

# Experiment 3

## Title

Responsive web design and animations

## Aim

To develop a responsive multi-page website using media queries to ensure compatibility across different screen sizes and devices, and to make it dynamic using animations.

## Software

Code Editor ([VSCode](#)/[Notepad++](#))

Web Browser ([Firefox](#)/[Chrome](#))

## Theory

Responsive Web Design is about creating web pages that look good on all devices. We will use HTML and CSS to automatically resize, hide, shrink, or enlarge a website to make it look good on any device (desktops, tablets, and phones).

## The Viewport Meta Tag

- The viewport is the user's visible area of a web page. Without this tag, mobile browsers will render the page as if it were on a desktop (often 980px wide) and then shrink it down, making text tiny.
- **Code:** `<meta name="viewport" content="width=device-width, initial-scale=1.0">`
  - `width=device-width`: Sets the page width to follow the screen-width of the device.
  - `initial-scale=1.0`: Sets the initial zoom level when the page is first loaded.

## Media Queries

Media queries allow you to apply CSS rules only if certain conditions are true.

**@media screen (min-width: 320px) and (max-width: 768px)**

AT-RULE

MEDIA TYPE

MEDIA FEATURE

OPERATOR

MEDIA FEATURE

- **Common Breakpoints:**

- *Extra small (phones): < 600px*
- *Small (tablets): ≥ 600px*
- *Medium (laptops): ≥ 768px*
- *Large (desktops): ≥ 992px*

```
@media only screen and (max-width: 600px) {  
    body { background-color: lightblue; }  
}
```

[CSS Tricks detailed reference](#)

## Accessibility

Accessibility is the practice of making your websites usable by as many people as possible. We traditionally think of this as being about people with disabilities, but the practice of making sites accessible also benefits other groups such as those using mobile devices, or those with slow network connections.

Further references: [MDN docs W3C Accessibility Tips](#)

Include 3 different accessibility tips from the above W3C link in your multi-page website and mention which tips you applied in your submission.

### ACCESSIBILITY TIPS APPLIED IN LAB 3:

1. Provide sufficient contrast between foreground and background : Task 2.1 - Changed the Submit Data button from light gray (#ccc) to dark blue (#0d6efd) to meet WCAG contrast requirements.
2. Provide visible focus indication for keyboard users : Task 2.2 - Added outline and outline-offset to the button's : focus state so keyboard users can see where they are.
3. Provide meaningful text alternatives for image : Task 2.4 - Added alt="A cute

kitten" to the image so screen readers can describe it to visually impaired users.

## Animation

To make an animation work, you need two main ingredients:

1. The Timeline (**@keyframes**): This defines *what* happens at specific moments.
2. The Configuration (Animation Properties): This tells the element *how* to play that timeline (speed, loops, delay, etc.)

When defining the **@keyframes** you give the animation a name and use percentages to mark points in time.

- **0% (or from)**: The start of the animation.
- **100% (or to)**: The end of the animation.
- **In-between (e.g., 50%)**: What happens in the middle.

Once you have written the keyframes, nothing happens until you assign that animation to an HTML element. In the CSS selector for the element to be animated, include different animation-\* properties that you want to change:

- Which animation name to use.
- How long it should take (**duration**).
- How many times it should repeat (**iteration-count**).

## [CSS Tricks Reference](#)

## Conclusion

We have successfully applied responsive and accessibility-oriented web design principles to a dynamic multi-page website.

