

Here are simple HTML and CSS projects to help you practice and enhance your skills:

1. Personal Portfolio Page

Description: Create a basic portfolio page to showcase your work.

Features:

- Header with navigation links (Home, About, Projects, Contact)
- About section with a brief introduction
- Projects section displaying thumbnails and descriptions
- Contact form for inquiries

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Portfolio</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
    header { background: #333; color: #fff; padding: 1rem; text-align:
center; }
    nav a { color: #fff; margin: 0 15px; text-decoration: none; }
    .section { padding: 2rem; }
    .projects img { width: 100px; height: 100px; object-fit: cover; }
    form { max-width: 400px; margin: 0 auto; }
    input, textarea { width: 100%; padding: 10px; margin: 5px 0; }
  </style>
</head>
<body>
  <header>
    <nav>
      <a href="#about">About</a>
      <a href="#projects">Projects</a>
      <a href="#contact">Contact</a>
    </nav>
  </header>
  <div class="section" id="about">
```

```

    <h1>About Me</h1>
    <p>Introduction text here.</p>
</div>
<div class="section" id="projects">
    <h1>Projects</h1>
    <div class="projects">
        
        <p>Project 1 description.</p>
    </div>
    <div class="projects">
        
        <p>Project 2 description.</p>
    </div>
</div>
<div class="section" id="contact">
    <h1>Contact</h1>
    <form>
        <input type="text" placeholder="Your Name" required>
        <input type="email" placeholder="Your Email" required>
        <textarea rows="4" placeholder="Your Message"></textarea>
        <button type="submit">Send</button>
    </form>
</div>
</body>
</html>

```

2. Landing Page

Description: Design a simple landing page for a fictional product or service.

Features:

- Hero section with a call-to-action button
- Features section
- Testimonials section
- Footer with contact information

Code Snippet:

html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">

```

```

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Landing Page</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
        .hero { background: #4CAF50; color: #fff; text-align: center;
padding: 5rem 1rem; }
        .hero button { background: #fff; color: #4CAF50; border: none;
padding: 10px 20px; cursor: pointer; }
        .section { padding: 2rem; text-align: center; }
        footer { background: #333; color: #fff; text-align: center; padding:
1rem; }
    </style>
</head>
<body>
    <div class="hero">
        <h1>Welcome to Our Service</h1>
        <p>Discover the best solution for your needs.</p>
        <button>Get Started</button>
    </div>
    <div class="section">
        <h2>Features</h2>
        <p>Feature 1</p>
        <p>Feature 2</p>
        <p>Feature 3</p>
    </div>
    <div class="section">
        <h2>Testimonials</h2>
        <p>"This service is amazing!" - User A</p>
        <p>"I love it!" - User B</p>
    </div>
    <footer>
        <p>Contact us at info@example.com</p>
    </footer>
</body>
</html>

```

3. Simple Blog Layout

Description: Create a blog layout with a list of posts.

Features:

- Header with navigation
- Blog posts with titles and excerpts

- Sidebar with recent posts

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Blog</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
    header { background: #333; color: #fff; padding: 1rem; text-align:
center; }
    .container { display: flex; }
    .main { flex: 2; padding: 2rem; }
    .sidebar { flex: 1; padding: 2rem; background: #f4f4f4; }
    .post { margin-bottom: 1rem; }
    .post h2 { margin: 0; }
    .post p { margin: 0.5rem 0; }
  </style>
</head>
<body>
  <header>
    <h1>My Blog</h1>
  </header>
  <div class="container">
    <div class="main">
      <div class="post">
        <h2>Blog Post 1</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Proin ac est id erat facilisis bibendum.</p>
      </div>
      <div class="post">
        <h2>Blog Post 2</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Proin ac est id erat facilisis bibendum.</p>
      </div>
    </div>
    <div class="sidebar">
      <h3>Recent Posts</h3>
```

```

        <ul>
            <li><a href="#">Blog Post 1</a></li>
            <li><a href="#">Blog Post 2</a></li>
        </ul>
    </div>
</div>
</body>
</html>

```

4. CSS Grid Layout

Description: Use CSS Grid to create a simple grid layout.

Features:

- Header
- Grid of items
- Footer

Code Snippet:

html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Grid Layout</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
        .container { display: grid; grid-template-columns: repeat(auto-fill,
minmax(150px, 1fr)); gap: 10px; padding: 1rem; }
        .item { background: #4CAF50; color: #fff; text-align: center;
padding: 20px; }
        footer { background: #333; color: #fff; text-align: center; padding:
1rem; }
    </style>
</head>
<body>
    <header>
        <h1>Grid Layout</h1>
    </header>
    <div class="container">
        <div class="item">Item 1</div>

