

## Week 1

### Week 1 - Fix the code

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>Tech Shop Online</h1>
    <h2>Our Mission</h2>
    <p>Our mission is to provide affordable, high-quality electronics
to customers around the world.</p>
    <h3>Contact Information</h3>
    <p>We are committed to providing excellent customer service. Reach
us at contact@techshop.com or call 1-800-123-4567.</p>
    <footer><p>2025 Tech Shop Online. All rights
reserved.</p></footer>
    <h2>Why Choose Us?</h2>
    <p>At Tech Shop Online, we offer the best deals on electronics,
fast shipping, and excellent customer support. We make sure that every
product you purchase meets our highest quality standards.</p>
    <p>We offer a range of premium laptops, including the latest
models with high-speed processors and long-lasting batteries. Perfect
for both work and play!</p>
</body>
</html>
```

### Week 1 - Type the code

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Tech Conference 2025</title>
</head>
<body>
```

```

<h1>Tech Conference 2025</h1>
<h3><em>Innovating the Future of Technology</em></h3>
<h2>Conference Introduction</h2>
<p>The <b>exciting</b> and <b>innovative</b> Tech Conference 2025 is a gathering of the brightest minds in <i>technology.</i> This event aims to showcase cutting-edge advancements and provide a platform for professionals to share their knowledge.</p>
<h3>Event Details</h3>
<p>This year's conference will feature <strong>global speakers</strong> and offer numerous <em>networking opportunities</em> for attendees to connect, learn, and collaborate.</p>
<h2>Schedule</h2>
<h3>Day 1</h3>
<p>Session 1:<b> Artificial Intelligence in 2025</b></p>
<p>Session 2:<b> Future of Machine Learning</b></p>
<h3>Day 2</h3>
<p>Session 1:<b> Blockchain Revolution</b></p>
<p>Session 2:<b> Innovation in Cybersecurity</b></p>
<h2>Speakers</h2>
<h5><span>Dr. John Doe</span></h5>
<p><span><b>CEO, Tech Innovations Inc.</b></span></p>
<h5><span>Jane Smith</span></h5>
<p><span><i><b>Lead Engineer, AI Solutions</b></i></span></p>
<footer><h6>Contact Us</h6>
<p>For more information, email us at <u>contact@techconference.com</u> or visit our website at <span>www.techconference2025.com.</span></p></footer>
</body>
</html>

```

## Week 1 - Practice 1

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Top Tourist Destinations</title>
</head>
<body>
    <h1>Top Tourist Destinations</h1>
    <ol>
        <li><h2><b>Paris</b></h2>
            <p><ul><li><b>Description</b>: Known for its romantic ambiance and iconic landmarks like the Eiffel Tower.</li></ul></p>
            <ul><li><img alt="paris" src = "paris.jpg" width="300px" /></li></ul>
        </li>
    </ol>

```

```

<li><h2><b>New York</b></h2>
<p><ul><li><b>Description</b>: Famous for the Statue of Liberty, Times Square, and Central Park.</li></ul></p>
<ul><li><img src = "new-york.jpg" width = "300" alt ="New York"></li></ul>
</li>

<li><h2><b>Tokyo</b></h2>
<p><ul><li><b>Description</b>: A blend of modern skyscrapers and traditional temples, with vibrant districts like Shibuya.</li></ul></p>
<ul><li><img src = "tokyo.jpg" width ="300" alt="Tokyo"></li></ul>
</li>

<li><h2><b>Rome</b></h2>
<p><ul><li><b>Description</b>: Known for its ancient history, including landmarks like the Colosseum and Vatican City.</li></ul></p>
<ul><li><img src = "rome.jpg" width = "300" alt ="Rome"></li></ul>
</li>

<li><h2><b>London</b></h2>
<p><ul><li><b>Description</b>: Famous for its rich history, the Tower of London, Buckingham Palace, and the London Eye.</li></ul></p>
<ul><li><img src = "london.jpg" width = "300" alt="london"></li></ul>
</li>
</ol>
</body>
</html>

```

Week 1 - Practice 2

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Tech Conference 2025</title>
</head>
<body>
<header>
<h1>Tech Conference 2025</h1>
<h3><b><em>Innovating the Future of Technology</em></b></h3>
<h2>Conference Introduction</h2>
<p><b>exciting</b> <b>innovative</b> <i>technology</i></p>
<h3>Event Details</h3>
<p><strong>global speakers</strong> <em>networking opportunities</em></p>
<h2>Schedule</h2>
<h3>Day 1</h3>
<p>Session 1: <b>Artificial Intelligence in 2025</b></p>

```

```

<p>Session 2: <b>Futures of Machine Learning</b></p>
<h2>Speakers</h2>
<h5>Dr. John Doe</h5>
<span><b>CEO, Tech Innovations Inc.</b></span>
<h5>Jane Smith</h5>
<span><i><b>Lead Engineer, AI Solutions</b></i></span>
<footer>
<h6>Contact Us</h6>
<p>For more information, email us at <u>contact@techconference.com</u>
or visit our
website at <span>www.techconference2025.com</span></p>
</footer>
</header>
</body>
</html>

```

Week 1 - Challenge

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>The Reading Nook - Your Favorite Bookstore</title>
</head>
<body>
<header>
<h1>Welcome to The Reading Nook</h1>
<h3>Where every book has a story</h3>
</header>
<h2>Our Mission</h2>
<p>At The Reading Nook, we are dedicated to providing <b>quality
books</b> and
<i><u>literature</u></i> to all the readers</p>
<h3>New Arrivals</h3>
<p><i>The Silent Sea</i> - A misy-read for all the fans of mystery and
adventure.</p>
<h2>Customer Reviews</h2>
<h3><span><b>Jane Doe</b></span></h3>

```

<p>Rating: 5/5. One of the best books I've read this year. Highly recommended <span style="text-decoration: underline;">mystery</span> genre lovers to try it.</p>

```
<footer>
<h6>Contact Us</h6>
<p>Email: <u>info@readingnook.com</u> | Phone: <span><u>123-456-7890</u></span></p>
</footer>
<h2>Why Choose Us</h2>
<p>We offer a <strong>wide selection</strong> of books at
<em>affordable prices</em>.</p>
</body>
</html>
```

## Week 2

### WEEK-2 TYPE THE CODE

```
<!-- write your code here -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Top Tourist Destinations</title>
</head>
<body>
    <h1>Top Tourist Destinations</h1>
    <ol>
        <li>
            <h2>Paris</h2>
            <ul>
                <li>
                    <p><b>Description:</b>
                        Known for its romantic ambiance and iconic
                        landmarks like the Eiffel Tower.
                    </p>
                </li>
                <li>
                    
                </li>
            </ul>
        </li>

        <li>
            <h2>New York</h2>
            <ul>
                <li>
                    <p><b>Description:</b>
                        Famous for the Statue of Liberty, Times Square,
                        and Central Park.
                    </p>
                </li>
                <li>

```

```
        
    </li>
</ul>
</li>

<li>
    <h2>Tokyo</h2>
    <ul>
        <li>
            <p><b>Description:</b>
                A blend of modern skyscrapers and
                traditional temples, with vibrant districts
                like Shibuya.
            </p>
        </li>
        <li>
            
        </li>
    </ul>

</li>
<li>
    <h2>Rome</h2>
    <ul>
        <li>
            <p><b>Description:</b>
                Known for its ancient history, including
                landmarks like Colasseum and Vatican City.
            </p>
        </li>
        <li>
            
        </li>
    </ul>

</li>
<li>
```

```
<h2>London</h2>
<ul>
  <li>
    <p><b>Description:</b>
      Famous for its rich history, the Tower of London,
      Buckingham Palace, and the London Eye.
    </p>
  </li>
  <li>
    
  </li>
</ul>

</li>
</ol>

</body>
</html>
```

## WEEK-2 FIX THE CODE

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Luxury Stays - Book Your Dream Room</title>
    <meta name="description" content="Luxury Stays offers the finest rooms
and amenities.">
</head>
<body>

    <header>
        <h1>Welcome to Luxury Stays</h1>
        <h2>Your Dream Stay Awaits</h2>
        
        <nav>
            <ul>
                <li><a href="#about">About the Hotel</a></li>
                <li><a href="#rooms">Rooms and Pricing</a></li>
                <li><a href="#services">Hotel Services</a></li>
                <li><a href="#gallery">Gallery</a></li>
                <li><a href="#contact">Contact Us</a></li>
                <!-- <li><a href="#special-offers">Special Offers</a></li>
-->
            </ul>
        </nav>
    </header>

    <section id="about">
        <h2>About the Hotel</h2>
        <p>Luxury Stays offers the finest hospitality with exceptional
service, comfortable rooms, and world-class amenities.
        <!-- <strong>Best stay ever</strong> -->
    </section>

    <section id="rooms">
        <h2>Rooms and Pricing</h2>
        <table border="1">
```

```

<thead>
  <tr>
    <th>Room Type</th>
    <th>Price per Night</th>
    <th>Description</th>
  </tr>
</thead>
<tbody>
  <tr>
    <td>Single Room</td>
    <td>$200</td>
    <td>A comfortable room with a single bed and basic amenities.</td>
  </tr>
  <tr>
    <td>Double Room</td>
    <td>$150</td>
    <td>A spacious room with a king-size bed and a beautiful view.</td>
  </tr>
  <tr>
    <td>Suite</td>
    <td>$250</td>
    <td>A luxurious suite with a living area, jacuzzi, and premium amenities.</td>
  </tr>
  <tr>
    <td>Presidential Suite</td>
    <td>$500</td>
    <td>An extravagant suite with multiple rooms, a private balcony, and butler service.</td>
  </tr>
</tbody>
</table>
</section>

<section id=services>
  <h2>Hotel Services</h2>
  <ul>
    <li>Free Wi-Fi</li>

```

```
<li>24/7 Room Service</li>
<li>Swimming Pool</li>
<li>Gym</li>
<li>Airport Shuttle</li>
</ul>
</section>
<section id="gallery">
    <h2>Gallery</h2>
    
    

    
    <!--  <!-- -->
</section>
<aside>
    <h3>Customer Reviews</h3>
    <h4>John Doe</h4>
    <p><q>The Stay was amazing. The service was exceptional,
        and the room was perfect!</q></p>
    <h4>Jane Smith</h4>
    <p><q>A wonderful exprerience. Highly recommend
        the suite for a luxury stay!</q></p>
    <h4>Mark Johnson</h4>
    <p><q>Great location, fantastic amenities, and friendly
        staff. Will definitely return!</q></p>

</aside>

<footer id="contact">
    <h2>Contact Us</h2>
    <p>For booking inquiries, contact us at: info@luxurystays.com</p>
    <p>&copy 2025 Luxury Stays. All rights reserved.</p>
</footer>

<section>
    <h2>Register Now</h2>
    <ul>
        <li>Standard Pass - $99</li>
        <li>VIP Pass - $199</li>
    </ul>
</section>
```

```
<li>Student Pass - $49</li>
</ul>
</section>
```

