

School of Computing and Information Technology, University of Technology, Jamaica

Module Name: Web Programming

Module Number: CIT2011

Title: Storyboard

Name: Dominique Watson

ID #: 2404230

Tutor: Ms. Christine Anuli

Student Details

Full Name: Dominique Watson

ID#: 2404230

Course: CIT2011- Web Programming

Day/Time: Thursday, 1pm-3pm

Github Repository: https://github.com/dominiquewatson/dwatson_2404230_utech_ia2

Business Description

Logo:



Business Name: Book Haven

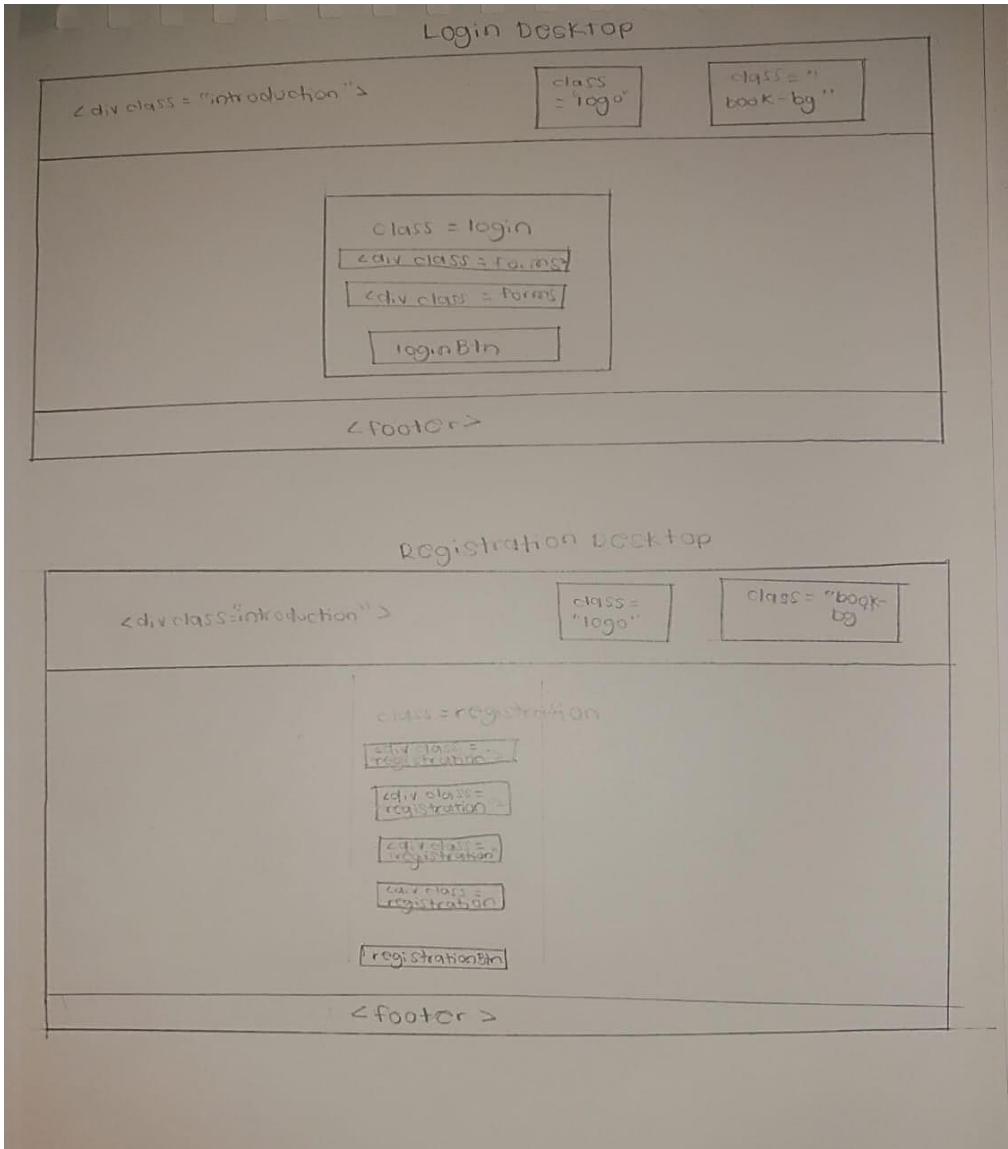
Slogan: Reading is an adventure

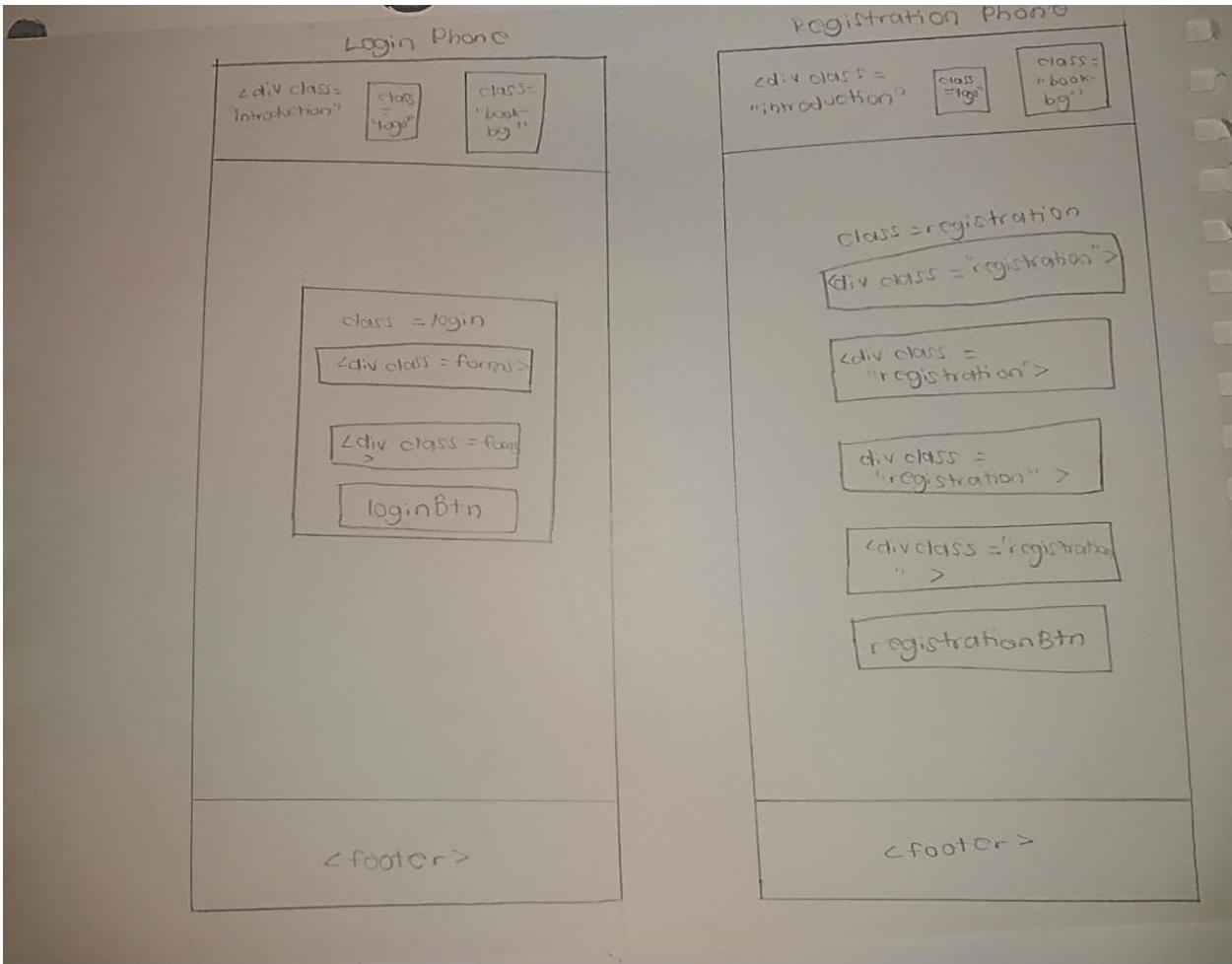
Mission Statement: Dedicated to provide affordable, diverse, and high-quality books that encourage learning and creativity across Jamaica.

