sql)) {
        mysqli_close($con);
        echo '<script>
            alert("Successfully Updated!");
        </script>';
    }
}

```

Figure 60:Update Quantity

Update shopping cart total price

Once the customer has paid for all the books to purchase, the system will then update the shopping cart's total price. The total price will be set to 0 as in to clear the shopping cart. First the UPDATE statement is stored in the \$reset_total_price. The statement is used to update the shopping cart table by setting the total price of the shopping cart of the respective customer to 0.

```
// Reset shopping cart price to 0 after checkout
$reset_total_price = "UPDATE shopping_cart SET Total_Price = 0 WHERE Cus_ID = $cus_id";
$reset_total_price_result = mysqli_query($con, $reset_total_price);
```

Figure 61: Update total price

4.1.4 Delete - HTML Form, PHP

Admin Dashboard –Delete Book

This deletes book function will collect book information with the HTML form POST method from Book Editing Form as shown in the figure below. The delete book function will run after the form is submitted by “Delete” button. PHP POST method is used to collect the ISBN from the HTML form. SQL delete query is used delete the specified book from book database with a WHERE clause. A message box will notify admin when the book had been successfully removed from the database.

```
if(isset($_POST['delete-book']))
{
    include("conn.php");
    $ISBN = $_POST['old_isbn'];
    $sql = "DELETE FROM `book` WHERE `ISBN` = '$ISBN'";

    if(mysqli_query($con, $sql)) {
        mysqli_close($con);
        echo '<script>
            alert("Book Successfully Removed!");
        </script>';
    }
}
```

Figure 62: Book Editing form of admin_search_book.php

Shopping Cart–Delete Book

This deletes book function in the shopping cart page will be activated using the POST method. When the user clicks on the delete cart icon, the isset function will be triggered and the book that users have selected will be deleted from their shopping cart. An SQL delete query is used delete the specified book from shopping cart database with a WHERE clause. Once the user clicks on the delete icon, the book that users have selected to be deleted will be removed from the shopping cart page.

```
if(isset($_POST['delete-cart'])){  
    include("conn.php");  
  
    $Book_ID = $_POST['Book_ID'];  
  
    // get cart id to delete book from cart item table  
    $get_cart_id = "SELECT Cart_ID FROM shopping_cart WHERE Cus_ID = '$cus_id'";  
    $cart_id_result = mysqli_query($con, $get_cart_id);  
    $cart_id = mysqli_fetch_array($cart_id_result);  
    $Cart_ID = $cart_id['Cart_ID'];  
  
    // delete book from cart  
    $del_book_sql = "DELETE FROM `shopping_cart_item` WHERE Book_ID = '$Book_ID' AND Cart_ID = '$Cart_ID'";  
    mysqli_query($con, $del_book_sql);
```

Figure 63: Shopping Cart–Delete Book

Delete all shopping cart items

Once a payment is successfully made, the system will remove every item that is in the shopping cart. A delete query is created with keywords DELETE, INNER JOIN and WHERE. INNER JOIN is used to join tables together according to the matching customer id. Once the delete statement is done then the customer will be redirected to the My Purchase page.

```
//delete shopping cart item after checkout  
$delete = "DELETE shopping_cart_item.* FROM shopping_cart INNER JOIN shopping_cart_item  
ON shopping_cart.Cart_ID = shopping_cart_item.Cart_ID WHERE Cus_ID = '$cus_id';  
  
$delete_result = mysqli_query($con, $delete);  
  
echo '<script> window.location.href = "MyPurchase.php"; </script>';
```

Figure 64: delete shopping cart items

4.1.5 Login - HTML Form, PHP

Login page

The login function is the heart and soul of keeping the system well managed and filter the contents according to the user's profile. This login function collects the data through the HTML form POST method and by user's input data submitted through the "LOGIN" button.

```

<h1>Login</h1>

<div>
    Username:
    <input type = "text" name = "username" class = "box">
</div>

<div>
    Password:
    <input type = "password" name = "password" class = "box">
</div>

<div class="error_msg" id="error_msg"><?php echo $error; ?></div>

<div class="buttons">
    <button class="white-btn" type="submit">Login</button>
    <button class="reset-btn" id="reset-btn" name="reset-btn" type="reset">Reset</button>
</div>

<div class="create_link">
    Don't have an account?
    <p><a href="create_account.html">Create one now!</a></p>
</div>

```

Figure 65: Login html

After user has clicked on the "LOGIN" button, the data is submitted to the php in the same page. In this Login page's php code, it is mainly to figure out whether the credentials entered matches the username and password of an existing user. First, the code must include the main connection to the database used, by using the include command. "conn.php" is the file that creates a connection to the database server. The php code states that if the request method of the server is "POST" which is used in the login form, it will then take the username and password inputted and store it in a variable with the \$_POST method, then use the variable in a MySQL query statement to find for the customer. If the query returns exactly 1 result that matches the input username and password, the username is stored into a session variable that will be used in other pages and the user will be linked to the Main Page "home.php". If no results are found, then an error message will pop up to tell the user that he/she has entered incorrect login details.

```
<?php
    include("conn.php");
    session_start();
    $error = "";

    if($_SERVER["REQUEST_METHOD"] == "POST") {
        // username and password sent from form

        $myusername = mysqli_real_escape_string($con,$_POST['username']);
        $mypassword = mysqli_real_escape_string($con,$_POST['password']);

        $sql = "SELECT Cus_ID FROM customer WHERE Name = '$myusername' and Password = '$mypassword'";
        $result = mysqli_query($con,$sql);
        $row = mysqli_fetch_array($result,MYSQLI_ASSOC);

        $count = mysqli_num_rows($result);

        // If result matched $myusername and $mypassword, table row must be 1 row

        if($count == 1) {
            $_SESSION['login_user'] = $myusername;

            header("location: home.php");
        }else {
            $error = "Your Login Name or Password is invalid";
        }
        if(isset($_POST['reset-btn'])) {
            $error = "";
        }
    }
?>
```

Figure 66: Login php

Logout

In every page after a user has logged into their account, there will be a logout button located at the top right corner. Once the user has clicked on the logout button, the php code starts running, and destroys the current session. It will then redirect the user back to login.php to login once again.

```
<?php
    session_start();

    if(session_destroy()) {
        header("Location: login.php");
    }
?>
```

Figure 67: logout php

4.1.6 Signup – HTML Form, PHP

Create Account

The Create Account or as known as Sign Up function collects data from the HTML form POST method input of a new user. All of the input fields of the form the user enters data in the html form and then clicks on the submit button which will not submit the form yet but will use JavaScript to check the input data. All input data fields are normal text fields whereas there will be 2 different checkboxes for users to check. One of the checkboxes is “subscribe to newsletter” which is optional for users, and another one “I agree to Terms and Conditions” which is required to be checked. All the other input boxes are required for the users to fill in following the format given, the username should be at least 4 characters to 14 characters long, the email must follow the email format that includes ‘@’ and ‘.com’, phone number should only contain numbers and length between 10 to 11 numbers, last but not least the password should be at least 4 characters or more.

Before submitting the form, all input will be checked through the JavaScript to make sure that all fields are entered accordingly. Then it will send the data to “insert.php”, which will then collect the data inputted and create a query out of it by using the \$_POST method collecting the input data from the form

```
<form action="insert.php" method="POST" id="create_form" name="create_form" onsubmit="return check_password()">
<section class="create_container">
    <h1>Create An Account</h1>
    <div>
        <p>Name: </p>
        <input type="text" name="username" minlength="4" maxlength="14" required="required" title="Username must be between 4 to 14 characters long." placeholder="Enter a cool username">
    </div>
    <div>
        <p>Email: </p>
        <input type="email" name="email" required="required" title="Enter your Email" placeholder="Enter Email">
    </div>
    <div>
        <p>Phone number: </p>
        <input type="tel" name="phonenumer" pattern="[0-9]{10,11}" required="required" title="Should contain only numbers and length between 10 to 11" placeholder="Enter Phone Number (xxx xxx xxxx)">
    </div>
    <div>
        <p>Shipping address: </p>
        <input type="text" name="shipping_address" required="required" title="Enter your Shipping address" placeholder="Enter Shipping address">
    </div>
    <div>
        <p>Password: </p>
        <input type="password" id="password" name="password" minlength="4" required="required" title="Set a password that is 4 characters or more" placeholder="Enter password">
    </div>
    <div>
        <p>Confirm password: </p>
        <input type="password" id="cfm_password" name="cfm_password" required="required" title="Confirm password" placeholder="Confirm your password">
    </div>
    <input type="checkbox" id="subscribe" name="subscribe" onclick="cb_changeValue()" value="0">
    <label>Subscribe to newsletter</label><br>
    <input type="checkbox" class="tnc_agree" id="tnc_agree" name="tnc_agree" required>
    <label> I Agree to Terms and Conditions</label><br>
    <div class="buttons">
        <button type="submit" class="submit-btn">Create Account</button>
        <button type="reset" class="reset-btn">Reset</button>
        <a href="home.php"><button type="button" class="white-btn">Cancel</button></a>
    </div>
</div>
</section>
</form>
```

Figure 68: Create account html

The query will be executed in the MySQL database and insert a new row that consists of the input data. If the data is successfully inserted in, it will then show a pop-up message “Details recorded!” and redirect the user to the login.php page and proceed to login, else it will show an error message.

```
$sql="INSERT INTO customer (Name, Password, Contact, Email, Subscription, Shipping_Address)
VALUES
('$_POST[username]', '$_POST[password]', '$_POST[phonenumerber]', '$_POST[email]', '$_POST[subscribe]', '$_POST[shipping_address]')";
```

Figure 69: Insert.php

\$check_existing is a query that runs after the user submitted the form. This PHP statement will collect the row that the username matches the username inputted by the user. By using mysqli_num_rows it will calculate the number of rows that is obtained from the query and store it in \$exist_rows variable. Next, an if else statement is used to determine what action will be carried out when the value stored in \$exist_rows is not 0 vice versa. If the value is not 0, a JavaScript will be echoed out with an alert message informing the user that the username is already taken and require them to choose another username, it will then redirect users back to create_account.html page; if \$exist_rows’s value is 0 then the query to insert new customer details will run with mysqli_query command and it will echo an alert message “Details recorded!” followed by redirecting the users to create_done.html page. Finally, the connection to database is closed by using mysqli_close(\$con) command.

```
$check_existing = mysqli_query($con,"SELECT * FROM customer WHERE Name = '$_POST[username]'");
$exist_rows = mysqli_num_rows($check_existing);

if (!$exist_rows == 0)
{
    echo '<script>alert("Username already taken, please choose another username.");
    window.location.href = "create_account.html";
    </script>';
}
else
{
    mysqli_query($con,$sql);
    echo '<script>alert("Details recorded!");
    window.location.href = "create_done.html";
    </script>';
}

mysqli_close($con);
```

Figure 70: Check existing user

4.1.7 Self-created CSS and JavaScript

script.js

This Self-created JavaScript file is responsible for checking the admin value from the home page. The id “adminvalue” is taken from an input with a hidden type and stored into the id “adminvalue” for the JavaScript to use and manipulate. An admin value of 1 means that the user is an admin and an admin value of 0 means that the user is not an admin.

If the adminvalue is “1” then the JavaScript file will get the id of “adminbutton” from the home.php file and style it to a block which means that the adminbutton will be visible on the home page. If the adminvalue is other than “1”, the adminbutton will display none which means that it will not be visible in the home.php page. This is done so that non-admin users will not see the admin button and cannot access admin functionalities.

The “login_button” id is also taken from the home.php page and manipulated in the JavaScript page. If the adminvalue is “0” which means a user has logged in or the adminvalue is “1” which means an admin has logged in, the display style of the login button will be none. This means that the login button will not display when a user has logged in. If the user is not logged in, the button will appear for the user to log in.

Lastly if the adminvalue is set to nothing, the JavaScript will get the element of the “logout_btn” id and set the display style to none. This means that the logout button will not show if there is no login. Else, if the adminvalue has a value which can be either a 0 for users or a 1 for admin, the logout button will have a style display of block. This means that the logout button will show if the user is logged in.

There are few functionalities that are created as well in the JavaScript file. One of the examples is the display_contact_edit function. This function will take the id “contact_edit_form” from the account.php page and change it to a block when the user was to click the edit contact icon. The values of the old contact will be displayed, and users are able to change their contact number.

```
// Admin button
var adminvalue = document.getElementById("adminvalue").value;

if (adminvalue == "1"){
    document.getElementById('adminbutton').style.display = "block";
} else{
    document.getElementById('adminbutton').style.display = "none";
};

// Login and Sign up button display
if (adminvalue == "0" || adminvalue == "1"){
    document.getElementById('login_button').style.display = "none";
}

// Logout button display
if (adminvalue == "") {
    document.getElementById('logout_btn').style.display = "none";
} else{
    document.getElementById('logout_btn').style.display = "block";
}
```

Figure 71:Check admin value

The cancelEdit_contact() function on the other hand uses the style display none property. This means that it will remove the content when the user were to click the cancel button. The function will get the id of “contact_edit_form” from the account.php file and display as none on click.