## WEEK-2 PRACTICE AT HOME-1

```
<!-- write your code here -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Product Catalog</title>
</head>
<body>
    <h1>Welcome to Our Product Catalog</h1>
    <p>
        Welcome to our product catalog. Here you will find a selection
        of the latest and most popular products available in our
        store.
    </p>

    <h2>Our Products</h2>
    <ul>
        <li>Electronics</li>
        <li>Accessories</li>
        <li>Home Appliances</li>
    </ul>

    <table border="1">
        <thead>
            <tr>
                <th>Product Name</th>
                <th>Price</th>
                <th>Description</th>
            </tr>
        </thead>
        <tbody>
```

```
<tr>
    <td>Laptop</td>
    <td>$1200</td>
    <td>A high-performance laptop</td>
</tr>
<tr>
    <td>Smartphone</td>
    <td>$800</td>
    <td>A latest model smartphone</td>
</tr>
<tr>
    <td>Headphones</td>
    <td>$150</td>
    <td>Noise-cancelling headphones</td>
</tr>
<tr>
    <td>Smartwatch</td>
    <td>$250</td>
    <td>Fitness tracking smartwatch</td>
</tr>
<tr>
    <td>Tablet</td>
    <td>$500</td>
    <td>A lightweight tablet with great features</td>
</tr>

</tbody>
</table>

</body>
</html>
```

## WEEK-2 PRACTICE AT HOME -2

```
<!-- write your code here -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>E-Commerce Product Catalog</title>
</head>
<body>
    <header>
        <h1>Welcome to Our E-Commerce Store</h1>
        <p>We offer a wide variety of products to suit all your needs.</p>
    </header>

    <section>
        <h2>Product Categories</h2>
        <ul>
            <li>Electronics</li>
            <li>Clothing</li>
            <li>Home Appliances</li>
        </ul>
    </section>
    <section>
        <h2>Product Catalog</h2>
        <table border="1">
            <thead>
                <tr>
                    <th>Product Name</th>
                    <th>Price</th>
                    <th>Description</th>
                    <th>Image</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>Laptop</td>
                    <td>$1200</td>
```

```

        <td>A high-performance laptop</td>
        <td></td>
    </tr>
    <tr>
        <td>Smartphone</td>
        <td>$800</td>
        <td>Latest model smartphone</td>
        <td></td>
    </tr>
    <tr>
        <td>Headphones</td>
        <td>$150</td>
        <td>Noise-cancelling headphones</td>
        <td></td>
    </tr>
    <tr>
        <td>Smartwatch</td>
        <td>$250</td>
        <td>Fitness tracking smartwatch</td>
        <td></td>
    </tr>
    <tr>
        <td>Tablet</td>
        <td>$500</td>
        <td>A lightweight tablet with great features</td>
        <td></td>
    </tr>
</tbody>
</table>
</section>

<section>
    <h2>Best-Selling Products</h2>
    <article>
        <h3>Laptop</h3>

```

```
<p>A high performance laptop for work and environment.</p>

<ul>
    <li>Fast Processor</li>
    <li>Large screen</li>
    <li>Long battery life</li>
</ul>
</article>
<article>
    <h3>Smartphone</h3>
    <p>Latest model smartphone with advanced features.</p>
    
    <ul>
        <li>High-resolution camera</li>
        <li>Fast performance</li>
        <li>Long battery life</li>
    </ul>
</article>
<article>
    <h3>Smartwatch</h3>
    <p>Fitness tracking smartwatch for a healthy lifestyle.</p>
    
    <ul>
        <li>Track steps and heart rate</li>
        <li>Water-resistant</li>
        <li>Long battery life</li>
    </ul>
</article>

</section>

<sidebar>
    <h3>Special Offers</h3>
    <ul>
        <li>50% off on selected electronics</li>
        <li>Buy 1, Get 1 Free on accessories</li>
    </ul>
</sidebar>
```

```
        <li>Seasonal sale: Up to 40% off on home appliances</li>
    </ul>
</sidebar>

<section>
    <h2>Customer Testimonials</h2>
    <h3>John Doe</h3>
    <p><q>I am thrilled with my new laptop from this store.  

        The delivery was fast, and the product is amazing!</q></p>

    <h3>Jane Smith</h3>
    <p><q>The smartphone I bought exceeded my expectations. Great  

        performance and value for money!</q></p>

    <h3>Mark Johnson</h3>
    <p><q>The smartphone I purchased is exactly what I needed  

        to stay fit. Highly recommended!</q></p>

</section>

<footer>
    <p>Contact us at <u>info@ecommerce.com</u></p>
    <p>2025 E-Commerce Store. All rights reserved.</p>
</footer>

</body>
</html>
```

## WEEK-2 CHALLENGE YOURSELF

```
<!-- write your code here -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Tech Innovators 2025 - Event Page</title>
</head>
<body>
    <header>
        <h1>Tech Innovators 2025</h1>
        <h2>Innovating the Future of Technology</h2>
        
    </header>

    <section>
        <h2>About the Event</h2>
        <p>Tech Innovators 2025 is a premier event that brings minds
            in technology. Join us to explore the latest trends and
            innovation in the tech industry, network with thought leaders,
            and gain insights into the future of technology.
        </p>
    </section>

    <section>
        <h2>Event Schedule</h2>
        <table border="1">
            <thead>
                <tr>
                    <th>Time</th>
                    <th>Session</th>
                    <th>Speaker</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>9:00 AM</td>
```

```

        <td>Keynote: The Future of AI</td>
        <td>Dr. John Doe</td>
    </tr>

    <tr>
        <td>11:00 AM</td>
        <td>Blockchain for Business</td>
        <td>Jane Smith</td>
    </tr>

    <tr>
        <td>1:00 PM</td>
        <td>Innovation in Cloud Computing</td>
        <td>Mark Johnson</td>
    </tr>

    <tr>
        <td>3:00 PM</td>
        <td>Tech Startups: How to Disrupt the Market </td>
        <td>Alice Brown</td>
    </tr>

    <tr>
        <td>5:00 PM</td>
        <td>Clouding Remarks & Networking</td>
        <td>Event Organization</td>
    </tr>

</tbody>
</table>
</section>

```

```

<section>
    <h2>Speakers</h2>
    <article>
        <h3>Dr. John Doe</h3>
        <p>Dr. John Doe is a renowned expert in AI with over
            20 years of experience in the field. He will be
            discussing the future of artificial intelligence and

```

```
        its impact on various industries.  
    </p>  
      
    </article>  
  
<article>  
    <h3>Jane Smith</h3>  
    <p>Jane Smith is a leading blockchain expert  
and a thought leader in decentralized technologies. She will  
discuss the impact of blockchain on business and finance.  
    </p>  
      
    </article>  
  
  
<article>  
    <h3>Mark Johnson</h3>  
    <p>Mark Johnson is a cloud computing visionary and CEO  
of CloudTech Innovations. He will share his insights into  
the latest  
        advancements in cloud infrastructure.  
    </p>  
      
    </article>  
  
  
</section>  
  
<section>  
    <h2>Venue</h2>  
    <p>The event will be held at the Grand Tech Conference  
Center, located in downtown Silicon Valley.  
    </p>  
      
</section>  
  
<aside>
```

```
<h3>Our Sponsors</h3>
<ul>
  <li>TechCorp</li>
  <li>Innovate Solutions</li>
  <li>Global Technologies</li>
</ul>
</aside>

<footer>
  <h2>Contact Us</h2>
  <p>For more information, email us at
info@techinnovators2025.com</p>
  <p>Copyrights, 2025 Tech Innovators 2025. All rights reserved.</p>

</footer>

<section>
  <h2>Register Now</h2>
  <ul>
    <li>General Admissions: $1000</li>
    <li>VIP Admissions: $250</li>
    <li>Student Admissions: $50</li>
  </ul>
</section>

</body>
</html>
```

## WEEK 3

### WEEK-3 TYPE THE CODE

```
<!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>
    /* Custom styles here */

    .profile-card{
      display: flex;
      align-items: center;
      flex-direction: column;
      box-shadow: 0 4px 8px rgba(0,0,0,0.1);
      border-radius: 0.75rem;
      margin:0px;
      overflow: hidden;
      background-color: rgb(255,255,255);
    }

    .headerClass{
      color:white;
      background-color:rgb(76,174,79);
      width:97vw;
      text-align: center;
      border-radius: 0.75rem;
      padding:20px;
      margin-top: 0px;
      margin-bottom: 20px;
    }

  }

  p{
    text-align: center;
  }
```

```
.button{  
    background-color: rgb(76,174,79);  
    color:white;  
    border-radius: 0.5rem;  
    border:none;  
    border-color: none;  
    padding:8px;  
    margin-top: 20px;  
    margin-bottom: 20px;  
  
}  
  
.button:hover{  
    background-color: rgb(69,170,63);  
}  
</style>  
</head>  
<body>  
  
<div class="profile-card">  
    <header class="headerClass">  
        <h2>John Doe</h2>  
  
    </header>  
  
    <section >  
        <p>Web Developer at XYZ Company.</p>  
        <p>Loves coding and coffee.</p>  
    </section>  
  
    <button class="button">  
        Contact  
    </button>  
</div>  
  
</body>  
</html>
```

### WEEK-3 FIX THE CODE

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>SkillUp Login</title>
    <link rel="stylesheet" href="./styles.css">
</head>
<body>
    <div class="login-form">
        <h1>Welcome to SkillUp-Learn, Grow, Succeed!</h1>
        <form>
            <h2>Login Form</h2>
            <username>
                Username
            </username>
            <input type="text" name="username" id="username">
            <br>
            <password>
                Password
            </password>
            <input type="password" name="password" id="password">
            <br>
            <button type="submit" id="submitButton">Login</button>

        </form>
    </div>
    <script> src="script.js" </script>
</body>
</html>
```

```
.login-form {
    background-color: rgb(255,255,255);
    border-radius: 5px;
    margin-top: 50px;
    padding: 30px;
}

h1 {
    color: rgba(255, 0, 0);
    font-size: 100px;
}

input[type="text"], input[type="password"] {
    width: 250px;
    height: 50px;
    font-size: 16px;
    padding: 20px;
    margin: 10px 0;
    border: 1px solid;
    background-color: transparent;
}

button[type="submit"] {

    padding: 15px 30px;
    width: 250px;
    cursor: non-resizable;
    text-align: center;
}

#submitButton{
    background-color: rgb(0,0,255);
    font-size: 16px;
    color: white;

}

input[type="text"]:focus, input[type="password"]:focus {
    outline: none;
}
```

