



A • P • I • I • T

Inspire love for learning 

WEBSITE ASSIGNMENT

LEVEL 4

COMP40004 : WEB DEVELOPMENT AND OPERATING SYSTEMS 2

Individual Assignment: COM24A3

Pamila Sendila Imaduwa Gamage	CB015690
-------------------------------	----------

Contents

1.	LINK TO THE WEBSITE	3
2.	INTRODUCTION	3
2.1.	INTORDUCTION TO THE REPORT	3
2.2.	DEVELOPMENT PROCESS.....	3
3.	WIRE FRAMES.....	4
3.1.	ORDER PAGE	4
3.2.	CHECKOUT PAGE	5
4.	TESTING AND VALIDATION	6
4.1.1.	Order.html (ordering page).....	6
4.1.2.	Checkout.html.....	6
4.2.	CSS VALIDATION.....	7
4.2.1.	Checkout.css	7
4.2.2.	Order.css	7
4.3.	Lighthouse Testing	8
4.3.1.	ORDER PAGE.....	8
4.3.2.	CHECKOUT PAGE	8
4.4.	ACCESSIBILITY USING WAVE	9
5.	Diagrams.....	10
5.1.	ACTIVITY DIAGRAMS.....	10
5.1.1.	ADD TO CART FUNCTION	10
5.1.2.	SAVE TO FAVOURITES	10
5.1.3.	APPLY FAVOURTIES.....	11
5.1.4.	ADJUST QUANTITY IN CART	11
5.1.5.	CHECKOUT PROCESS	12
5.2.	SWIM LANE DIAGRAMS	13
5.2.1.	ORDERING PAGE DIAGRAM	13
5.2.2.	CHECKOUT PAGE DIAGRAM	14
5.3.	TEST CASE DIAGRAMS	15
3.	FACED CHALLENGES AND SOLUTIONS.....	17
4.	REFERENCES	18

1. LINK TO THE WEBSITE

Github hosted site (as per requirement)

<https://quandaledinglejunior.github.io/grass/>

Vercel Hosted site (to show favicon functioning)

<https://grass-dun.vercel.app/>

2. INTRODUCTION

2.1. INTORDUCTION TO THE REPORT

I have developed a gaming website to share information about consoles, peripherals and upcoming games and has a FAQ section for the users. The goal was to keep the site simple, responsive, and easy to use.

I used HTML, CSS and JavaScript for the design and functionality and deployed it to GitHub Pages. I used tools like VS Code and Live Server to write and test the code (Also Vercel to double check issues). For testing I have used W3C HTML and CSS Validators to verify clean code, Lighthouse for performance and accessibility and WAVE for accessibility testing.

This report will cover the website structure, the testing and validation I did on the site, the challenges I have faced, how I fixed those issues and what I learned during the process.

2.2. DEVELOPMENT PROCESS

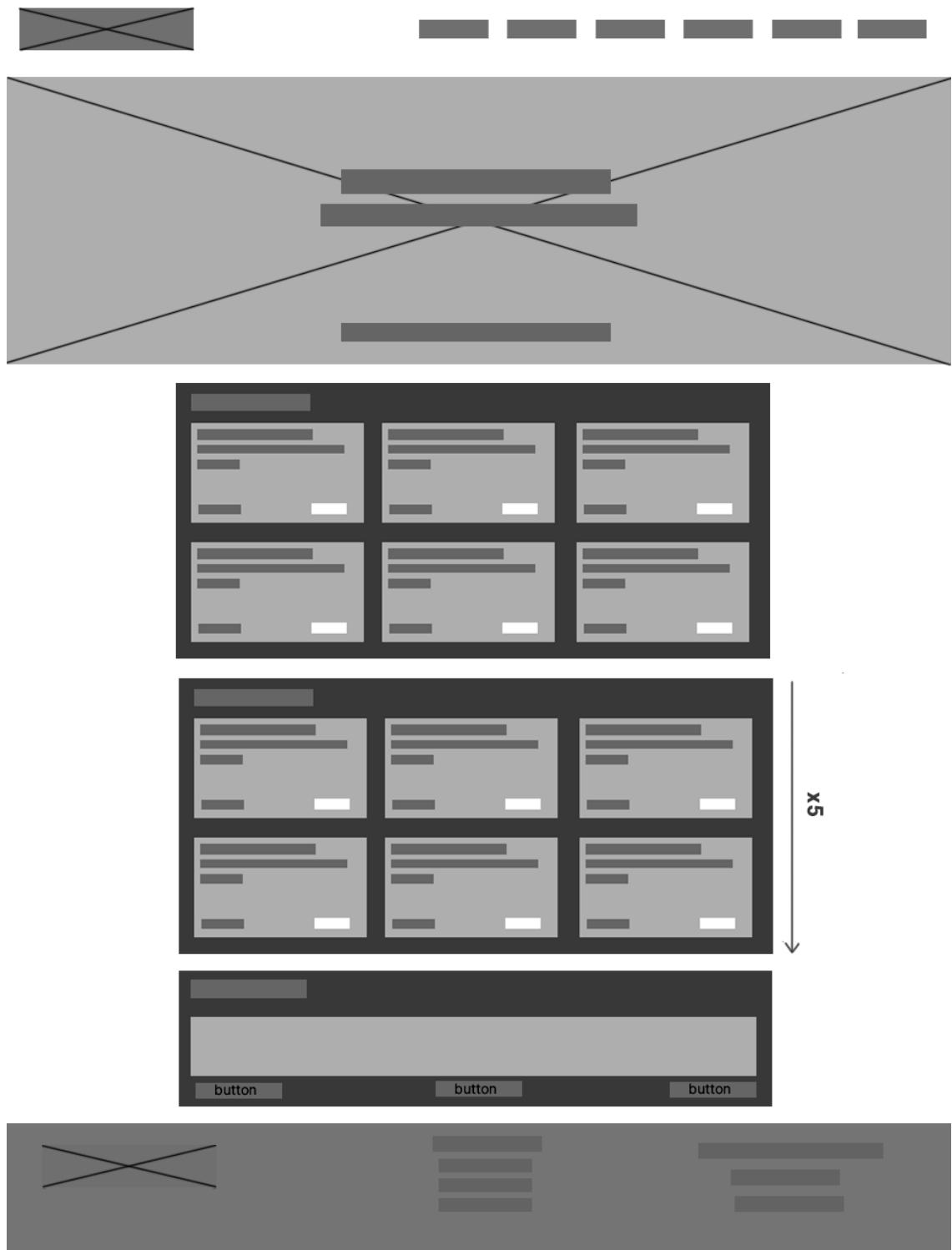
The project files are stored in a simple folder structure where HTML, CSS and JavaScript files are in the root directory and images are inside subfolders.