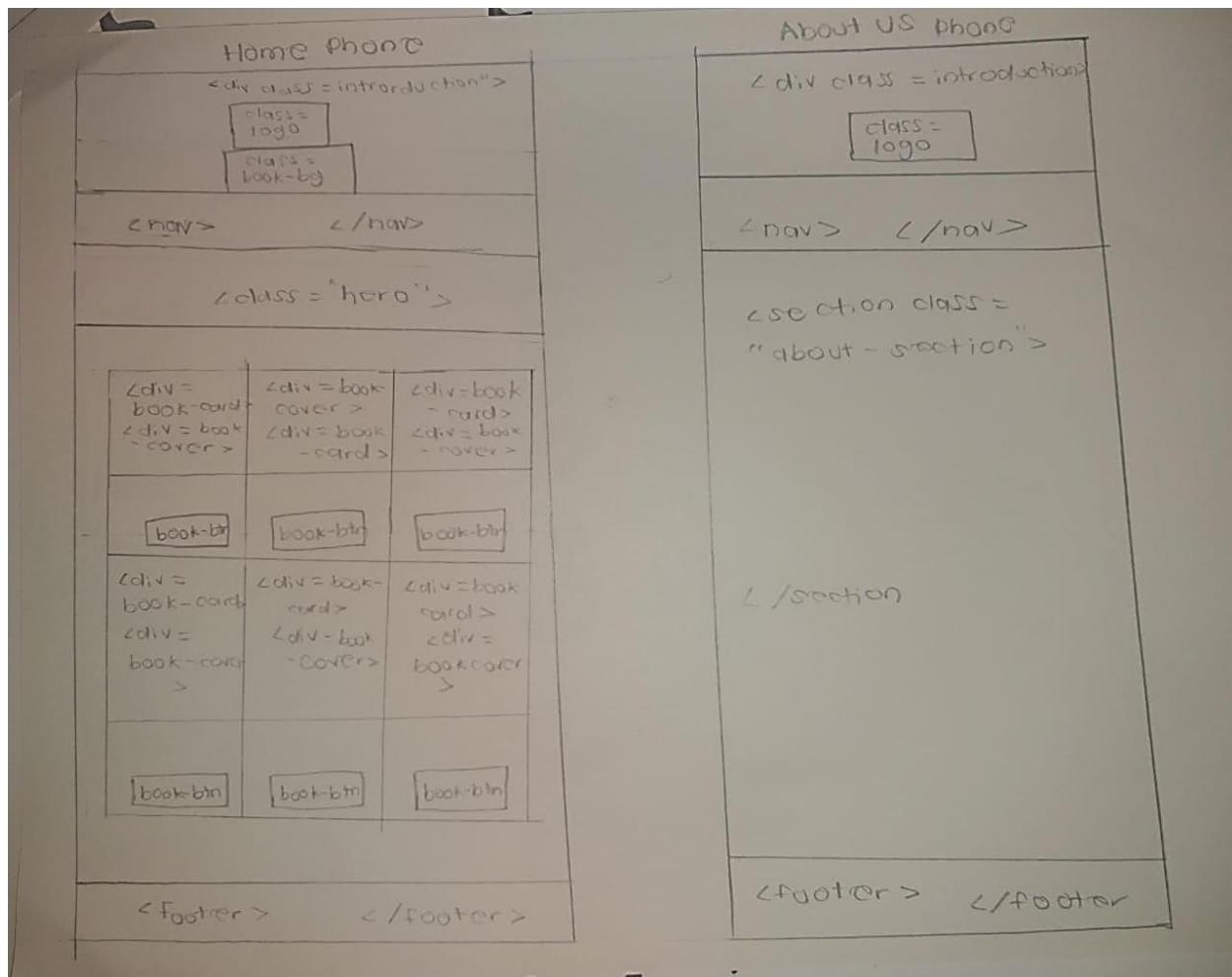
Vision Statement: To become Jamaica's leading online bookstore, recognized for promoting literacy, supporting local authors, and creating a generation of readers.

Indented Audience: General public and Book Lovers.

Wireframe







HOME DESKTOP

```
<div class="introduction">
```

卷之三

1990

~~class = bookbag~~

< class="hero" >

```

graph TD
    book[book] --> bookCard[book-card]
    book --> bookAttr[book-attr]
    book --> bookInfo[book-info]
    book --> bookTimes[book-times]
    book --> bookWords[book-words]
    book --> bookBin[book-bin]
    bookAttr --> color[color]
    bookAttr --> size[size]
  
```

$\text{Lohn} = \text{book-}$
 $\text{card} >$
 $\text{Lohn} = \text{book-}$
 $\text{color} >>$

```
cdiv = book  
cccdv =  
cdiv = book  
cccdv =  
  
bookinfo  
bookinfo  
bookinfo
```

```
<div>=book  
    <card>  
        <div>=book  
            <cover>>  
  
<book info>  
<book price>  
  
    book-btn
```