```
}

label {
    font-size: 18px;
    font-weight: bold;
    display: block;
    text-align: left;
    color: red;
}

span {
    display: none;
}

form{
    display: flex;
    align-items: center;
    flex-direction: column;
}
```

### WEEK-3 PRACTICE AT HOME - QN 1

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Contact Us Web Page</title>
    <link rel="stylesheet" href="./style.css" >
</head>
<body>
    <!-- write your code here -->
    <header>
        <h1>Contact Us</h1>
    </header>
    <nav class="navcont">
        <ul class="ultext" >
            <a href="#home">HOME</a>
            <a href="products">PRODUCTS</a>
            <a href="#about">ABOUT</a>
            <a href="#contact">CONTACT</a>
        </ul>
    </nav>
    <section class="container">
        <h1>Get in Touch</h1>
        <p>Feel free to reach out to us for any queries or feedback.</p>
        <form class="contact-form">
            <input type="text" placeholder="Your Name">
            <input type="email" placeholder="Your Email">
            <input type="text" placeholder="Your message">
            <button type="submit">Send Message</button>
        </form>

    </section>
    <footer>
        <p>2024 Online Store. All Rights Reserved.</p>
    </footer>
</body>
</html>
```

```
header{
    background-color: rgb(0, 115, 230);
    padding: 16px;

    margin-bottom: 0;
}

header h1{
    color:white;
    text-align: center;
}

nav{
    display:flex;
    align-items: center;
    justify-content: center;
    background-color: rgb(77, 146, 215);
    margin-top: 0;

}

a{
    text-decoration: none;
    color:white;
    margin-left: 20px;
}

.ultext{
    text-transform: uppercase;
}

.container{
    display: flex;
    align-items: center;
    flex-direction: column;
    max-width: 1200px;
}

.contact-form{
```

```
    display: flex;
    align-items: center;
    border-radius: 0.75rem;
    flex-direction: column;

}

input{
    border: none;

    margin:10%;
    width:150%
}

button{
    text-align: center;
    color: rgb(255,255,255);
    background-color: rgb(0,115,230);
    border:none;
    border-radius: 0.2rem;
    padding:5px
}

footer{
    margin-top: 7%;
    color:rgb(255,255,255);
    padding:5px;
    text-align: center;
    background-color: rgb(0,115,230);
}
```

**WEEK-3 PRACTICE AT HOME QN-2**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Restaurant Menu</title>
    <link rel="stylesheet" href=".//styles.css">
</head>
<body>
    <!-- write your code here -->
    <section class="menu-section">
        <header>
            <h1>Delicious Restaurant</h1>
        </header>
        <nav>

            <a href="#home">Home</a>
            <a href="#menu">Menu</a>
            <a href="#about">About</a>
            <a href="#contact">Contact</a>

        </nav>

    </section>

    <section class="container">

        <h2 class="section-title">Starters</h2>
        <section class="menu-items">

            <div class="item">
                <h2>Garlic Bread</h2>
                <p>Crispy bread with garlic and herbs.</p>
                <h2 class="price">$5.99</h2>
            </div>
            <div class="item">

        
```

```
<h2>Tomato Soup</h2>
<p>Fresh tomato soup with basil.</p>
<h2 class="price">$4.99</h2>
</div>

</section>

<h2 class="section-title">Main Course</h2>
<section class="menu-items">

<div class="item">
    <h2>Grilled Chicken</h2>
    <p>Served with vegetables and fries.</p>
    <h2 class="price">$12.99</h2>
</div>
<div class="item">
    <h2>Veggie Paste</h2>
    <p>Pasta with fresh vegetables and herbs.</p>
    <h2 class="price">$10.99</h2>
</div>

</section>

<h2 class="section-title">Desserts</h2>
<section class="menu-items">

<div class="item">
    <h2>Chocolate Cake</h2>
    <p>Rich and moist chocolate cake.</p>
    <h2 class="price">$6.99</h2>
</div>
```

```
</div>
<div class="item">
    <h2>Ice Cream</h2>
    <p>Choice of vanilla, chocolate, or strawberry.</p>
    <h2 class=price>$3.99</h2>
</div>

</section>

</section>

<footer>
    <p>&copy 2024 Delicious Restaurant. All rights reserved.</p>
</footer>
</body>
</html>
```

```
header{
    background-color:
    rgb(195, 7, 7); padding:
    16px;
    margin-bottom: 0px;
}
header h1{
    text-align: center;
    color: rgb(255,255,255);

}
nav{
    display: flex;
    align-items:
    center; justify-
    content: center;
    margin-top: 0px;
    background-color: rgb(91, 3, 3);
    padding: 1%;

}
a{
    text-decoration:
    none;
    color:rgb(255,255
    ,255); margin-
    left:20px;

}
.container{
    display:f
    lex;
    flex-direction:
    column; align-
    items: center;
    justify-content:
    center;

}
.menu-
    items{
        display:
        flex;
```

```
align-items:  
center; justify-  
content:  
center; }
```

```
.item{  
    width:500px;  
    background-color:  
        rgb(255,255,255);    border:  
        2px solid    rgb(237, 230,  
        230);            border-radius:  
        0.75rem;  
    margin-right: 1%;  
    padding: 1%;  
  
}  
  
.section-title{  
    color:rgb(195,  
    7, 7);  
}  
  
footer{  
    margin-top: 1%;  
    background-color:  
        rgb(195, 7, 7); color:  
        rgb(255,255,255);  
    text-align:  
        center;  
    padding:  
        0.2%;  
}  
  
.price{  
    color:rgb(195, 7, 7);  
  
}
```

## WEEK 3 - CHALLENGE YOURSELF

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CSS Introduction Project</title>
    <link rel="stylesheet" href="./style.css">

</head>
<body>
    <!-- Write your HTML here -->
    <nav class="navbar">

        <a href="#introduction">Introduction</a>
        <a href="csssyntax">CSS Syntax</a>
        <a href="cssproperties">CSS Properties</a>

    </nav>

    <main>
        <section class="card">
            <h2 class="heading">Introduction to CSS</h2>
            <p>CSS stands for <span style="color: rgb(237, 185, 89);">Cascading Style Sheets. </span>
                It is used to style and layout web pages, CSS allows you
                to apply styles to <span style="font-weight: bold;">web
                pages.</span>
                More importantly, CSS enables you to do this independently
                of the HTML that makes up the web page.

            </p>
            <h3 class="subheading">CSS Syntax</h3>
            <p>The syntax of CSS consists of a selector
                and a declaration block.
            </p>

        <div class="code-block">
```

```
<p>selector { </p>
<p>&nbsp; &nbsp; &nbsp;property: value;</p>
<p>}</p>

</div>

<h3 class="subheading">CSS Properties</h3>
<p>
    There are numerous CSS properties that can be used to
    style elements. For example, the <span style="color:
rgb(247, 67, 67);">color</span> property defines
    the text color, and the <span style="color: rgb(247, 67,
67);">font-size</span> property
    defines the size of the text.
</p>

</section>
</main>

<footer class="footer">
    <p>&copy; 2024 CSS Introduction Project</p>
</footer>

</body>
</html>
```

```
body{
    background-color: rgb(52, 73, 94);
}

.navbar{
    background-color: rgb(44, 62, 80);
    display: flex;
    align-items: center;
    justify-content: center;
    padding: 1%;
    margin-bottom: 2%;
}

a{
    text-decoration: none;
    margin-left: 10%;
    color:rgb(255,255,255);
    padding: 0.5%;
}

.card{
    background-color: rgb(255,255,255);
    padding: 0.5%;
}

.heading{
    color:rgb(192,57,43)
}

.subheading{
    color: rgb(41,128,185);
}

.code-block{
    background-color: gainsboro;
    padding: 0.5%;
}

.footer{
    margin-top: 4%;
```

```
background-color: rgb(44, 62, 80);  
text-align: center;  
color:rgb(255,255,255);  
padding: 1%;  
}  
  
a:hover{  
background-color: rgb(157, 238, 37);  
color:black;  
}
```

## WEEK 4

### **WEEK-4 TYPE THE CODE**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Building Material Inventory Management System</title>
  <link rel="stylesheet" href="./style.css">
</head>
<body>

  <div class="formdiv">
    <h3>Building Material Inventory</h3>
    <form>

      <label for="materialName">Material Name:</label>
      <input type="text" id="materialName" placeholder="Enter material name" >
      <label for="category">Material Category:</label>
      <select id="category">
        <option value="Wood">Wood</option>
        <option value="Cement">Cement</option>
        <option value="Steel">Steel</option>
        <option value="Bricks">Bricks</option>
        <option value="Glass">Glass</option>
      </select>
      <label for="quantity">Quantity</label>
      <input id="quantity" type="number" placeholder="Enter quantity">
      <label for="supplierName">Supplier Name:</label>
      <input id="supplierName" placeholder="Enter supplier name" type="text">
      <label for="costPerUnit">Cost Per Unit(RS) :</label>
      <input id="costPerUnit" type="number" placeholder="Enter cost per unit">
      <button type="button" id="addMaterialBtn">Submit Material</button>
      <span id="error-message"></span>
    </form>

  </div>

<section>
  <table id="materialTable">
    <caption>Material List:</caption>
    <thead id="headerRow">
```

```
<tr>
<th>Material Name</th>
<th>Category</th>
<th>Quantity</th>
<th>Supplier</th>
<th>Cost Per Unit (RS)</th>

</tr>
</thead>
<tbody>

</tbody>

</table>

</section>

<div class="summary">
<h3>Inventory Summary:</h3>
<h3>Total Products: <span id="totalProducts">0</span></h3>
<h3>Most Expensive Product: <span id="mostExpensive">None</span></h3>

</div>

<script src=".//script.js"></script>
</body>
</html>
```

```
form{
display: flex; width:60%;

justify-content: center; flex-direction: column;
}

.formdiv{ display: flex;
align-items: center; justify-content: center; flex-direction: column;

}

input{
margin-bottom: 1%;
}

select{
margin-bottom: 1%;
}

#addMaterialBtn{
background-color: rgb(76,175,80); padding: 1%; color:rgb(255,255,255);
font-size: large; font-weight: bold;

}

thead{
background-color: rgb(76,175,80);
}

.summary{ display: flex;
align-items: center; justify-content: center; flex-direction: column;
```

```
}

#error-message{ opacity:0; color:red;
text-align: center; font-weight: bold;
}

.headerTxt{
text-align: center;
}
#materialTable{ margin-top: 2%;
margin-left: 20%; width:60%


}

#headerRow{
background-color: rgb(76,175,80); color:rgb(255,255,255);