I First of all I created the desktop view of the website and then the mobile view. I used Visual Studio Code for coding and Live Server to see the changes in real time. I have used Git to push updates to GitHub where the website was hosted and made accessible online.

The website has a responsive layout with interactive hover effects and a clean structure to have better accessibility and user experience. The goal was to create a smooth experience on both mobile and desktop views (with a higher focus on mobile).

3. WIRE FRAMES

3.1. ORDER PAGE



3.2. CHECKOUT PAGE



4. TESTING AND VALIDATION

4.1.1. Order.html (ordering page)

Showing results for uploaded file order.html

Checker Input

Show source outline image report

Check by No file chosen

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 5 milliseconds.

4.1.2. Checkout.html

Showing results for uploaded file order.html

Checker Input

Show source outline image report

Check by No file chosen

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 5 milliseconds.

4.2. CSS VALIDATION

4.2.1. Checkout.css

W3C CSS Validator results for checkout.css (CSS level 3 + SVG)

Congratulations! No Error Found.

This document validates as [CSS level 3 + SVG](#) !

4.2.2. Order.css

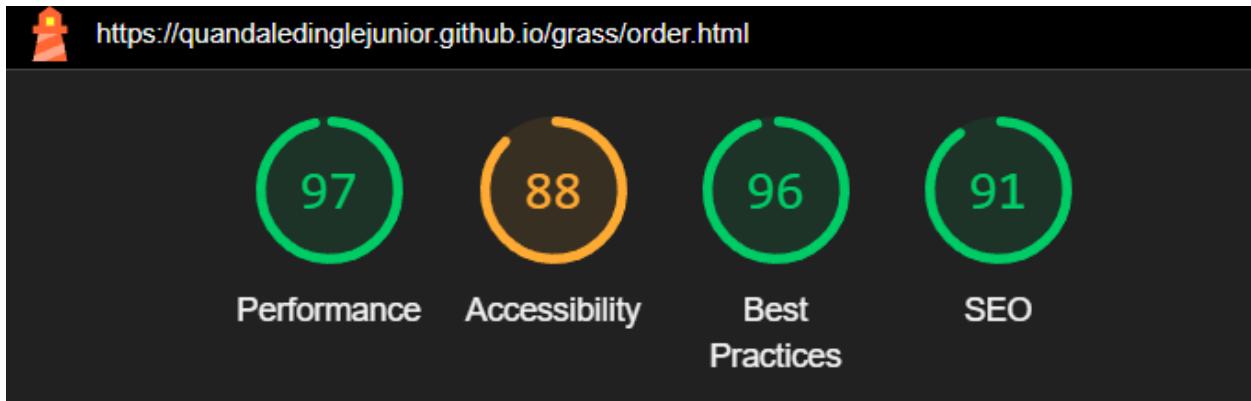
W3C CSS Validator results for order.css (CSS level 3 + SVG)

Congratulations! No Error Found.