```
//account
function display_contact_edit(){
    document.getElementById("contact_edit_form").style.display = "block";
}
```

Figure 72:Display contact edit function

```
function cancelEdit_contact(){
    document.getElementById("contact_edit_form").style.display = "none";
}
```

Figure 73: cancelEdit_contact function

Contact:

012345

New value:

Figure 74: Cancel button on account.php

Create Account JavaScript

Another area that JavaScript was inserted in the assignment would be in the create_account.html. The function cb_changevalue() helps to change the value each time the checkbox is checked. A JavaScript code document.getElementById("subscribe").value is used to get the value of checkbox id "subscribe". For example, when the user registers an account and decides to subscribe to the newsletter, the user will tick the checkbox and the value will be changed to 1, else it will remain 0.

On submission of the form, a check_password() functionality will run to check if the user's password is same as their confirmation password. First it uses document.forms["create_form"]["password"].value to obtain the value input by user in the form for the password and do the same for password confirmation which has an id of "cfm_password". It then assigns these values into variables password1 and password2. Next, in order to remove whitespaces before and after the values the "trim()" function is used. If both passwords are the same, the JavaScript will return true and the form will be submitted, and the account of the user will be created. Else, the user will be alerted that their password is not the same and require them to reenter their password.

```
<script>
    function cb_changeValue()
    {
        if (document.getElementById("subscribe").value == 0){
            document.getElementById("subscribe").value = 1;
        }else{
            document.getElementById("subscribe").value = 0
        }
    }
    function check_password()
    {
        let password1 = document.forms["create_form"]["password"].value;
        let password2 = document.forms["create_form"]["cfm_password"].value;

        password1 = password1.trim();
        password2 = password2.trim();

        if (password1 == password2)
        {
            return true;
        }
        else
        {
            alert("Input password must be the same!!!");
            document.getElementById("password").value="";
            document.getElementById("cfm_password").value="";
            return false;
        }
    }
</script>
```

Figure 75: create_account.html script

Admin Dashboard – New Book

This New Book function allow admin to view the HTML form to insert book information by clicking the “NEW” button. A JavaScript function add_book() will be executed once the button is clicked. The getElementById() is used to manipulate the display style of the element according to the id attribute. Hence, the HTML form will be displayed.

```
<div class="admin-book-col">
  <form method="post">
    <button type="submit" class="gold-btn" name="new-book" onclick ="add_book()">NEW</button>
  </div>

  <!-------Javascript----->

<script>
  function add_book(){
    document.getElementById("new-book-form").style.display="block";
  }
</script>
```

Figure 76: New Book section of admin.php

Home Page - Header

This is the header of the home page. The minimum height and width of the header is set to cover the whole page. For the background image of header, a linear gradient is used to cover the background image. RGBA colors in the linear gradient is used to create smooth transitions between them. The background image is centered and resize to cover the whole header. The header is positioned relative to its current window (Easy Tutorials, 2021).

```
.header{ /* class need to have ". " */
  min-height: 100vh; /* vh = 1% of the height of the viewport */
  width: 100%;
  background-image: linear-gradient(rgba(4,9,30,0.7),rgba(4,9,30,0.7)),url(images/banner.png); /* linear-gradient = transition of 2 color from top to bottom */
  background-position: center;
  background-size: cover;
  position: relative; /* Other content will not be adjusted to fit into any gap left by the element. */
}
```

Figure 77: Header section of style.css

Navigation Bar

Figure below shows the navigation bar for all pages. First, the navigation bar is set to a fixed height so it won't exceed a certain height, there will also be paddings to space the content towards the middle. With the display flex and the flex-wrap, it can ensure that the navigation bar will not overflow, and it will go to the second line. The search logo's font size is increased slightly to 30 pixels and its font color is changed to white so that it is more visible. Moreover, all the elements in the navigation bar except for the logout button have a golden line that expands for 70% of the maximum width under the link when the cursor is pointing at it. The letter spacing is set to 1 pixel to make it easier to read the links. Moreover, the logos in the navigation bar are set to a bigger font size for better visibility.

```
nav{  
    display: flex;  
    height: 120px;  
    width: 100%;  
    align-items: center;  
    justify-content: space-between;  
    padding: 0 50px 0 100px;  
    flex-wrap: wrap;  
}  
.header nav .search_logo {  
    font-size: 30px;  
}  
.header nav .search_logo a{  
    color: ■#ffff;  
}  
nav ul{  
    display: flex;  
    list-style: none;  
}  
nav ul li:not(.Logout){  
    margin: 0 5px; /*put more space between navlinks */  
}  
nav ul li:not(.Logout) a{  
    color: ■#ffff;  
    text-decoration: none; /*removes the annoying line below*/  
    font-size: 15px;  
    font-weight: 600;  
    padding: 8px 15px;  
    letter-spacing: 1px;  
    transition: all 0.5s ease;  
}  
nav ul li:not(.Logout)::after{ /* inserts something after the content of each selected elements */  
    content: ''; /*to generate line effect*/  
    width: 0%; /*hide the line when no hover*/  
    height: 2px;  
    background: ■#FFD700;  
    display: block;  
    margin: auto; /* horizontally center the element within its container */  
    transition: 0.5s; /* transition time*/  
}  
nav ul li:not(.Logout):hover::after{  
    width: 70%; /* show line when hover */  
}  
nav ul li .fas, .fab{  
    font-size: 20px;  
}
```

Figure 78: Navigation Bar CSS part I

The logout button in the navigation bar is similar to the guest's login and sign-up button. The border is set to red with a thickness of 2 pixels and solid style. Padding is given to let the button look wider and have more space for the content inside. Background is set to transparent to keep it consistent and when hovered above it the cursor will turn to pointer. The contents inside are set to the default font color which is white and with bolder and bigger font, as well as letter-spacing and transition to maintain the consistency. Once the logout button is hovered, the background turns red in 0.5 seconds.

```
nav ul li.logout {  
    border: 2px solid #red;  
    padding: 8px 16px;  
    background: transparent;  
    cursor: pointer;  
}  
nav ul li.logout a{  
    color: #fff;  
    text-decoration: none;  
    font-size: 15px;  
    font-weight: 600;  
    letter-spacing: 1px;  
    transition: 0.5s;  
}  
nav ul li.logout:hover{  
    background: #red;  
    transition: 0.5s;  
}  
nav ul li.logout a:hover{  
    transition: all 0.5s ease;  
}
```

Figure 79: Navigation Bar CSS part 2

Footer

For the footer, flex-basis is used to make each column 31% of the screen width. Furthermore, border-radius is used to make the column has round corners. When a column is hovered, there will be a box-shadow around within 0.5 seconds of transition (Easy Tutorials, 2021).

```
.footer {
  width: 100%;
  text-align: center;
  padding: 200px 50px 100px 50px;
}

.footer-col{
  flex-basis: 31%; /* make each box 31% on screen*/
  background: #A5633633;
  border-radius: 10px; /* round corner */
  margin-bottom: 5%;
  padding: 20px 12px;
  transition: 0.5s;
}

.footer .fas{
  color: #7f5539;
  font-size: 40px;
}

.footer h3{
  text-align: center;
  font-weight: 600;
  margin:10px 0;
}

.footer-col:hover{
  box-shadow: 0 0 20px 0px rgba(0,0,0,0.2); /* (left-right) (up-down)
  (blur intensity) (spread) (color)*/
}
```

Figure 80: Footer section of style.css

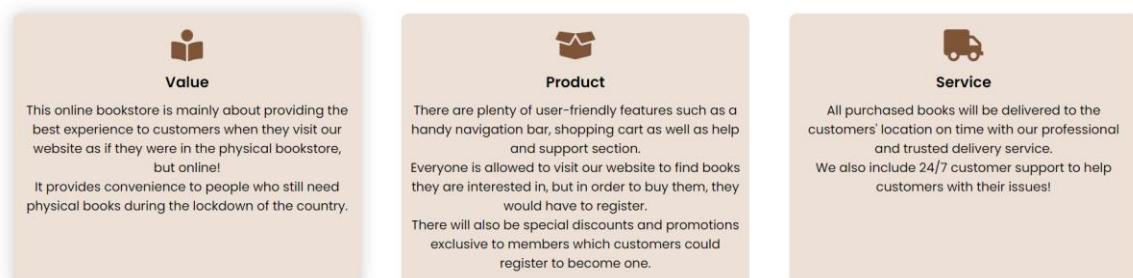


Figure 81: Footer section

Feedback – Feedback Form

In the text area of the feedback form, the background color will change according to the value entered or not. Furthermore, the font color of placeholder inside the text area has been specified. For the star ratings radio buttons, there will be shadows when the star is not selected. When the stars are hovered, the cursor will change into pointer. If a star is selected, the star will turn into gold color with a transition.

```
.feedback-form textarea:valid{
    background-color: #fec89a;
}
.feedback-form textarea:invalid{
    background-color: #ffadad99;
}
.feedback-form textarea::placeholder{
    color: #fff; /* placeholder = word inside the field */
}
.feedback-form h4{
    color: #fec89a;
}
.rating input{
    font-size: 30px;
    text-shadow: 1px 1px 0 #fec89a;
    cursor: pointer;
    width: 35px;
}
.rating input:checked{ /* check for radio button */
    color: #ffd700;
    text-shadow: 1px 1px 0 #1b1b1b;
    transition: 0.5s;
}
```

Figure 82: Feedback section of style.css

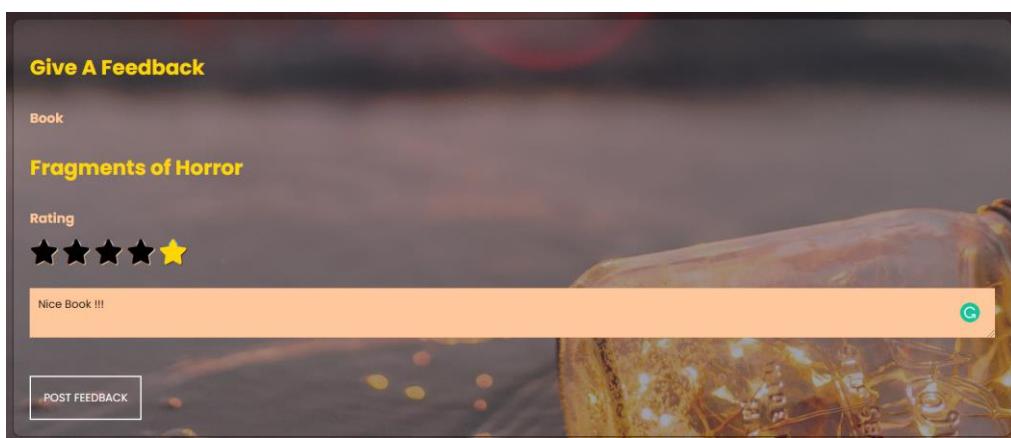


Figure 83: Feedback form of feedback.php

Gold Button

The gold button will have a white border and text initially. The background of the button will be transparent at first and transition to gold color after the button is hovered. The border color and text color will also change at the same time (Easy Tutorials, 2021).

```
.gold-btn{  
    margin: 10px 0;  
    color: #fff;  
    border: 2px solid #fff;  
    padding: 15px;  
    background: transparent;  
    cursor: pointer; /* point logo (index finger out) */  
}  
.gold-btn:hover{  
    border: 2px solid #ffd700;  
    background: #ffd700;  
    transition: 0.5s;  
    color: #1b1b1b;  
}
```

Figure 84: Feedback button of style.css

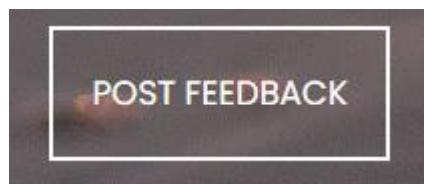


Figure 85: Feedback button

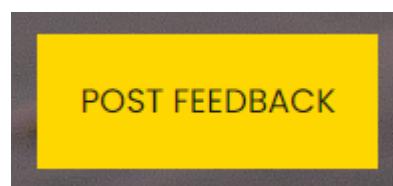


Figure 86: Hovered Feedback button

Product – Book Details

The display property of the elements in book details is set to flex. Then the elements are centered in the box with justify-content. All the boxes are separated by using the margin-bottom. The box round corners are using the border-radius.

```
.detail-box{  
    width: 50%;  
    margin: auto;  
    border: 1px solid #023E8A;  
    background-color: #03045eE6;  
    margin-bottom: 20px;  
    border-radius: 30px;  
}  
.detail-box h5, .detail-box h6, .detail-box .fas, .detail-box .fab{  
    display: flex;  
    justify-content: center;  
    color: #caf0f8;  
    font-size: 15px;  
}
```

Figure 87: Book Details section of product.php



Figure 88: Book Details section

Admin – Report

The arrangement of the admin three report section is done with the display flex property. The property of justify-content is set to space-around which all columns will have before, between, and after the boxes.

