Hamburger

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

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

  <title>Pure CSS Hamburger</title>

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

  <style>

    /\* Reset + base \*/

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

  }

  body, html {

    height: 100%;

    font-family: 'Segoe UI', sans-serif;

    background: #111;

    color: #fff;

  }

  /\* Navbar setup \*/

  .navbar {

    position: relative;

    z-index: 10;

  }

  /\* Hide checkbox \*/

  #menu-toggle {

    display: none;

  }

  /\* Hamburger icon \*/

  .menu-btn {

    position: fixed;

    top: 1.5rem;

    right: 1.5rem;

    width: 35px;

    height: 25px;

    cursor: pointer;

    display: flex;

    flex-direction: column;

    justify-content: space-between;

    z-index: 20;

  }

  .menu-btn span {

    height: 4px;

    width: 100%;

    background: white;

    border-radius: 5px;

    transition: 0.4s ease;

  }

  /\* Fullscreen nav menu \*/

  .menu {

    position: fixed;

    top: 0;

    left: 0;

    height: 100vh;

    width: 100%;

    background: #222;

    transform: translateX(100%);

    transition: transform 0.5s ease;

    display: flex;

    justify-content: center;

    align-items: center;

  }

  .menu ul {

    list-style: none;

    text-align: center;

  }

  .menu ul li {

    margin: 20px 0;

  }

  .menu ul li a {

    color: #fff;

    text-decoration: none;

    font-size: 2rem;

    transition: color 0.3s;

  }

  .menu ul li a:hover {

    color: #f39c12;

  }

  /\* Toggle menu open \*/

  #menu-toggle:checked ~ .menu {

    transform: translateX(0);

  }

  /\* Animate icon to X \*/

  #menu-toggle:checked + .menu-btn span:nth-child(1) {

    transform: rotate(45deg) translate(5px, 5px);

  }

  #menu-toggle:checked + .menu-btn span:nth-child(2) {

    opacity: 0;

  }

  #menu-toggle:checked + .menu-btn span:nth-child(3) {

    transform: rotate(-45deg) translate(5px, -5px);

  }

  </style>

</head>

<body>

  <nav class="navbar">

    <input type="checkbox" id="menu-toggle" />

    <label for="menu-toggle" class="menu-btn">

      <span></span>

      <span></span>

      <span></span>

    </label>

    <div class="menu">

      <ul>

        <li><a href="#">MENU</a></li>

        <li><a href="#">TYPE of BURGERS </a></li>

        <li><a href="#">TOPPING</a></li>

        <li><a href="#">SPECIAL HAMBURGER </a></li>

      </ul>

    </div>

  </nav>

</body>

</html>

1 Pure CSS responsive Masonry Grid

Problem :

Create a masonry-style grid layout(like pintrest) without javascript,using only css . the gird

Should:

Have variable-height items

Be responsive(columns adjust based on viewport width).

Maintain consistent gaps between items.

Constraints :

Use css Grid or Flexbox (no column-count)

Must work in modern browsers (Chrome , firebox, edge.)

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>CSS Masonry & Modal (No JS)</title>

  <style>

    /\* --- Masonry Grid Styling --- \*/

    .masonry-container {

      display: grid;

      grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));

      gap: 16px;

      padding: 20px;

    }

    .masonry-item {

      background: #f7f7f7;

      padding: 16px;

      border-radius: 10px;

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

    }

    /\* Simulated different heights \*/

    .masonry-item:nth-child(even) {

      height: 150px;

    }

    .masonry-item:nth-child(odd) {

      height: 250px;

    }

    /\* --- Modal Styling (CSS only using :target) --- \*/

    .modal {

      position: fixed;

      top: 0; left: 0;

      width: 100vw; height: 100vh;

      background: rgba(0, 0, 0, 0.6);

      display: none;

      justify-content: center;

      align-items: center;

      z-index: 999;

    }

    .modal:target {

      display: flex;