```

```

        <div class="item">Item 2</div>
        <div class="item">Item 3</div>
        <div class="item">Item 4</div>
        <div class="item">Item 5</div>
    </div>
    <footer>
        <p>Footer Content</p>
    </footer>
</body>
</html>

```

5. Contact Form with Styles

Description: Create a styled contact form.

Features:

- Styled input fields and buttons
- Error messages

Code Snippet:

html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Contact Form</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
        .form-container { max-width: 600px; margin: 0 auto; padding:
2rem; }
        input, textarea { width: 100%; padding: 10px; margin: 5px 0; }
        button { background: #4CAF50; color: #fff; border: none; padding:
10px; cursor: pointer; }
        button:hover { background: #45a049; }
        .error { color: red; }
    </style>
</head>
<body>
    <div class="form-container">
        <h1>Contact Us</h1>
        <form>

```

```

        <input type="text" placeholder="Your Name" required>
        <input type="email" placeholder="Your Email" required>
        <textarea rows="4" placeholder="Your Message"></textarea>
        <button type="submit">Send</button>
    </form>
</div>
</body>
</html>

```

6. Responsive Navigation Bar

Description: Create a responsive navigation bar.

Features:

- Navigation links
- Responsive design with media queries

Code Snippet:

html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Responsive Navbar</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
        nav { background: #333; color: #fff; padding: 1rem; text-align:
center; }
        nav a { color: #fff; margin: 0 15px; text-decoration: none; }
        @media (max-width: 600px) {
            nav { font-size: 14px; }
            nav a { display: block; margin: 5px 0; }
        }
    </style>
</head>
<body>
    <nav>
        <a href="#">Home</a>
        <a href="#">About</a>
        <a href="#">Services</a>
        <a href="#">Contact</a>
    </nav>

```

```
</nav>
</body>
</html>
```

7. Simple Image Gallery

Description: Create a gallery of images with hover effects.

Features:

- Grid layout for images
- Hover effects

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Image Gallery</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
    .gallery { display: flex; flex-wrap: wrap; gap: 10px; padding: 1rem; }
    .gallery img { width: 200px; height: 150px; object-fit: cover;
transition: transform 0.3s; }
    .gallery img:hover { transform: scale(1.1); }
  </style>
</head>
<body>
  <div class="gallery">
    
    
    
    
  </div>
</body>
</html>
```

8. Pricing Table

Description: Design a pricing table with different plans.

Features:

- Table layout
- Highlighted pricing plan

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Pricing Table</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
    .pricing-table { display: flex; justify-content: center; padding: 2rem; }
    .plan { border: 1px solid #ccc; padding: 1rem; margin: 0 1rem; text-
align: center; }
    .plan.highlight { background: #4CAF50; color: #fff; }
    .plan h2 { margin: 0; }
  </style>
</head>
<body>
  <div class="pricing-table">
    <div class="plan">
      <h2>Basic</h2>
      <p>$10/month</p>
      <p>Feature 1</p>
      <p>Feature 2</p>
    </div>
    <div class="plan highlight">
      <h2>Pro</h2>
      <p>$30/month</p>
      <p>Feature 1</p>
      <p>Feature 2</p>
      <p>Feature 3</p>
    </div>
    <div class="plan">
      <h2>Premium</h2>
      <p>$50/month</p>
      <p>Feature 1</p>
      <p>Feature 2</p>
      <p>Feature 3</p>
    </div>
  </div>
```

```
        <p>Feature 4</p>
    </div>
</div>
</body>
</html>
```

9. Testimonial Slider

Description: Create a simple testimonial slider.