```
.admin-report{  
    width: 80%;  
    display: flex;  
    justify-content: space-around;  
    padding-bottom: 100px;  
}
```

Figure 89: Admin Report Section of style.css

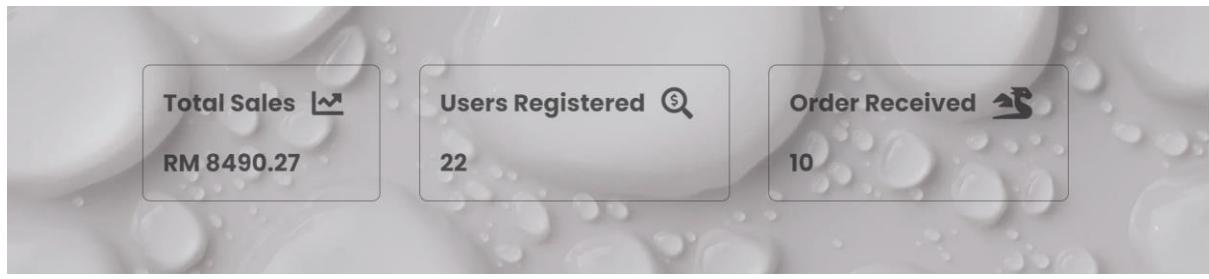


Figure 90: Admin Report Section of admin.php

5.0 Main Section (User Guidance & Sample Screen)

5.1 Guest Accessible Page

Guest Header Homepage

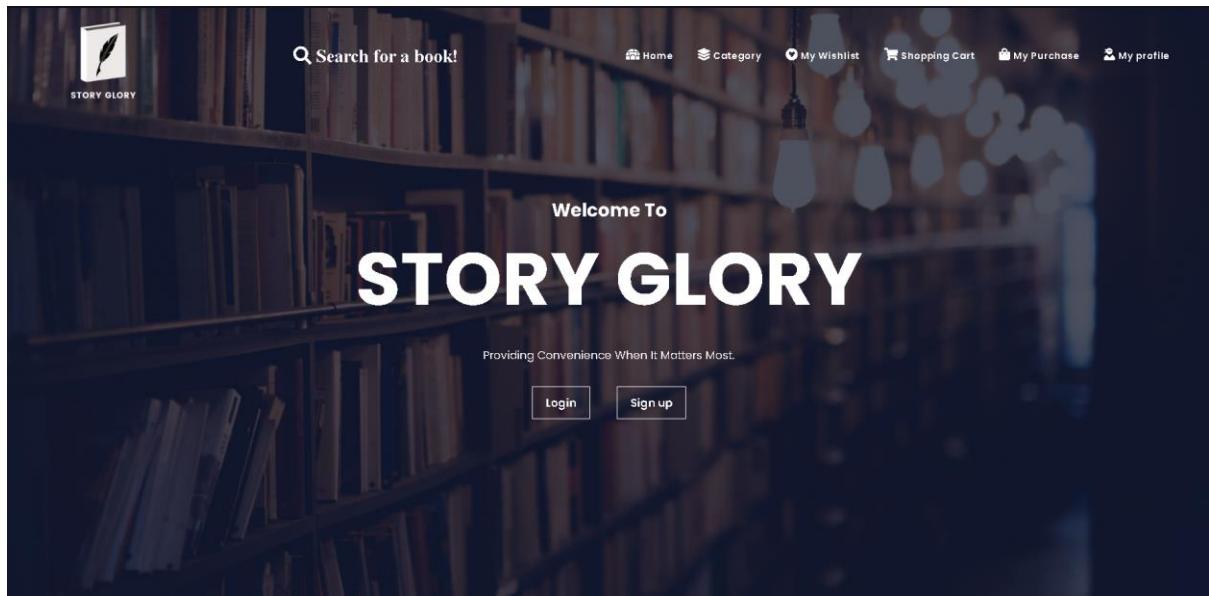


Figure 91: Header Homepage

This is the home page that the guest will be greeted during the first visit to the Story Glory website. A header section is present, and guest can navigate to any of the icons, and it will direct them to respective pages. The header consists of a search for book function, a home button, and category that guest can access. However, the My Wishlist, shopping cart, my purchase and my profile buttons will direct the guest to the login page because these functions require an account to access. Guest can also login by clicking on the login button below the Story Glory text. If guest do not have an account, guest can create one by clicking on the sign-up button.

Body Homepage



Figure 92:Body homepage

In the body section of the homepage, guest can view the most recent books published on the website. However, for guest to view more information about the product, guest would have to login to an account.

Footer Homepage

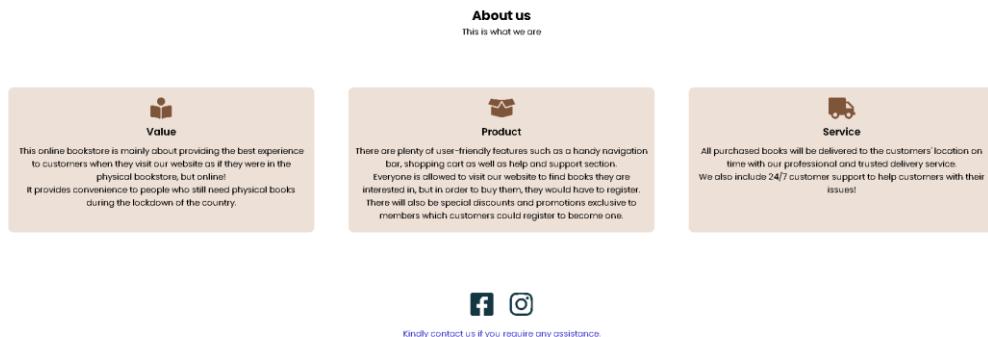


Figure 93: Footer Homepage

The footer section is where the guest will learn a little about the website. The value, product, and service of the website will be displayed as well as their socials. A Facebook and Instagram icon is displayed at the bottom and if guest have any queries, guest can message the respective pages to get their question answered.

Social Media



Figure 94: Story Glory's Facebook

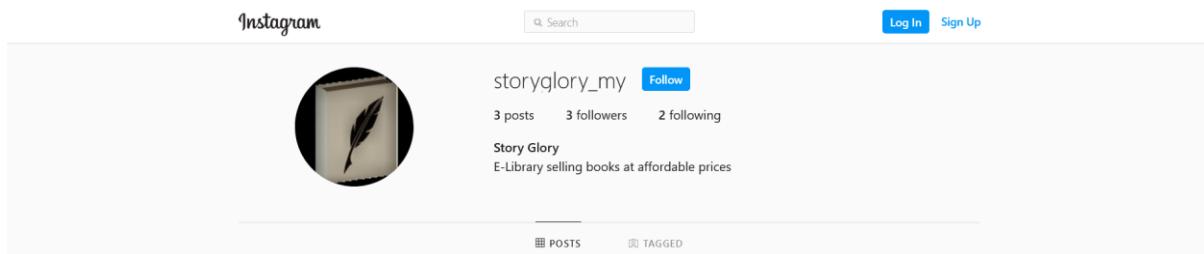


Figure 95: Story Glory's Instagram

These are the social medias of Story Glory which users can approach them through the social media icon in the footer section. Users can contact these social media if any assistance is needed, or any issues occurred.

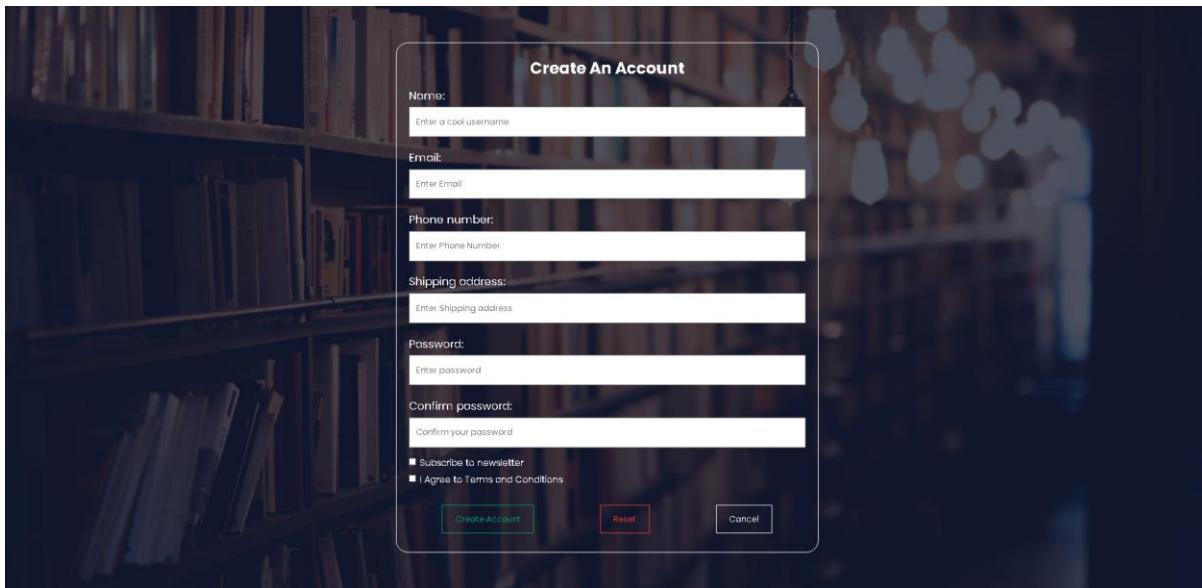
Category page

The screenshot shows a dark-themed web application for a bookstore. At the top, there's a navigation bar with links for Home, Category, My Wishlist, Shopping Cart, My Purchase, and My profile. On the left, a sidebar lists genres: Academic, Motivational, Fantasy, Romance, Horror, Sci-fi, History, and Kids. Below this is a 'Reset Genre' button. The main content area features a search bar with placeholder text 'Enter a book's name' and a magnifying glass icon. Underneath the search bar are buttons for 'Reset Search Value' and 'Sorted by: ||'. A section titled 'Genres' shows 'Showing 1 - 15 of 50 results'. To the right of this are sorting options: 'Sort by: A-Z | Price: low to high | Price: high to low | Most recent | Reset Sorting |'. The main area displays a grid of 15 book covers, each with its title and author. The books include titles like 'IT ENDS WITH US' by Colleen Hoover, 'SCOUT' by Julia Galef, 'ENCOURAGING WORDS FROM JOHN CENA', 'The Witches' by Lemony Snicket, 'ACTING WITH POWER' by Deborah Greenfield, 'THE END OF HER' by Shari Lapena, 'ARTEMIS' by Andy Weir, 'THE SECRET LANGUAGE OF YOUR BODY' by Inna Segal, 'Sapiens: A Brief History of Humankind' by Yuval Noah Harari, 'BRAIN GAMES clever kids' by Susanna Clarke, 'PIRANESI', 'NAT'S WHAT I RECKON' by Liane Moriarty, 'NINE PERFECT STRANGERS', and 'BEAUTIFUL WORLD, WHERE ARE YOU' by Sally Rooney. Navigation arrows at the bottom indicate the user can scroll through more books.

Figure 96: Category Page

Guest can access this category page and search for a book name by inputting the name of the book and clicking on the search icon. There is a reset search value button below to remove the search input that the guest has entered. At the left navigation bar of the category, the section consists of the genres that the guest can choose from. Some of the genres consist of academic, motivational, fantasy, and many more. Besides the sort by, there are hyperlinks for users to sort their books as well. Guest can sort the book by price, most recent, and reset the sorting of the book. At most, 15 books will be displayed at a time, and guests can click the arrow button to navigate through the books. Unfortunately, guests will not be able to view book details. When a guest clicks on a book, the guest will be directed to the login page.

Create Account



The image shows a 'Create An Account' form window centered over a dark, slightly blurred background of bookshelves. The form has a white header 'Create An Account'. It contains six input fields with placeholder text: 'Name' (Enter a cool username), 'Email' (Enter Email), 'Phone number' (Enter Phone Number), 'Shipping address' (Enter Shipping address), 'Password' (Enter password), and 'Confirm password' (Confirm your password). Below these fields are two checkboxes: 'Subscribe to newsletter' and 'I Agree to Terms and Conditions'. At the bottom are three buttons: 'Create Account' (highlighted with a green border), 'Reset' (with a red border), and 'Cancel'.

Figure 97: Create Account