    }

    .modal-content {

      background: white;

      padding: 24px;

      border-radius: 12px;

      width: 90%;

      max-width: 400px;

      box-shadow: 0 10px 25px rgba(0,0,0,0.2);

    }

    .button {

      display: inline-block;

      padding: 10px 16px;

      background-color: royalblue;

      color: white;

      text-decoration: none;

      border-radius: 6px;

      margin: 20px;

    }

    .close {

      display: inline-block;

      margin-top: 20px;

      padding: 8px 14px;

      background-color: crimson;

      color: white;

      border-radius: 6px;

      text-decoration: none;

    }

  </style>

</head>

<body>

  <!-- Open Modal Button -->

  <a href="#openModal" class="button">Open Modal</a>

  <!-- CSS-only Modal -->

  <div id="openModal" class="modal" role="dialog" aria-modal="true">

    <div class="modal-content">

      <p>This is a pure CSS modal dialog! 🧡</p>

      <a href="#" class="close">Close</a>

    </div>

  </div>

  <!-- Masonry Grid -->

  <div class="masonry-container">

    <div class="masonry-item">Item 1</div>

    <div class="masonry-item">Item 2</div>

    <div class="masonry-item">Item 3</div>

    <div class="masonry-item">Item 4</div>

    <div class="masonry-item">Item 5</div>

    <div class="masonry-item">Item 6</div>

    <div class="masonry-item">Item 7</div>

    <div class="masonry-item">Item 8</div>

  </div>

</body>

</html>

5 Responsive Multi-Column form with Flexfox

Problem :

Design a multi-column form that:

Stacks into a single column on mobile.

Align labels and inputs neatly.

Handles validation errors without breaking Layout.

Constraints:

Use flexbox(no floats or tables).

Must be accessible (<fieldset>,<legend>).

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Responsive Multi-Column Form</title>

  <style>

    body {

      font-family: Arial, sans-serif;

      padding: 20px;

    }

    fieldset {

      border: 1px solid #ccc;

      padding: 20px;

      border-radius: 10px;

      max-width: 700px;

      margin: auto;

    }

    legend {

      font-size: 1.2em;

      padding: 0 10px;

      font-weight: bold;

    }

    .form-row {

      display: flex;

      gap: 20px;

      flex-wrap: wrap;

    }

    .form-group {

      flex: 1;

      min-width: 250px;

      display: flex;

      flex-direction: column;

      margin-bottom: 16px;

    }

    label {

      margin-bottom: 6px;

      font-weight: bold;

    }

    input, select, textarea {

      padding: 10px;

      border: 1px solid #ccc;

      border-radius: 6px;

    }

    @media (max-width: 600px) {

      .form-row {

        flex-direction: column;

      }

    }

  </style>

</head>

<body>

  <form>

    <fieldset>

      <legend>Student Registration</legend>

      <div class="form-row">

        <div class="form-group">

          <label for="fname">First Name</label>

          <input type="text" id="fname" name="fname" required>

        </div>

        <div class="form-group">

          <label for="lname">Last Name</label>

          <input type="text" id="lname" name="lname" required>

        </div>

      </div>

      <div class="form-row">

        <div class="form-group">

          <label for="email">Email</label>

          <input type="email" id="email" name="email" required>

        </div>

        <div class="form-group">

          <label for="course">Course</label>

          <select id="course" name="course">

            <option>B.Tech</option>

            <option>MCA</option>

            <option>Diploma</option>

          </select>

        </div>

      </div>

    </fieldset>

  </form>

</body>

</html>

6 CSS-only Parallax Scrolling Effect

Problem:

Create a Parallax scrolling Effect where background images move slower than foreground content.

Constraints:

No JavaScript.

Must work on mobile( no background-attachment fixed).