Features:

- Slide through testimonials
- Basic transition effect

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Testimonial Slider</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
        .slider { width: 300px; margin: 0 auto; overflow: hidden; }
        .slides { display: flex; transition: transform 0.5s; }
        .slide { min-width: 300px; padding: 1rem; text-align: center;
background: #f4f4f4; }
    </style>
</head>
<body>
    <div class="slider">
        <div class="slides">
            <div class="slide">"Great service!" - User A</div>
            <div class="slide">"Loved it!" - User B</div>
            <div class="slide">"Highly recommend!" - User C</div>
        </div>
    </div>

    <script>
        let slides = document.querySelector('.slides');
        let index = 0;
```

```

function showSlide() {
    index++;
    if (index >= slides.children.length) {
        index = 0;
    }
    slides.style.transform = `translateX(-${index * 300}px)`;
}

setInterval(showSlide, 3000);
</script>
</body>
</html>

```

10. Countdown Timer

Description: Create a countdown timer.

Features:

- Display days, hours, minutes, and seconds
- Update every second

Code Snippet:

html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Countdown Timer</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 0; padding: 0; text-align:
center; }
        .timer { font-size: 2rem; }
    </style>
</head>
<body>
    <h1>Countdown Timer</h1>
    <div class="timer" id="countdown"></div>

    <script>
        function updateCountdown() {

```

```

    let endDate = new Date('2024-12-31T23:59:59');
    let now = new Date();
    let timeLeft = endDate - now;

    let days = Math.floor(timeLeft / (1000 * 60 * 60 * 24));
    let hours = Math.floor((timeLeft % (1000 * 60 * 60 * 24)) / (1000 *
60 * 60));
    let minutes = Math.floor((timeLeft % (1000 * 60 * 60)) / (1000 *
60));
    let seconds = Math.floor((timeLeft % (1000 * 60)) / 1000);

    document.getElementById('countdown').textContent = `${days}d
${hours}h ${minutes}m ${seconds}s`;

    if (timeLeft < 0) {
        clearInterval(timerInterval);
        document.getElementById('countdown').textContent =
'EXPIRED';
    }
}

let timerInterval = setInterval(updateCountdown, 1000);
</script>
</body>
</html>

```

11. Simple To-Do List

Description: Create a basic to-do list where users can add and remove tasks.

Features:

- Add new tasks
- Remove tasks
- Basic styling

Code Snippet:

html

```

<!DOCTYPE html>
<html lang="en">
<head>

```

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<title>To-Do List</title>
<style>
  body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
  .container { max-width: 400px; margin: 0 auto; padding: 1rem; }
  ul { list-style: none; padding: 0; }
  li { padding: 10px; border-bottom: 1px solid #ddd; }
  button { background: #4CAF50; color: #fff; border: none; padding:
5px 10px; cursor: pointer; }
  button:hover { background: #45a049; }
</style>
</head>
<body>
  <div class="container">
    <h1>To-Do List</h1>
    <input type="text" id="taskInput" placeholder="Add a new task">
    <button onclick="addTask()">Add Task</button>
    <ul id="taskList"></ul>
  </div>

  <script>
    function addTask() {
      const input = document.getElementById('taskInput');
      const task = input.value;
      if (task) {
        const listItem = document.createElement('li');
        listItem.textContent = task;
        const removeBtn = document.createElement('button');
        removeBtn.textContent = 'Remove';
        removeBtn.onclick = function() {
          this.parentElement.remove();
        };
        listItem.appendChild(removeBtn);
        document.getElementById('taskList').appendChild(listItem);
        input.value = "";
      }
    }
  </script>
</body>
</html>
```

12. Responsive Blog Post

Description: Create a single blog post page with responsive design.

Features:

- Title and content
- Sidebar for related posts
- Responsive layout

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Blog Post</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
    .container { display: flex; flex-direction: column; margin: 0 auto;
max-width: 800px; padding: 1rem; }
    .main { flex: 3; padding: 1rem; }
    .sidebar { flex: 1; padding: 1rem; background: #f4f4f4; }
    @media (max-width: 600px) {
      .container { flex-direction: column; }
      .sidebar { order: -1; }
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="main">
      <h1>Blog Post Title</h1>
      <p>Blog post content goes here. Lorem ipsum dolor sit amet,
consectetur adipiscing elit.</p>
    </div>
    <div class="sidebar">
      <h2>Related Posts</h2>
      <ul>
        <li><a href="#">Related Post 1</a></li>
        <li><a href="#">Related Post 2</a></li>
      </ul>
    </div>
  </div>
</body>
</html>
```

```
        </ul>
    </div>
</div>
</body>
</html>
```

13. Login Form

Description: Design a simple login form with user-friendly styling.