Guest can create their account from the create account page. To navigate to this page, guests can click on the “Sign Up” button on the home page or navigate to this page through the hyperlink on the login page. To create an account, the guests would need to enter their name, email, phone number, shipping address, password, and confirmation password. The input that the guest enter must be in the proper format. For example, if the password does not match the confirmation password, the guest will not be able to create their account.

Here is the list of required input formats that guests will have to follow:

1. The username must be between 4 to 14 characters long
2. Email must be according to format (must include @ and .com)
3. Phone number should only contain numbers and should be 10 to 11 numbers long
4. Password should be at least four characters long

Guest can choose whether to subscribe to the newsletter of the website or not by checking the checkbox. However, the terms and conditions checkbox must be checked by the guest to create their account. If the guest did not check the agree to terms and condition check box, a message would alert the guest, telling the guest that the terms and conditions need to be checked. Once all the credentials of the guest are correct, the guest can click on the create account button to create their account. If the input is wrong, the guest can reset it by clicking on the reset button. If the guest wants to exit out of the register page, the cancel button is available for the guest to

be directed to the homepage of the website. Once account creation is done, guests will be redirected to another page.

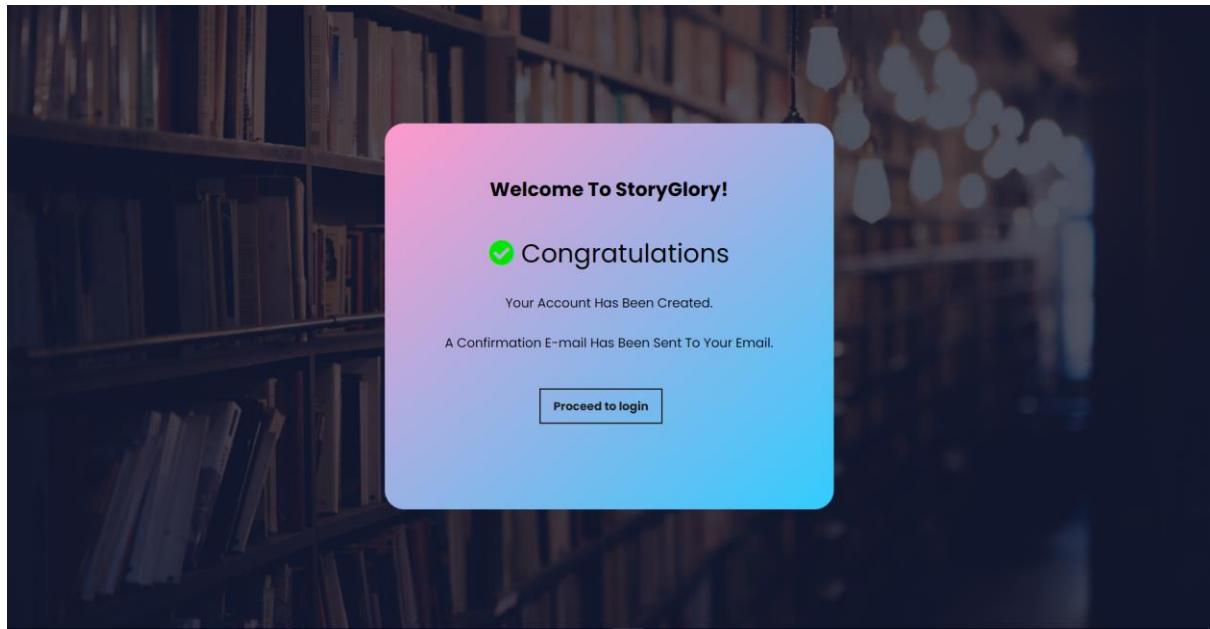


Figure 98: Create Done

This page informs the user that account creation is successful and that users can proceed to login once clicked on the “Proceed to Login” button.

Login Page

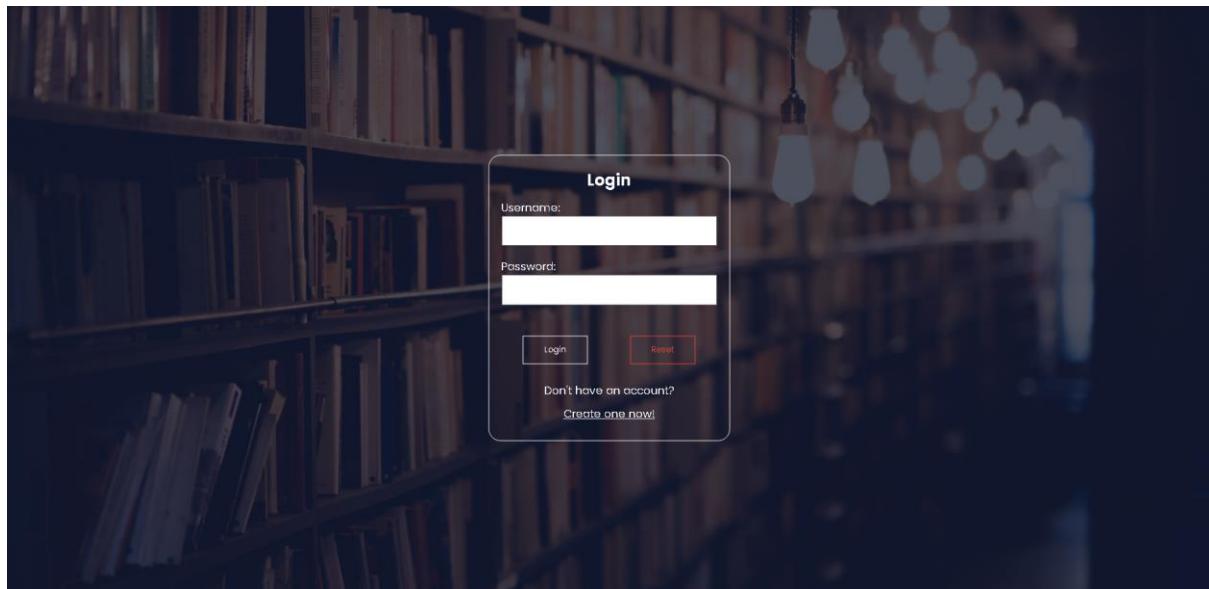


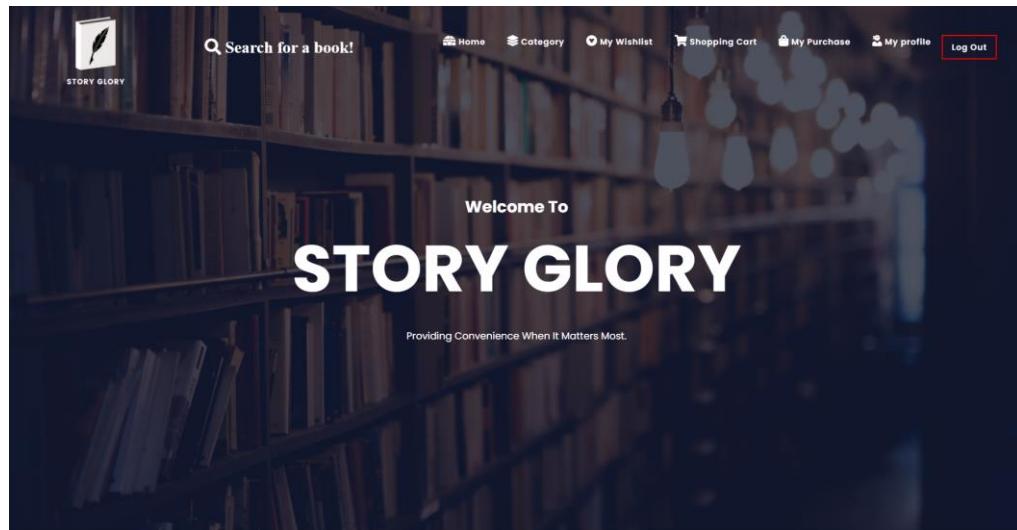
Figure 99: Login Page

The user can login to their account by entering the credentials of their registered account in the respective input boxes. Users are required to enter their username and password. If the username and password match with an account in the Story Glory database, the user can click on the login button to login and be directed to the homepage of the website. To clear the input that the user has keyed in, the users can click on the reset button. If the user does not have an account, users can register one by clicking on the create account hyperlink below.

5.2 Customer Accessible Page

Once the guest has successfully logged in to their account, guests are considered as customer. The first page that will greet the customer is the homepage.

Customer Homepage




Value

This online bookstore is mainly about providing the best experience to customers when they visit our website as if they were in the physical bookstore, but online! It provides convenience to people who still need physical books during the lockdown of the country.


Product

There are plenty of user-friendly features such as a handy navigation bar, shopping cart as well as help and support section. Everyone is allowed to visit our website to find books they are interested in, but in order to buy them, they would have to register. There will also be special discounts and promotions exclusive to members which customers could register to become one.


Service

All purchased books will be delivered to the customers' location on time with our professional and trusted delivery service. We also include 24/7 customer support to help customers with their issues!

[Facebook](#) [Instagram](#)
Kindly contact us if you require any assistance.

Figure 100: Customer Homepage

On every page, there will always be a header that consists of different navigational links, as well as a footer that includes the about us section that has information regarding the company, followed by links to the company's social media.

Similar to the guest accessible pages, the header includes navigational links that link to the Homepage, Category page, My Wishlist page, Shopping Cart page, My Purchase page, My profile page, and a Log out button. The difference of the navigational bar compared to the guest version is that every single page can be accessed, and there is an extra log out button at the right; while the guest can only access the homepage and the Category page, the other pages require the guests to log in as a customer to access it.

The main body of the homepage would be the most recent published books section. It shows the customer five of the most recently published books. Customers can click on the books that pique their interest, and it will link them to the product page, where it will show the details of the book.

Finally, the footer is the part where the about us section is located. It includes the value, product, and service that the company offers, also with a brief description in each of the boxes. Below the about us are icons that link to the company's social media, including Facebook and Instagram.

Customer Category page

Genres

Showing 1 - 15 of 50 results

Sort by: A-Z | Price: low to high | Price: high to low | Most recent | Reset Sorting |

Genre	Book Title	Author
Academic	IT ENDS WITH US	COLLEEN HOOVER
Motivational	THE SCOUT MINDSET	JULIA GALEF