## Tasks

# 1.

```
/* 1.1 Custom Media Query */
.mobile-alert {
    display: none; /* Hidden by default on Desktop */
    background-color: #fee2e2;
    color: #b91c1c;
    padding: 10px;
    text-align: center;
    border: 1px solid #f87171;
    margin-bottom: 10px;
}

/* TODO 1.1: Write a Media Query for Mobile */
/* For mobile screens that have max-width 576px. Change the display property so the mobile-alert becomes visible. */
@media (max-width: 576px) {
    .mobile-alert { display: block; }
}

/* 1.2 Flexbox Responsive Menu */
.nav-menu {
    list-style: none; padding: 0; margin: 0;
    background-color: var(--dark);
    display: flex; /* Flex row by default */
    justify-content: space-around;
}
.nav-menu li a { display: block; padding: 15px; color: white; text-decoration: none; }

/* TODO 1.2: Stack Menu Vertically on Mobile */
/* Inside a media query (max-width: 600px):
   1. Flex direction should change when device is phone.
   2. Now the vertical stacking will work, but all the different boxes mix into one another.
      Target appropriate element and add a border to show each option as distinct.
*/
@media (max-width: 600px) {
    .nav-menu {
        flex-direction: column;
    }
    .nav-menu li {
        border-bottom: 1px solid #475569;
    }
}
```

```
/* 1.5 Device Breakpoints (NEW) */
.device-box {
    background-color: #e2e8f0; /* Default (Mobile) */
    padding: 20px;
    text-align: center;
    border: 2px solid #94a3b8;
    margin-bottom: 10px;
    font-weight: bold;
    color: #475569;
}

/* TODO 1.5: Tablet & Desktop Breakpoints */
/* 1. Create a media query for Tablet (min-width: 768px).
   Inside it, change .device-box background-color to yellow */
@media (min-width: 768px) {
    .device-box { background-color: yellow; }
}

/* 2. Create a media query for Desktop (min-width: 992px).
   Inside it, change .device-box background-color to green */
@media (min-width: 992px) {
    .device-box { background-color: green; }
}

/* 1.6 Screen Orientation */
.landscape-box {
    background-color: #ffddd7;
    color: #881337;
    padding: 15px;
    text-align: center;
    border: 2px dashed #fb7185;
    display: none; /* Hidden by default (Portrait) */
    font-weight: bold;
}

/* TODO 1.6: Orientation Media Query */
/* Write a media query for (orientation: landscape).
   Inside it, change display property to make it visible */
@media (orientation: landscape) {
    .landscape-box { display: block; }
}
```

## TASK 1: RESPONSIVE DESIGN

# 1. Media Queries & Flexbox

**SUCCESS (1.1):** You are viewing this on a mobile screen width!

**Subtask 1.1:** Resize your browser window to be very small. The red alert box above should appear only when narrow.

---

**Subtask 1.2:** Fix the menu below. On mobile, these items should stack vertically instead of squishing horizontally.

Home

Products

About

Contact

## Bootstrap Grid System

For these tasks, modify the HTML classes below directly. No CSS writing needed.

**Subtask 1.3:** The three boxes below are currently stacked (block elements). Use Bootstrap classes `row` (on parent) and `col-4` (on children) to make them sit side-by-side in equal columns.



**Subtask 1.4:** Responsive Utilities. Hide the box below on mobile phones but show it on desktops using Bootstrap classes.

I should disappear on mobile screens!

## Advanced Media Queries

**Subtask 1.5:** Breakpoints. Write CSS media queries so the box below changes color:

Default (Gray) → Tablet (Yellow) → Desktop (Green)

Resize window to see color change!

**Subtask 1.6:** Orientation. Write a CSS media query so the box below *only* appears when the screen is in Landscape mode.

I am visible because you are in LANDSCAPE mode.

2.

```
● ● ●

/* --- TASK 2: ACCESSIBILITY --- */

.btn-accessible {
    /* TODO 2.1: Fix Contrast */
    /* Current: #ccc background with white text fails WCAG (Web Content Accessibility Guidelines). */
    /* Change background-color to a darker blue (e.g., #0d6efd) */
    background-color: #0d6efd;

    color: white; border: none; padding: 10px 20px;
    font-size: 1rem; cursor: pointer; border-radius: 4px;
}

/* TODO 2.2: Add Focus State */
/* Target .btn-accessible using pseudo-class */
/* Add outline and outline-offset - This helps keyboard users see where they are. */
.btn-accessible:focus {
    outline: 2px solid #0d6efd;
    outline-offset: 2px;
}
```

**TASK 2: ACCESSIBILITY**

## 2. Visual & Assistive A11y

**Subtask 2.1 & 2.2:** The button below has poor contrast and no focus state. Fix this in the CSS block.

Submit Data

**Subtask 2.3: ARIA Labels.** The button below only contains an icon (X). A screen reader doesn't know what it does. Add an `aria-label="Close"` attribute to the HTML tag below.