}

#headerRow th{ text-align: center;
padding: 1%;
}

caption{
font-weight: bold; font-size: large;
}

.trow{
text-align: center;
}

label{
font-weight: bold; font-size: large;}
```

```
document.getElementById("addMaterialBtn").addEventListener("click", ()=>{

const name=document.getElementById("materialName").value.trim(); const
qty=document.getElementById("quantity").value.trim();
const sn=document.getElementById("supplierName").value.trim(); const
cpi=document.getElementById("costPerUnit").value.trim(); const
errMsg=document.getElementById("error-message");
const cat=document.getElementById("category"); const
catVal=cat.options[cat.selectedIndex].text;

if(!name || !catVal || !sn || !cpi || !qty){
errMsg.textContent="Please fill out all fields!";
errMsg.style.opacity=1;
return;
}

else{
errMsg.style.opacity=0;
const table=document.getElementById("materialTable");
const tbody=document.querySelector("#materialTable tbody"); const
trow=document.createElement("tr");
const tp=document.getElementById("totalProducts"); const
eIItem=document.getElementById("mostExpensive"); let
arr=[name,catVal,qty,sn,cpi];

for(let i=0;i<5;i++){
let tdata=document.createElement("td"); if(i==4){
let c=parseFloat(arr[i]).toFixed(2); tdata.textContent=`RS ${c}`;
}
else{
tdata.textContent=arr[i];
}

trow.appendChild(tdata);
}

trow.classList.add("trow"); tbody.appendChild(trow);

let n=table.rows.length-1;
```

```
tp.textContent=n;
const data=getTableData(tbody); eItem.textContent=`${data.name} (R ${data.cost})`;
document.getElementsByTagName("form")[0].reset();
}

const getTableData=(tbody)=>{ let mc=0;
let ei="";
for(let row of tbody.rows) {
const name=row.cells[0].textContent.trim(); const
txt=row.cells[4].textContent.trim();
const cost=parseFloat(txt.replace(/[^\\d.]/g,"")).toFixed(2);
if(cost>mc) {
mc=cost; ei=name;
}
return {name:ei,cost:mc}
}
```

## **WEEK 4-PRACTICE AT HOME-1**

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Habit Tracker</title>
<link rel="stylesheet" href=".style.css">

</head>
<body>
<div class="main-cont">
<form>
<h1>Habit Tracker</h1>
<div class="formdiv">
<input type="checkbox" id="chkWater">
<label for="chWater">Drink Water</label>
</div>

<div class="formdiv">
<input type="checkbox" id="chkExercise">
<label for="chkExercise">Exercise</label>
</div>

<div class="formdiv">
<input type="checkbox" id="chkReading">
<label for="chkReading">Read a book</label>
</div>

<div class="btnDiv">
<button id="mark-progress" type="button">Mark Progress</button>
<button id="reset-progress" type="button">Reset</button>
</div>

</form>

<p id="summary">You have completed <span id="count">0</span> habits today.</p>
</div>
<script src=".script.js"></script>
</body>
</html>
```

```
.main-cont{ width:50%;  
  
background-color: whitesmoke; border:2px solid white;  
border-radius: 1rem; margin-left: 30%;  
  
}  
form{  
display: flex; margin-left:36%;  
justify-content: center; flex-direction: column;  
  
}  
  
.btnDiv{  
display: flex;  
align-items: center;  
  
}  
  
#mark-progress{  
background-color: rgb(41,168,71); margin-right: 2%;  
}  
#reset-progress{ background-color: red;  
}  
  
button{  
margin-top: 2%; color:white; padding: 2%;  
}  
  
p{  
  
margin-left: 34%;  
}  
.formdiv{  
margin-bottom: 1%;}
```

```
document.getElementById("mark-
progress").addEventListener("click", ()=>{

const dw=document.getElementById("chkWater"); const
de=document.getElementById("chkExercise"); const
rb=document.getElementById("chkReading"); const
sumTxt=document.getElementById("count"); let n=0;

if(dw.checked) { n=n+1
}
if(de.checked) { n=n+1
}
if(rb.checked) { n=n+1
}

sumTxt.textContent=n;

})
```

```
document.getElementById("reset-
progress").addEventListener("click", ()=>{
document.getElementsByTagName("form")[0].reset();
const sumTxt=document.getElementById("count"); let n=0;
sumTxt.textContent=n;

})
```

## WEEK-4 PRACTICE AT HOME -2

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href=".//styles.css">
    <title>Multiplication Table Generator</title>
</head>
<body>
    <div class="mulClass">
        <h3>Multiplication Table Generator</h3>
        <label for="number">Enter a number:</label>
        <input id="number" type="number" placeholder="Enter a number">
        <label for="start">Start Value:</label>
        <input id="start" type="number" placeholder="Start of range">
        <label for="end">End Value:</label>
        <input id="end" type="number" placeholder="End of range">
        <button type="button" id="generateBtn">Generate Table</button>

        <span id="errormessage">All fields are required!</span>
        <div id="table">

        </div>
    </div>
    <script src=".//script.js"></script>

</body>
</html>
```

```
body{
    display: flex;
    align-items: center;
    justify-content: center;
}

.mulClass {
    display: flex;
    flex-direction: column;
    background-color: white;
    box-shadow: 2px 2px 2px rgb(224, 220, 220);
    border-radius: 1rem;
    width: 40%;
    padding: 2%;
    min-height: fit-content;
}

#generateBtn{
    background-color: rgb(77, 176, 80);
    color: rgb(255, 255, 255);
    padding: 1%;
}

input{
    margin-bottom: 2%;
    padding: 1%;
}

label{
    font-weight: bold;
}

#table{
    display: flex;
    flex-direction: column;
}
```

```
#errormessage{  
    color:red;  
    opacity:0;  
    text-align: center;  
}  
  
 
```

```
document.getElementById("generateBtn").addEventListener("click", ()=>{

    const maindiv=document.getElementsByClassName("mulClass")[0];
    const num=document.getElementById("number").value.trim();
    const start=document.getElementById("start").value.trim();
    const end=document.getElementById("end").value.trim();
    const errtxt=document.getElementById("errormessage");

    if(!num || !start || !end){
        errtxt.style.opacity=1;
        return;
    }

    else{
        errtxt.style.opacity=0;
        let i=parseInt(start);
        let j=parseInt(end);
        let n=parseInt(num);
        const tdiv=document.getElementById("table");
        tdiv.innerHTML="";
        const ptxt=document.createElement("p");
        ptxt.textContent='Multiplication Table of ${n}';
        ptxt.style.fontWeight="bold";
        tdiv.appendChild(ptxt);
        for(let k=i;k<=j;k++){
            let mele=document.createElement("p");
            let res=n*k;
            mele.textContent=`${n} x ${k} = ${res}`;
            tdiv.appendChild(mele);
        }
    }
})
```

## **WEEK 4-CHALLENGE YOURSELF**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <!-- Write your code -->
    <h1>Grade Calculator</h1>
    <div>

        <span id="marksLabel">Enter your marks (0-100): </span>
        <input id="marks" type="number">
        <button type="button" id="getGradeButton">Convert</button>
        <p id="gradeResult"></p>
    </div>
    <script src=".//script.js"></script>
</body>
</html>
```

```
document.getElementById("getGradeButton").addEventListener("click", ()=>{

    const mark=parseInt(document.getElementById("marks").value.trim());
    const result=document.getElementById("gradeResult");

    if(!mark) {
        result.textContent="Enter any marks to get the grades.";
        return;
    }

    else if(!(mark>=0 && mark<=100)) {
        result.textContent="The marks should be between 0-100.";
        return;
    }

    else{
        if(mark>=90) {
            result.textContent=`For marks ${mark}, your grade is: A`;

        }
        else if(mark>=80 && mark<=89) {
            result.textContent=`For marks ${mark}, your grade is: B`;
        }

        else if(mark>=70 && mark<=79) {
            result.textContent=`For marks ${mark}, your grade is: C`;
        }

        else if(mark>=60 && mark<=69) {
            result.textContent=`For marks ${mark}, your grade is: D`;
        }

        else{
            result.textContent=`For marks ${mark}, your grade is: F`;
        }
    }
})
```

## **WEEK-4 FIX THE CODE**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
  scale=1.0">
  <title>Music Album Collection Manager</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <header>
    <h3>Music Album Collection</h3>
  </header>

  <form>
    <label for="albumName">Album Name: </label>
    <input id="albumName" type="text" required placeholder="Enter album
      name">
    <br><br>

    <label for="releaseYear">Release Year: </label>
    <input id="releaseYear" type="number" required placeholder="Enter
      release year">
    <br><br>

    <label for="genre">Genre: </label>
    <select id="genre">
      <option value="">Select genre</option>
      <option value="Pop">Pop</option>
      <option value="Rock">Rock</option>
      <option value="Jazz">Jazz</option>
      <option value="Classical">Classical</option>
    </select>
    <br><br>

    <input type="checkbox" id="favorite">
    <label for="favorite">Mark as Favorite </label>
    <br><br>

    <button id="addAlbumBtn" type="button">Add Album</button>
    <span id="error-message"></span>
  </form>

  <h2>Album List:</h2>
```

```
<ul id="albumList"></ul>

<h2>Favorite Albums:</h2>
<ul id="favoriteAlbumList"></ul>

<script src="script.js"></script>
</body>
</html>
```

```
body{
    font: 18px Arial,
    sans-serif;
    background: lightblue;
    padding: 10px;
}
```

```
h1 {
    font-family: 'Times New
    Roman'; font-size:
    40px;
}
```

```
#albumList {
    /* color: blue; */
    background-color:
    lightgray; text-
    align: center;
    font-size: 14px;
}
```

```
ul {
    margin-left:
    10px; padding-
    left: 20px;
    list-style-type:
    square;
}
```

```
input[type="text"]
    { padding: 5px
    10px; width:
    180px;
}
```

```
button {
    background-color:
    rgb(76,175,80); color:
    red;
    padding: 10px
    20px; border:
    none;
    /* cursor: not-allowed; */
}
```

```
#error-
    message {
        color:
        #ff0000;
```

```
font-weight:  
normal; font-  
size: 14px;  
display:  
block;  
text-align: left;
```

```
}

#favoriteAlbumList li {
    /* color:
    green; */
    font-size:
    12px; font-
    weight:
    light;
}

input[type="numb
er"] { width:
120px;
padding: 8px;
margin-top:
5px;
background-color: yellow;
}
```

```

document.getElementById("addAlbumBtn").addEventListener("click", ()=>{
  const an=document.getElementById("albumName").value.trim();
  const ry=document.getElementById("releaseYear").value.trim();
  const gl=document.getElementById("genre");
  const genre=gl.options[gl.selectedIndex].text;
  const fav=document.getElementById("favorite");
  const em=document.getElementById("error-message");

  if(!an || !ry || !(gl.value.trim())){
    em.textContent="Please fill out all fields!";
    document.getElementsByTagName("form")[0].reset();
    return;
  }

  else if(fav.checked){
    em.textContent="";
    const favoriteAlbumList=document.getElementById("favoriteAlbumList");
    const liEle=document.createElement("li");
    // const hEle1=document.createElement("h2");
    // const hEle2=document.createElement("h2");

    liEle.innerHTML=`<strong>${an}</strong><br><em>Released: ${ry} | Genre: ${genre}
    `;
    // liEle.appendChild(hEle1);
    // liEle.appendChild(hEle2);
    favoriteAlbumList.appendChild(liEle);
    document.getElementsByTagName("form")[0].reset();
    return;
  }
  else {
    em.textContent="";
    em.style.opacity=0;
    const albumList=document.getElementById("albumList");
    const liEle=document.createElement("li");
    // const hEle1=document.createElement("h2");
    // const hEle2=document.createElement("h2");
    // hEle1.innerHTML=an;
    // hEle2.innerHTML=`<em>Released: </em> ${ry} | <em>Genre: </em> ${genre}`;
    liEle.innerHTML=`<strong>${an}</strong><br><em>Released: ${ry} | Genre: ${genre}
    `;
  }
}

```