Fantasy	JOHN CENA'S DO YOUR BEST, EVERY DAY, TO DO YOUR BEST, EVERY DAY	JOHN CENA
Romance	THE WITCHES	ROALD DAHL
Horror	ACTING WITH POWER	DEBORAH GREENFIELD
Sci-fi	SHARI LAPENA	SHARI LAPENA
History	THE END OF HER	SHARI LAPENA
Kids	THE MARTIAN	ANDY WEIR
Reset Genre	ARTEMIS	ANDY WEIR
	THE SECRET LANGUAGE OF YOUR BODY	INNA SEGAL
	SAPIENS	YUVAL NOAH HARARI
	BRAIN GAMES FOR KIDS	SUSANNA CLARKE
	PIRANESI	SUSANNA CLARKE
	NAT'S WHAT I RECKON	LIANE MORIARTY
	NINE PERFECT STRANGERS	LIANE MORIARTY
	BEAUTIFUL WORLD, WHERE ARE YOU	SALLY ROONEY

Figure 101: Customer Category page

In the customer category page, it's just like the guest category page. Customers may enter the book's title to search for it. Customers may also select a particular genre and sort by either Alphabetical order, price from low to high, price from high to low and by the most recent published books. These sorting and filters may be reset manually by clicking on a simple button each located under the search bar, genre list and sort by list.

When a genre, sort by or book title is filtered then the books will be shown as according to the filter. Customers could also navigate through the next 15 books by clicking on the small arrow located at the bottom of the last row of books. By clicking on a book that the customer is interested, it will then redirect the customer to the product details page.

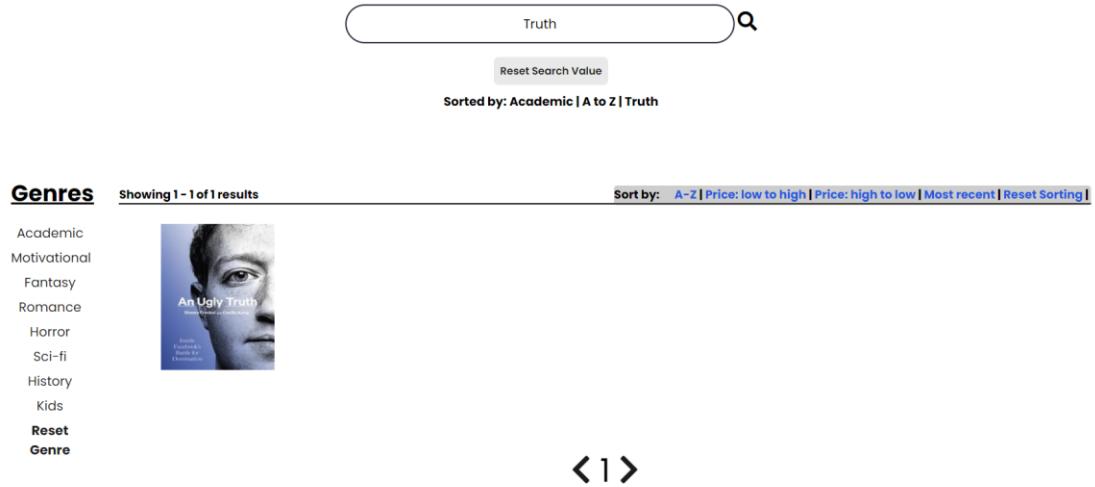


Figure 102: Filtered category page

Figure above shows an example of the category page that is filtered with “Academic” genre, sorted by Alphabetical order and filtered with the Book title that includes “Truth”.

Product Page

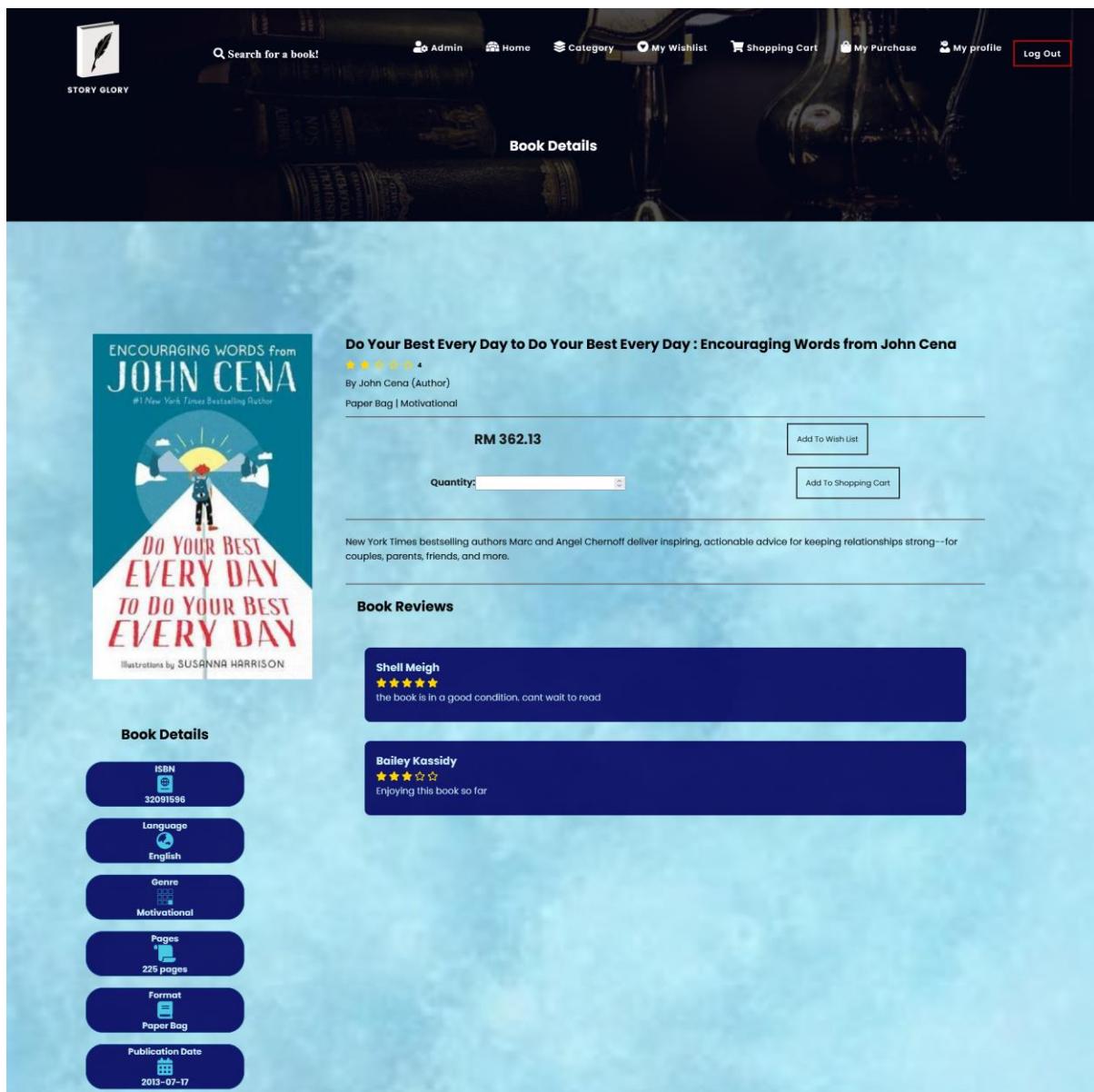


Figure 103: Product Details

The main body of the product page includes a big book cover at the left, the book title, overall ratings, book author, book format, price of the book, description or summary about the book, the ISBN of the book, general language of the book, its genre, number of pages, as well as publication date. All of the info about the book is listed down here in the product page. If interested in buying either now or next time, customers may enter the quantity desired to purchase and click on the “Add to Wish List” or “Add to Shopping Cart”.

If the customer decides to add the book to Wishlist or Shopping Cart, it will notify the customer that the book has been added to the Wishlist or the Shopping Cart.

My Wishlist

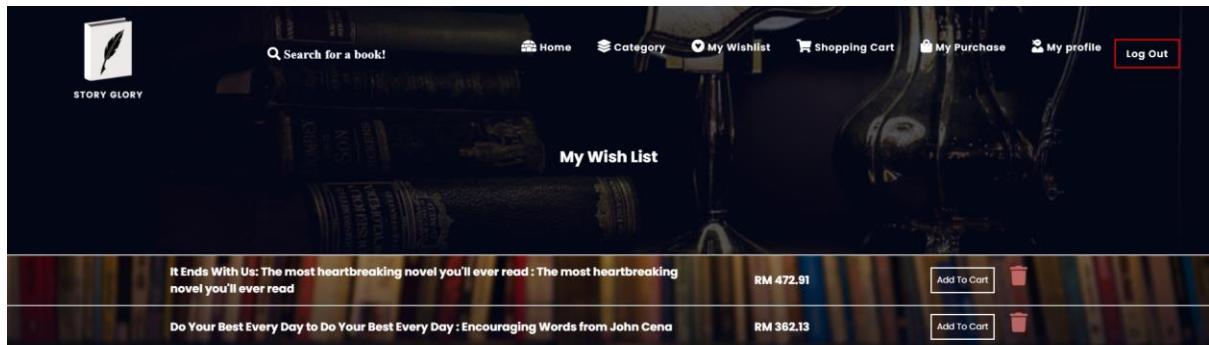


Figure 104: Wishlist page

In the My Wishlist page, header is same as any other page accessible by the customer. The main body of the My Wishlist page consists of the books that the customer has added to their Wishlist. It includes the book titles, book price, an “Add to cart” button to add the book to the shopping cart if the customer is interested in buying the book, and also a trashcan icon to remove the book from the customer’s Wishlist if the customer is not interested in the book anymore.

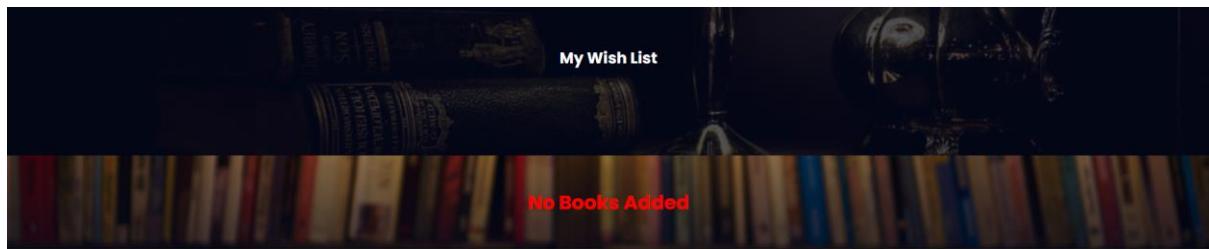
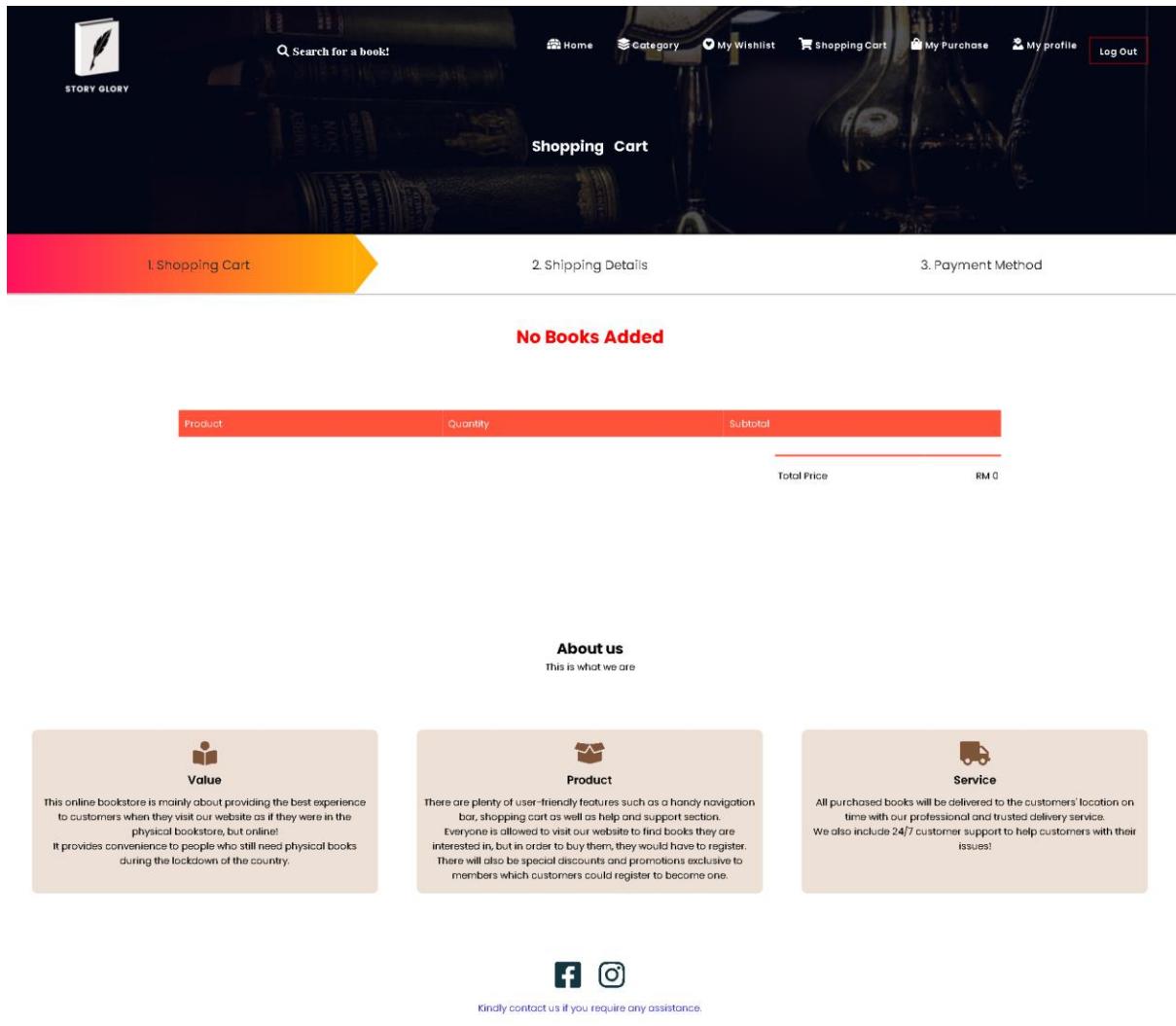


Figure 105: My Wishlist (Empty)

Figure above shows the My Wishlist page without books. A message “No Books Added” will show up in the middle of the screen.

Shopping Cart (empty)



The screenshot shows the Story Glory website's shopping cart page. At the top, there is a navigation bar with links for Home, Category, My Wishlist, Shopping Cart, My Purchase, My profile, and Log Out. The main header features a dark background with book images and the text "STORY GLORY". Below the header, a large red arrow points from left to right, containing the text "1. Shopping Cart", "2. Shipping Details", and "3. Payment Method". The main content area displays a message "No Books Added" in red. Below this, there is a table with columns for Product, Quantity, Subtotal, Total Price, and a red RM 0. Underneath the table, there is a section titled "About us" with three sub-sections: "Value", "Product", and "Service". Each sub-section contains text and icons.

Product	Quantity	Subtotal
		RM 0

About us

This is what we are

This online bookstore is mainly about providing the best experience to customers when they visit our website as if they were in the physical bookstore, but online! It provides convenience to people who still need physical books during the lockdown of the country.

There are plenty of user-friendly features such as a handy navigation bar, shopping cart as well as help and support section. Everyone is allowed to visit our website to find books they are interested in, but in order to buy them, they would have to register. There will also be special discounts and promotions exclusive to members which customers could register to become one.

All purchased books will be delivered to the customers' location on time with our professional and trusted delivery service. We also include 24/7 customer support to help customers with their issues!

Kindly contact us if you require any assistance.

Figure 106: Shopping Cart (empty)

The shopping cart page can be accessed by the customer when the customer click on the shopping cart icon at the navigation bar which can be found at the header section. If no books are added, a text will be displayed telling the user that no books are added.

Page | 101

Shopping Cart