X

**Subtask 2.4: Alt Text.** The image below is missing an alt attribute. Add `alt="A cute kitten"` so screen readers can pick up the image description.

LinkedIn Post

3.

**TASK 3: ANIMATION**

## 3. Transitions & Keyframes

**Subtask 3.1:** Simple Hover. Make this card scale up slightly (zoom in) when you hover over it using `transition` and `transform`.

Hover Me

**Subtask 3.2 & 3.3:** Infinite Spinner. Define the keyframes in CSS and apply the animation to the circle below.



**Subtask 3.4:** Pulse. Create a 'pulse' keyframe in CSS (opacity changing) and apply it to this red dot.



```
● ○ ●

/* --- TASK 3: ANIMATION --- */

/* 3.1 Hover Transition */
.anim-card {
    background: white; border: 1px solid #ddd; padding: 20px;
    width: 150px; text-align: center; cursor: pointer;

    /* TODO 3.1: Add Transition */
    /* Add 'transition: transform 0.3s ease;' */
    transition: transform 0.3s ease;
}

/* TODO 3.1 (Cont): Add Transform to .anim-card on Hover */
/* Add 'transform: scale(1.1);' */
.anim-card:hover {
    transform: scale(1.1);
}

/* 3.2 Loading Spinner Keyframes */
.spinner {
    width: 40px; height: 40px;
    border: 4px solid #f3f3f3;
    border-top: 4px solid var(--primary);
    border-radius: 50%;
}

/* TODO 3.2: Define Keyframes */
/* Create @keyframes named 'spin' */
@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}
/* 0% { transform: rotate(0deg); } -> 100% { transform:rotate(360deg); }*/

/* TODO 3.3: Apply Animation */
/* Target .spinner class above (or write a new rule) */
/* Add 'animation: spin 1s linear infinite;' */
.spinner {
    animation: spin 1s linear infinite;
}

/* TODO 3.4: Pulse Animation */
.pulse-circle {
    width: 30px; height: 30px; background-color: red; border-radius: 50%;
}
/* Define @keyframes named 'pulse' that changes opacity from 1 to 0.5 */
/* Apply it to .pulse-circle with 1s duration, infinite alternate */
@keyframes pulse {
    0% { opacity: 1; }
    100% { opacity: 0.5; }
}

.pulse-circle {
    animation: pulse 1s infinite alternate;
}
```

## 4.1

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Lab 3: Responsive, Ally & Animation</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
        <link rel="stylesheet" href="style.css">
</head>
<body>

    <header>
        <h1>Lab 3: Responsive Web Design and Animations</h1>
        <p class="lead">Complete the tasks below. Edit the CSS in the <code>&lt;head&gt;</code> and HTML body where indicated.</p>
        <br>
    </header>

    <section>
        <div class="task-container">
            <span class="task-badge">Task 1: Responsive Design</span>
            <h3>1. Media Queries & Flexbox</h3>

            <div class="mobile-alert">
                <strong>SUCCESS (1.1):</strong> You are viewing this on a mobile screen width!
            </div>
            <p><strong>Subtask 1.1:</strong> Resize your browser window to be very small. The red alert box above should appear only when narrow.</p>