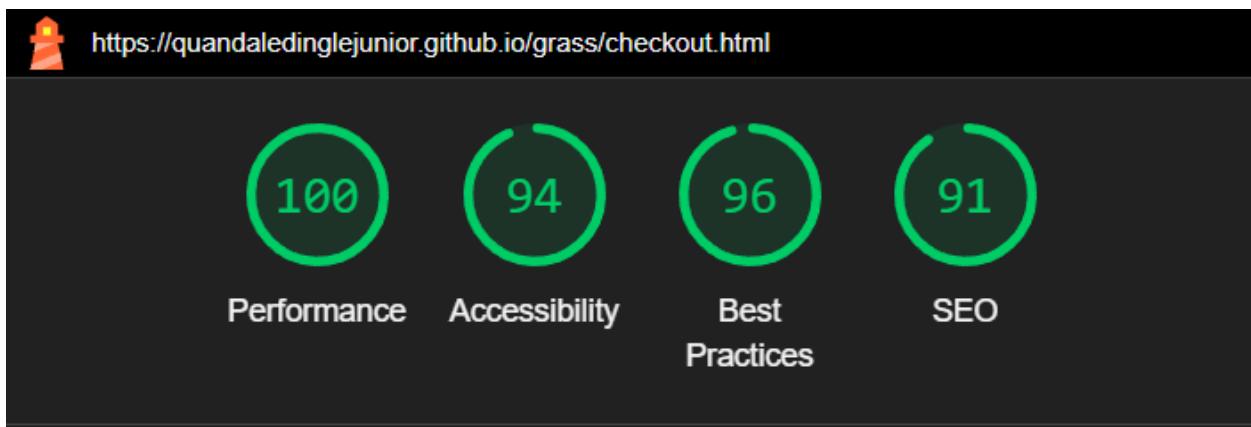
This document validates as [CSS level 3 + SVG](#) !

4.3. Lighthouse Testing

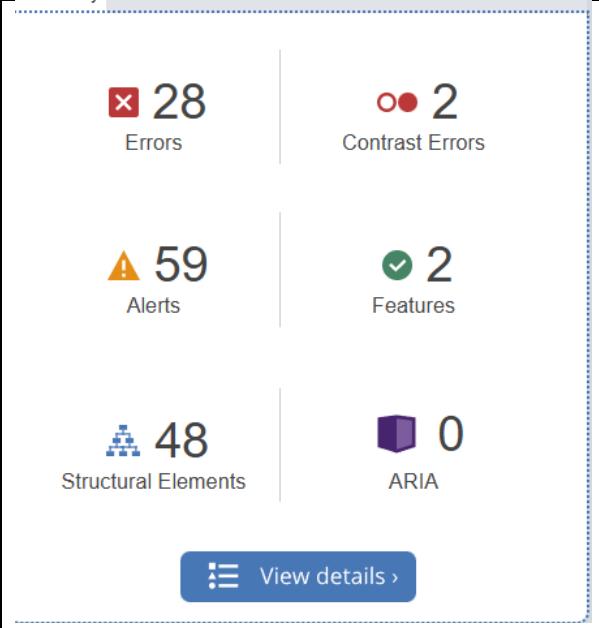
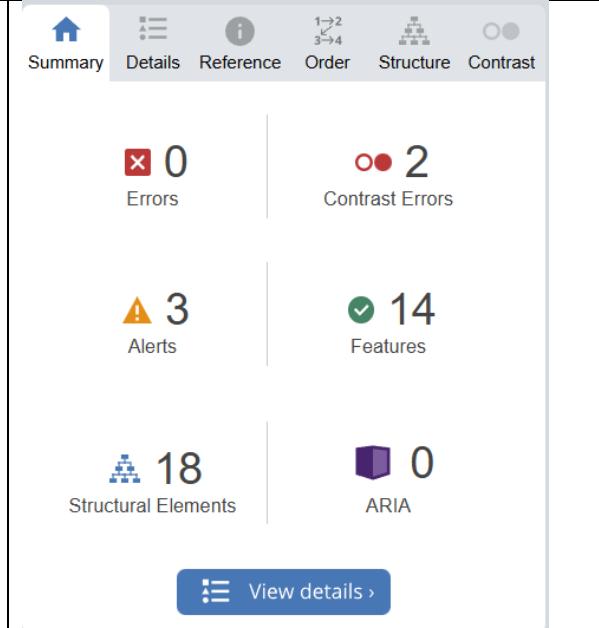
4.3.1. ORDER PAGE