```
</em> `;  
    // liEle.appendChild(hEle1);  
    // liEle.appendChild(hEle2);  
    albumList.appendChild(liEle);  
    document.getElementsByTagName("form")[0].reset  
(); return;  
  
}  
}
```

## Week 5

### Type the code

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Highlight Paragraph</title>
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
<style>
    .highlight {
        background-color: rgb(255, 255, 0); /* exact yellow for test match */
    }
</style>
</head>
<body class="p-4">

    <p id="textPara">This is a sample paragraph. Click the button below to highlight or remove the highlight.</p>
    <button id="highlightBtn" class="btn btn-primary">Highlight</button>

    <script>
        document.addEventListener("DOMContentLoaded", function () {
            const paragraph = document.getElementById("textPara");
            const button = document.getElementById("highlightBtn");

            // Toggle background color on click
            button.addEventListener("click", function () {
                paragraph.classList.toggle("highlight");
            });
        });
    </script>
</body>
</html>
```

### Practice at home-1

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

```
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Layered Image Display</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet"/>
/>
<style>
.image-container {
  position: relative;
  width: 300px;
  height: 200px;
  margin-bottom: 20px;
}

.layered-img {
  position: absolute;
  width: 100%;
  height: 100%;
  object-fit: cover;
}
</style>
</head>
<body class="p-4">

<div class="image-container">
  
  
  
</div>

<div class="btn-group">
  <button id="btn1" class="btn btn-primary">Image 1</button>
  <button id="btn2" class="btn btn-success">Image 2</button>
  <button id="btn3" class="btn btn-danger">Image 3</button>
</div>

<script>
document.addEventListener("DOMContentLoaded", () => {
  const buttons = ["btn1", "btn2", "btn3"];
  const images = ["img1", "img2", "img3"];

  document.getElementById("img1").style.zIndex = "1";

```

```

document.getElementById("img2").style.zIndex = "2";
document.getElementById("img3").style.zIndex = "3";

buttons.forEach((btnId, index) => {
  document.getElementById(btnId).addEventListener("click", () => {
    images.forEach((imgId, imgIndex) => {
      document.getElementById(imgId).style.zIndex =
        imgIndex === index ? "10" : "1";
    });
  });
});
});

</script>

```

```

</body>
</html>

```

## **Practice at home -2**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Font Style Toggles</title>
  <link
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
    rel="stylesheet"
  />
  <style>
    #editor {
      width: 100%;
      height: 200px;
      font-size: 12px;
      color: black;
      font-weight: normal;
      font-style: normal;
    }
  </style>
</head>
<body class="p-4">

<div class="mb-3 d-flex align-items-center gap-2">

```

```
<select id="fontSizeSelect" class="form-select w-auto">
  <option value="12px">12px</option>
  <option value="14px">14px</option>
  <option value="16px">16px</option>
  <option value="18px">18px</option>
  <option value="20px">20px</option>
</select>

<select id="fontColorSelect" class="form-select w-auto">
  <option value="black">Black</option>
  <option value="red">Red</option>
  <option value="blue">Blue</option>
  <option value="green">Green</option>
</select>

<div class="btn-group">
  <button id="boldBtn" class="btn btn-outline-dark">Bold</button>
  <button id="italicBtn" class="btn btn-outline-dark">Italic</button>
  <button id="resetBtn" class="btn btn-outline-secondary">Reset</button>
</div>
</div>

<textarea id="editor" class="form-control"></textarea>

<script>
  document.addEventListener("DOMContentLoaded", () => {
    const fontSizeSelect = document.getElementById("fontSizeSelect");
    const fontColorSelect = document.getElementById("fontColorSelect");
    const boldBtn = document.getElementById("boldBtn");
    const italicBtn = document.getElementById("italicBtn");
    const resetBtn = document.getElementById("resetBtn");
    const editor = document.getElementById("editor");

    // Set default values
    fontSizeSelect.value = "12px";
    fontColorSelect.value = "black";

    // Change font size
    fontSizeSelect.addEventListener("change", () => {
      editor.style.fontSize = fontSizeSelect.value;
    });

    // Change font color
    fontColorSelect.addEventListener("change", () => {
```

```
editor.style.color = fontColorSelect.value;
});

// Toggle bold
boldBtn.addEventListener("click", () => {
boldBtn.classList.toggle("active");
editor.style.fontWeight = boldBtn.classList.contains("active") ? "bold" : "normal";
});

// Toggle italic
italicBtn.addEventListener("click", () => {
italicBtn.classList.toggle("active");
editor.style.fontStyle = italicBtn.classList.contains("active") ? "italic" : "normal";
});

// Reset styles
resetBtn.addEventListener("click", () => {
fontSizeSelect.value = "12px";
fontColorSelect.value = "black";
boldBtn.classList.remove("active");
italicBtn.classList.remove("active");
editor.style.fontSize = "12px";
editor.style.color = "black";
editor.style.fontWeight = "normal";
editor.style.fontStyle = "normal";
});
});

```

</script>

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

## Week 6

### Fix the code

```
// /models/taskModels.js

const mongoose = require('mongoose');
const taskSchema=new mongoose.Schema({
  title:{type:String, required:[true,'Task title is required']},
  completed:{type:Boolean, default:false}
});
module.exports=mongoose.model('Task', taskSchema);
```

```
// /routers/taskRoutes.js

const express=require('express');
const router=express.Router();
const {
  createTask,
  getAllTasks,
  getTaskById,
} = require('../controllers/taskController');
router.post('/tasks',createTask);
router.get('/tasks',getAllTasks);
router.get('/tasks/:id', getTaskById);
module.exports=router;
```

```
// index.js

const express=require('express');
const mongoose=require('mongoose');
const taskRoutes=require('./routers/taskRoutes');
const app=express();
const PORT=8080;
app.use(express.json());
app.use('/',taskRoutes);
mongoose.connect('mongodb://127.0.0.1:27017/appdb', {
  useNewUrlParser:true,
  useUnifiedTopology:true,
})
.then(()=> {
  console.log('Connected to MongoDB');
```

```

    app.listen(PORT, ()=> {
      console.log(`Server is running on port ${PORT}`);
    });
})
.catch((err)=> {
  console.error('MongoDB connection error:',err);
});

```

## FIX THE CODE

```

// /controllers/itemController.js

const Item = require('../models/itemModel');
const createItem = async (req, res) => {
  const { name, quantity } = req.body;

  if (!name || quantity == null) {
    return res.status(400).json({ message: 'Name and quantity is required' });
  }
  try {
    const newItem = new Item({
      name,
      quantity
    });

    const savedItem = await newItem.save();
    res.status(201).json(savedItem);
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
};

const getAllItems = async (req, res) => {
  try {
    const items = await Item.find();
    res.status(200).json(items);
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
};

module.exports = {
  createItem,

```

```
    getAllItems
};

// /models/itemModel.js

const mongoose = require('mongoose');

const itemSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true
  },
  quantity: {
    type: Number,
    required: true
  }
});

const Item = mongoose.model('Item', itemSchema);

module.exports = Item;

// /routers/itemRouter.js

const express = require('express');
const { createItem, getAllItems } = require('../controllers/itemController');

const router = express.Router();

router.post('/', createItem);

router.get('/', getAllItems);

module.exports = router;

// index.js

const express = require('express');
const mongoose = require('mongoose');
const itemRoutes = require('./routes/itemRoutes');
```

```

const app = express();

app.use(express.json());

mongoose.connect('mongodb://127.0.0.1:27017/itemDB', {
  useNewUrlParser: true,
  useUnifiedTopology: true
})
.then(() => console.log('Connected to MongoDB'))
.catch(err => console.log(err));

app.use('/items', itemRoutes);

const port = 8080;
app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});

```

## **PRACTICE AT HOME 1**

```

// /controllers/productController.js

const Product=require('../models/productModel');
exports.createProduct=async(req,res,next) => {
  try{
    const {name,price,category,inStock}=req.body;
    const product=new Product({name,price,category,inStock});
    const savedProduct=await product.save();
    res.status(201).json(savedProduct);
  }catch(err) {
    next(err);
  }
};

exports.getAllProducts=async(req,res,next) => {
  try{
    const products=await Product.find();
    res.status(200).json(products);
  }catch(err){
    next(err);
  }
};

// /middlewares/errorHandler.js

```

```

const errorHandler=(err,req,res,next) => {
  console.error(err.stack);
  const statusCode=res.statusCode !== 200 ? res.statusCode : 500;
  res.status(statusCode).json({
    error:{
      message:err.message,
      stack:process.env.NODE_ENV==='production'? 'Stack trace hidden' : err.stack
    }
  });
};

module.exports=errorHandler;

// /models/productModel.js

const mongoose=require('mongoose');
const productSchema=new mongoose.Schema({
  name:{type:String, required:true},
  price:{type:Number, required:true},
  category:{type:String, required:true},
  inStock:{type:Boolean, default:true}
});
module.exports=mongoose.model('Product',productSchema);

// /routers/productRouter.js

const express=require('express');
const router=express.Router();
const ProductController=require('../controllers/productController');
router.post('/products',ProductController.createProduct);
router.get('/products',ProductController.getAllProducts);
module.exports=router;

// index.js

const express=require('express');
const mongoose=require('mongoose');
const productRoutes=require('./routers/productRoutes');
const errorHandler=require('./middlewares/errorHandler');
const app=express();
const PORT=8080;
app.use(express.json());
mongoose.connect('mongodb://127.0.0.1:27017/productsDB', {
  useNewUrlParser:true,
  useUnifiedTopology:true
}

```

```

})
.then(() => console.log('Connected to MongoDB'))
.catch((err) => console.error('MongoDB connection failed:',err));
app.use('/products',productRoutes);
app.use(errorHandler);
app.listen(PORT, () => {
  console.log(` Server is running on port ${PORT}`);
});

```

## **PRACTICE AT HOME 2**

```

// /controllers/movieController.js

const Movie=require("../models/movieModel");
exports.createMovie=async(req,res) => {
  try{
    const movie=new Movie(req.body);
    const savedMovie=await movie.save();
    res.status(201).json(savedMovie);
  }catch(error){
    res.status(500).json({message:error.message});
  }
};

exports.getAllMovies=async(req,res) => {
  try{
    const movies=await Movie.find();
    res.status(200).json(movies);
  }catch(error) {
    res.status(500).json({message:error.message});
  }
};

exports.getMovieById=async(req,res) => {
  try{
    const movie=await Movie.findById(req.params.id);
    if(!movie){
      return res.status(404).json({message:"Movie not found"});
    }
    res.status(200).json(movie);
  }catch(error) {
    res.status(500).json({message:error.message});
  }
};

exports.updateMovie=async(req,res) => {

```

```

try{
    const updatedMovie=await Movie.findByIdAndUpdate(req.params.id,req.body, {
        new:true,
        runValidators:true,
    });
    if(!updatedMovie){
        return res.status(404).json({message:"Movie not found"});
    }
    res.status(200).json(updatedMovie);
}catch(error){
    res.status(500).json({message:error.message});
}
};

exports.deleteMovie=async(req,res) => {
    try{
        const deletedMovie=await Movie.findByIdAndDelete(req.params.id);
        if(!deletedMovie){
            return res.status(404).json({message:"Movie not found"});
        }
        res.status(200).json({message:"Movie deleted successfully"});
    }catch(error){
        res.status(500).json({message:error.message});
    }
};

```

// /models/movieModel.js