            <hr>
            <p><strong>Subtask 1.2:</strong> Fix the menu below. On mobile, these items should stack vertically instead of squishing horizontally.</p>
            <nav>
                <ul class="nav-menu">
                    <li><a href="lab3(1).html">Home</a></li>
                    <li><a href="products.html">Products</a></li>
                    <li><a href="about.html">About</a></li>
                    <li><a href="contact.html">Contact</a></li>
                </ul>
            </nav>

            <hr>
            <h3>Bootstrap Grid System</h3>
            <p class="small text-muted">For these tasks, modify the HTML classes below directly. No CSS writing needed.</p>
        </div>
    </section>

```

**<p><strong>Subtask 1.3:</strong>** The three boxes below are currently stacked (block elements). Use Bootstrap classes `<code>row</code>` (on parent) and `<code>col-4</code>` (on children) to make them sit side-by-side in equal columns.</p>

```
<div class="row">

    <div class="col-4">
        <div class="demo-box">Column 1</div>
    </div>
    <div class="col-4">
        <div class="demo-box">Column 2</div>
    </div>
    <div class="col-4">
        <div class="demo-box">Column 3</div>
    </div>
</div>
```

**<p class="mt-3"><strong>Subtask 1.4:</strong>** Responsive Utilities. Hide the box below on mobile phones but show it on desktops using Bootstrap classes.</p>

```
<div class="demo-box bg-warning text-dark d-none d-md-block">
    I should disappear on mobile screens!
</div>
```

```
<hr>
<h3>Advanced Media Queries</h3>
```

**<p><strong>Subtask 1.5:</strong>** Breakpoints. Write CSS media queries so the box below changes color: <br>

```
<small>Default (Gray) &rarr; Tablet (Yellow) &rarr; Desktop (Green)</small></p>
<div class="device-box">Resize window to see color change!</div>
```

**<p class="mt-4"><strong>Subtask 1.6:</strong>** Orientation. Write a CSS media query so the box below <em>only</em> appears when the screen is in Landscape mode.</p>

```
<div class="landscape-box">I am visible because you are in LANDSCAPE mode.</div>
```

```
</div>
</section>
```

```
<section>
```

```
    <div class="task-container">
        <span class="task-badge">Task 2: Accessibility</span>
        <h3>2. Visual & Assistive Ally</h3>
```

**<p><strong>Subtask 2.1 & 2.2:</strong>** The button below has poor contrast and no focus state. Fix this in the CSS block.</p>

```
<button class="btn-accessible">Submit Data</button>
```

```
<hr>
```

**<p><strong>Subtask 2.3: ARIA Labels.</strong>** The button below only contains an icon (X). A screen reader doesn't know what it does. Add an `<code>aria-label="Close"</code>` attribute to the HTML tag below.</p>

```

<button aria-label="Close" style="padding: 5px 10px; font-weight: bold;">&times;

<hr>
<p><strong>Subtask 2.4: Alt Text.</strong> The image below is missing an alt attribute. Add <code>alt="A cute kitten"</code> so screen readers can pick up the image description.</p>


</div>
</section>

<section>
<div class="task-container">
<span class="task-badge">Task 3: Animation</span>
<h3>3. Transitions & Keyframes</h3>

<p><strong>Subtask 3.1:</strong> Simple Hover. Make this card scale up slightly (zoom in) when you hover over it using <code>transition</code> and <code>transform</code>.</p>
<div class="anim-card">Hover Me</div>

<hr>
<p><strong>Subtask 3.2 & 3.3:</strong> Infinite Spinner. Define the keyframes in CSS and apply the animation to the circle below.</p>
<div class="spinner"></div>

<hr>
<p><strong>Subtask 3.4:</strong> Pulse. Create a 'pulse' keyframe in CSS (opacity changing) and apply it to this red dot.</p>
<div class="pulse-circle"></div>
</div>
</section>

<section>
<div class="task-container">
<span class="task-badge">Task 4: Multi-page Site</span>
<h3>4. Site Structure & Navigation</h3>