<footer> </footer>

About Us

<div class=

introduction >

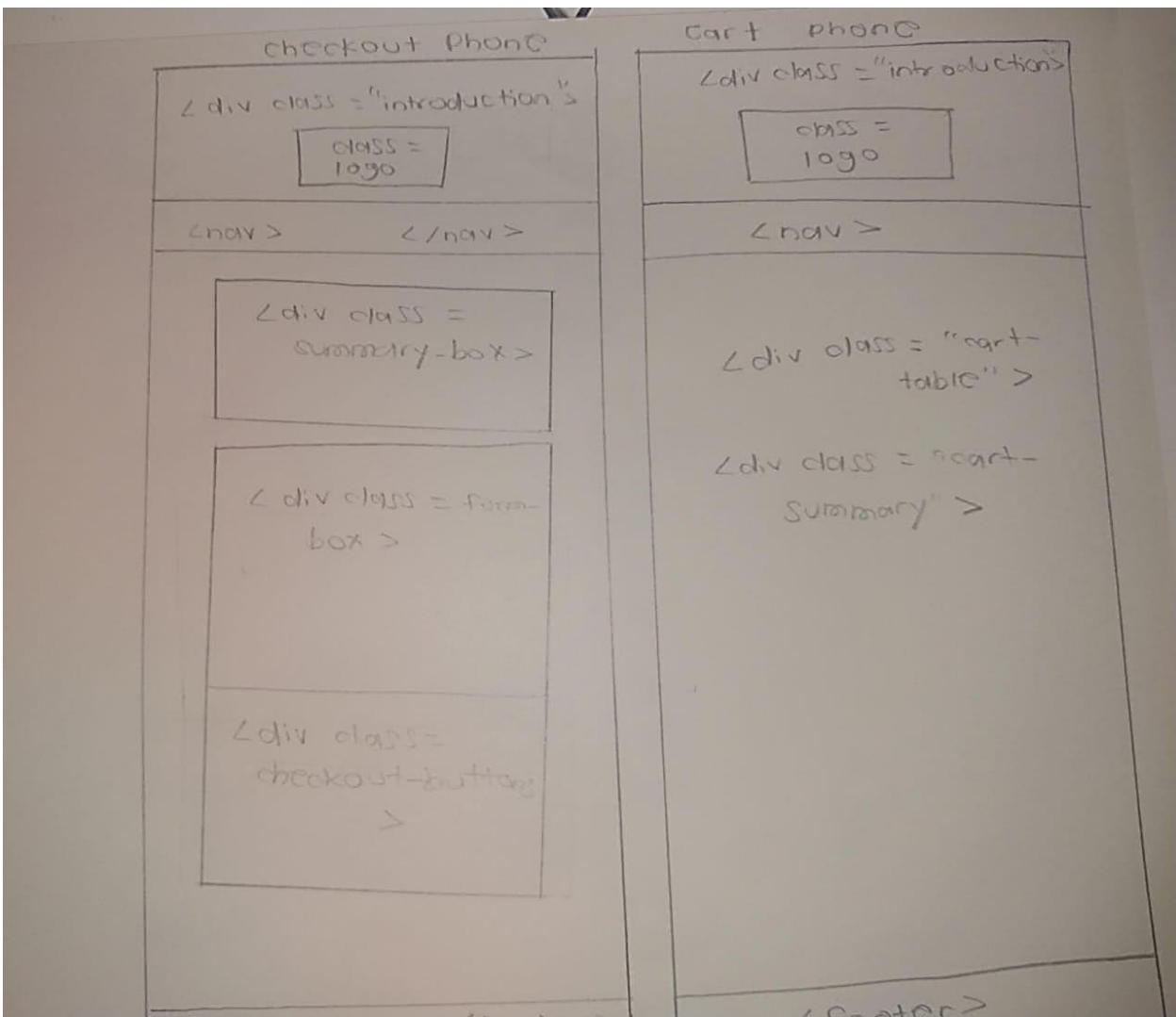
1050

<non>

</section>

< footer >

</footer>



check out desktop

<div class="introduction">

class =
1990

<nav>

<div class =
summary-box>

<div class =
form-box>

<div class =