Features:

- Username and password fields
- Submit button

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Login Form</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 0; padding: 0; display:
flex; justify-content: center; align-items: center; height: 100vh; }
    .form-container { max-width: 300px; width: 100%; padding: 2rem;
border: 1px solid #ccc; border-radius: 5px; box-shadow: 0 0 10px
rgba(0,0,0,0.1); }
    input { width: 100%; padding: 10px; margin: 5px 0; border: 1px solid
#ddd; border-radius: 3px; }
    button { background: #4CAF50; color: #fff; border: none; padding:
10px; cursor: pointer; border-radius: 3px; }
    button:hover { background: #45a049; }
  </style>
</head>
<body>
  <div class="form-container">
    <h1>Login</h1>
    <form>
      <input type="text" placeholder="Username" required>
      <input type="password" placeholder="Password" required>
      <button type="submit">Login</button>
```

```
        </form>
    </div>
</body>
</html>
```

14. CSS Card Layout

Description: Create a layout with styled cards for content.

Features:

- Cards with image, title, and description
- Responsive design

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Card Layout</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
    .container { display: flex; flex-wrap: wrap; gap: 20px; padding:
1rem; }
    .card { background: #fff; border: 1px solid #ddd; border-radius: 5px;
overflow: hidden; width: calc(33.333% - 20px); box-shadow: 0 0 10px
rgba(0,0,0,0.1); }
    .card img { width: 100%; }
    .card-content { padding: 15px; }
    .card-content h3 { margin: 0; }
    @media (max-width: 600px) {
      .card { width: 100%; }
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="card">
      
      <div class="card-content">
        <h3>Card Title 1</h3>
```



```

        <p>Description of the card content.</p>
    </div>
</div>
<div class="card">
    
    <div class="card-content">
        <h3>Card Title 2</h3>
        <p>Description of the card content.</p>
    </div>
</div>
<div class="card">
    
    <div class="card-content">
        <h3>Card Title 3</h3>
        <p>Description of the card content.</p>
    </div>
</div>
</div>
</body>
</html>

```

15. Newsletter Signup

Description: Design a simple newsletter signup form.

Features:

- Email input field
- Signup button

Code Snippet:

html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Newsletter Signup</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 0; padding: 0; display:
flex; justify-content: center; align-items: center; height: 100vh; }
        .signup-container { max-width: 400px; width: 100%; padding: 2rem;
border: 1px solid #ddd; border-radius: 5px; box-shadow: 0 0 10px

```

```

    rgba(0,0,0,0.1); }
    input { width: 100%; padding: 10px; margin: 5px 0; border: 1px solid
#ddd; border-radius: 3px; }
    button { background: #4CAF50; color: #fff; border: none; padding:
10px; cursor: pointer; border-radius: 3px; }
    button:hover { background: #45a049; }
  </style>
</head>
<body>
  <div class="signup-container">
    <h1>Newsletter Signup</h1>
    <form>
      <input type="email" placeholder="Your Email" required>
      <button type="submit">Subscribe</button>
    </form>
  </div>
</body>
</html>

```

16. Profile Card

Description: Create a profile card with user information.

Features:

- Profile picture
- User details

Code Snippet:

html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Profile Card</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 0; padding: 0; display:
flex; justify-content: center; align-items: center; height: 100vh; }
    .profile-card { max-width: 300px; width: 100%; border: 1px solid
#ddd; border-radius: 5px; text-align: center; padding: 1rem; box-shadow:
0 0 10px rgba(0,0,0,0.1); }
    .profile-card img { width: 100px; height: 100px; border-radius:

```

```

50%; }
.profile-card h2 { margin: 10px 0; }
</style>
</head>
<body>
  <div class="profile-card">
    
    <h2>John Doe</h2>
    <p>Web Developer</p>
  </div>
</body>
</html>

```

17. Custom Checkboxes and Radios

Description: Style custom checkboxes and radio buttons.