<p><strong>Subtask 4.1: External CSS.</strong> Currently, all styles are in the <code>&lt;head&gt;</code> of this file. To share these responsive styles with the new pages you are about to create, you must move them to an external file.</p>
<ol>
<li>Create a new file named <code>style.css</code> in the same folder.</li>
<li>Cut everything <em>inside</em> the <code>&lt;style&gt;</code> tags above (from <code>:root</code> down to the end of animations) and paste it into <code>style.css</code>.</li>
<li>Delete the <code>&lt;style&gt;</code> tags from this HTML file.</li>
<li>Link the new CSS file here using: <code>&lt;link rel="stylesheet" href="style.css"&gt;</code>.</li>
</ol>
</p>

<hr>

<p><strong>Subtask 4.2: Create Pages.</strong> Create three new HTML files in the same folder: <code>products.html</code>, <code>about.html</code>, and <code>contact.html</code>. <br> Copy the HTML structure (Head, Navbar, Footer) to these files so they share the same layout and are <strong>responsive</strong> (thanks to the external CSS).</p>
</p>

<hr>

<p><strong>Subtask 4.3: Navigation.</strong> Update the navigation menu in <strong>ALL</strong> four files (including this one, which acts as "Home") so the links work properly.<br> Current code: <code>&lt;a href="#"&gt;</code><br> Expected code example: <code>&lt;a href="products.html"&gt;Products&lt;/a&gt;</code></p>
</div>
</section>

<footer style="margin-top: 50px; text-align: center; color: #666;">
<p>Web Programming, Nishant Shankar &copy; 2026</p>
</footer>

</body>
</html>

```

```
/* =====
   BASE STYLES (Do not modify this section)
===== */
:root {
  --primary: #3b82f6;
  --dark: #1e293b;
  --light: #f8fafc;
}

body {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  line-height: 1.6;
  color: var(--dark);
  max-width: 960px;
  margin: 0 auto;
  padding: 20px;
  background-color: var(--light);
}

h1 { border-bottom: 2px solid var(--primary); padding-bottom: 10px; margin-bottom: 30px; }
section { margin-bottom: 50px; }

.concept-card {
  background: white;
  border-left: 5px solid var(--primary);
  padding: 15px 20px;
  margin-bottom: 20px;
  box-shadow: 0 2px 4px rgba(0,0,0,0.05);
}

code { background-color: #eee; padding: 2px 5px; border-radius: 4px; color: #d63384; font-size: 0.9em; }

.task-container {
  background: #fff;
  border: 2px dashed #cbd5e1;
  padding: 25px;
  border-radius: 8px;
  position: relative;
}

.task-badge {
  position: absolute;
```

```
    top: -15px;
    left: 20px;
    background: var(--dark);
    color: white;
    padding: 5px 15px;
    font-weight: bold;
    border-radius: 20px;
    text-transform: uppercase;
    font-size: 0.8rem;
}

/* Helper for Bootstrap Visuals */
.demo-box {
    background-color: #e0e7ff;
    border: 1px solid #6366f1;
    padding: 15px;
    text-align: center;
    font-weight: bold;
    color: #312e81;
    margin-bottom: 10px;
}

/* =====
STUDENT WORKSPACE
Complete the TODOs below.
===== */

/* --- TASK 1: RESPONSIVE DESIGN & BOOTSTRAP --- */

/* 1.1 Custom Media Query */
.mobile-alert {
    display: none; /* Hidden by default on Desktop */
    background-color: #fee2e2;
    color: #b91c1c;
    padding: 10px;
    text-align: center;
    border: 1px solid #f87171;
    margin-bottom: 10px;
}

/* TODO 1.1: Write a Media Query for Mobile */
/* For mobile screens that have max-width 576px. Change the display property so the mobile-alert becomes visible. */
@media (max-width: 576px) {
    .mobile-alert { display: block; }
}
```

```
/* 1.2 Flexbox Responsive Menu */
.nav-menu {
    list-style: none; padding: 0; margin: 0;
    background-color: var(--dark);
    display: flex; /* Flex row by default */
    justify-content: space-around;
}
.nav-menu li a { display: block; padding: 15px; color: white; text-decoration: none; }

