**Exercise: Design Patterns**

**Create a Reusable Card UI Component**

Build a card component that is visually consistent, responsive, and reusable for any content type (e.g., product, blog post, service offering).

* Implement a scalable card layout using CSS Grid or Flexbox.
* Maintain accessibility and responsive behavior.
* Reuse the card component across different sections.

**HTML:**

<div class="card">

<img src="https://via.placeholder.com/300x180" alt="Sample Image" />

<div class="card-content">

<h3>Card Title</h3>

<p>This is a short card description. It's responsive and reusable.</p>

<button class="btn">Learn More</button>

</div>

</div>

**CSS / SCSS:**

.card {

display: flex;

flex-direction: column;

border: 1px solid #ddd;

border-radius: 0.5rem;

overflow: hidden;

background: #fff;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

max-width: 320px;

transition: transform 0.3s ease;

&:hover {

transform: translateY(-5px);

}

img {

width: 100%;

object-fit: cover;

}

.card-content {

padding: 1rem;

h3 {

font-size: 1.25rem;

margin-bottom: 0.5rem;

}

p {

font-size: 1rem;

color: #555;

}

.btn {

margin-top: 1rem;

background: #4F46E5;

color: white;

padding: 0.5rem 1rem;

border: none;

border-radius: 0.25rem;

cursor: pointer;

&:hover {

background: #4338ca;

}

}

}

}

**Testing:**

1. Add multiple cards to a responsive grid layout.
2. Test responsiveness on mobile, tablet, and desktop.
3. Use keyboard and screen reader to check accessibility.
4. Refactor into a reusable component in your codebase (function, class, or template).

**Quiz:**

* What parts of the card layout could be turned into a reusable SCSS mixin or component?
* How would you convert this into a component in React, Angular, or another framework?
* How does this design handle varying amounts of content?

**Accessibility Enhancements**

Add semantic elements and ARIA attributes:

* <article> is a landmark element for screen readers.
* aria-labelledby and aria-label improve navigation context for assistive tech.

**<article** class="card" role="group" **aria-labelledby**="card-title-1">

<img src="..." alt="Sample of a product" />

<div class="card-content">

<h3 id="card-title-1">Card Title</h3>

<p>This is a short card description.</p>

<button class="btn" aria-label="Learn more about Card Title">Learn More</button>

</div>

**</article>**

**Reusable SCSS Mixins/Components**

You can extract shared styles like this:

@mixin button-style($bg, $hover-bg) {

background: $bg;

color: white;

padding: 0.5rem 1rem;

border: none;

border-radius: 0.25rem;

cursor: pointer;

&:hover {

background: $hover-bg;

}

}

Usage:

.btn {

@include button-style(#4F46E5, #4338ca);

}

**Testing Checklist**

|  |  |
| --- | --- |
| **Test** | **Expected Result** |
| **Resize screen** | Cards stack or realign fluidly |
| **Tab key navigation** | Focus flows to button in each card |
| **Screen reader** | Can read title, description, and button clearly |
| **Mobile touch** | Tap feedback works, layout remains intact |
| **Content overflow** | Longer titles/descriptions don’t break layout |

**Quiz Answers**

1. **What parts could become SCSS mixins or components?**
   * Button styling
   * Card shadow and padding
   * Grid layout behavior
   * Typography rules
2. **How would you convert this into a component?**
   * Create a function or class in JS/TS
   * Use slots or props for content
   * Isolate styling via CSS Modules or component-scoped styles
3. **How does the design handle varying content?**
   * Flexible height layout supports longer/shorter text
   * object-fit: cover ensures image consistency
   * Responsive behavior allows reflow instead of truncate