Features:

- Custom designs for checkboxes and radios

Code Snippet:

html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Custom Checkboxes and Radios</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
    .custom-checkbox, .custom-radio { display: flex; align-items: center;
margin-bottom: 10px; }
    .custom-checkbox input[type="checkbox"], .custom-radio
input[type="radio"] { display: none; }
    .custom-checkbox label, .custom-radio label { position: relative;
padding-left: 25px; cursor: pointer; }
    .custom-checkbox label::before, .custom-radio label::before
{ content: ""; position: absolute; left: 0; top: 0; width: 20px; height: 20px;
border: 2px solid #ddd; border-radius: 3px; background: #fff; }
    .custom-radio label::before { border-radius: 50%; }
    .custom-checkbox input[type="checkbox"]:checked + label::before
{ background: #4CAF50; border-color: #4CAF50; }

```

```

        .custom-radio input[type="radio"]:checked + label::before
    { background: #4CAF50; border-color: #4CAF50; }
    </style>
</head>
<body>
    <div class="custom-checkbox">
        <input type="checkbox" id="checkbox1">
        <label for="checkbox1">Custom Checkbox</label>
    </div>
    <div class="custom-radio">
        <input type="radio" id="radio1" name="radio-group">
        <label for="radio1">Custom Radio</label>
    </div>
</body>
</html>

```

18. Footer with Social Media Links

Description: Create a footer with social media icons.

Features:

- Social media links
- Responsive design

Code Snippet:

```

html

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Footer with Social Media</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
        footer { background: #333; color: #fff; text-align: center; padding:
1rem; }
        .social-icons { list-style: none; padding: 0; margin: 0; }
        .social-icons li { display: inline; margin: 0 10px; }
        .social-icons a { color: #fff; text-decoration: none; font-size: 20px; }
        .social-icons a:hover { color: #4CAF50; }
    </style>
</head>

```

```

<body>
  <footer>
    <ul class="social-icons">
      <li><a href="#" class="facebook">Facebook</a></li>
      <li><a href="#" class="twitter">Twitter</a></li>
      <li><a href="#" class="instagram">Instagram</a></li>
    </ul>
  </footer>
</body>
</html>

```

19. Subscription Box

Description: Create a subscription box with a call-to-action button.

Features:

- Email input
- Subscription button

Code Snippet:

html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Subscription Box</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
    .subscription-box { max-width: 500px; width: 100%; padding: 2rem;
margin: 0 auto; border: 1px solid #ddd; border-radius: 5px; text-align:
center; }
    input { width: 80%; padding: 10px; margin: 10px 0; border: 1px solid
#ddd; border-radius: 3px; }
    button { background: #4CAF50; color: #fff; border: none; padding:
10px; cursor: pointer; border-radius: 3px; }
    button:hover { background: #45a049; }
  </style>
</head>
<body>
  <div class="subscription-box">
    <h1>Subscribe Now</h1>

```

```

        <p>Stay updated with our latest news and offers.</p>
        <input type="email" placeholder="Your Email" required>
        <button type="submit">Subscribe</button>
    </div>
</body>
</html>

```

20. Content Accordion

Description: Create an accordion to expand and collapse content sections.

Features:

- Expand and collapse content
- Animated transitions

Code Snippet:

html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Content Accordion</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
        .accordion { border: 1px solid #ddd; border-radius: 5px; overflow:
hidden; }
        .accordion-item { border-bottom: 1px solid #ddd; }
        .accordion-header { background: #f4f4f4; padding: 10px; cursor:
pointer; }
        .accordion-content { padding: 0 10px; display: none; }
        .accordion-content.show { display: block; }
    </style>
</head>
<body>
    <div class="accordion">
        <div class="accordion-item">
            <div class="accordion-header">Section 1</div>
            <div class="accordion-content">Content for section 1.</div>
        </div>
        <div class="accordion-item">

```

```

    <div class="accordion-header">Section 2</div>
    <div class="accordion-content">Content for section 2.</div>
  </div>
</div>

<script>
  const headers = document.querySelectorAll('.accordion-header');
  headers.forEach(header => {
    header.addEventListener('click', () => {
      const content = header.nextElementSibling;
      content.classList.toggle('show');
    });
  });
</script>
</body>
</html>

```

21. Modal Window

Description: Create a modal window that opens when a button is clicked.

Features:

- Modal content
- Close button

Code Snippet:

html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Modal Window</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
    .modal { display: none; position: fixed; top: 0; left: 0; width: 100%;
height: 100%; background: rgba(0,0,0,0.5); justify-content: center; align-
items: center; }
    .modal-content { background: #fff; padding: 1rem; border-radius:
5px; width: 80%; max-width: 500px; }
    .close { float: right; cursor: pointer; }
  </style>

