# **Responsiveness Strategy Document**

### **Application: Angular E-Commerce Platform**

**Purpose**: This document defines strategies and best practices for ensuring the application is responsive across various screen sizes and devices.

## **🎯 Goals of Responsiveness**

* Optimize user experience across mobile, tablet, and desktop devices.
* Maintain functional parity across all screen sizes.
* Ensure accessibility and usability without horizontal scrolling or UI distortion.

## **🧱 Technology Stack**

| **Component** | **Tool/Library** |
| --- | --- |
| UI Framework | Bootstrap 5.x |
| Component Lib | Angular Components |
| Layout System | Bootstrap Grid |
| Styling | SCSS / CSS |
| Icons | Font Awesome / Bootstrap Icons |

## **📐 Breakpoints Used**

We follow Bootstrap's default breakpoints:

| **Device** | **Width Range** | **Bootstrap Class** |
| --- | --- | --- |
| Extra small | < 576px | col- |
| Small | ≥ 576px | col-sm- |
| Medium | ≥ 768px | col-md- |
| Large | ≥ 992px | col-lg- |
| Extra large | ≥ 1200px | col-xl- |
| XXL | ≥ 1400px | col-xxl- |

## **🧰 Implementation Guidelines**

### **✅ Grid Layouts**

* Use row and col-\*-\* classes to create responsive layouts.
* Prefer col-md-6 col-lg-4 col-xl-3 for product cards to stack differently on each screen size.

### **✅ Images**

* Use img-fluid to make images scale with the container.
* Set fixed height and object-fit: contain for product thumbnails.

### **✅ Buttons & Forms**

* Use w-100 on buttons for full-width buttons on mobile.
* Form elements stack vertically on small screens using form-group and grid classes.

### **✅ Spacing**

* Use responsive margin/padding: mt-2, mb-md-3, px-lg-5, etc.
* Avoid hardcoded values like px: 30px in favor of Bootstrap spacing utilities.

### **✅ Text and Headings**

* Use text-center, text-md-start for alignment across breakpoints.
* Use display-\*, h1-h6 with responsive sizing classes like fs-5, fs-md-3.

## **🔁 Dynamic Component Behavior**

* Use ngIf to show/hide sections based on screen size (e.g., navigation drawer).
* Use media queries in SCSS for fine control.

scss

CopyEdit

@media (max-width: 576px) {

.custom-nav {

display: none;

}

}

## **⏳ Loader Strategy**

* Show loader (spinner or skeleton) using Bootstrap spinners while:  
  + Fetching products
  + Authenticating user
  + Saving updates

html

CopyEdit

<div class="text-center" \*ngIf="loading">

<div class="spinner-border text-primary" role="status">

<span class="visually-hidden">Loading...</span>

</div>

</div>

## **📱 Mobile-First Design Principles**

* Design for mobile first and scale up to desktop.
* Optimize for touch (larger buttons, no hover-only interactions).
* Avoid modals/popup-heavy UI.

## **🧪 Testing Strategy**

* Test layouts on Chrome DevTools across:  
  + iPhone SE, iPhone 14
  + iPad
  + Galaxy S20
  + 13" and 15" Laptops
  + 1920x1080 desktops
* Use tools like:  
  + BrowserStack
  + Responsively App
  + Lighthouse audits

## **🧩 Component Responsiveness Summary**

| **Component** | **Strategy** |
| --- | --- |
| Header/Navbar | Collapsible menu using navbar-expand-\* |
| Product Cards | Bootstrap grid (col-md-6 col-lg-4) |
| Forms | Responsive form-group layout |
| Modals | Scrollable and mobile-friendly |
| Footer | Stack content vertically on small screens |