4.3.2. CHECKOUT PAGE



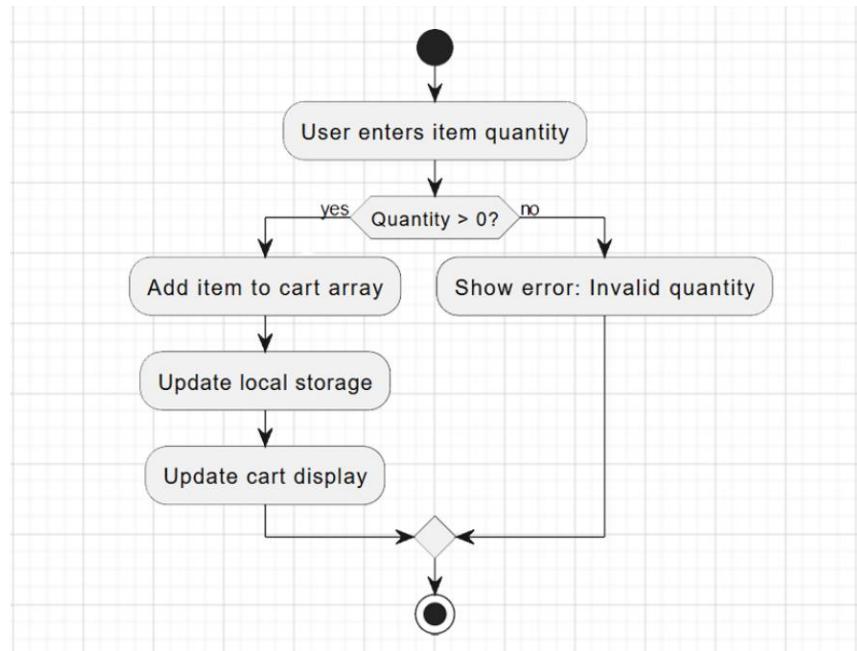
4.4. ACCESSIBILITY USING WAVE

	
<p>Order Page (Labels not being named, it does not suit the website hence they haven't been named)</p>	<p>Checkout Page (low contrast errors)</p>