```

```

    </style>
</head>
<body>
  <button onclick="openModal()">Open Modal</button>
  <div class="modal" id="myModal">
    <div class="modal-content">
      <span class="close" onclick="closeModal()">&times;</span>
      <h2>Modal Title</h2>
      <p>Some content for the modal.</p>
    </div>
  </div>

  <script>
    function openModal() {
      document.getElementById('myModal').style.display = 'flex';
    }
    function closeModal() {
      document.getElementById('myModal').style.display = 'none';
    }
  </script>
</body>
</html>

```

22. Navigation Bar

Description: Design a horizontal navigation bar with links.

Features:

- Navigation links
- Responsive design

Code Snippet:

html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Navigation Bar</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
    nav { background: #333; color: #fff; padding: 10px; }

```



```

    nav ul { list-style: none; margin: 0; padding: 0; display: flex; }
    nav ul li { margin-right: 20px; }
    nav ul li a { color: #fff; text-decoration: none; }
    nav ul li a:hover { text-decoration: underline; }
  </style>
</head>
<body>
  <nav>
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">About</a></li>
      <li><a href="#">Services</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </nav>
</body>
</html>

```

23. Pricing Table

Description: Create a pricing table with different plans.

Features:

- Pricing options
- Plan details

Code Snippet:

html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Pricing Table</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
    .pricing-table { display: flex; justify-content: center; gap: 20px;
padding: 2rem; }
    .pricing-card { background: #fff; border: 1px solid #ddd; border-
radius: 5px; padding: 1rem; width: 200px; text-align: center; }
    .pricing-card h2 { margin: 10px 0; }
    .pricing-card p { margin: 0; }

```

```

.pricing-card button { background: #4CAF50; color: #fff; border:
none; padding: 10px; cursor: pointer; border-radius: 3px; margin-top:
10px; }
.pricing-card button:hover { background: #45a049; }
</style>
</head>
<body>
  <div class="pricing-table">
    <div class="pricing-card">
      <h2>Basic Plan</h2>
      <p>$19/month</p>
      <p>Basic features</p>
      <button>Sign Up</button>
    </div>
    <div class="pricing-card">
      <h2>Pro Plan</h2>
      <p>$39/month</p>
      <p>Advanced features</p>
      <button>Sign Up</button>
    </div>
    <div class="pricing-card">
      <h2>Premium Plan</h2>
      <p>$59/month</p>
      <p>All features</p>
      <button>Sign Up</button>
    </div>
  </div>
</body>
</html>

```

24. Event Countdown Timer

Description: Design a countdown timer for an event.

Features:

- Days, hours, minutes, and seconds countdown

Code Snippet:

html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">

```

```

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Countdown Timer</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 0; padding: 0; display:
flex; justify-content: center; align-items: center; height: 100vh;
background: #f4f4f4; }
        .countdown { text-align: center; }
        .countdown div { font-size: 2rem; margin: 0.5rem; }
    </style>
</head>
<body>
    <div class="countdown">
        <div id="days">00</div>
        <div id="hours">00</div>
        <div id="minutes">00</div>
        <div id="seconds">00</div>
    </div>

    <script>
        function updateCountdown() {
            const endDate = new Date('2024-12-31T23:59:59').getTime();
            const now = new Date().getTime();
            const timeLeft = endDate - now;

            const days = Math.floor(timeLeft / (1000 * 60 * 60 * 24));
            const hours = Math.floor((timeLeft % (1000 * 60 * 60 * 24)) /
(1000 * 60 * 60));
            const minutes = Math.floor((timeLeft % (1000 * 60 * 60)) / (1000 *
60));
            const seconds = Math.floor((timeLeft % (1000 * 60)) / 1000);

            document.getElementById('days').textContent = days;
            document.getElementById('hours').textContent = hours;
            document.getElementById('minutes').textContent = minutes;
            document.getElementById('seconds').textContent = seconds;
        }