```

const mongoose=require("mongoose");
const movieSchema=new mongoose.Schema({
    title: {
        type:String,
        required:[true, "Movie title is required"],
    },
    director: {
        type:String,
        required:[true,"Director is required"],
    },
    releaseYear: {
        type:Number,
        required:[true,"Release year is required"],
    },
    genre: {
        type:String,
        required:[true,"Genre is required"],
    }
});

```

```

    },
});

const Movie=mongoose.model("Movie",movieSchema);
module.exports=Movie;

// routers/movieRoutes.js

const express=require("express");
const router=express.Router();
const {
  getMovieById,
  updateMovieById,
  deleteMovieById,
} = require("../controllers/movieController");
router.get("/movies/:id",getMovieById);
router.put("/movies/:id",updateMovieById);
router.delete("/movies/:id",deleteMovieById);
module.exports=router;

//index.js

const express=require("express");
const mongoose=require("mongoose");
const movieRoutes=require("./routers/movieRoutes");
const app=express();
app.use(express.json());
mongoose
  .connect("mongodb://localhost:27017/movies",{
    useNewUrlParser:true,
    useUnifiedTopology:true,
  })
  .then(() => console.log("MongoDB connected"))
  .catch((err) => console.log(err));
app.use("/",movieRoutes);
app.use((err,req,res,next) => {
  console.error(err.stack);
  res.status(500).json({error:"Something went wrong!"});
});
const PORT=process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});

```

## **CHALLENGE YOURSELF**

```
// /controllers/customerController.js

const Customer=require('../models/customerModel');
exports.createCustomer=async(req,res,next) => {
    try{
        const {name,email,phone,isActive}=req.body;
        if(!name || !email || !phone){
            return res.status(404).json({message:'Name, email, and phone are required'});
        }
        const newCustomer=new Customer({name,email,phone,isActive});
        const savedCustomer=await newCustomer.save();
        res.status(201).json(savedCustomer);
    }catch(err){
        next(err);
    }
};

exports.getAllCustomers=async(req,res,next) => {
    try{
        const customers=await Customer.find();
        res.status(200).json(customers);
    }catch(err) {
        next(err);
    }
};

exports.updateCustomer=async(req,res,next) => {
    try{
        const {id}=req.params;
        const {name,email,phone,isActive}=req.body;
        if(!name || !email || !phone) {
            return res.status(404).json({message: 'Name, email, and phone are required to update
the customer'});
        }
        const updated=await Customer.findByIdAndUpdate(
            id,
            {name,email,phone,isActive},
            {new:true}
        );
        if(!updated){
            return res.status(404).json({message:'Customer not found'});
        }
        res.status(200).json(updated);
    }catch(err){

```

```

        next(err);
    }
};

// /middlewares/errorHandler.js

const errorHandler=(err,req,res,next)=> {
    console.error(err.stack);
    const status=res.statusCode !== 200 ? res.statusCode : 500;
    res.status(status).json({
        message:err.message,
        ...(process.env.NODE_ENV === 'development' && {stack:err.stack})
    });
};

module.exports=errorHandler;

// /models/customerModel.js

const mongoose=require('mongoose');
const customerSchema=new mongoose.Schema({
    name:{type:String,required:true},
    email:{type:String,required:true},
    phone:{type:String,required:true},
    isActive:{type:Boolean,default:true}
});
module.exports=mongoose.model('Customer',customerSchema);

// /routers/customerRoutes.js

const express=require('express');
const router=express.Router();
const customerController=require('../controllers/customerController');
router.post('/customers',customerController.createCustomer);
router.get('/customers',customerController.getAllCustomers);
router.put('/customers/:id',customerController.updateCustomer);
module.exports=router;

// index.js

const express=require('express');
const mongoose=require('mongoose');
const customerRoutes=require('./routers/customerRoutes');
const errorHandler=require('./middlewares/errorHandler');
const app=express();

```

```
const PORT=8080;
app.use(express.json());
mongoose.connect('mongodb://127.0.0.1:27017/customersDB', {
  useNewUrlParser:true,
  useUnifiedTopology:true
})
.then(() => console.log('MongoDB connected'))
.catch(err => console.error('MongoDB connection error:',err));
app.use('/customers',customerRoutes);
app.use(errorHandler);
app.listen(PORT, () => {
  console.log(` Server running on http://localhost:${PORT}`);
});
```

# WEEK 7

## TYPE THE CODE

```
// src/components/MedicalShop.jsx

import React, { useState } from 'react';
const MedicalShop = () => {
  const [inventory, setInventory] = useState([
    { name: 'Paracetamol', quantity: 50, price: 0.5 },
    { name: 'Aspirin', quantity: 30, price: 0.8 },
    { name: 'Band-Aid', quantity: 100, price: 0.2 },
  ]);
  const [formData, setFormData] = useState({
    name: '',
    quantity: '',
    price: '',
  });
  const [search, setSearch] = useState('');
  const handleChange = (e) => {
    const { placeholder, value } = e.target;
    setFormData((prev) => ({
      ...prev,
      [placeholder.toLowerCase()]: value,
    }));
  };
  const handleAddMedicine = () => {
    const { name, quantity, price } = formData;
    if (!name || !quantity || !price) {
      alert('Please fill in all fields');
      return;
    }
    const newMedicine = { name, quantity, price };
    setInventory((prev) => [...prev, newMedicine]);
    setFormData({ name: '', quantity: '', price: '' }); // reset form
  };
  const filteredInventory = inventory.filter((item) =>
    item.name.toLowerCase().includes(search.toLowerCase())
  );
  return (
    <div className="container my-4">
      <h2 className="text-center mb-4">Medical Shop Inventory</h2>
      {/* Search */}
      <input
        type="text"
        placeholder="Search Medicine"
        className="form-control mb-3"
        value={search}
        onChange={(e) => setSearch(e.target.value)}
      />
      {/* Form */}
      <div className="row mb-3">
        <div className="col-md-4">
          <input
            type="text"
            placeholder="Name"
            className="form-control mb-2"
            value={formData.name}
            onChange={handleChange}
          />
        </div>
        <div className="col-md-4">
          <input
            type="number"
            placeholder="Quantity"
            className="form-control mb-2"
          />
        </div>
      </div>
    </div>
  );
}
```

```

        value={formData.quantity}
        onChange={handleChange}
    />
</div>
<div className="col-md-4">
    <input
        type="number"
        placeholder="Price"
        step="0.01"
        className="form-control mb-2"
        value={formData.price}
        onChange={handleChange}
    />
</div>
</div>
<button
    className="btn btn-success w-100 mb-4"
    onClick={handleAddMedicine}
>
    Add Medicine
</button>
{ /* Table */}
<table className="table table-striped table-bordered">
    <thead className="table-dark">
        <tr>
            <th>Name</th>
            <th>Quantity</th>
            <th>Price</th>
        </tr>
    </thead>
    <tbody>
        {filteredInventory.length > 0 ? (
            filteredInventory.map((item, index) => (
                <tr key={index}>
                    <td>{item.name}</td>
                    <td>{item.quantity}</td>
                    <td>{item.price}</td>
                </tr>
            )))
        ) : (
            <tr>
                <td colSpan="3" className="text-center">
                    No medicines found
                </td>
            </tr>
        )}
    </tbody>
</table>
</div>
);
};

export default MedicalShop;

// src/App.js

import React from 'react';
import MedicalShop from './components/MedicalShop';
import './App.css';
function App() {
    return (
        <div>
            <MedicalShop />
        </div>
    );
}

```

```
}

export default App;
```

#### PRACTICE AT HOME

```
// src/components/Bookstore.jsx

import React, { useState } from 'react';
const initialInventory = [
  { id: 1, title: 'The Great Gatsby', author: 'F. Scott Fitzgerald', stock: 12, price: 10.5 },
  { id: 2, title: '1984', author: 'George Orwell', stock: 8, price: 9.0 },
  { id: 3, title: 'To Kill a Mockingbird', author: 'Harper Lee', stock: 15, price: 12.0 },
];
export default function Bookstore() {
  const [inventory, setInventory] = useState(initialInventory);
  const [newBook, setNewBook] = useState({
    title: '',
    author: '',
    stock: '',
    price: '',
  });
  const handleChange = (e) => {
    const { name, value } = e.target;
    setNewBook((prev) => ({ ...prev, [name]: value }));
  };
  const handleAddBook = () => {
    const { title, author, stock, price } = newBook;
    if (
      title.trim() === '' ||
      author.trim() === '' ||
      stock === '' ||
      price === ''
    ) {
      alert('Please fill in all fields');
      return;
    }
    const nextId =
      inventory.length === 0
        ? 1
        : Math.max(...inventory.map((b) => b.id)) + 1;
    const book = {
      id: nextId,
      title: title.trim(),
      author: author.trim(),
      stock: Number(stock),
      price: Number(price),
    };
    setInventory((prev) => [...prev, book]);
    // clear inputs
    setNewBook({
      title: '',
      author: '',
      stock: '',
      price: '',
    });
  };
  const formatPrice = (n) =>
    `$$ {Number(n).toFixed(2)} `;
  return (
    <div className="container my-5">
      <h1 className="fw-bold text-primary text-center">Bookstore Inventory</h1>
```

```

<p className="text-muted text-center">
    Manage your book collection with ease
</p>
<h2 className="mt-4 mb-3">Available Books</h2>
<div className="row g-3">
    {inventory.map((book) => (
        <div className="col-md-4" key={book.id}>
            <div className="card h-100 shadow-sm">
                <div className="card-body">
                    <h5 className="card-title text-success">{book.title}</h5>
                    <p className="card-subtitle mb-2 text-muted">by
{book.author}</p>
                    <p className="mb-1">
                        <span className="fw-bold">Stock:</span> {book.stock}
                    </p>
                    <p className="mb-0">
                        <span className="fw-bold">Price:</span>
                    {formatPrice(book.price)}
                    </p>
                </div>
                <div className="card-footer bg-transparent border-0">
                    <span className="badge bg-info">ID: {book.id}</span>
                </div>
            </div>
        )))
    </div>
<h2 className="mt-5 mb-3">Add a New Book</h2>
<div className="card shadow-sm">
    <div className="card-body">
        <div className="row g-3">
            <div className="col-md-3">
                <input
                    className="form-control"
                    type="text"
                    name="title"
                    placeholder="Book Title"
                    value={newBook.title}
                    onChange={handleChange}
                />
            </div>
            <div className="col-md-3">
                <input
                    className="form-control"
                    type="text"
                    name="author"
                    placeholder="Author"
                    value={newBook.author}
                    onChange={handleChange}
                />
            </div>
            <div className="col-md-2">
                <input
                    className="form-control"
                    type="number"
                    name="stock"
                    placeholder="Stock"
                    min="0"
                    value={newBook.stock}
                    onChange={handleChange}
                />
            </div>
            <div className="col-md-2">
                <input

```

