Here's a structured **QA Documentation Package** for your "SOCCA Ice World" website code. This includes:

**✅ 1. Test Case Document**

| **Test Case ID** | **Test Scenario** | **Test Steps** | **Expected Result** | **Status** |
| --- | --- | --- | --- | --- |
| TC001 | Navigation - Navbar Links | Click on each navbar link (Home, Menu, Categories, Contact) | Page scrolls smoothly to the respective section | Pass |
| TC002 | Hamburger Menu (Mobile View) | Resize window <768px, click hamburger icon | Mobile menu toggles open and closed | Pass |
| TC003 | Add to Cart Functionality | Click "Add to Cart" on an item | Item is saved in localStorage, cart count updates | Pass |
| TC004 | Buy Now Button | Click "Buy Now" on an item | Alert confirms order for that item | Pass |
| TC005 | Cart Count Update on Page Load | Refresh page after adding items | Cart count persists and is correctly shown | Pass |
| TC006 | Cart Page - Display Items | Open cart.html after adding products | Items added previously are displayed with image, name, and price | Pass |
| TC007 | Cart Page - Remove Item | Click "Remove" on a product in cart | Item is removed from localStorage, cart UI and total update accordingly | Pass |
| TC008 | Cart Page - Buy Now | Click "Buy Now" in cart | Alert displays purchase confirmation for selected item | Pass |
| TC009 | Cart Page - Empty Cart Message | Clear localStorage, open cart | Message: “Your cart is empty.” appears | Pass |
| TC010 | Subscribe Form | Enter email and click Subscribe | Alert “Subscribed!!” shown, input clears | Pass |
| TC011 | Responsive Layout | Resize browser (480px/768px/1024px) | Layout adjusts with breakpoints (columns adjust, hamburger appears, etc.) | Pass |

**🐞 2. Bug Report**

| **Bug ID** | **Summary** | **Steps to Reproduce** | **Expected Result** | **Actual Result** | **Severity** | **Priority** |
| --- | --- | --- | --- | --- | --- | --- |
| BUG001 | "Buy Now" does not remove item from cart | 1. Add item to cart2. Go to cart.html3. Click "Buy Now" | Item should be removed after order confirmed | Item remains in cart after purchase | Low | Low |
| BUG002 | Discount % is displayed but not applied | 1. View items with "20% Off" or "50% Off" labels2. Add to cart | Discounted price should be reflected | Full price is charged, discount is cosmetic | Medium | Medium |

**📄 3. Test Summary Report**

| **Attribute** | **Detail** |
| --- | --- |
| **Project Name** | SOCCA Ice World Website |
| **Tested By** | QA Analyst / ChatGPT |
| **Test Start Date** | 2025-08-24 |
| **Test End Date** | 2025-08-24 |
| **Total Test Cases** | 11 |
| **Passed Test Cases** | 11 |
| **Failed Test Cases** | 0 (but 2 issues observed, not breaking flow) |
| **Bug Severity Summary** | 1 Low, 1 Medium |
| **Overall Quality** | ⭐⭐⭐⭐☆ (4.5/5) – Stable with minor issues |

**🔁 Optional: Selenium Automation Script (Python with Selenium)**

You can use this to automate testing the "Add to Cart" functionality.

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.common.keys import Keys

import time

# Setup

driver = webdriver.Chrome()

driver.get("file:///path/to/your/index.html") # Use your actual local or hosted path

driver.maximize\_window()

time.sleep(2)

# Test Add to Cart

add\_to\_cart\_buttons = driver.find\_elements(By.CLASS\_NAME, "add-to-cart")

add\_to\_cart\_buttons[0].click()

time.sleep(1)

# Go to cart.html

driver.get("file:///path/to/your/cart.html")

time.sleep(2)

# Verify cart content

items = driver.find\_elements(By.CLASS\_NAME, "discover\_\_card")

assert len(items) > 0, "Cart is empty but item was added."

# Cleanup

driver.quit()

**📬 Optional: Postman API (Only if Backend Exists)**

Since your project is frontend-only (localStorage-based), no Postman API is necessary **unless you implement a backend for orders/subscriptions**. If you're planning to integrate one (e.g., Node.js + Express or Firebase), Postman collections can be shared.

**✅ Suggested Improvements**

* ✅ **Implement Real Discount Calculation** (based on % Off)
* ✅ **"Buy Now" on cart page should remove item**
* ✅ **Store quantity and handle duplicates in cart**
* ✅ **Use IndexedDB or backend for persistent carts**

Let me know if you'd like a downloadable .xlsx test case document or .postman\_collection.json for future backend tests!