checkout-buttons>

<footer>

</footer>

Cart Desktop

<div class = "introduction">

class
= 1990

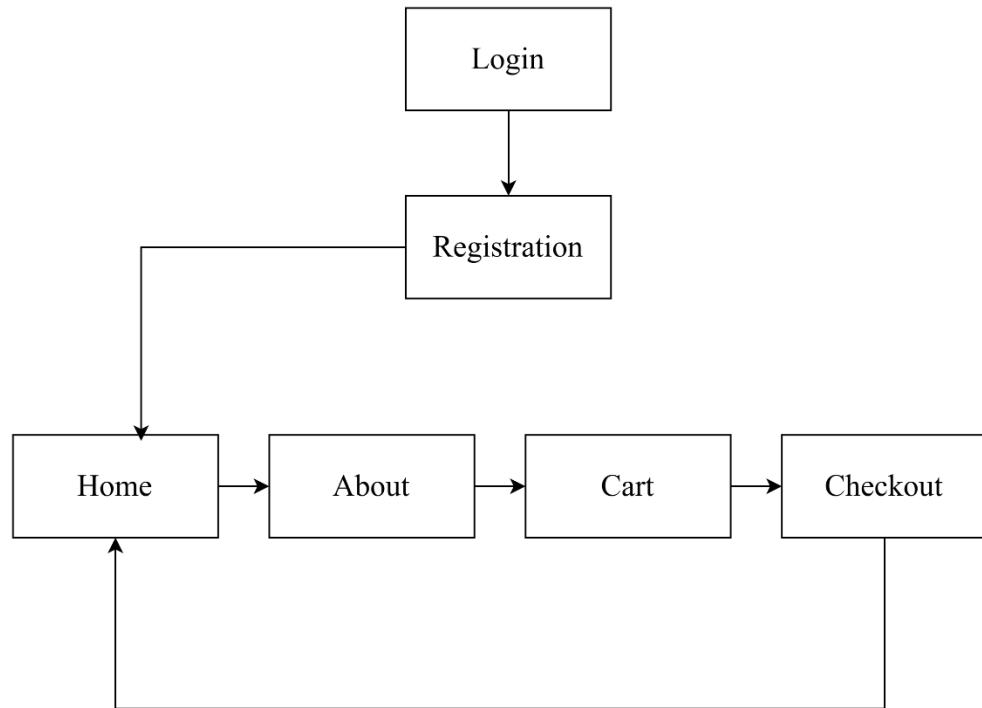
<nav>

<div class = "cart-table">

<div class = "cart-summary">

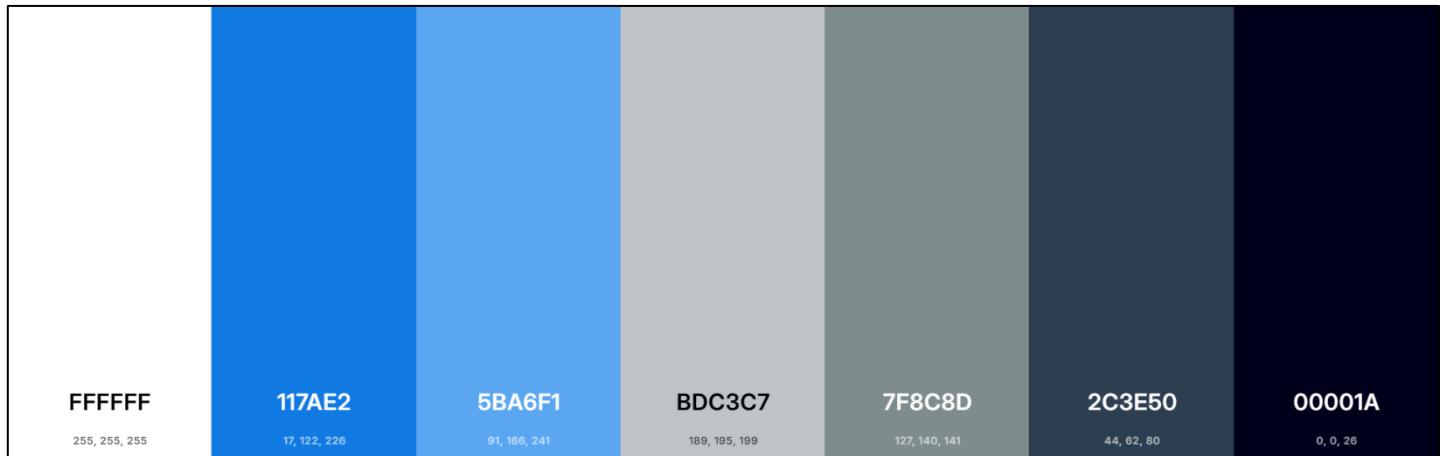
<footer>

Sitemap



Colour Palette

Website Layout:



Color Name	HEX	RGB
White	#FFFFFF	(255, 255, 255)
Azure Blue	#117AE2	(17, 122, 226)
Cool Horizon	#5BA6F1	(91, 166, 241)
Pale Slate	#BDC3C7	(189, 195, 199)
Olive Grey	#7F8C8D	(127, 140, 141)
Charcoal Blue	#2C3E50	(44, 62, 80)
Ink Black	#00001A	(0, 0, 26)

Buttons on Checkout Page:



Color Name	HEX	RGB
Blue Bell	#1798EE	(23, 152, 238)
Jade Green	#1BC532	(27, 197, 50)
Burnt Tangerine	#DC1B05	(220, 27, 5)
Fuchsia Flame	#E71987	(231, 25, 135)
Grey Olive	#7F8C8D	(127, 140, 141)

Justification of Colour Palette: The palette combines neutral blue tones (White, Pale Slate, Olive Grey, Charcoal Blue, Ink Black) with vibrant accent colours (Azure Blue, Cool Horizon, Blue Bell, Jade Green, Burnt Tangerine, Fuchsia Flame) to create a balanced and visually appealing interface. The neutral blue shades provide readability and a clean layout for the entire website, allowing users to feel calm as they're searching for books. On the other hand, the bright accent colours guide the user's attention to key actions like buttons on the checkout page. Overall, this palette selection was made to promote user engagement.

Icons and their Dimensions

Icon Specifications:

Font Family: Material Icons

Font Weight: Inherited from parent (700 or bold)

Dimensions: Controlled by font-size

Display: Inline-flex with vertical alignment

Navigation Items Using Icons:

Home Link - Home icon



About Link – Info icon



Cart Link - Shopping cart icon



Checkout Link - Payment/credit card icon



Logo:

Dimensions: Height: 60px, Width: auto

Styling: Border-radius: 5px

Location: Header introduction section

Book Cover Images:

Dimensions:

Grid view: 100% width × 300px height

Cart view: 80px width × 100px height

Object Fit: Cover

Border Radius: 6px (cart), none (grid)

Transition: Scale transform on hover

Font Weights:

Primary: 700 (bold) - applied to body

Labels: Bold (inherited weight)

Buttons: Bold

Font Sizes Hierarchy:

Header H1: 2.5rem (introduction), 2rem (main header)

Hero H1: 3.5rem

Section Titles: 2.2rem

Login/Form Headers: 1.8rem

Book Titles: 1.2rem

Body Text: 1rem

Author/Secondary Text: 0.9rem

User-Defined Functions Documentation

Creator: Dominique Watson

Login Functions

- **Login Validation**

Purpose: Validates user login credentials before granting access.

Description: This script checks the username and password entered in the login form. It ensures both fields are filled in, compares the input to the predefined valid credentials (**manager123** and **password123**), and displays either a success or error message. On successful login, the user is redirected to **register.html** after a 1-second delay.

Registration Functions

- **Registration Form Validation**

Purpose: Ensures all registration fields are correctly filled before account creation.

Description: This function validates the registration form fields: first name, last name, email, password, and confirm password. It checks for empty fields, validates email format, ensures passwords match, and enforces a minimum password length of 8 characters. If validation passes, a success message is displayed and the user is redirected to **home.html** after 1 second.

- **Empty Field Check**

Purpose: Prevents form submission with missing information.

Description: Before processing either form, the script ensures that all required input fields are filled. If any field is empty, an error message is displayed, and the submission is halted.

- **Password Match Check**

Purpose: Ensures user enters matching passwords during registration.

Description: The script compares the password and confirm password fields. If they do not match, the user is informed immediately and the form does not submit.

- **Password Strength Check**

Purpose: Enforces a minimum security requirement.

Description: The script checks that the password contains at least **8 characters**. If not, an error message is displayed and the form is rejected.

- **Email Format Validation**

Purpose: Ensures that the email address provided is valid.

Description: Uses a regular expression to check if the email matches the correct format (example@example.com). If the pattern does not match, the user is prompted to enter a valid email.

- **Success Redirect**

Purpose: Redirects the user after successful login or registration.

Description: After showing a success message, the script waits **1 second** before automatically sending the user to the next page (**register.html or home.html**), giving time for the confirmation message to appear.

Home Functions

- Add to Cart

Purpose: Handles adding books to the shopping cart

Description: This function enables “Add to Cart” behavior on an e-commerce webpage.