```

        className="form-control"
        type="number"
        name="price"
        placeholder="Price ($)"
        step="0.01"
        min="0"
        value={newBook.price}
        onChange={handleChange}
      />
    </div>
    <div className="col-md-2 d-grid">
      <button
        className="btn btn-success w-100"
        onClick={handleAddBook}
      >
        Add Book
      </button>
    </div>
  </div>
</div>
</div>
);
}

// src/App.js

import React from 'react';
import Bookstore from './components/Bookstore';
export default function App() { return <Bookstore />; }

```

### CHALLENGE YOURSELF

```

// src/components/MusicStore.jsx

import React, { useState } from 'react';
const initialItems = [
  { id: 1, name: 'Acoustic Guitar', quantity: 10, price: 150 },
  { id: 2, name: 'Electric Keyboard', quantity: 5, price: 300 },
  { id: 3, name: 'Drum Set', quantity: 3, price: 500 },
];
export default function MusicStore() {
  const [inventory, setInventory] = useState(initialItems);
  const [newItem, setNewItem] = useState({ name: '', quantity: '', price: '' });
  const onChange = (e) => {
    const { name, value } = e.target;
    setNewItem((prev) => ({ ...prev, [name]: value }));
  };
  const addNewItem = () => {
    const { name, quantity, price } = newItem;
    if (name.trim() === '' || quantity === '' || price === '') {
      alert('Please fill in all fields');
      return;
    }
    const nextId =
      inventory.length === 0 ? 1 : Math.max(...inventory.map((i) => i.id)) + 1;
    setInventory((prev) => [
      ...prev,
      {
        id: nextId,
        name: name.trim(),
        quantity: Number(quantity),
        price: Number(price),
      }
    ]);
  };
}

```

```

        },
    ]);
    setNewItem({ name: '', quantity: '', price: '' });
}
const money = (n) => `$$ {Number(n).toFixed(2)} `;
return (
    <div className="container my-4">
        {/* Header bar styled to resemble screenshot */}
        <div className="card shadow-sm mb-3">
            <div className="card-body bg-dark text-warning text-center">
                <h1 className="m-0">Music Store Inventory</h1>
            </div>
        </div>
        <div className="card h-100 shadow-sm">
            <div className="card-body">
                <div className="table-responsive">
                    <table className="table table-bordered table-hover shadow-sm m-0">
                        <thead className="table-warning">
                            <tr>
                                <th scope="col">ID</th>
                                <th scope="col">Instrument / Album</th>
                                <th scope="col">Quantity</th>
                                <th scope="col">Price ($)</th>
                            </tr>
                        </thead>
                        <tbody>
                            {inventory.map((item) => (
                                <tr key={item.id}>
                                    <td>
                                        {item.id}
                                    </td>
                                    <td>{item.name}</td>
                                    <td>{item.quantity}</td>
                                    <td>{money(item.price)}</td>
                                </tr>
                            )))
                        </tbody>
                    </table>
                </div>
                {/* Add form */}
                <h2 className="text-info text-center mt-4 mb-3">Add New
Instrument</h2>
                <div className="row g-2 align-items-stretch">
                    <div className="col-md-5">
                        <input
                            className="form-control border-info"
                            type="text"
                            name="name"
                            placeholder="Instrument / Album Name"
                            value={newItem.name}
                            onChange={onChange}
                        />
                    </div>
                    <div className="col-md-2">
                        <input
                            className="form-control border-info"
                            type="number"
                            name="quantity"
                            placeholder="Quantity"
                            min="0"
                            value={newItem.quantity}
                            onChange={onChange}
                        />
                    </div>
                </div>
            </div>
        </div>
    </div>
)

```

```

        <div className="col-md-3">
          <input
            className="form-control border-info"
            type="number"
            name="price"
            placeholder="Price ($)"
            step="0.01"
            min="0"
            value={newItem.price}
            onChange={onChange}
          />
        </div>
        <div className="col-md-2 d-grid">
          <button className="btn btn-warning w-100 fw-bold"
            onClick={addItem}>
            Add Item
          </button>
        </div>
      </div>
    </div>
  </div>
);
}

// src/App.js

import React from 'react';
import MusicStore from './components/MusicStore';
export default function App() {
  return <MusicStore />;
}

```

#### FIX THE CODE

```

// src/components/FoodOrder.jsx

import React, { useState } from 'react';
import 'bootstrap/dist/css/bootstrap.min.css';
const initialOrders = [
  { id: 1, item: 'Burger', quantity: 2, price: 150 },
  { id: 2, item: 'Pizza', quantity: 1, price: 300 },
  { id: 3, item: 'Pasta', quantity: 3, price: 200 },
];
export default function FoodOrder() {
  const [orders, setOrders] = useState(initialOrders);
  const [newOrder, setNewOrder] = useState({ item: '', quantity: '', price: '' });
  const handleNewOrderChange = (e) => {
    const { name, value } = e.target;
    setNewOrder((prev) => ({ ...prev, [name]: value }));
  };
  const handleAddOrder = () => {
    const { item, quantity, price } = newOrder;
    if (item.trim() === '' || quantity === '' || price === '') {
      alert('Please fill in all fields');
      return;
    }
    const nextId = orders.length === 0 ? 1 : Math.max(...orders.map((o) => o.id)) + 1;
    const toAdd = {
      id: nextId,
      item: item.trim(),
    };
    setOrders((prev) => [...prev, toAdd]);
    setNewOrder({ item: '', quantity: '', price: '' });
  };
}

```

```

        quantity: Number(quantity),
        price: Number(price),
    };
    setOrders((prev) => [...prev, toAdd]);
    setNewOrder({ item: '', quantity: '', price: '' });
};

return (
    <div className="container my-4">
        <h1 className="text-primary text-center mb-4">Food Order Management</h1>
        <div className="table-responsive">
            <table className="table table-striped table-bordered" role="table">
                <thead>
                    <tr>
                        <th>ID</th>
                        <th>Food Item</th>
                        <th>Quantity</th>
                        <th>Price ($)</th>
                    </tr>
                </thead>
                <tbody>
                    {orders.map((o) => (
                        <tr key={o.id}>
                            <td>{o.id}</td>
                            <td>{o.item}</td>
                            <td>{o.quantity}</td>
                            <td>{o.price}</td>
                        </tr>
                    )))
                </tbody>
            </table>
        </div>
        <h2 className="text-success text-center mt-5 mb-3">Add New Food Order</h2>
        <div className="row g-3">
            <div className="col-md-5">
                <input
                    type="text"
                    className="form-control"
                    placeholder="Food Item"
                    name="item"
                    value={newOrder.item}
                    onChange={handleNewOrderChange}
                />
            </div>
            <div className="col-md-3">
                <input
                    type="number"
                    className="form-control"
                    placeholder="Quantity"
                    name="quantity"
                    value={newOrder.quantity}
                    onChange={handleNewOrderChange}
                    min="0"
                />
            </div>
            <div className="col-md-2">
                <input
                    type="number"
                    className="form-control"
                    placeholder="Price ($)"
                    name="price"
                    value={newOrder.price}
                    onChange={handleNewOrderChange}
                    step="0.01"
                    min="0"
                />
            </div>
        </div>
    </div>
)

```

```
        />
    </div>
    <div className="col-md-2 d-grid">
        <button className="btn btn-success w-100" onClick={handleAddOrder}>
            Add Order
        </button>
    </div>
</div>
</div>
);
}

// src/App.js

import React from 'react';
import FoodOrder from './components/FoodOrder';
export default function App() {
    return <FoodOrder />;
}
```

# WEEK 8

TYPE THE CODE

```
// src/App.js

import React from 'react';
class App extends React.Component {
  constructor(props) {
    super(props);
    const employees = [
      'John Doe',
      'Jane Smith',
      'Mike Johnson',
      'Emily Davis',
      'James Brown',
      'Jennifer Wilson',
      'Paul Garcia',
      'Laura Martinez',
    ];
    this.state = {
      query: '',
      employees,
      filteredEmployees: employees,
    };
  }
  componentDidUpdate(prevProps, prevState) {
    if (prevState.query !== this.state.query) {
      const q = this.state.query.toLowerCase();
      const filtered = this.state.employees.filter((n) =>
        n.toLowerCase().includes(q)
      );
      this.setState({ filteredEmployees: filtered });
    }
  }
  handleSearch = (e) => {
    this.setState({ query: e.target.value });
  };
  render() {
    return (
      <div className="container py-5">
        <h1 className="text-center mb-4">Employee Directory</h1>
        <div className="d-flex justify-content-center mb-3">
          <input
            type="text"
            className="form-control w-50"
            placeholder="Search employees..."
            value={this.state.query}
            onChange={this.handleSearch}
          />
        </div>
        <ul className="list-unstyled d-flex flex-column align-items-center gap-2">
          {this.state.filteredEmployees.map((name) => (
            <li
              key={name}
              className="w-50 text-center py-2 border rounded bg-light"
            >
              {name}
            </li>
          )));
        </ul>
      </div>
    );
  }
}
```

```
export default App;
```

FIX THE CODE

```
// src/components/Stopwatch.jsx

import React from 'react';
export default class Stopwatch extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      elapsedMs: 0,
      isRunning: false,
      laps: [],
    };
    this.intervalId = null;
  }
  componentWillUnmount() {
    if (this.intervalId) clearInterval(this.intervalId);
  }
  // mm:ss.cs where cs = centiseconds
  formatTime = (ms) => {
    const totalCs = Math.floor(ms / 10);
    const minutes = Math.floor(totalCs / 600);
    const seconds = Math.floor((totalCs % 600) / 100);
    const cs = totalCs % 100;
    const mm = String(minutes).padStart(2, '0');
    const ss = String(seconds).padStart(2, '0');
    const cc = String(cs).padStart(2, '0');
    return `${mm}:${ss}.${cc}`;
  };
  start = () => {
    if (this.state.isRunning) return;
    this.setState({ isRunning: true });
    const startAt = performance.now() - this.state.elapsedMs;
    this.intervalId = setInterval(() => {
      const now = performance.now();
      this.setState({ elapsedMs: now - startAt });
    }, 10);
  };
  stop = () => {
    if (!this.state.isRunning) return;
    this.setState({ isRunning: false });
    if (this.intervalId) {
      clearInterval(this.intervalId);
      this.intervalId = null;
    }
  };
  reset = () => {
    this.stop();
    this.setState({ elapsedMs: 0, laps: [] });
  };
  addLap = () => {
    if (!this.state.isRunning) return;
    this.setState((prev) => ({ laps: [...prev.laps, prev.elapsedMs] }));
  };
  render() {
    const { elapsedMs, isRunning, laps } = this.state;
    return (
      <div className="d-flex justify-content-center align-items-start py-5">
        <div className="border rounded p-4 text-center" style={{ width: 320 }}>
          <h3 className="mb-3">Stopwatch</h3>
          <div className="display-6 mb-3" aria-label="time">
```