/* TODO 1.2: Stack Menu Vertically on Mobile */
/* Inside a media query (max-width: 600px):
   1. Flex direction should change when device is phone.
   2. Now the vertical stacking will work, but all the different boxes mix into one another.
      Target appropriate element and add a border to show each option as distinct.
*/
@media (max-width: 600px) {
    .nav-menu {
        flex-direction: column;
    }
    .nav-menu li {
        border-bottom: 1px solid #475569;
    }
}

/* 1.5 Device Breakpoints (NEW) */
.device-box {
    background-color: #e2e8f0; /* Default (Mobile) */
    padding: 20px;
    text-align: center;
    border: 2px solid #94a3b8;
    margin-bottom: 10px;
    font-weight: bold;
    color: #475569;
}

/* TODO 1.5: Tablet & Desktop Breakpoints */
/* 1. Create a media query for Tablet (min-width: 768px).
   Inside it, change .device-box background-color to yellow */
@media (min-width: 768px) {
    .device-box { background-color: yellow; }
}
```

```
/* 2. Create a media query for Desktop (min-width: 992px).
   Inside it, change .device-box background-color to green */
@media (min-width: 992px) {
    .device-box { background-color: green; }
}

/* 1.6 Screen Orientation */
.landscapex-box {
    background-color: #ffddd7;
    color: #881337;
    padding: 15px;
    text-align: center;
    border: 2px dashed #fb7185;
    display: none; /* Hidden by default (Portrait) */
    font-weight: bold;
}

/* TODO 1.6: Orientation Media Query */
/* Write a media query for (orientation: landscape).
   Inside it, change display property to make it visible */
@media (orientation: landscape) {
    .landscapex-box { display: block; }
}

/* --- TASK 2: ACCESSIBILITY --- */

.btn-accessible {
    /* TODO 2.1: Fix Contrast */
    /* Current: #ccc background with white text fails WCAG (Web Content Accessibility Guidelines). */
    /* Change background-color to a darker blue (e.g., #0d6efd) */
    background-color: #0d6efd;

    color: white; border: none; padding: 10px 20px;
    font-size: 1rem; cursor: pointer; border-radius: 4px;
}

/* TODO 2.2: Add Focus State */
/* Target .btn-accessible using pseudo-class */
/* Add outline and outline-offset - This helps keyboard users see where they are. */
.btn-accessible:focus {
    outline: 2px solid #0d6efd;
    outline-offset: 2px;
}
```

```

/* --- TASK 3: ANIMATION --- */

/* 3.1 Hover Transition */
.anim-card {
    background: white; border: 1px solid #ddd; padding: 20px;
    width: 150px; text-align: center; cursor: pointer;

    /* TODO 3.1: Add Transition */
    /* Add 'transition: transform 0.3s ease;' */
    transition: transform 0.3s ease;
}

/* TODO 3.1 (Cont): Add Transform to .anim-card on Hover */
/* Add 'transform: scale(1.1);' */
.anim-card:hover {
    transform: scale(1.1);
}

/* 3.2 Loading Spinner Keyframes */
.spinner {
    width: 40px; height: 40px;
    border: 4px solid #f3f3f3;
    border-top: 4px solid var(--primary);
    border-radius: 50%;
}

/* TODO 3.2: Define Keyframes */
/* Create @keyframes named 'spin' */
@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}
/* 0% { transform: rotate(0deg); } -> 100% { transform: rotate(360deg); } */

/* TODO 3.3: Apply Animation */
/* Target .spinner class above (or write a new rule) */
/* Add 'animation: spin 1s linear infinite;' */
.spinner {
    animation: spin 1s linear infinite;
}