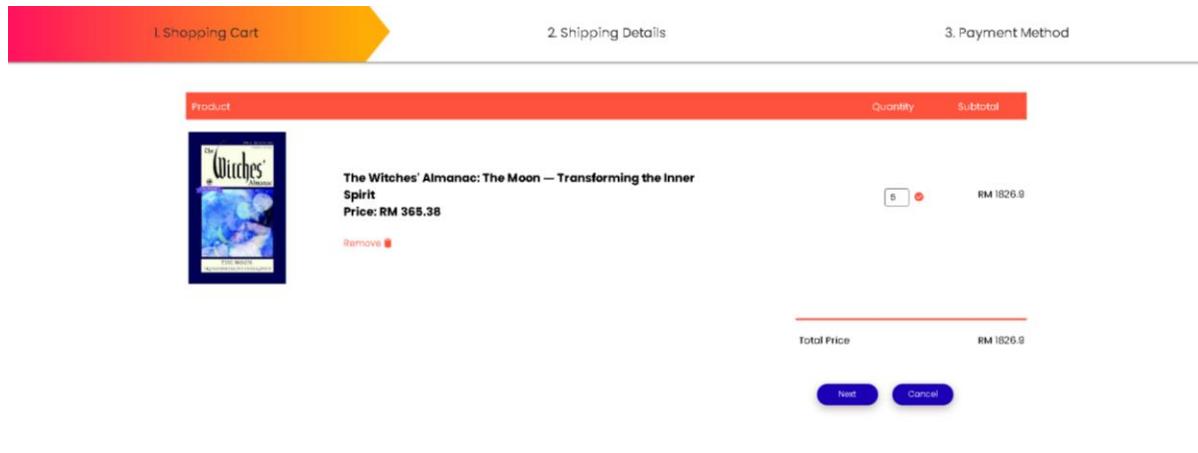


Figure 107: Shopping Cart

If customers have added books into their shopping cart, customers can access the shopping cart page and view the books that has been added into their shopping cart. In the figure, it shows that the user has added one book with a quantity of five titled “The Witches’ Almanac: The Moon — Transforming the Inner Spirit”. The subtotal of the book can be viewed as well as the total price. Users can also remove the book by clicking on the remove button to remove the book from their shopping cart. There are two buttons on this webpage which are the next and cancel. If the user does not want to proceed further, users can press cancel to return to the homepage. If users want to procced to the next step of payment, users can press the next button and the users will be directed to the shipping details page.

Shipping Details

The screenshot shows a web form titled "Shipping Details". At the top, there is a navigation bar with three items: "1. Shopping Cart" (red), "2. Shipping Details" (orange, which is the current active step), and "3. Payment Method" (yellow). Below the title, there are input fields for "Full Name", "Phone Number", "Address", "Zip/Postal Code", and "City". To the right, a summary table displays the "Cart" contents with a subtotal of RM1826.9, shipping at RM10.00, and a total of RM1836.9. At the bottom, there are two buttons: "Next" and "Cancel".

Figure 108:Shipping Details

The shipping details is where users can input their shipping details such as their full name, phone number, address zip/Postal code, city, as well as choose the type of delivery method for their item. The details of their purchase will be displayed at the right navigation bar such as the subtotal, shipping, and total price. When users are satisfied with their input, users can proceed to the payment method by clicking the next button. Else, users can click cancel to be directed to the previous page which is the shopping cart page.

Payment Method

The screenshot shows a payment method page with three tabs at the top: "1. Shopping Cart" (red background), "2. Shipping Details" (orange background), and "3. Payment Method" (yellow background). The "3. Payment Method" tab is active.

Billing Address:

- Full Name: [Input field]
- Email: [Input field]
- Address: [Input field]
- City: [Input field] (Kuala Lumpur)
- State: [Input field] (Selangor)
- Zip: [Input field] (47000)

Payment:

- Accepted Cards: [Image showing logos for Visa, MasterCard, American Express, and Discover]
- Name on Card: [Input field]
- Credit card number: [Input field] (111-2222-3333-4444)
- Exp Month: [Input field] (September)
- Exp Year: [Input field] (2021)
- CVV: [Input field] (372)

Cart:

Item	Quantity
The Witches' Almanac: The Moon — Transforming the Inner Spirit	1
Total	RM 36.9

Shipping address same as billing

[Continue to checkout](#)

Figure 109: Payment Method

In the payment method page, users must enter their full name, email, address, city, state, zip code, payment details to proceed to payment. Same as the previous page, the information of the user's purchase will be shown at the right navigational bar such as the title of the book, subtotal, and total price of the book as well as the quantity of the book. After all the information of the users are keyed in, the users can make their payment by clicking on the continue to checkout button. Once the continue to checkout page has been made, users will be directed to the My Purchase page to view their purchase and the delivery status of it.

My Purchase

My Purchase

Elihu Medway

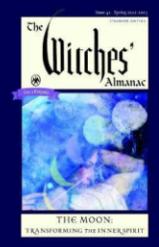
Delivery Status
Returned

	MEGAMONSTER	x 1	98.79	Feedback
	Death to Jar Sauce : Rad Recipes for Champions	x 5	470.75	Feedback

Order Total: 569.54

[Received](#) [Cancel Order](#)

Delivery Status

	The Witches' Almanac: The Moon — Transforming the Inner Spirit	x 5	1826.9	Feedback
---	--	-----	--------	--------------------------

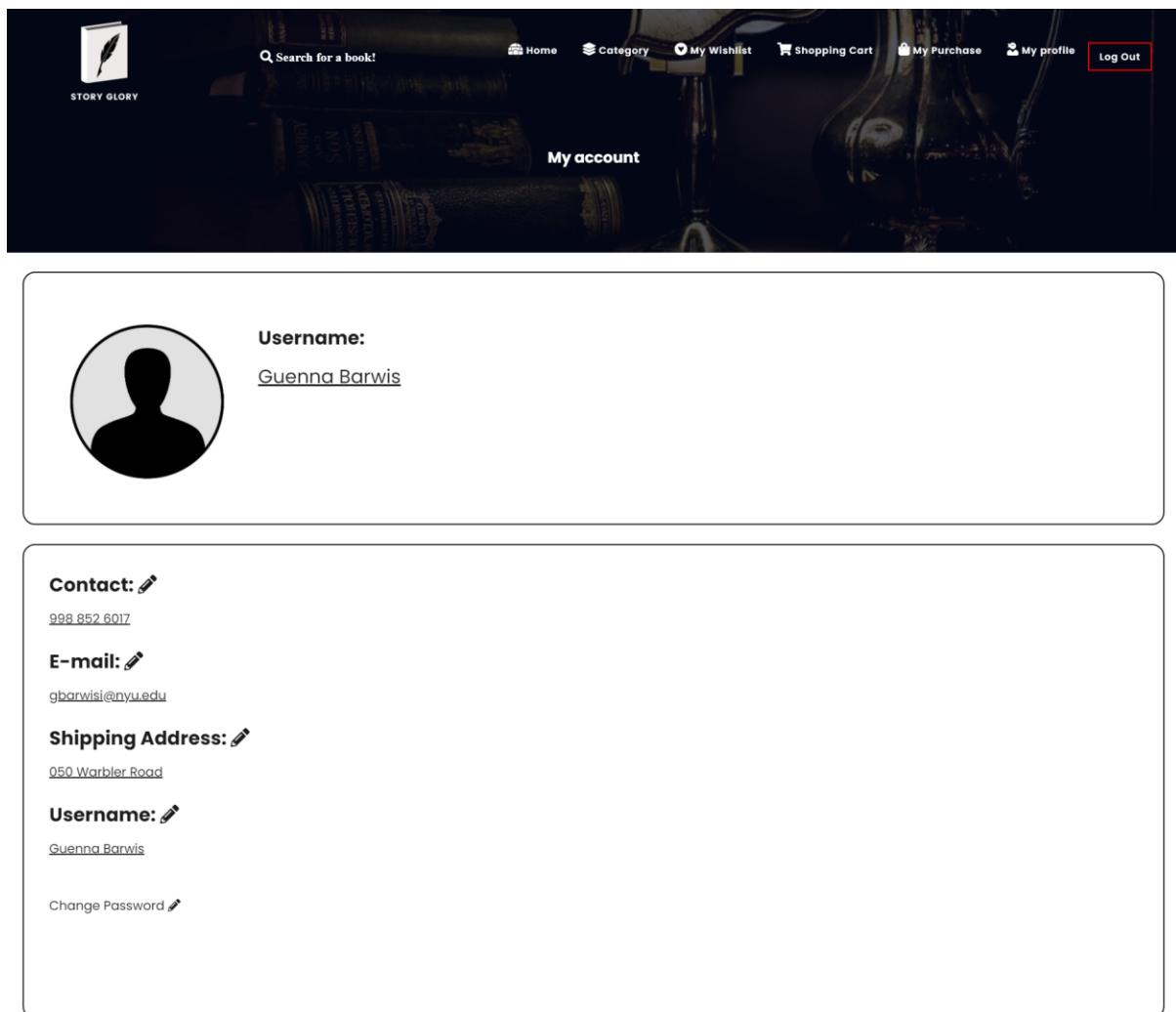
Order Total: 1836.9

[Received](#) [Cancel Order](#)

Figure 110: My Purchase

My purchase page will show all the details of the books that the user has purchased. If the user did not purchase any book, then a text telling the user that no books has been purchased will show up. The details of the books such as the title, quantity, and the subtotal as well as the grand total will be shown. The user can keep track of the delivery status of the book at the top right corner of the screen. Users can provide their feedback to a particular book after by clicking on the feedback icon as well. The receive button is to notify admin that the users has receive their product whereas the cancel order button is for the user to cancel or refund an order that has been made.

My Profile Page



The screenshot shows the 'My Profile' page of a book store website. At the top, there's a navigation bar with links for Home, Category, My Wishlist, Shopping Cart, My Purchase, My profile, and Log Out. A search bar says 'Search for a book!'. Below the navigation is a banner with the text 'My account' and several book covers. The main content area has two sections: one for user profile information (username, contact, email, shipping address) and another for password management.