        setInterval(updateCountdown, 1000);
        updateCountdown();
    </script>
</body>
</html>

```

25. Basic Contact Form

Description: Design a basic contact form with fields for name, email, and message.

Features:

- Name, email, and message fields
- Submit button

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Contact Form</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 0; padding: 0; display:
flex; justify-content: center; align-items: center; height: 100vh; }
    .contact-form { max-width: 600px; width: 100%; padding: 2rem;
border: 1px solid #ddd; border-radius: 5px; }
    input, textarea { width: 100%; padding: 10px; margin: 5px 0; border:
1px solid #ddd; border-radius: 3px; }
    button { background: #4CAF50; color: #fff; border: none; padding:
10px; cursor: pointer; border-radius: 3px; }
    button:hover { background: #45a049; }
  </style>
</head>
<body>
  <div class="contact-form">
    <h1>Contact Us</h1>
    <form>
      <input type="text" placeholder="Your Name" required>
      <input type="email" placeholder="Your Email" required>
      <textarea rows="5" placeholder="Your Message" required></
textarea>
      <button type="submit">Send Message</button>
    </form>
  </div>
</body>
```

</html>

26. CSS Grid Layout

Description: Create a layout using CSS Grid with multiple sections.

Features:

- Grid layout with rows and columns

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>CSS Grid Layout</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
    .grid-container { display: grid; grid-template-columns: repeat(3, 1fr);
gap: 10px; padding: 1rem; }
    .grid-item { background: #f4f4f4; padding: 20px; text-align: center;
border: 1px solid #ddd; }
  </style>
</head>
<body>
  <div class="grid-container">
    <div class="grid-item">Item 1</div>
    <div class="grid-item">Item 2</div>
    <div class="grid-item">Item 3</div>
    <div class="grid-item">Item 4</div>
    <div class="grid-item">Item 5</div>
    <div class="grid-item">Item 6</div>
  </div>
</body>
</html>
```

27. CSS Flexbox Layout

Description: Create a layout using Flexbox with multiple sections.

Features:

- Flexbox layout with rows and columns

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>CSS Flexbox Layout</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
    .flex-container { display: flex; flex-wrap: wrap; gap: 10px; padding:
1rem; }
    .flex-item { background: #f4f4f4; padding: 20px; text-align: center;
border: 1px solid #ddd; flex: 1; }
  </style>
</head>
<body>
  <div class="flex-container">
    <div class="flex-item">Item 1</div>
    <div class="flex-item">Item 2</div>
    <div class="flex-item">Item 3</div>
    <div class="flex-item">Item 4</div>
    <div class="flex-item">Item 5</div>
    <div class="flex-item">Item 6</div>
  </div>
</body>
</html>
```

28. Simple Table

Description: Create a simple table with headers and data rows.

Features:

- Table with headers and data

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<title>Simple Table</title>
<style>
  body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
  table { width: 100%; border-collapse: collapse; margin: 1rem 0; }
  th, td { border: 1px solid #ddd; padding: 8px; text-align: left; }
  th { background: #f4f4f4; }
</style>
</head>
<body>
  <table>
    <thead>
      <tr>
        <th>Header 1</th>
        <th>Header 2</th>
        <th>Header 3</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Data 1</td>
        <td>Data 2</td>
        <td>Data 3</td>
      </tr>
      <tr>
        <td>Data 4</td>
        <td>Data 5</td>
        <td>Data 6</td>
      </tr>
    </tbody>
  </table>
</body>
</html>

```

29. Image Gallery

Description: Create a simple image gallery with thumbnails.

Features:

- Thumbnail images
- Grid layout

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Image Gallery</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
    .gallery { display: grid; grid-template-columns: repeat(auto-fill,
minmax(150px, 1fr)); gap: 10px; padding: 1rem; }
    .gallery img { width: 100%; height: auto; border: 1px solid #ddd;
border-radius: 5px; }
  </style>
</head>
<body>
  <div class="gallery">
    
    
    
    
  </div>
</body>
</html>
```

30. Responsive Design Example

Description: Create a responsive layout that adjusts for mobile devices.

Features:

- Responsive grid layout

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
```



```
scale=1.0">
  <title>Responsive Design</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
    .container { display: flex; flex-wrap: wrap; gap: 10px; padding:
1rem; }
    .box { background: #f4f4f4; padding: 20px; text-align: center; flex: 1;
min-width: 200px; }
  </style>
</head>
<body>
  <div class="container">
    <div class="box">Box 1</div>
    <div class="box">Box 2</div>
    <div class="box">Box 3</div>
    <div class="box">Box 4</div>
  </div>
</body>
</html>
```