When a user clicks an **Add to Cart** button, the script retrieves product details (title and price), stores the product in the browser’s **localStorage**, and updates quantity if the item already exists in the cart.

Cart Functions

- Item Subtotal Calculation

Purpose: Calculates the raw cost of each cart item before discounts or taxes.

Description: For every item in the cart, the script multiplies the item’s price by its quantity:

$$\text{itemSubtotal} = \text{price} \times \text{quantity}$$

This value is displayed in the cart and also added to the overall cart subtotal.

- Discount Calculation

Purpose: Applies a 5% discount on eligible items.

Description:

The script checks if an item’s subtotal is greater than **5000**. If yes, a **5% discount** is applied:

$$\text{discount} = \text{itemSubtotal} \times 0.05$$

If not, discount remains **0**. The total discount for all items is also summed and displayed in the order summary.

- **Tax Calculation**

Purpose: Calculates the 15% tax to be added after discounts.

Description:

Tax is applied to the discounted item amount:

$$\text{tax} = (\text{itemSubtotal} - \text{discount}) \times 0.15$$

Each item's tax value contributes to the overall tax total in the summary section.

- **Total Per Item Calculation**

Purpose: Computes the final cost of each item after discount and tax.

Description:

The total is determined by:

$$\text{total} = \text{itemSubtotal} - \text{discount} + \text{tax}$$

This final amount is displayed as the item's overall cost in the cart.

- **Cart Summary Calculations**

Purpose: Displays the overall order totals.

Description: The script accumulates the following across all cart items:

Subtotal: Sum of all item subtotals

Total Discount: Sum of all discounts applied

Total Tax: Sum of all tax amounts

Grand Total: $\text{grand total} = \text{subtotal} - \text{total discount} + \text{total tax}$

- **Update Cart Display**

Purpose: Renders and updates the shopping cart page.

Description:

This function reads cart data from **localStorage** and dynamically displays each item in a cart table. It calculates and updates the subtotal, discount, tax, and total for each item and the overall order summary. The function also refreshes the cart UI whenever quantities are changed or items are removed.

- **Quantity Update Listener**

Purpose: Updates item quantity in real time.

Description:

Listens for changes in the quantity input fields. When a user edits an item's quantity, the cart array is updated and the cart display operates to reflect new prices, discounts, and totals. The updated cart is saved back to **localStorage**.

- **Remove Item Listener**

Purpose: Removes items from the shopping cart.

Description:

When a user clicks the "Remove" button for an item, that item is deleted from the cart array. The cart display is then refreshed and the updated cart is saved to **localStorage**.

- **Checkout Redirect**

Purpose: Navigates the user to the checkout page.

Description:

When the checkout button is clicked, the page automatically redirects the user to **checkout.html** to continue the purchase process.

- **Initial Cart Render**

Purpose: Displays the cart when the page loads.

Description:

On page load, the script reads cart data from **localStorage** and calls the **updateCartDisplay()** function so all items and calculations appear immediately.

Checkout

- **Cart Totals Calculation**

Purpose: Computes all financial values for the checkout summary.

Description:

The script loops through all cart items and calculates:

- **Item Subtotal:** $\text{price} \times \text{quantity}$
- **Item Discount:** 5% applied if subtotal > 5000
- **Item Tax:** 15% applied after discount
- **Item Total:** subtotal – discount + tax

All item values are added together to produce the overall **subtotal**, **discount**, **tax**, and **grand total**, which are then displayed on the checkout page.

- **Display Totals**

- **Purpose:** Shows all calculated values to the customer.

Description:

After computing totals, the script updates the checkout summary using:

- **co-subtotal**

- **co-discount**
- **co-tax**
- **co-total**

All values are formatted to two decimal places for clarity.

- **Confirm Button**

Purpose: Confirms the user's order details.

Description:

When clicked, this button simply displays a confirmation message — no redirection or data change occurs. It is used to verify that the user approves the checkout information.

- **Cancel Button**

Purpose: Clears the form without affecting the cart.

Description:

Displays a message indicating the form was cleared. (If you later add form-clearing logic, this button should reset all user-entered fields.)

- **Clear All Button**

Purpose: Clears the entire cart and resets the checkout page.

Description:

This button removes the cart from **localStorage**, shows an alert confirming the action, and reloads the page to show an empty checkout.

- **Checkout Submission**

Purpose: Validates user payment and completes the purchase process.