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>CSS Parallax</title>

  <style>

    \* {

      margin: 0;

      padding: 0;

    }

    .parallax {

      background-image: url('https://images.unsplash.com/photo-1506744038136-46273834b3fb');

      height: 100vh;

      background-attachment: fixed;

      background-position: center;

      background-repeat: no-repeat;

      background-size: cover;

    }

    .content {

      height: 100vh;

      background-color: white;

      display: flex;

      justify-content: center;

      align-items: center;

      font-size: 2em;

      color: #333;

    }

  </style>

</head>

<body>

  <div class="parallax"></div>

  <div class="content">Welcome to Sanjivani 💙</div>

  <div class="parallax"></div>

  <div class="content">Scroll more! 🚀</div>

</body>

</html>

7 Custom Styled Slider (Cross-Browser)

Problem:

Style and <input type=”range”>to:

Have a custom thumb and track design.

Show a dynamic tooltip displaying the current value.

Work consistently across Chrome,firbox, and Safari.

Constraints:

NO External libraries.

Must be accessible(aria\*attributes).

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Custom CSS Range Slider</title>

  <style>

    body {

      font-family: 'Segoe UI', sans-serif;

      display: flex;

      justify-content: center;

      align-items: center;

      height: 100vh;

      background: #f4f4f4;

    }

    .slider-container {

      position: relative;

      width: 300px;

    }

    input[type="range"] {

      -webkit-appearance: none;

      width: 100%;

      height: 10px;

      background: linear-gradient(90deg, #00c6ff, #0072ff);

      border-radius: 5px;

      outline: none;

      margin: 0;

    }

    input[type="range"]::-webkit-slider-thumb {

      -webkit-appearance: none;

      appearance: none;

      width: 24px;

      height: 24px;

      border-radius: 50%;

      background: #fff;

      border: 2px solid #0072ff;

      cursor: pointer;

      box-shadow: 0 0 4px rgba(0,0,0,0.2);

      transition: transform 0.2s ease;

    }

    input[type="range"]::-moz-range-thumb {

      width: 24px;

      height: 24px;

      border-radius: 50%;

      background: #fff;

      border: 2px solid #0072ff;

      cursor: pointer;

      box-shadow: 0 0 4px rgba(0,0,0,0.2);

    }

    .tooltip {

      position: absolute;

      top: -35px;

      left: 50%;

      transform: translateX(-50%);

      background: #0072ff;

      color: #fff;

      padding: 6px 10px;

      border-radius: 6px;

      font-size: 14px;

      font-weight: bold;

      white-space: nowrap;

      pointer-events: none;

    }

    .tooltip::after {

      content: '';

      position: absolute;

      bottom: -6px;

      left: 50%;

      transform: translateX(-50%);

      border: 6px solid transparent;

      border-top-color: #0072ff;

    }

    /\* Responsive tooltip movement using CSS trick \*/

    input[type="range"] {

      --val: 50;

    }

    .slider-container::before {

      content: attr(data-value);

      position: absolute;

      top: -40px;

      left: calc(var(--val) \* 1% - 16px);

      background: #0072ff;

      color: white;

      padding: 5px 10px;

      font-size: 13px;

      border-radius: 4px;

    }

    /\* Responsive only \*/

    @media (max-width: 400px) {

      .slider-container {

        width: 90%;

      }

    }

  </style>

</head>

<body>

  <div class="slider-container" data-value="50">

    <input

      type="range"

      min="0"

      max="100"

      value="50"

      aria-valuemin="0"

      aria-valuemax="100"

      aria-valuenow="50"

      aria-label="Custom styled slider"

      disabled

    >

    <div class="tooltip">50</div>

  </div>

</body>

</html>

8 CSS-only card flip Animation(3D Transform)

Problem:

Implement a 3D card flip animation where hovering over a card flips it to reveal content on the back.

Constraints:

No javascript.

Must preserve 3D perspective.

