**Lab 5: Layout website**

**1. Requirements**

Students will design and develop a modern, single-page website that includes a navigation bar, hero banner, about section, services, and a contact form.

**CSS Topics Applied**

* box-sizing: border-box for consistent layout control
* CSS position (relative, absolute, fixed) to create dropdown menus and overlays
* Flexbox(flex-direction, justify-content, align-items, …) to arrange layout

**Output**

By the end of the lab, each student will have a functional, company website using only HTML and CSS (no frameworks).

**Files Required**

* index.html
* style.css
* Optional: images/ folder

**2. Theory**

**2.1 Box-Sizing**

CSS box-sizing defines how the total width and height of an element is calculated.

**Default (content-box):**

* The width/height includes only the content.
* Padding and borders are added *outside* of the width, which can cause layout issues.

**border-box:**

* Padding and border are included inside the width/height.
* This makes layout more predictable, especially for columns or grid-based layouts.

**2.2 Position**

CSS position allows elements to be placed differently from their default flow. This is useful **f**or dropdowns, sticky headers, or floating buttons.

|  |  |
| --- | --- |
| **Value** | **Description** |
| static | Default value. Elements follow normal document flow. |
| relative | Positions the element relative to its normal position. |
| absolute | Positions relative to the nearest ancestor that is not static. |
| fixed | Positions relative to the **viewport**; stays in the same place when scrolling. |
| sticky | Scrolls normally, then “sticks” when reaching a threshold (browser support varies). |

**Common Use Cases:**

* Fixed navigation bars: position: fixed
* Dropdown menus: position: absolute
* Overlay modals or tooltips

**2.3 Flexbox**

Flexbox (Flexible Box Layout) is a one-dimensional layout model that helps distribute space and align content.

**On the Flex Container (Parent Element)**

| **Property** | **Description** |
| --- | --- |
| display: flex | Enables the flex context |
| flex-direction | Defines the main axis: row, row-reverse, column, column-reverse |
| justify-content | Aligns items along the main axis: flex-start, center, space-between, space-around, space-evenly |
| align-items | Aligns items on the cross axis: stretch, center, flex-start, flex-end, baseline |
| flex-wrap | Allows items to wrap onto multiple lines: nowrap, wrap, wrap-reverse |
| gap | Controls spacing between items (modern browsers only) |

**On Flex Items (Children)**

| **Property** | **Description** |
| --- | --- |
| flex-grow | Defines how much an item will grow relative to others |
| flex-shrink | Defines how an item will shrink if space is tight |
| flex-basis | Sets the initial size of the item before remaining space is distributed |
| flex | Shorthand for flex-grow flex-shrink flex-basis |
| align-self | Overrides align-items for a specific item |
| order | Controls the display order of flex items (lower value = shown earlier) |

**3. Step-by-Step Guide**

**Step 1: Drop Down Menu**

A screenshot of a computer

AI-generated content may be incorrect.

**1 Set Up the HTML Structure**

* Use a <nav> element to contain the navigation.
* Divide it into two main parts:
  + A **logo section** using a <div class="logo">, which includes an image and a brand name.
  + A **menu list** using <ul> with <li> items for each page link.
* To create dropdowns, nest an additional <ul> inside a <li>.

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

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

  <title>Agriculture Website</title>

  <link href="style.css" rel="stylesheet"/>

</head>

<body>

  <header>

    <nav id="navmenu" class="navmenu">

      <div class="logo">

        <img src="https://cdn-icons-png.flaticon.com/512/5900/5900287.png" alt="Logo" />

        <span>AGRI CULTURE</span>

      </div>

      <ul>

        <li><a href="index.html">Home</a></li>

        <li><a href="about.html">About Us</a></li>

        <li><a href="services.html">Our Services</a></li>

        <li><a href="testimonials.html">Testimonials</a></li>

        <li><a href="blog.html">Blog</a></li>

        <li class="dropdown">

          <a href="#"><span>Dropdown</span> <i class="bi bi-chevron-down toggle-dropdown"></i></a>

          <ul>

            <li><a href="#">Dropdown 1</a></li>

            <li class="dropdown">

              <a href="#"><span>Deep Dropdown</span> <i class="bi bi-chevron-right toggle-dropdown"></i></a>

              <ul>

                <li><a href="#">Deep Dropdown 1</a></li>

                <li><a href="#">Deep Dropdown 2</a></li>

                <li><a href="#">Deep Dropdown 3</a></li>

                <li><a href="#">Deep Dropdown 4</a></li>

                <li><a href="#">Deep Dropdown 5</a></li>

              </ul>

            </li>

            <li><a href="#">Dropdown 2</a></li>

            <li><a href="#">Dropdown 3</a></li>

            <li><a href="#">Dropdown 4</a></li>

          </ul>

        </li>

        <li><a href="contact.html">Contact</a></li>

      </ul>

    </nav>

  </header>

</body>

</html>

**2: Style the Navigation with CSS**

* Use display: flex to arrange the logo and menu items.
* Apply justify-content: center and align-items: center to center content horizontally and vertically.
* Hide dropdown submenus with display: none and show them on hover with li:hover > ul.