Description: When the user clicks the checkout button:

- The script validates required fields (name, address, amount paid).
- Ensures the payment is **not less than** the total cost.
- Calculates **change** using:

$$\text{change} = \text{amountPaid} - \text{total}$$

- Displays a full purchase summary (name, address, total cost, amount paid, and change).
- Clears the cart from localStorage.
- Redirects the user to **home.html**.

- **Close Button**

Purpose: Returns the user to the cart page.

Description: When clicked, the user is redirected back to **cart.html**, allowing them to review or adjust their items.

Bibliography

Barnes, J. L. (2017). Bad Blood. Barnes and Noble. URL:

<https://www.barnesandnoble.com/w/bad-blood-jennifer-lynn-barnes/1123632264>

Barnes, J. L. (2020). The Inheritance Games. Amazon. URL:

<https://www.amazon.com.au/Inheritance-Games-Jennifer-Lynn-Barnes-ebook/dp/B086X175RY>

Chbosky, S. (1999). The Perks of Being a Wallflower. Goodreads.

URL:<https://www.goodreads.com/book/show/4327066-the-perks-of-being-a-wallflower>

Chen, J. (2020). I Am Not Jessica Chen. Barnes and Noble. URL:

<https://www.barnesandnoble.com/w/i-am-not-jessica-chen-ann-liang/1145648218>

Collins, S. (2008). The Hunger Games. Amazon. URL: <https://www.amazon.com/Hunger-Games-Book/dp/0439023483>

Coloor.(2025). Palette Generator.Coloor.

URL: <https://coolors.co/>

Envato.(2021).Color Psychology | Red, Blue, Yellow, Green, Pink, White, Black, Gray and Gold. YouTube. URL: <https://www.youtube.com/watch?v=8J0nbuunDXc>

Green, J. (2019). Turtles All the Way Down. Barnes and Noble.

URL:<https://www.barnesandnoble.com/w/turtles-all-the-way-down-john-green/1126619413>

Hazelwood, A. (2024). Problematic Summer Romance. Penguin Random House. URL:

[https://www.penguinrandomhouse.com/books/800620/problematic-summer-romance-by-
ali-hazelwood/](https://www.penguinrandomhouse.com/books/800620/problematic-summer-romance-by-ali-hazelwood/)

Jackson, H. (2022). Five Survive. Barnes and Noble. URL:

<https://www.barnesandnoble.com/w/five-survive-holly-jackson/1141932413>

Jackson, H. (2024). The Reappearance Of Rachel Price. Penguin Random Huse. URL:

[https://www.penguinrandomhouse.com/books/668984/the-reappearance-of-rachel-price-
by-holly-jackson/](https://www.penguinrandomhouse.com/books/668984/the-reappearance-of-rachel-price-by-holly-jackson/)

Lo, M. (2021). Last Night At The Telegraph Club. Amazon. URL:

<https://www.amazon.com/Last-Night-at-Telegraph-Club/dp/0525555250>

McManus, K. M. (2017). One Of Us Is Lying. Barnes and Noble. URL:

<https://www.barnesandnoble.com/w/one-of-us-is-lying-karen-m-mcmanus/1124568207>

McManus, K. M. (2020). The Cousins. Barnes and Noble. URL:

<https://www.barnesandnoble.com/w/the-cousins-karen-m-mcmanus/1136585461>

Russell, R. R. (2009). Dork Diaries. Barnes and Noble. URL:

<https://www.barnesandnoble.com/w/tales-from-a-not-so-smart-miss-know-it-all-rachel-renee-russell/1110199529>

Silver, E. (2020). Wild Love. Apple Books. URL: <https://books.apple.com/us/book/wild-love/id6468767368>

Thomas, A. (2017). The Hate U Give. Barnes and Noble.

URL:<https://www.barnesandnoble.com/w/the-hate-u-give-angie-thomas/1124651146>

White, E. B. (1952). Charlotte's Web. Amazon. URL:<https://www.amazon.com/Charlottes-Web-B-White-1952-01-21/dp/B01K3IMJRK>

Yoon, N. (2015). Everything, Everything. Penguin Random House. URL:

<https://www.penguinrandomhouse.com/books/240386/everything-everything-by-nicola-yoon/>

Yoon, N. (2016). *The Sun Is Also A Star*. Penguin Random House. Barnes and Noble. URL:

<https://www.barnesandnoble.com/w/the-sun-is-also-a-star-nicola-yoon/1123596765>