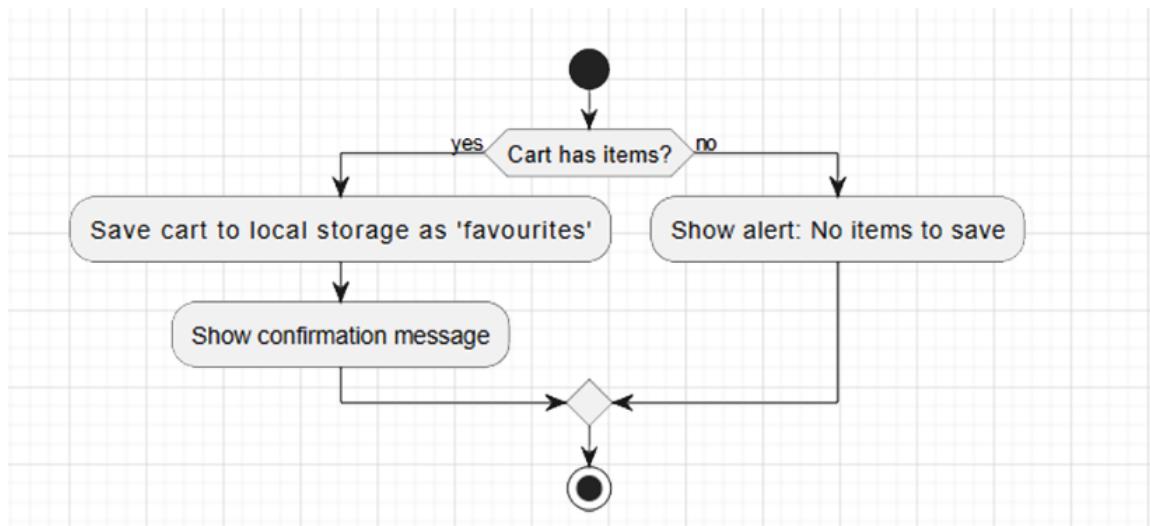
5. Diagrams

5.1. ACTIVITY DIAGRAMS

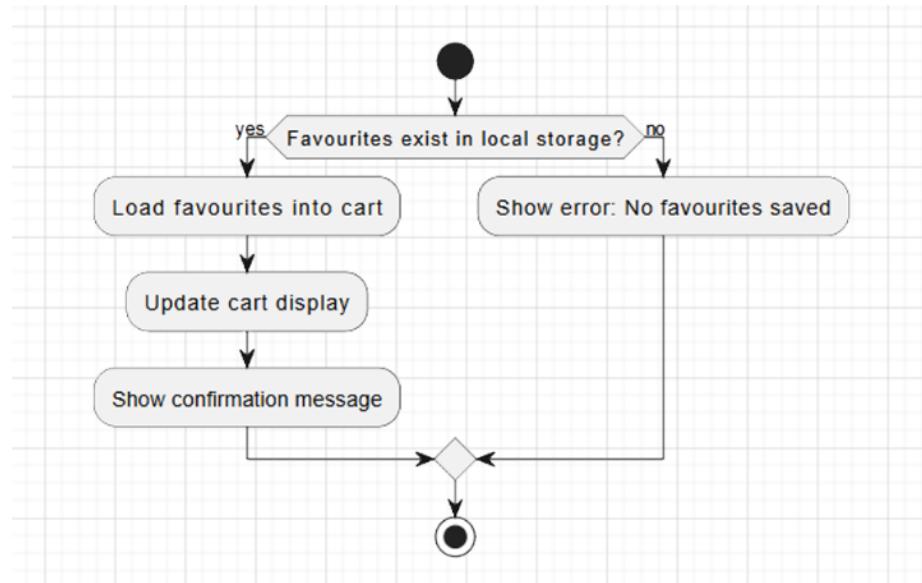
5.1.1. ADD TO CART FUNCTION



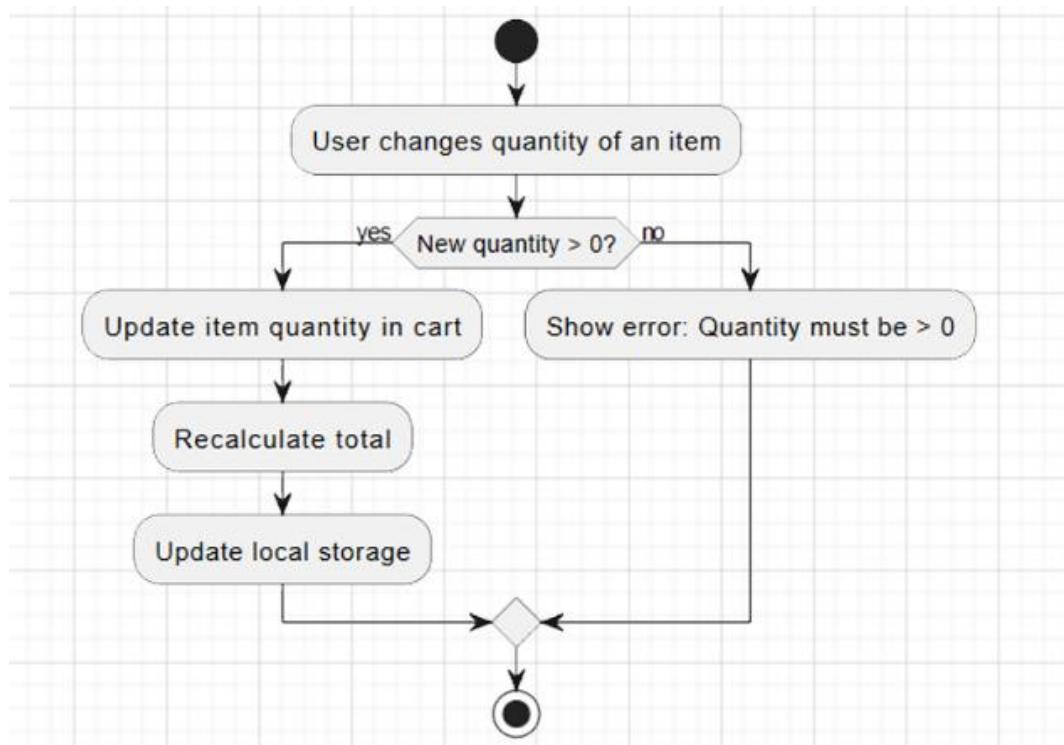
5.1.2. SAVE TO FAVOURITES



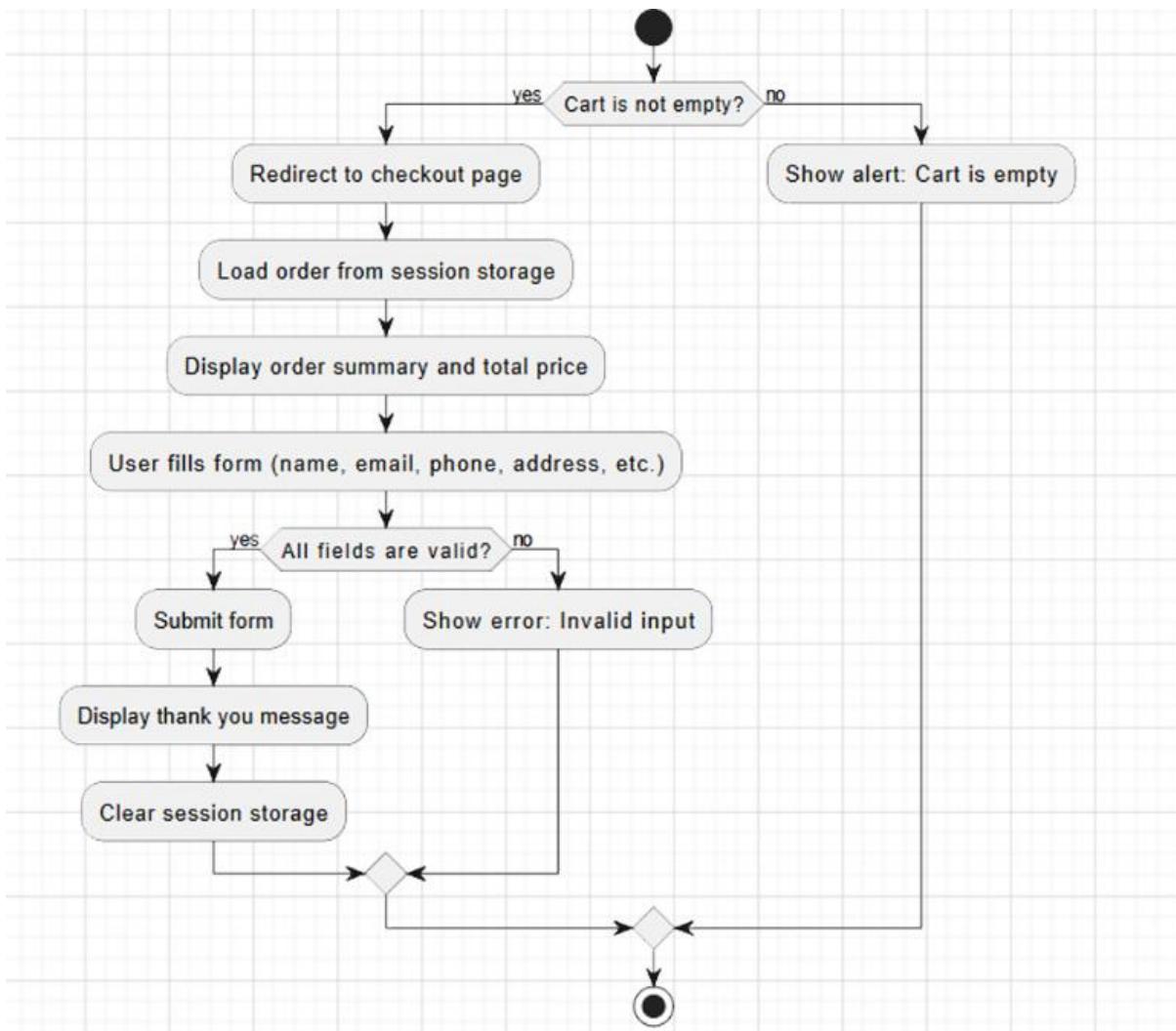
5.1.3. APPLY FAVOURITES



5.1.4. ADJUST QUANTITY IN CART

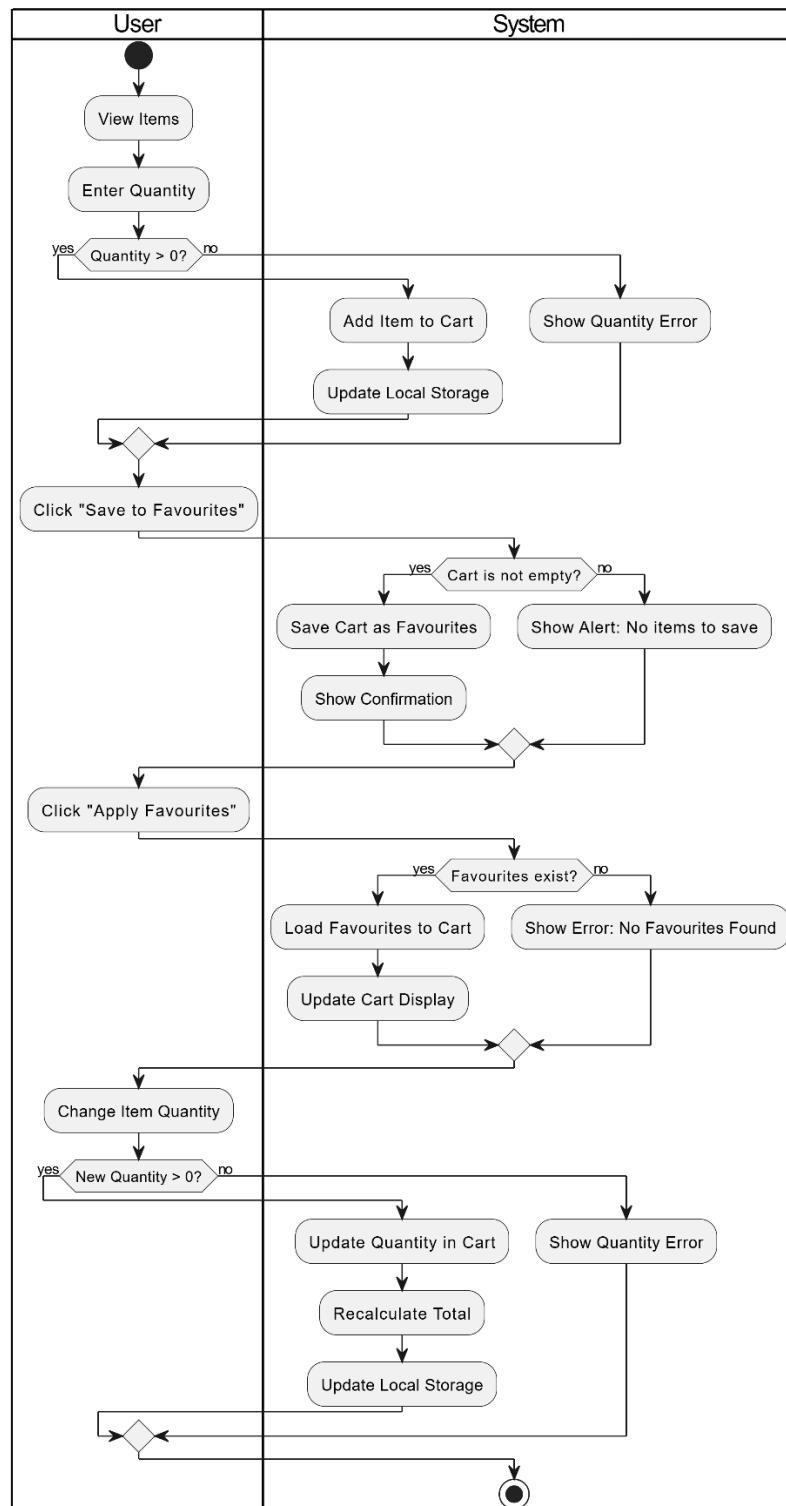


5.1.5. CHECKOUT PROCESS

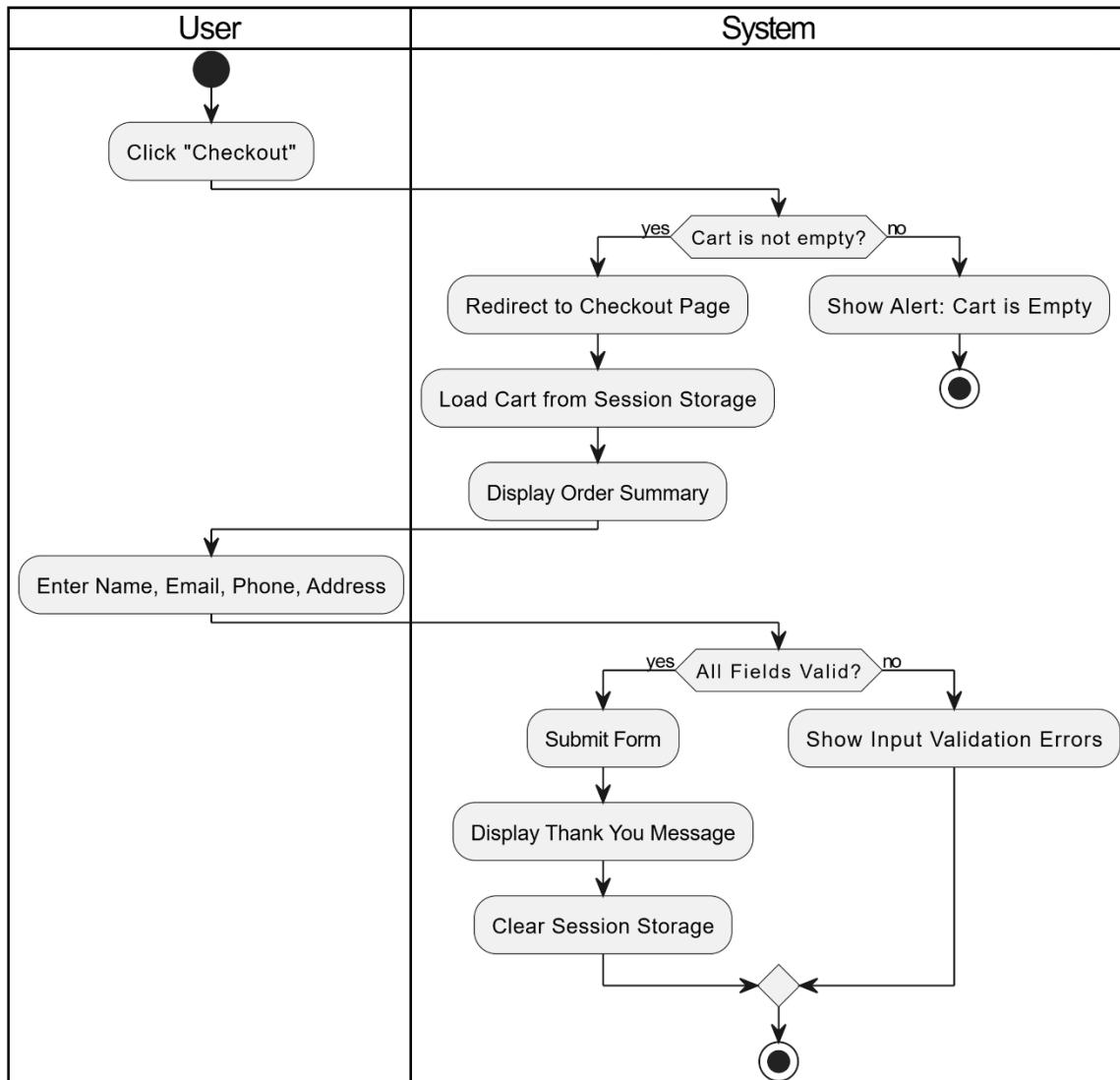


5.2. SWIM LANE DIAGRAMS

5.2.1. ORDERING PAGE DIAGRAM



5.2.2. CHECKOUT PAGE DIAGRAM



5.3. TEST CASE DIAGRAMS

Test No.	Test Case & Scenario	User Input	Expected Outcome	Actual Outcome	Status
TC01	Add item to cart with valid quantity	Select item, enter quantity 3, click "Add to Cart"	Item is added to cart and stored in local storage	Item successfully added and stored	Pass
TC02	Add item to cart with zero quantity	Select item, enter quantity 0, click "Add to Cart"	Alert shown: "Quantity must be greater than 0"	Alert appears	Pass
TC03	Add item to cart with negative quantity	Select item, enter quantity -2, click "Add to Cart"	Alert shown: "Quantity must be greater than 0"	Alert appears	Pass
TC04	Save order as favorites when cart has items	Cart has items, click "Save as Favorites"	Items are saved in local storage under favorites	Favorites saved	Pass