/\* General reset \*/

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

  }

  body {

    font-family: Arial, sans-serif;

  }

  /\* Centered Navigation \*/

  header {

    background: #fff;

    padding: 20px 0;

  }

  .navmenu {

    display: flex;

    flex-direction: column;

    align-items: center;

  }

  .logo {

    display: flex;

    align-items: center;

    gap: 10px;

    margin-bottom: 15px;

  }

  .logo img {

    width: 40px;

    height: 40px;

  }

  .logo span {

    font-size: 20px;

    font-weight: bold;

    color: #3a7f2e;

  }

  .navmenu ul {

    list-style: none;

    display: flex;

    justify-content: center;

    gap: 30px;

    position: relative;

  }

  .navmenu ul li {

    position: relative;

  }

  .navmenu ul li a {

    text-decoration: none;

    color: #333;

    padding: 10px 15px;

    display: inline-block;

    transition: color 0.3s;

  }

  .navmenu ul li a:hover {

    color: #3a7f2e;

  }

  /\* Dropdown styles \*/

  .navmenu ul li ul {

    display: none;

    position: absolute;

    top: 100%;

    left: 0;

    flex-direction: column;

    background: #fff;

    padding: 10px 0;

    box-shadow: 0 2px 6px rgba(0, 0, 0, 0.1);

    z-index: 1000;

    min-width: 180px;

  }

  .navmenu ul li:hover > ul {

    display: flex;

  }

  .navmenu ul li ul li {

    width: 100%;

  }

  .navmenu ul li ul li a {

    padding: 8px 20px;

    white-space: nowrap;

  }

  .navmenu ul li ul li:hover > a {

    background: #f0f0f0;

  }

  /\* Deep dropdown styles \*/

  .navmenu ul li ul li.dropdown:hover > ul {

    display: flex;

    top: 0;

    left: 100%;

  }

**Step 2: Creating a Flexbox Layout**

A green field with a hill in the background

AI-generated content may be incorrect.

**1 Set Up the HTML Structure**

* + Wrap all your content inside a main container (e.g., .container) for easier styling.
  + Use semantic elements: <h1>, <p>, <ul> with <li> for better structure and accessibility.
  + Add Font Awesome icons using <i> tags for a more modern look.

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

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

    <title>Plants Make Life Better</title>

    <!-- Link to external CSS -->

    <link rel="stylesheet" href="style.css" />

    <!-- Font Awesome CDN for icons -->

    <link

      rel="stylesheet"

      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.0/css/all.min.css"

    />

  </head>

  <body>

    <div class="container">

      <!-- Left Section: Text content -->

      <div class="left-content">

        <h1>Plants Make Life Better</h1>

        <p>Lorem ipsum dolor sit amet consectetur...</p>

        <ul>

          <li><i class="fa-solid fa-check"></i> Lorem ipsum dolor sit amet</li>

          <li>

            <i class="fa-solid fa-check"></i> Velit explicabo vitae repellendu

          </li>

          <li><i class="fa-solid fa-check"></i> Repellat aliquam nihil illo</li>

        </ul>

        <button class="contact-button">GET IN TOUCH</button>

      </div>

      <!-- Right Section: Image with Play Icon -->

      <div class="right-content">

        <img

          src="figure.jpg"

          alt="Farmer"

        />

      </div>

    </div>

  </body>

</html>

**2: Style the Navigation with CSS**

* + Use display: flex on the .container to align .left-content and .right-content horizontally.
  + Use flex: 1 to ensure both sections take up equal space.
  + Use gap for spacing between the two sections.
  + Use flex-wrap: wrap to make the layout responsive (stack vertically on small screens).