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>3D Card Flip (CSS Only)</title>

  <style>

    body {

      margin: 0;

      height: 100vh;

      display: flex;

      justify-content: center;

      align-items: center;

      background: #f3f4f6;

      font-family: 'Segoe UI', sans-serif;

    }

    .card-container {

      width: 300px;

      height: 200px;

      perspective: 1000px; /\* 👈 enables 3D \*/

    }

    .card {

      width: 100%;

      height: 100%;

      position: relative;

      transform-style: preserve-3d;

      transition: transform 0.6s ease-in-out;

    }

    .card-container:hover .card {

      transform: rotateY(180deg);

    }

    .card-front,

    .card-back {

      position: absolute;

      width: 100%;

      height: 100%;

      backface-visibility: hidden;

      border-radius: 10px;

      box-shadow: 0 4px 10px rgba(0, 0, 0, 0.15);

      display: flex;

      justify-content: center;

      align-items: center;

      font-size: 20px;

      color: white;

    }

    .card-front {

      background: #0072ff;

    }

    .card-back {

      background: #00c6ff;

      transform: rotateY(180deg);

    }

  </style>

</head>

<body>

  <div class="card-container" aria-label="Flipping card for details">

    <div class="card">

      <div class="card-front" aria-hidden="false">

        Front Side 😎

      </div>

      <div class="card-back" aria-hidden="true">

        Back Side ✨

      </div>

    </div>

  </div>

</body>

</html>

9 Responsive CSS Grid with Aspect Ratio Tiles

Problem:

Build a grid layout where each tile:

Maintain a 1:1 aspect ratio.

Expands on hover without disrupting the grid.

Works in a responsive multi-column layout.

Constraints:

No Hardcoded heights.

Must use CSS gird.

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Aspect Ratio Grid</title>

  <style>

    \* {

      box-sizing: border-box;

      margin: 0;

      padding: 0;

    }

    body {

      background: #f0f0f0;

      padding: 2rem;

      font-family: Arial, sans-serif;

    }

    .grid {

      display: grid;

      grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));

      gap: 1rem;

    }

    .tile {

      aspect-ratio: 1 / 1; /\* Maintains 1:1 ratio \*/

      background-color: #0077ff;

      border-radius: 12px;

      transition: transform 0.3s ease;

      position: relative;

    }

    .tile:hover {

      transform: scale(1.05); /\* Expands on hover \*/

      z-index: 2;

    }

  </style>

</head>

<body>

  <div class="grid">

    <div class="tile"></div>

    <div class="tile"></div>

    <div class="tile"></div>

    <div class="tile"></div>

    <div class="tile"></div>

    <div class="tile"></div>

  </div>

</body>

</html>

10 Pure CSS Tooltip with Arrow

Problem:

Create a tooltip that:

Appears on hover.

Has a dynamic arrow pointing to the element.

Adjust position based on available space.

Constraints:

No javaScript.

Must work with any text length.

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Pure CSS Tooltip with Arrow</title>

  <style>

    body {

      font-family: Arial, sans-serif;

      padding: 100px;

      display: flex;

      justify-content: center;

    }

    .tooltip-wrapper {

      position: relative;

      display: inline-block;

      cursor: pointer;

    }

    .tooltip-text {

      position: absolute;

      bottom: 125%; /\* Position above \*/

      left: 50%;

      transform: translateX(-50%);

      background-color: #333;

      color: #fff;

      padding: 0.5rem 1rem;

      border-radius: 5px;

      white-space: nowrap;

      opacity: 0;

      pointer-events: none;

      transition: opacity 0.3s ease;

      z-index: 10;

    }

    .tooltip-text::after {

      content: "";

      position: absolute;

      top: 100%;

      left: 50%;

      transform: translateX(-50%);

      border-width: 6px;

      border-style: solid;

      border-color: #333 transparent transparent transparent;

    }

    .tooltip-wrapper:hover .tooltip-text {

      opacity: 1;

    }

    /\* Optional Responsive Adjustments (Bottom & Top Tooltip Flip) \*/

    @media (max-width: 600px) {

      .tooltip-text {

        bottom: auto;

        top: 125%;

      }

      .tooltip-text::after {

        top: auto;

        bottom: 100%;

        border-color: transparent transparent #333 transparent;

      }

    }

  </style>

</head>

<body>

  <div class="tooltip-wrapper">

    Hover over me

    <div class="tooltip-text">This is a dynamic pure CSS tooltip ✨</div>

  </div>

</body>

</html>