A close up of a logo

Description automatically generated

**CSS Styling**

**Disclaimer: The content is curated from online/offline resources and used for educational purpose only**

**LAB MANUAL**

**Responsive Two-Column Layout Design using Flexbox and Media Queries**

**Objective:**

The objective of this activity is to teach learners how to build a responsive two-column layout using CSS Flexbox combined with media queries. This exercise will help participants understand how flex containers and flex items behave, how to distribute space between columns, and how to ensure the layout adapts gracefully to different screen sizes, stacking columns vertically on smaller devices. By the end, learners should be able to create flexible, clean, and responsive two-column web designs suitable for web pages and applications.

**Equipment Required:**

* A computer with a text editor (e.g., VS Code, Sublime Text, Notepad++)
* A modern web browser such as Chrome, Firefox, or Edge with developer tools

**Prerequisites:**

* Basic knowledge of HTML structure and tags
* Familiarity with CSS selectors, properties, and values
* Basic understanding of CSS Flexbox concepts and syntax
* Basic understanding of CSS media queries for responsive design
* Ability to save and open .html files in a web browser

**Problem Statement:**

Your task is to design a web page that features a two-column layout using CSS Flexbox. The layout should divide the page into two sections arranged horizontally side-by-side on larger screens, with equal or adjustable widths. On smaller screen widths (such as mobile devices), the two columns should stack vertically to enhance readability and usability. You will implement the flex container and flex items, apply necessary flex properties, and write media queries to adapt the layout based on the screen size.

**Procedure:**

1. Create a new file named ***responsive\_two\_column.html***.
2. Define the basic HTML structure including <head> and <body>.
3. Inside the <body>, create a container <div> representing the flex container.
4. Add two child <div> elements for the left and right columns within the container.
5. Add content placeholders such as headings, paragraphs, or images in each column.
6. In the <head>, include a <style> section or link an external CSS file for styling.
7. Use Flexbox properties on the container (display: flex; flex-direction: row;) and set widths or flex-grow for the columns.
8. Create a media query targeting screen widths less than 768px (or your chosen breakpoint) that changes the flex container direction to column and sets the columns to full width.
9. Save and open the file in a browser, then resize the browser window to test responsiveness.

**Code**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8" />**

**<meta name="viewport" content="width=device-width, initial-scale=1.0" />**

**<title>Responsive Two-Column Layout</title>**

**<!-- Linking the external CSS file -->**

**<link rel="stylesheet" href="styles.css" />**

**</head>**

**<body>**

**<div class="container">**

**<!-- Left Column -->**

**<div class="column left-column">**

**<h2>Left Column</h2>**

**<p>**

**This is the left column content area. You can add text, images, or other HTML elements here.**

**On wider screens, it appears side-by-side with the right column.**

**</p>**

**<p>**

**The left column background is a light blue to visually distinguish it.**

**</p>**

**</div>**

**<!-- Right Column -->**

**<div class="column right-column">**

**<h2>Right Column</h2>**

**<p>**

**This is the right column content area. It also supports any HTML content.**

**When viewed on smaller screens, this column will stack below the left column.**

**</p>**

**<p>**

**The right column background is a soft orange color for differentiation.**

**</p>**

**</div>**

**</div>**

**</body>**

**</html>**

***styles.css***

***/\* Basic reset and font \*/***

***body {***

***font-family: Arial, sans-serif;***

***margin: 0;***

***padding: 20px;***

***background-color: #fafafa;***

***color: #333;***

***}***

***/\* Flex container \*/***

***.container {***

***display: flex; /\* Use flexbox layout \*/***

***flex-direction: row; /\* Arrange children in a row (horizontal) \*/***

***gap: 20px; /\* Space between columns \*/***

***max-width: 900px;***

***margin: 0 auto; /\* Center horizontally \*/***

***background: white;***

***padding: 20px;***

***box-shadow: 0 4px 12px rgba(0,0,0,0.1);***

***border-radius: 8px;***

***}***

***/\* Left and right columns \*/***

***.column {***

***flex: 1; /\* Equal width \*/***

***padding: 20px;***

***box-sizing: border-box;***

***background-color: #f4f4f4;***

***border-radius: 6px;***

***}***

***/\* Optional: different background for left and right \*/***

***.left-column {***

***background-color: #e3f2fd;***

***}***

***.right-column {***

***background-color: #fff3e0;***

***}***

***/\* Heading styles \*/***

***h2 {***

***margin-top: 0;***

***color: #0d47a1;***

***}***

***/\* Responsive design: stack columns on small screens \*/***

***@media (max-width: 768px) {***

***.container {***

***flex-direction: column; /\* Stack columns vertically \*/***

***}***

***.column {***

***width: 100%;***

***padding: 15px;***

***margin-bottom: 15px;***

***}***

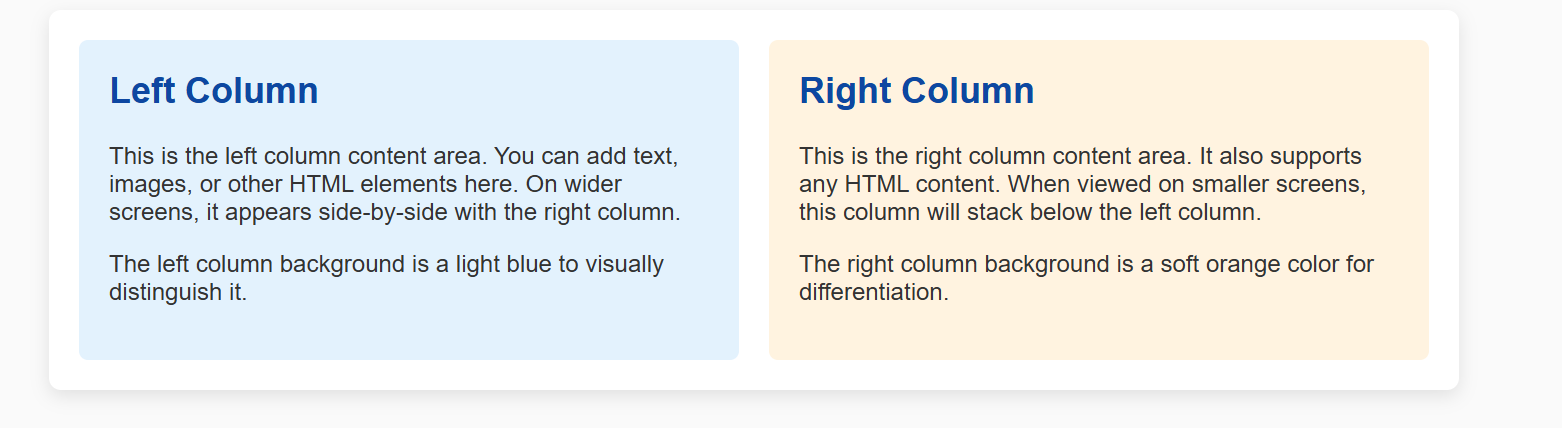
***.column:last-child {***

***margin-bottom: 0;***

***}***

***}***

**Output**

****