TC05	Save order as favorites with empty cart	Cart is empty, click "Save as Favorites"	Alert shown: "Cannot save an empty cart as favorites"	Alert appears	Pass
TC06	Apply saved favorites to cart	Click "Apply Favorites"	Previously saved favorites are loaded into cart	Cart is populated with saved favorite items	Pass
TC07	Check total value when items are added	Add items to cart	Total value is calculated and displayed	Total displayed correctly	Pass
TC08	Check total value when cart is empty	Ensure cart is empty	Total shows 0 or appropriate empty message	Total displayed as 0	Pass
TC09	Buy now with empty cart	Cart is empty, click "Buy Now"	Alert or prevention of redirection	Stays on same page, shows alert	Pass

TC10	Buy now with items in cart	Cart has items, click "Buy Now"	Redirects to payment page showing order summary	Successfully redirects and displays cart items	Pass
TC11	Order summary shows correct items	Add 2 items to cart, click "Buy Now"	All items from cart are shown in order summary	Order summary displays all correct items	Pass
TC12	Form validation - all details correct	Enter valid name, email, phone	On "Pay Now", shows success message	Success message shown	Pass
TC13	Form validation - invalid email	Enter invalid email, click "Pay Now"	Alert shown: "Enter a valid email address"	Alert appears	Pass