/* TODO 3.4: Pulse Animation */
.pulse-circle {
    width: 30px; height: 30px; background-color: red; border-radius: 50%;
}
/* Define @keyframes named 'pulse' that changes opacity from 1 to 0.5 */
/* Apply it to .pulse-circle with 1s duration, infinite alternate */
@keyframes pulse {
    0% { opacity: 1; }
    100% { opacity: 0.5; }
}

.pulse-circle {
    animation: pulse 1s infinite alternate;
}

```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>About - Lab 3</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
        <link rel="stylesheet" href="style.css">
</head>
<body>

    <header>
        <h1>About Us</h1>
        <p class="lead">Learn more about our company</p>
        <br>
    </header>

    <nav>
        <ul class="nav-menu">
            <li><a href="lab3(1).html">Home</a></li>
            <li><a href="products.html">Products</a></li>
            <li><a href="about.html">About</a></li>
            <li><a href="contact.html">Contact</a></li>
        </ul>
    </nav>

    <section>
        <div class="task-container">
            <h2>About Our Company</h2>
            <p>Welcome to the about page. This page shares the same responsive design as the home page thanks to the external CSS file.</p>

            <div class="concept-card">
                <h3>Our Mission</h3>
                <p>To provide excellent web development education and resources.</p>
            </div>

            <div class="concept-card">
                <h3>Our Team</h3>
                <p>We are a dedicated team of web developers passionate about teaching.</p>
            </div>

            <div class="concept-card">
                <h3>Our Values</h3>
                <p>Quality, accessibility, and responsive design are at the core of everything we do.</p>
            </div>
        </div>
    </section>

    <footer style="margin-top: 50px; text-align: center; color: #666;">
        <p>Web Programming, Nishant Shankar © 2026</p>
    </footer>

</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Contact - Lab 3</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
        <link rel="stylesheet" href="style.css">
</head>
<body>

    <header>
        <h1>Contact Us</h1>
        <p class="lead">Get in touch with us</p>
        <br>
    </header>

    <nav>
        <ul class="nav-menu">
            <li><a href="lab3(1).html">Home</a></li>
            <li><a href="products.html">Products</a></li>
            <li><a href="about.html">About</a></li>
            <li><a href="contact.html">Contact</a></li>
        </ul>
    </nav>

    <section>
        <div class="task-container">
            <h2>Contact Information</h2>
            <p>Welcome to the contact page. This page shares the same responsive design as the home page thanks to the external CSS file.</p>
            <div class="concept-card">
                <h3>Email</h3>
                <p>contact@example.com</p>
            </div>

            <div class="concept-card">
                <h3>Phone</h3>
                <p>+1 (555) 123-4567</p>
            </div>

            <div class="concept-card">
                <h3>Address</h3>
                <p>123 Web Street<br>Internet City, IC 12345</p>
            </div>

            <div class="concept-card">
                <h3>Contact Form</h3>
                <form>
                    <div class="mb-3">
                        <label for="name" class="form-label">Name</label>
                        <input type="text" class="form-control" id="name" required>
                    </div>
                    <div class="mb-3">
                        <label for="email" class="form-label">Email</label>
                        <input type="email" class="form-control" id="email" required>
                    </div>
                    <div class="mb-3">
                        <label for="message" class="form-label">Message</label>
                        <textarea class="form-control" id="message" rows="3" required></textarea>
                    </div>
                    <button type="submit" class="btn-accessible">Send Message</button>
                </form>
            </div>
        </div>
    </section>
```

```
<footer style="margin-top: 50px; text-align: center; color: #666;">
    <p>Web Programming, Nishant Shankar &copy; 2026</p>
</footer>

</body>
</html>
```

```
● ● ●

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Products - Lab 3</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <header>
        <h1>Products</h1>
        <p class="lead">Our product catalog</p>
        <br>
    </header>

    <nav>
        <ul class="nav-menu">
            <li><a href="lab3(1).html">Home</a></li>
            <li><a href="products.html">Products</a></li>
            <li><a href="about.html">About</a></li>
            <li><a href="contact.html">Contact</a></li>
        </ul>
    </nav>

    <section>
        <div class="task-container">
            <h2>Our Products</h2>
        </div>
    </section>
```

```

<p>Welcome to our products page. This page shares the same responsive design as the home page thanks to the external CSS file.</p>

<div class="row">
  <div class="col-md-4">
    <div class="concept-card">
      <h3>Product 1</h3>
      <p>Description of product 1</p>
    </div>
  </div>
  <div class="col-md-4">
    <div class="concept-card">
      <h3>Product 2</h3>
      <p>Description of product 2</p>
    </div>
  </div>
  <div class="col-md-4">
    <div class="concept-card">
      <h3>Product 3</h3>
      <p>Description of product 3</p>
    </div>
  </div>
</div>
<div style="margin-top: 50px; text-align: center; color: #666;">
  <p>Web Programming, Nishant Shankar © 2026</p>
</div>
</body>
</html>