Section	Details
User Profile	Username: <u>Guenna Barwis</u>
Contact	998 852 6017
E-mail	gbarwis@nyu.edu
Shipping Address	050 Warbler Road
Username	Guenna Barwis
Change Password 	

Figure 111: Customer Profile page

Besides that, customers can also view their own profile by clicking on “My Profile” in the navigation bar. In the main body of the My Profile page, the first part shows the customer’s user profile image and their username. The bottom part includes other details of the customer, including the customer’s contact info, email address, shipping address, username, as well as the password. All of the customer details are editable once clicked on the pencil icon.

New value:

Figure 112: Customer Profile Edit Box

After clicking on the pencil icon an input box will pop up. Customers may edit or change their details by entering a new detail in the input box and click on done. This will tell the system to

update the respective detail in the database. If the customer decides not to change the customer can click on the “Cancel” button.

5.3 Admin Accessible Page

Home Page (Admin)

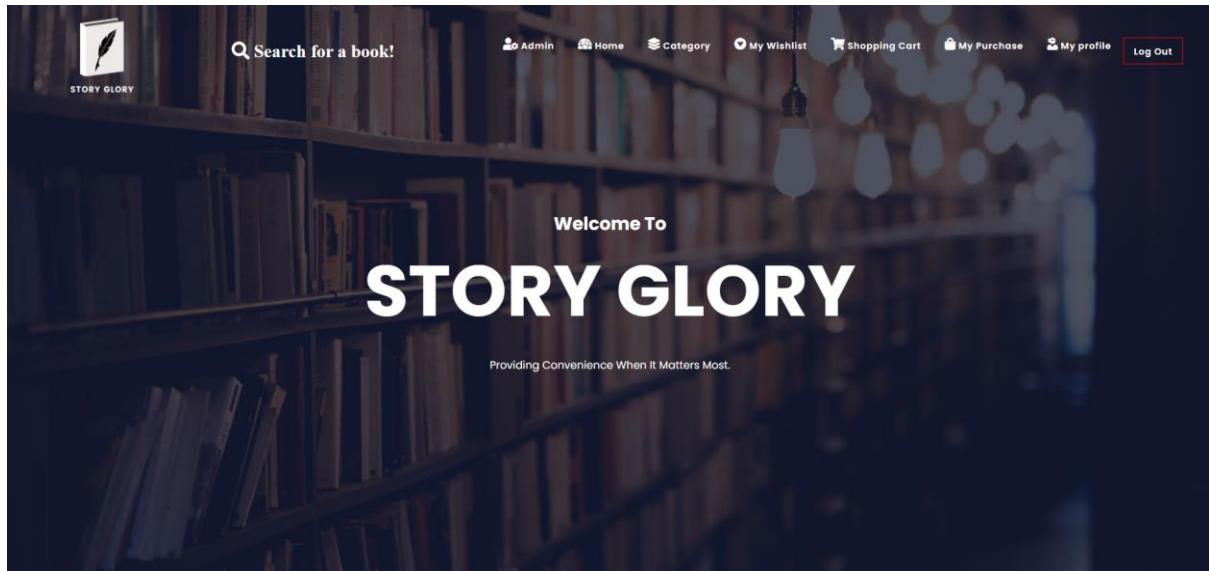


Figure 113: Home Page (Admin)

This is the home page admin will see when admin login. The admin dashboard will be available in the navigation bar. To access admin function, admin have to click on the “Admin” on the navigation bar. The admin dashboard will not be visible to customers and guests.

Admin Dashboard

The screenshot displays the Admin Dashboard of a book store website. At the top, there's a header bar with a logo, a search bar, and navigation links for Admin, Home, Category, My Wishlist, Shopping Cart, My Purchase, My profile, and Log Out. The main content area has a dark background with a water droplet pattern.

- Books:** This section shows statistics: Total Sales RM 8490.27, Users Registered 24, and Order Received 10. It includes a search bar and buttons for SEARCH, SHOW ALL, UPDATE, and DELETE, along with a NEW button.
- Delivery Status:** This section also features a search bar and buttons for SEARCH, SHOW ALL.
- Set Admin:** This section includes a search bar and buttons for SEARCH, SHOW ALL.

This section is titled "About us" and describes the website's purpose and services.

- Value:** This card states that the online bookstore provides a better experience than physical bookstores during lockdowns.
- Product:** This card highlights user-friendly features like a handy navigation bar and a shopping cart.
- Service:** This card emphasizes prompt delivery and 24/7 customer support.



Kindly contact us if you require any assistance.

Figure 114: Admin Dashboard

This is the whole admin dashboard page. The header and footer will be similar with customer and guest. The only different is the admin dashboard in the navigation bar is available.

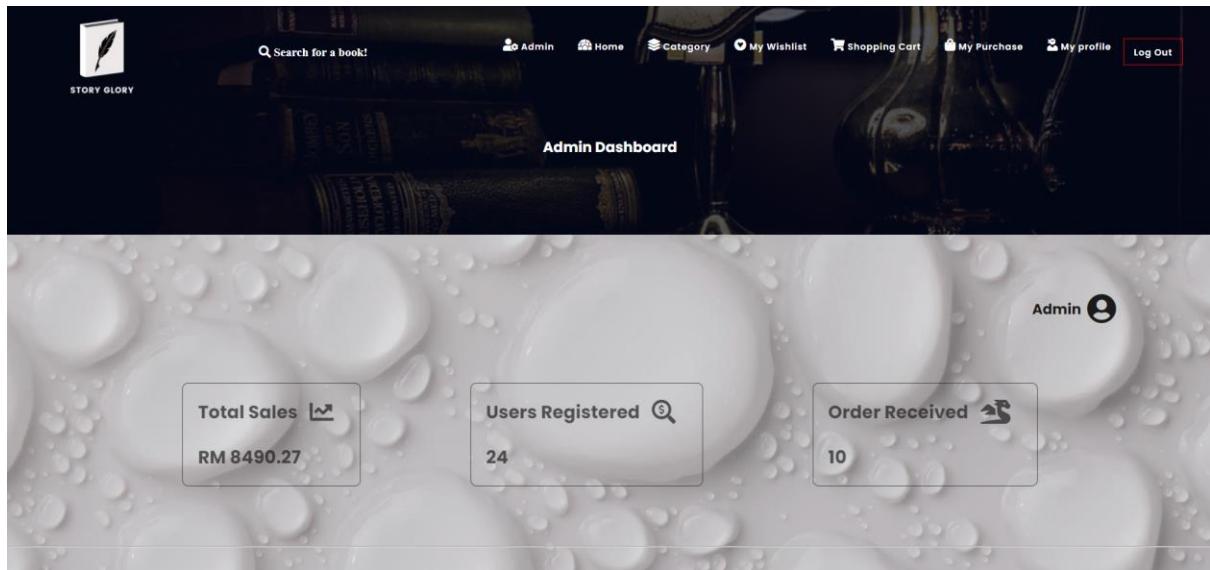


Figure 115: Admin Dashboard (Report)

This report section will show admin the total sales amount, total users registered, and the total order received.

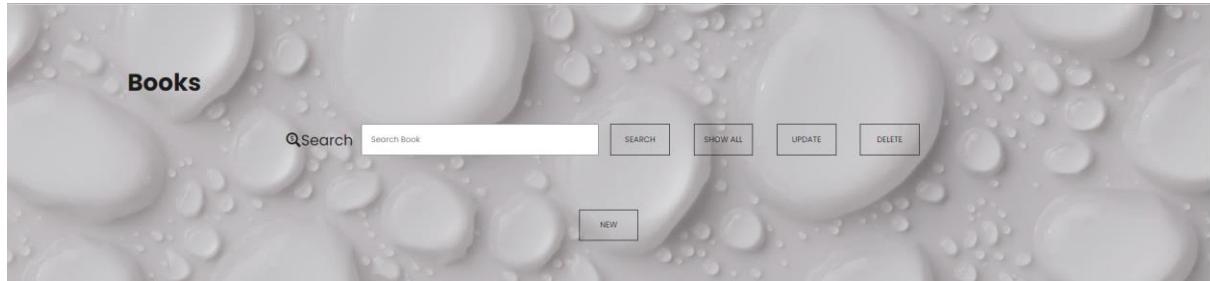
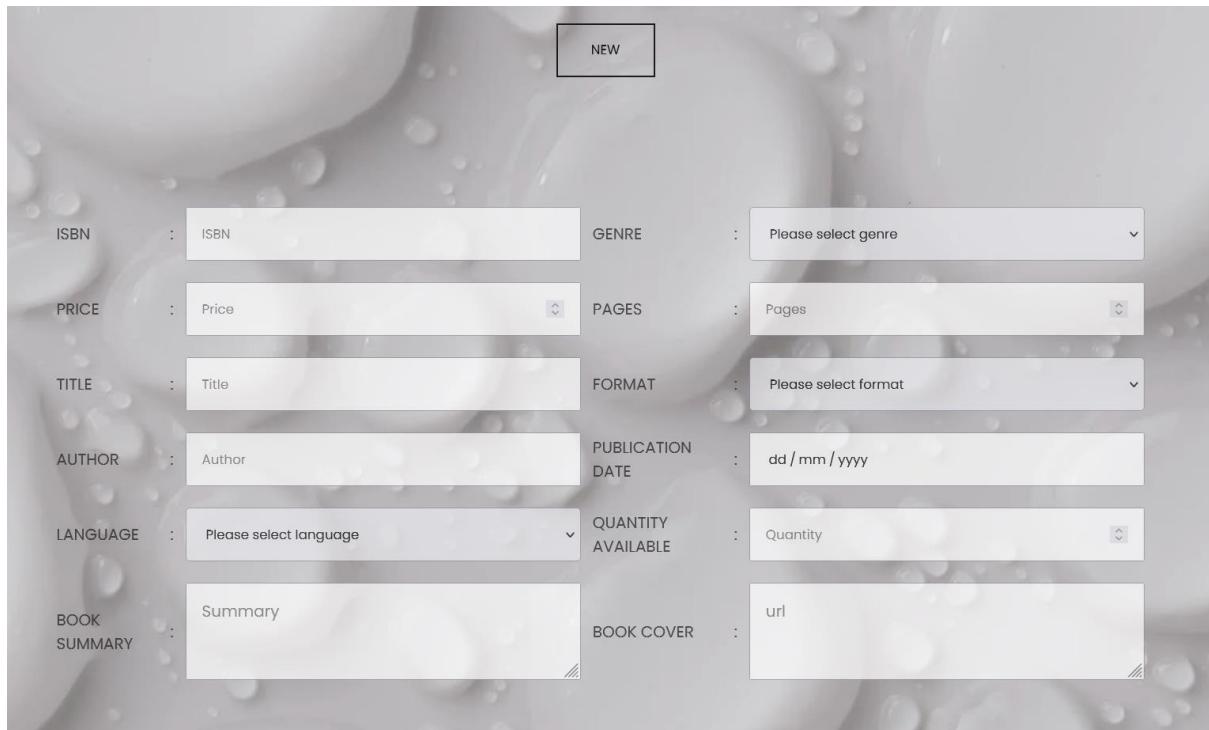


Figure 116: Admin Dashboard (Books)

This is the manage books section. If admin wants to insert a new book into the database, admin needs to click on the “NEW” button to view the insert book form.



The image shows a web form titled "NEW" at the top right. The form consists of two columns of input fields. The left column includes fields for ISBN, PRICE, TITLE, AUTHOR, LANGUAGE, and BOOK SUMMARY. The right column includes fields for GENRE, PAGES, FORMAT, PUBLICATION DATE, QUANTITY AVAILABLE, and BOOK COVER. Each field is preceded by a label and followed by a colon. The "ISBN" field contains "ISBN", the "PRICE" field contains "Price", the "TITLE" field contains "Title", the "AUTHOR" field contains "Author", the "LANGUAGE" field contains "Please select language", and the "BOOK SUMMARY" field contains "Summary". The "GENRE" field contains "Please select genre", the "PAGES" field contains "Pages", the "FORMAT" field contains "Please select format", the "PUBLICATION DATE" field contains "dd / mm / yyyy", the "QUANTITY AVAILABLE" field contains "Quantity", and the "BOOK COVER" field contains "url".

Figure 117: Insert Book Form

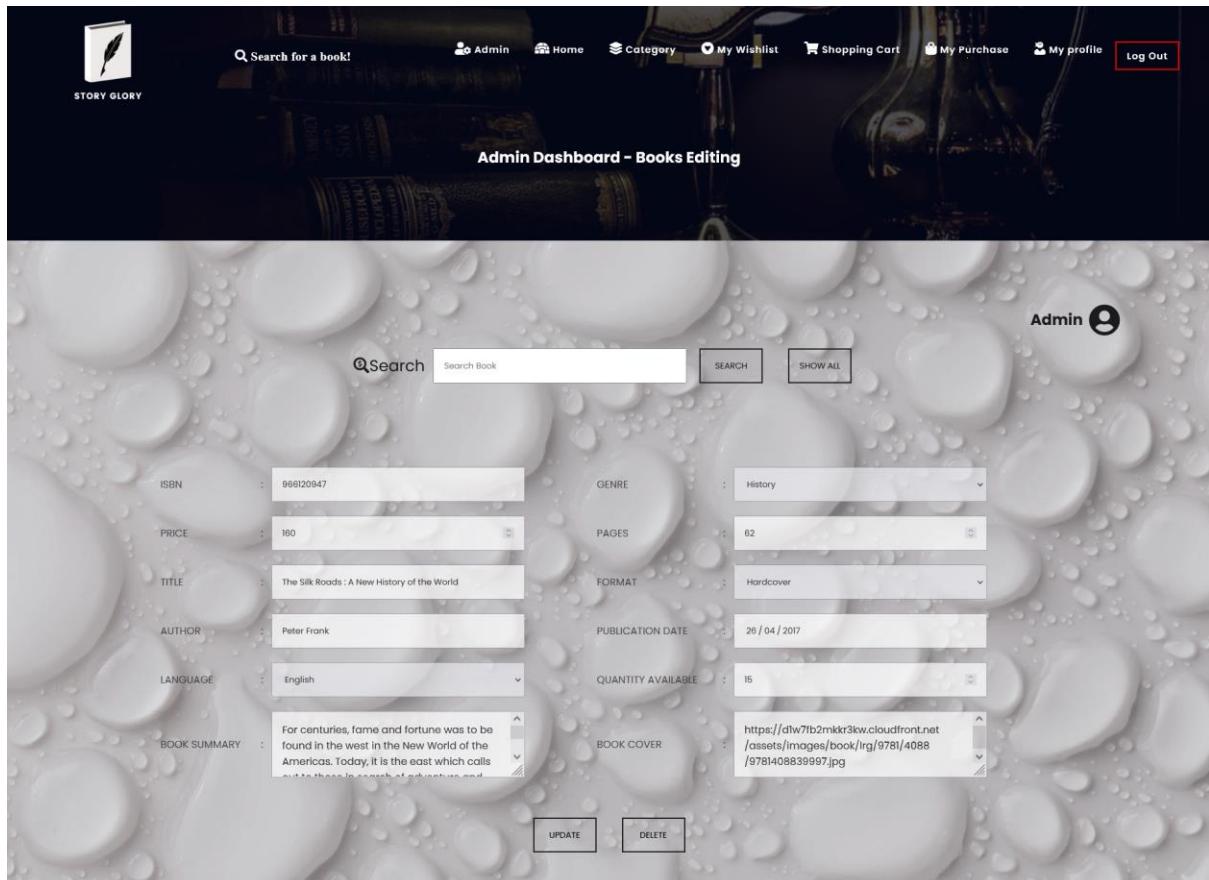
This is the insert book form displayed to the admin. Admin needs to complete book details and click the “NEW” button again to submit the form in order to insert the book.



The image shows a navigation bar with a search icon and the word "Search". To its right is a search input field labeled "Search Book". To the right of the search field are four buttons: "SEARCH", "SHOW ALL", "UPDATE", and "DELETE".

Figure 118: Manage Books

This is the section that allows the admin to search for books, show all the books, update books information and delete books. If admin wants to search, update or delete books, admin needs to enter a search value and click the button of action. The admin will soon be redirect to the books editing page which will display all the search results as shown in the figure below. On the other hand, if admin wants to show all the books the book editing page will shows all the books as illustrated below.



About us
This is what we are



Value