TC14	Form validation - empty required fields	Leave fields blank, click "Pay Now"	Alert: "Please fill out all required fields"	Alert appears	Pass
TC15	After successful payment, user gets confirmation message	Complete form and click "Pay Now"	Success message is shown	Message: "Payment successful!" appears	Pass

3. FACED CHALLENGES AND SOLUTIONS

Challenge:

Saving cart items as individual variables didn't make sense, so I had to figure out how to save an entire array of objects (items with name, quantity, price) into localStorage.

Solution:

I learned that I had to use `JSON.stringify()` to save the array and `JSON.parse()` to load it back. Before this I didn't know localStorage only handles strings.

Reference:

Youtu.be, 2025, <https://youtu.be/AUOzvFzdIk4?si=8vkPC43f0e2zICdI>. Accessed 25 Apr. 2025.

Challenge:

At first, clicking "Buy Now" would send the user to the checkout page even if they didn't pick anything.

Solution:

I added a check in JS to see if the `currentOrder` array was empty. If it was, I used an `alert()` and stopped the redirect with a `return`.

Challenge:

When changing quantities, the table and total price didn't update automatically, so the cart felt broken.

Solution:

I added `input.addEventListener('change', updateOrder)` to every quantity input so that anytime someone types a new number, it instantly updates the order summary and total.