```

        {this.formatTime(elapsedMs)}
    </div>
    <div className="d-flex gap-2 justify-content-center mb-3">
        <button
            className="btn btn-success"
            onClick={this.start}
            disabled={isRunning}
        >
            Start
        </button>
        <button
            className="btn btn-danger"
            onClick={this.stop}
            disabled={!isRunning}
        >
            Stop
        </button>
        <button
            className="btn btn-info text-white"
            onClick={this.reset}
            disabled={isRunning}
        >
            Reset
        </button>
    </div>
    <button
        className="btn btn-warning fw-semibold mb-3"
        onClick={this.addLap}
        disabled={!isRunning}
    >
        Add Lap
    </button>
    <ul className="list-unstyled m-0">
        {laps.map((ms, idx) => (
            <li key={idx} className="text-muted small">
                {this.formatTime(ms)}
            </li>
        )))
    </ul>
</div>
</div>
);
}
}

// src/App.js

import React from 'react';
import Stopwatch from './components/Stopwatch';
export default function App() {
    return <Stopwatch />;
}

```

## CHALLENGE YOURSELF

```

// src/App.js

import React from 'react';
import CropDashboard from './Components/CropDashboard';
import './App.css';
const initialCrops = [
    { name: 'Tomato', type: 'Vegetables', status: 'Growing' },
    { name: 'Apple', type: 'Fruits', status: 'Planted' },

```

```

        { name: 'Wheat', type: 'Grains', status: 'Harvested' },
        { name: 'Carrot', type: 'Vegetables', status: 'Growing' },
        { name: 'Banana', type: 'Fruits', status: 'Growing' },
    ];
export default function App() {
    return <CropDashboard crops={initialCrops} />;
}
// src/Components/CropDashboard.jsx

import React, { useState, useMemo } from 'react';
import CropList from './CropList';
export default function CropDashboard({ crops = [] }) {
    const [query, setQuery] = useState('');
    const filtered = useMemo(() => {
        const q = query.trim().toLowerCase();
        if (!q) return crops;
        return crops.filter((c) => c.name.toLowerCase().includes(q));
    }, [query, crops]);
    return (
        <div className="container py-5 d-flex justify-content-center">
            <div className="card shadow-sm p-4" style={{ maxWidth: 720, width: '100%' }>
                <h2 className="fw-bold text-uppercase mb-2">Crop Management System</h2>
                <hr className="border-2 border-success opacity-100 mt-0" />
                <div className="mb-3">
                    <label className="form-label fw-semibold">Search by Crop Name</label>
                    <input
                        type="text"
                        className="form-control border-success"
                        placeholder="Search crops..."
                        value={query}
                        onChange={(e) => setQuery(e.target.value)}
                    />
                </div>
                <CropList crops={filtered} />
            </div>
        </div>
    );
}

// src/Components/CropList.jsx

import React from 'react';
export default function CropList({ crops = [] }) {
    if (crops.length === 0) {
        return <div className="text-muted">No crops found.</div>;
    }
    return (
        <ul className="list-unstyled m-0">
            {crops.map((c) => (
                <li key={c.name} className="d-flex justify-content-between py-2 border-bottom">
                    <div className="d-flex align-items-center">
                        <span className="text-success me-2" aria-hidden="true">{' '}</span>
                        <span className="fw-semibold">{c.name}</span>
                    </div>
                    <div className="text-muted small">
                        - Type: {c.type} - Status: {c.status}
                    </div>
                </li>
            )))
        </ul>
    );
}

```

## Practice the Code

```
import React, {useState, useEffect} from 'react';

function App() {
  const [l, setL] = useState(true);
  const [t, setT] = useState([
    {id:1, t:'Complete React Assignment', p:'High', c: false},
    {id:2, t:'Review Bootstrap Documentation', p:'Medium', c: true},
    {id:3, t:'Debug Lifecycle Methods', p:'High', c: false},
    {id:4, t:'Setup React Dev Tools', p:'Low', c: false}
  ]);
  const [f, setF] = useState('All');
  const [nt, setNT] = useState('');
  const [np, setNP] = useState('High');
  const [sm, setSM] = useState(false);

  useEffect(() => {
    const task = setTimeout(() => setL(false), 2000);
    return () => clearTimeout(task);
  }, []);

  if(l) return <div className='spinner-border'></div>;

  let fd = t;
  if(f!=='All') {
    if(f==='Completed') fd = t.filter(task => task.c);
    else fd = t.filter(task => task.p === f);
  }

  const addTask = () => {
    if(!nt) return;
    setT([...t,{id: Date.now(), t: nt, p: np, c: false}]);
    setNT('');
    setNP('High');
    setSM(false);
  };

  const deleteTask = id => setT(t.filter(task => task.id !== id));
  const toggleComplete = id => setT(t.map(task => (task.id === id ? {...task, c: !task.c}: t)));

  return (
    <div>
      <h1>Task Manager</h1>
      <nav className='navbar'>Navbar</nav>
      <div>
        {[['All', 'High', 'Medium', 'Low', 'Completed']].map(fr => (
          <button key={fr} onClick={() => setF(fr)}>{fr}</button>
        )));
      </div>
    </div>
  );
}
```

```

        <button onClick={() => setSM(true)}>Add New Task</button>
        {sm && (
          <div>
            <input placeholder="Enter task title..." value={nt} onChange={ev =>
setNT(ev.target.value)}/>
            <label>Priority
              <select value={np} onChange={ev => setNP(ev.target.value)}>
                <option>High</option>
                <option>Medium</option>
                <option>Low</option>
              </select>
            </label>
            <button onClick={addTask}>Add Task</button>
          </div>
        )}
      <ul>
        {fd.map(task => (
          <li key={task.id}>
            <input type="checkbox" aria-label={task.t} checked={task.c} onChange={() =>
toggleComplete(task.id)}/>
            {task.t}
            <span className={badge ${task.p==='High'? 'bg-primary': task.p==='Medium'? 'bg-warning': 'bg-secondary'}}>{task.p}</span>
            <button onClick={() => deleteTask(task.id)}>Delete</button>
          </li>
        )));
      </ul>
    </div>
  );
}

export default App;

```

Week 9 - Skill builder

**Welcome.jsx**

```
import React from 'react';
import './Welcome.css';
function Welcome({ name }) {
  return (
    <div className='welcome'>
      <h2>{name}</h2>
    </div>
  );
}
export default Welcome;
```

**Welcome.css**

```
.welcome h2 {
  margin: 5px 0;
  color: #333;
}
```

**App.js**

```
import './App.css';
import Welcome from './components/Welcome';
function App() {
  return (
    <div className="App">
      <h1>Neo Players</h1>
      <h3>Top Students List</h3>
      <Welcome name="Sara" />
      <Welcome name="Chahal" />
      <Welcome name="Phillips" />
    </div>
  );
}
export default App;
```

**Week 9 - Challenge yourself**

**LifecycleLogs.js**

```
import React from 'react';
const LifecycleLogs = ({ logs, CL }) => {
  return (
    <div style={{ marginTop: '30px' }}>
      <h2>Lifecycle Logs</h2>
      <button onClick={CL}>Clear Log</button>
      <div
        style={{
          border: '1px solid #ccc',
          marginTop: '10px',
          padding: '10px',
          minHeight: '100px',
        }}
      >
```

```

{logs.length === 0 ? (
<p>No logs yet</p>
) : (
logs.map((l, ind) => <p key={ind}>{l}</p>)
)
</div>
</div>
);
};

export default LifecycleLogs;

```

## **CityFilter.js**

```

import React, { useEffect } from 'react';
const CityFilter = ({ currentFilter, onFilterChange, addLog, renderCount }) => {
useEffect(() => {
addLog('CityFilter Mounted');
}, [addLog]);
return (
<div style={{ marginBottom: '20px' }}>
<h2>Filter by Condition</h2>
<select
role="combobox"
value={currentFilter}
onChange={(e) => onFilterChange(e.target.value)}
>
<option value="all">All Conditions</option>
<option value="Sunny">Sunny</option>
<option value="Cloudy">Cloudy</option>
<option value="Rainy">Rainy</option>
<option value="Snowy">Snowy</option>
</select>
<p>Render Count: {renderCount}</p>
</div>
);
};
};

export default CityFilter;

```

## **Weathercard.js**

```

import React, { useEffect } from 'react';
const WeatherCard = ({ city, onToggle, addLog }) => {
useEffect(() => {
addLog(`WeatherCard Mounted: ${city.name}`)
}, [addLog, city]);
return (
<div style={{
border: '1px solid #ddd',
marginBottom: '10px',
padding: '10px',
borderRadius: '8px',
backgroundColor: '#f9f9f9'
}}>
<h3>{city.name}</h3>
<p>{city.temp} °C</p>
<span style={{
background: '#ddd',
padding: '4px 8px',
}}

```

```

borderRadius: '6px'
})>
{city.condition}
</span>
<div style={{ marginTop: '10px' }}>
<button onClick={() => onToggle(city.id)}>
{city.active === true ? 'Remove City' : 'Add City'}
</button>
</div>
<div>
<p>{city.condition}</p>
<p>{city.condition}</p>
</div>
</div>
);
};

export default WeatherCard;

```

## App.js

```

import React, { useState, useEffect, useCallback } from 'react';
import CityFilter from './components/CityFilter';
import WeatherCard from './components/WeatherCard';
import LifecycleLogs from './components/LifecycleLogs';
function App() {
const [c, setC] = useState([
{ id: 1, name: 'New York', condition: 'Sunny', temp: 22, active: true },
{ id: 2, name: 'London', condition: 'Cloudy', temp: 15, active: true },
{ id: 3, name: 'Tokyo', condition: 'Rainy', temp: 18, active: true },
{ id: 4, name: 'Paris', condition: 'Snowy', temp: 20, active: true }
]);
const [l, setL] = useState([]);
const [f, setF] = useState('all');
const [rc, setRC] = useState(1);
const AL = useCallback((m) => {
setL((p) => [...p, `${new Date().toLocaleTimeString()} - ${m}`])
}, []);
const FC = (v) => {
setF(v);
AL(`Filter changed to: ${v}`);
};
const TC = (id) => {
setC((p) => p.map((ct) => ct.id === id ? { ...ct, active: !ct.active } : ct));
AL(`City toggled: ${id}`);
};
const CL = () => setL(['Logs cleared']);
useEffect(() => { AL('App Mounted'); }, [AL]);
useEffect(() => { setRC((cnt) => cnt + 1); }, [1]);
const fc = f === 'all' ? c.filter((ct) => ct.active) : c.filter((ct) => ct.active && ct.condition === f);
return (
<div style={{ padding: '20px', fontFamily: 'Arial' }}>
<h1>Weather Dashboard</h1>
<CityFilter currentFilter={f} onFilterChange={FC} addLog={AL} renderCount={rc} />
<h2>Cities</h2>
{
fc.length > 0 ? (
fc.map((ct) => (
<WeatherCard key={ct.id} city={ct} onToggle={TC} addLog={AL} />

```

```
) )  
) : (  
<p>No cities to display</p>  
)  
>  
<LifecycleLogs logs={l} CL={CL} />  
</div>  
) ;  
}  
export default App;
```