/\* GENERAL LAYOUT \*/

body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

    background: #f0f8f5;

}

.container {

    display: flex;

    align-items: center;

    justify-content: space-between;

    padding: 40px;

    gap: 40px;

    flex-wrap: wrap;

}

/\* LEFT CONTENT \*/

.left-content {

    flex: 1;

}

.left-content h1 {

    font-size: 2.5em;

    color: #2f4f4f;

    margin-bottom: 10px;

}

.left-content p {

    font-size: 1.1em;

    color: #444;

    margin-bottom: 20px;

}

.left-content ul {

    list-style: none;

    padding: 0;

    margin-bottom: 20px;

}

.left-content li {

    margin-bottom: 10px;

    font-size: 1em;

    color: #333;

}

.left-content li i {

    color: green;

    margin-right: 8px;

}

/\* BUTTON \*/

.contact-button {

    padding: 12px 24px;

    background-color: green;

    color: white;

    border: none;

    border-radius: 6px;

    cursor: pointer;

    font-weight: bold;

    transition: background-color 0.3s ease;

}

.contact-button:hover {

    background-color: darkgreen;

}

/\* RIGHT CONTENT \*/

.right-content {

    flex: 1;

    position: relative;

}

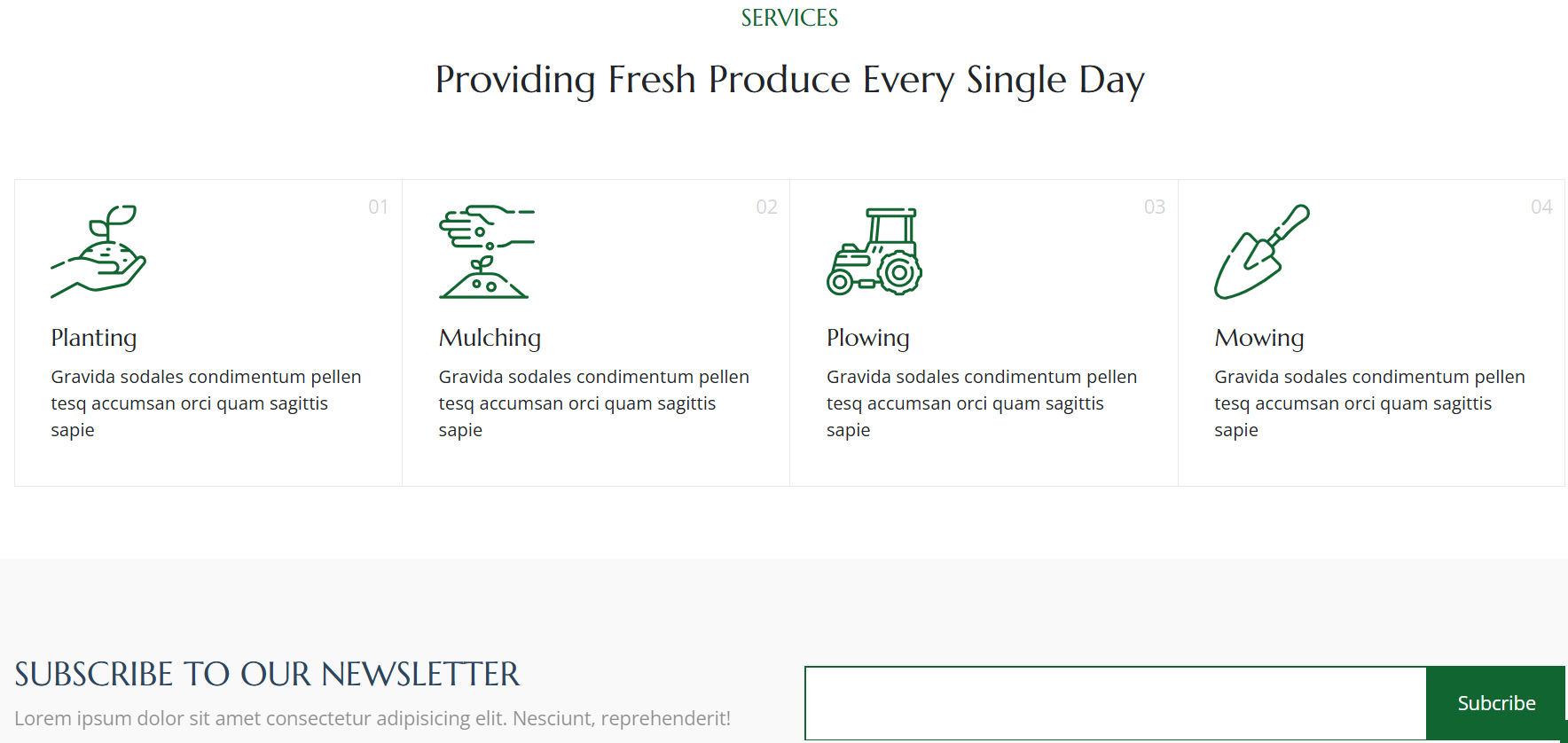
.right-content img {

    width: 100%;

    border-radius: 10px;