4. REFERENCES

- ❖ Canva. (2020). *Canva*. [online] Available at: https://www.canva.com/design/DAGZFTTyKaY/7LPzUDI7R_DxVpETg_Rpqw/edit?locale=en
- ❖ iEatWebsites (2021). *How to Add a Custom Fonts to Your Website (HTML and CSS)*. [online] YouTube. Available at: <https://www.youtube.com/watch?v=AAU25Fo4bFY>
- ❖ www.dafont.com. (n.d.). *TT Octosquares Font / dafont.com*. [online] Available at: <https://www.dafont.com/tt-octosquares.font>.
- ❖ Karim (2009). *is it possible to fix html that has unescaped < and > characters?* [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/1936887/is-it-possible-to-fix-html-that-has-unescaped-and-characters>.
- ❖ www.youtube.com. (n.d.). *Learn HTML forms in 8 minutes*  [online] Available at: <https://www.youtube.com/watch?v=2O8pkybH6po>.
- ❖ W3C (2013). *The W3C Markup Validation Service*. [online] W3.org. Available at: <https://validator.w3.org/>.
- ❖ W3C (n.d.). *The W3C CSS Validation Service*. [online] W3.org. Available at: <https://jigsaw.w3.org/css-validator/>.
- ❖ WebAIM (2025). *WAVE Web Accessibility Tool*. [online] wave.webaim.org. Available at: <https://wave.webaim.org/>.
- ❖ cloudconvert.com. (n.d.). *JPG to WEBP / CloudConvert*. [online] Available at: <https://cloudconvert.com/jpg-to-webp>.