```

## 4.3



The screenshot shows a Mac OS X window with a title bar containing three colored circular icons (red, yellow, green). The main content area displays the following HTML code:

```

| <nav>
  <ul class="nav-menu">
    <li><a href="lab3(1).html">Home</a></li>
    <li><a href="products.html">Products</a></li>
    <li><a href="about.html">About</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</nav>

```

# Lab 3: Responsive Web Design and Animations

Complete the tasks below. Edit the CSS in the `<head>` and HTML body where indicated.

## TASK 1: RESPONSIVE DESIGN

### 1. Media Queries & Flexbox

**Subtask 1.1:** Resize your browser window to be very small. The red alert box above should appear only when narrow.

**Subtask 1.2:** Fix the menu below. On mobile, these items should stack vertically instead of squishing horizontally.



### Bootstrap Grid System

For these tasks, modify the HTML classes below directly. No CSS writing needed.

**Subtask 1.3:** The three boxes below are currently stacked (block elements). Use Bootstrap classes `row` (on parent) and `col-4` (on children) to make them sit side-by-side in equal columns.



**Subtask 1.4:** Responsive Utilities. Hide the box below on mobile phones but show it on desktops using Bootstrap classes.



### Advanced Media Queries

**Subtask 1.5:** Breakpoints. Write CSS media queries so the box below changes color:

Default (Gray) → Tablet (Yellow) → Desktop (Green)



**Subtask 1.6:** Orientation. Write a CSS media query so the box below only appears when the screen is in Landscape mode.



## TASK 2: ACCESSIBILITY

### 2. Visual & Assistive A11y

**Subtask 2.1 & 2.2:** The button below has poor contrast and no focus state. Fix this in the CSS block.

**Submit Data**

**Subtask 2.3: ARIA Labels.** The button below only contains an icon (X). A screen reader doesn't know what it does. Add an `aria-label="Close"` attribute to the HTML tag below.



# Products

Our product catalog

Home

Products

About

Contact

## Our Products

Welcome to our products page. This page shares the same responsive design as the home page thanks to the external CSS file.

### Product 1

Description of product 1

### Product 2

Description of product 2

### Product 3

Description of product 3

Web Programming, Nishant Shankar © 2026

# About Us

Learn more about our company

Home

Products

About

Contact

## About Our Company

Welcome to the about page. This page shares the same responsive design as the home page thanks to the external CSS file.

### Our Mission

To provide excellent web development education and resources.

### Our Team

We are a dedicated team of web developers passionate about teaching.

### Our Values

Quality, accessibility, and responsive design are at the core of everything we do.

Web Programming, Nishant Shankar © 2026

# Contact Us

Get in touch with us

Home

Products

About

Contact

## Contact Information

Welcome to the contact page. This page shares the same responsive design as the home page thanks to the external CSS file.

### Email

contact@example.com

### Phone

+1 (555) 123-4567

### Address

123 Web Street  
Internet City, IC 12345

### Contact Form

Name

Email

Message

[Send Message](#)