}

**Step 3: Creating a Grid Layout**



**1 Set Up the HTML Structure**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Farm Services</title>

  <link rel="stylesheet" href="style.css">

</head>

<body>

  <h2>SERVICES</h2>

  <h3>Providing Fresh Produce Every Single Day</h3>

  <div class="services">

    <div class="service-box">

      <img src="https://img.icons8.com/ios/100/plant-under-sun--v1.png" alt="Planting">

      <h4>Planting</h4>

      <p>Gravida sodales condimentum pellen tesq accumsan orci quam sagittis sapie</p>

    </div>

    <div class="service-box">

      <img src="https://img.icons8.com/ios/100/hand-planting.png" alt="Mulching">

      <h4>Mulching</h4>

      <p>Gravida sodales condimentum pellen tesq accumsan orci quam sagittis sapie</p>

    </div>

    <div class="service-box">

      <img src="https://img.icons8.com/ios/100/tractor.png" alt="Plowing">

      <h4>Plowing</h4>

      <p>Gravida sodales condimentum pellen tesq accumsan orci quam sagittis sapie</p>

    </div>

    <div class="service-box">

      <img src="https://img.icons8.com/ios/100/gardening.png" alt="Mowing">

      <h4>Mowing</h4>

      <p>Gravida sodales condimentum pellen tesq accumsan orci quam sagittis sapie</p>

    </div>

  </div>

  <div class="newsletter">

    <h4>SUBSCRIBE TO OUR NEWSLETTER</h4>

    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Nesciunt, reprehenderit!</p>

    <form>

      <input type="email" placeholder="Enter your email">

      <button type="submit">Subscribe</button>

    </form>

  </div>

</body>

</html>

**2: Style the Navigation with CSS**

| **Concept** | **Explanation** |
| --- | --- |
| display: grid; | Turns the container into a grid layout. |
| grid-template-columns: repeat(4, 1fr); | Creates 4 equal-width columns. Each 1fr means one portion of available space. |
| gap: 1px; | Adds space between grid items (can be gap: 20px; for larger gaps). |
| grid-template-columns | Can be customized to create responsive layouts like 2 columns on tablets, 1 on mobile. |

body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

  }

  h2 {

    text-align: center;

    color: green;

    margin-top: 30px;

  }

  h3 {

    text-align: center;

    font-size: 28px;

    margin-bottom: 40px;

  }

  .services {

    display: grid;

    grid-template-columns: repeat(4, 1fr);

    gap: 1px;

    padding: 0 50px;

    margin-bottom: 60px;

  }

  .service-box {

    border: 1px solid #eee;

    padding: 30px 20px;

    text-align: center;

  }

  .service-box img {

    width: 60px;

    height: 60px;

    margin-bottom: 15px;

  }

  .service-box h4 {

    margin-bottom: 10px;

    font-size: 18px;

  }

  .service-box p {

    color: #555;

    font-size: 14px;

  }

  .newsletter {

    background-color: #f6f6f6;

    padding: 40px 50px;

    text-align: center;

  }

  .newsletter h4 {

    font-size: 24px;

    margin-bottom: 10px;

  }

  .newsletter p {

    margin-bottom: 20px;

    color: #666;

  }

  .newsletter input[type="email"] {

    padding: 10px;

    width: 300px;

    border: 1px solid #ccc;

    border-right: none;

    border-radius: 5px 0 0 5px;

  }

  .newsletter button {

    padding: 11px 20px;

    background-color: #126d35;

    color: white;

    border: none;

    border-radius: 0 5px 5px 0;

    cursor: pointer;

  }

  .newsletter button:hover {

    background-color: #0f5a2c;

  }

**Step 4: Footer**

A black screen with white text

AI-generated content may be incorrect.

**1 HTML Structure Suggestion**

* Use a <footer> tag for the entire section.
* Divide the content into columns using <div> elements with a common class (e.g., footer-col).
* Use headings (<h3> or <h4>) for section titles.
* Use <ul> and <li> for lists of links.
* Add contact information using <p> tags.
* Place social media icons in a separate <div> at the bottom right.
* Add a copyright notice at the bottom.

**2 CSS Styling Suggestions**

* Set the background color to dark (e.g., #181818).
* Use white or light gray text for contrast.
* Style the columns with display: flex for horizontal layout.
* Add padding and margin for spacing.
* Style headings with a different color or underline for emphasis.
* Style links to remove underlines and change color on hover.
* Style the social media icons with a background color and rounded corners.