This online bookstore is mainly about providing the best experience to customers when they visit our website as if they were in the physical bookstore, but online! It provides convenience to people who still need physical books during the lockdown of the country.



Product

There are plenty of user-friendly features such as a handy navigation bar, shopping cart as well as help and support section. Everyone is allowed to visit our website to find books they are interested in, but in order to buy them, they would have to register. There will also be special discounts and promotions exclusive to members which customers could register to become one.



Service

All purchased books will be delivered to the customers' location on time with our professional and trusted delivery service. We also include 24/7 customer support to help customers with their issues!

Kindly contact us if you require any assistance.

Figure 119: Books Editing Page

This is the books editing page which links to the admin dashboard page. The header and footer will be exactly same as the admin dashboard page.

Page | 113



Figure 120: Edit Books Button

The “UPDATE” allows an admin to update a book after the book information have been modified. The “DELETE” button will let the admin to remove the book from the database.

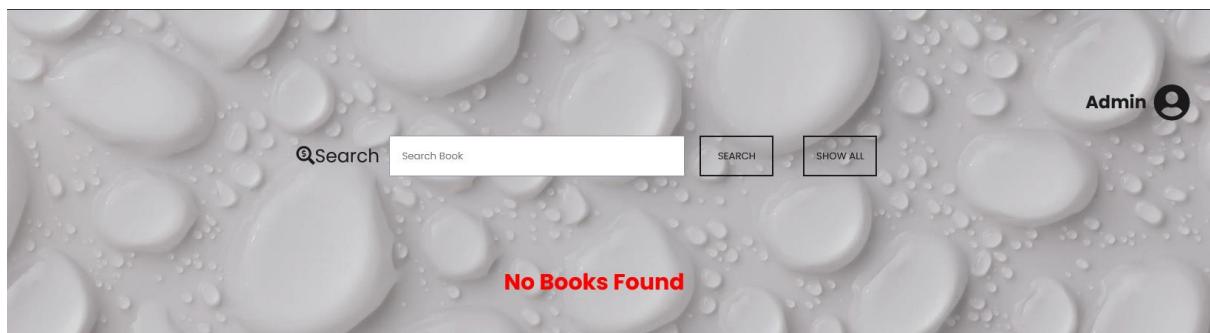


Figure 121: No Books Found

When there is no search result, the message of “No Books Found” will be shown on the screen.

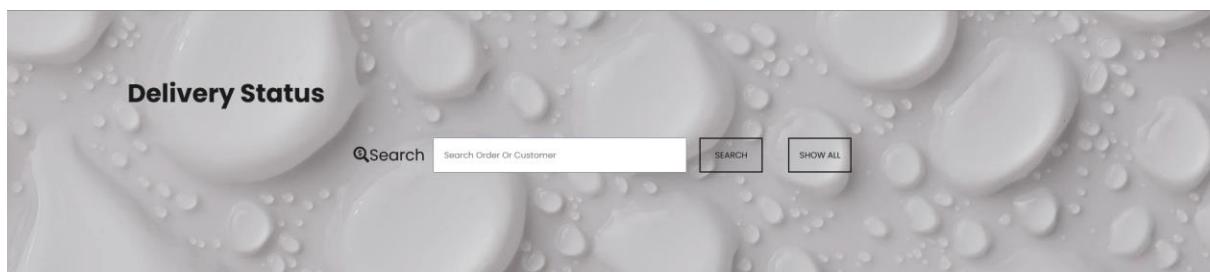


Figure 122: Admin Dashboard (Delivery Status)

This delivery status section of the admin dashboard allows admin to update the delivery status of the orders. To update an order, the admin can enter the Order_ID or customer to search order.

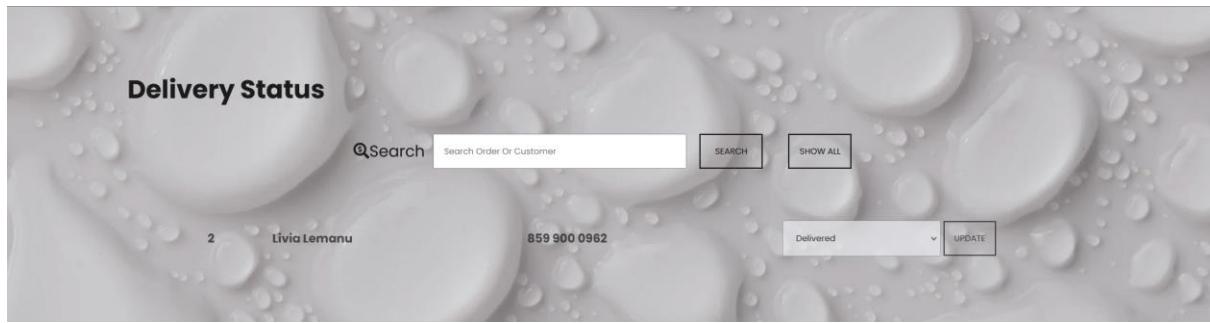


Figure 123: Delivery Status Search Result

The search result will be shown below as shown in Delivery Status Search Result figure below. If admin wish to view all the recent orders, admin can click on the “SHOW ALL” button to view recent orders.



Figure 124: Update Delivery Status

The delivery status of a particular order can be updated through selecting from the dropdown list and click on the “UPDATE” button.

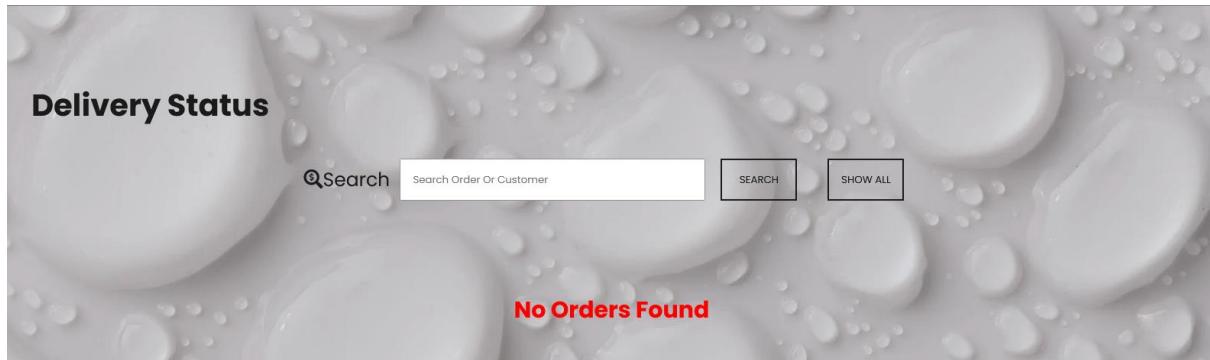


Figure 125: No Orders Found

When there is no search result, the message of “No Orders Found” will be shown on the screen.

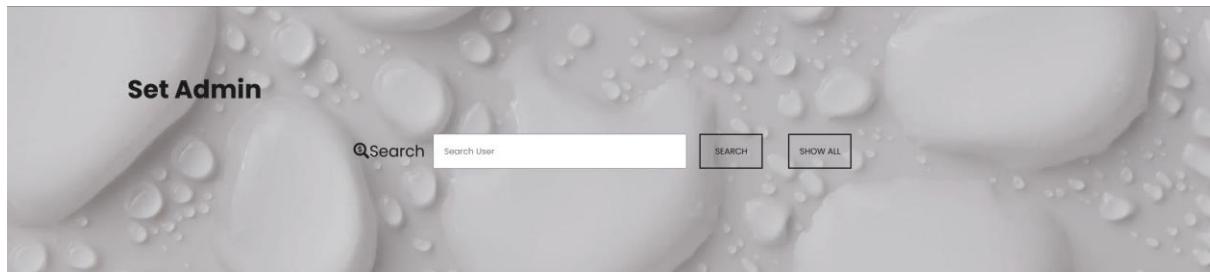


Figure 126: Admin Dashboard (Set Admin)

This is the set admin section of the admin dashboard which admin can set users to become an admin so the user can access the admin dashboard.

	User ID	Name	Role	Action
5	Virginie Askwith		Customer	<button>Set Admin</button>
1	Tamarah Sparrowe		Customer	<button>Set Admin</button>
22	Sophie		Admin	<button>Set Admin</button>

Figure 127: Set Admin Search Results

Admin can search for a user or show all the users. The search results will show the User_ID, Name, and the role of the users. To set a user as admin, admin can click on the “Set Admin” button from the search results.

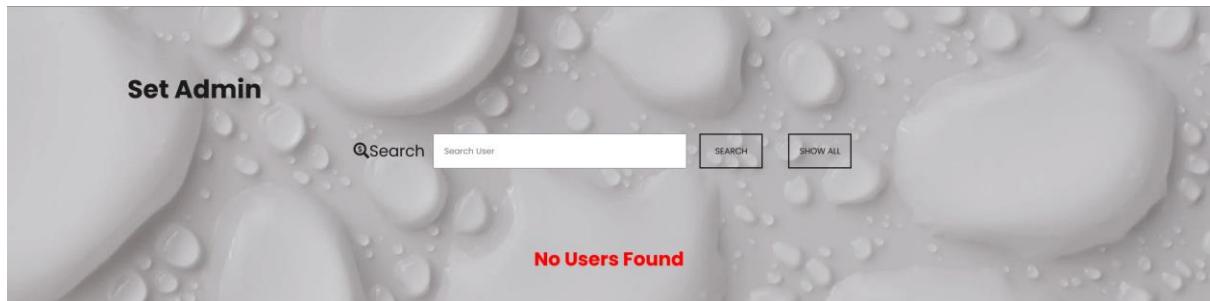


Figure 128: No Users Found

When there is no search result, the message of “No Users Found” will be shown on the screen.

6.0 Conclusion

In a nutshell, Story Glory Online Bookstore is constructed with a combination of HTML, CSS, PHP, JavaScript, and MySQL coding. All these coding languages are crucial for creating a responsive and well-designed website as well as connections to the Story Glory database. Story Glory provides a smooth, convenient, and functional website for customers to view the books that customers are interested in, add them to their Wishlist or Shopping Cart, and proceed to purchase the book. CRUD functions that support all the create, read, update, and delete functions have been implemented in the Story Glory Online Bookstore. This website will be serving the guest, customers, and admin with appropriate functions to support the daily business operations of Story Glory Online Bookstore.

7.0 Workload Matrix

Task	Maxim Hew Mun Sum	Teo Chen Ming	Teo Kai Yii
HTML & CSS	30	40	30
PHP & MySQL	20	35	45
JavaScript	30	50	20
Documentation	40	20	40

Figure 129: Workload Matrix

8.0 References

Tutorials, E. T. (2021, March 15). *How To Make Website Using HTML & CSS / Full Responsive Multi Page Website Design Step by Step.*

Retrieved October 4 , 2021, from <https://youtu.be/oYRda7UtuhA>

Souy, S. S. (2018, October 15). *How to Design Checkout Form HTML and CSS.* YouTube.

Retrieved October 4, 2021, from <https://youtu.be/MPbF0FOpAVs>

Jackson, P. J. (2021, October 7). *PHP MySQLi Functions: mysqli_query, mysqli_connect, mysqli_fetch_array.* Guru99. Retrieved October 14, 2021, from

<https://www.guru99.com/mysql-php-and-other-database-access-methods.html>

PHP exit() Function. (n.d.). W3schools. Retrieved November 12, 2021, from

https://www.w3schools.com/php/func_misc_exit.asp

PHP - MySQL Login. (n.d.). Tutorialspoint. Retrieved November 12, 2021, from

https://www.tutorialspoint.com/php/php_mysql_login.htm

JavaScript String trim(). (n.d.). W3schools. Retrieved November 12, 2021, from

https://www.w3schools.com/jsref/jsref_trim_string.asp

JavaScript String length. (n.d.). W3schools. Retrieved November 12, 2021, from

https://www.w3schools.com/jsref/jsref_length_string.asp