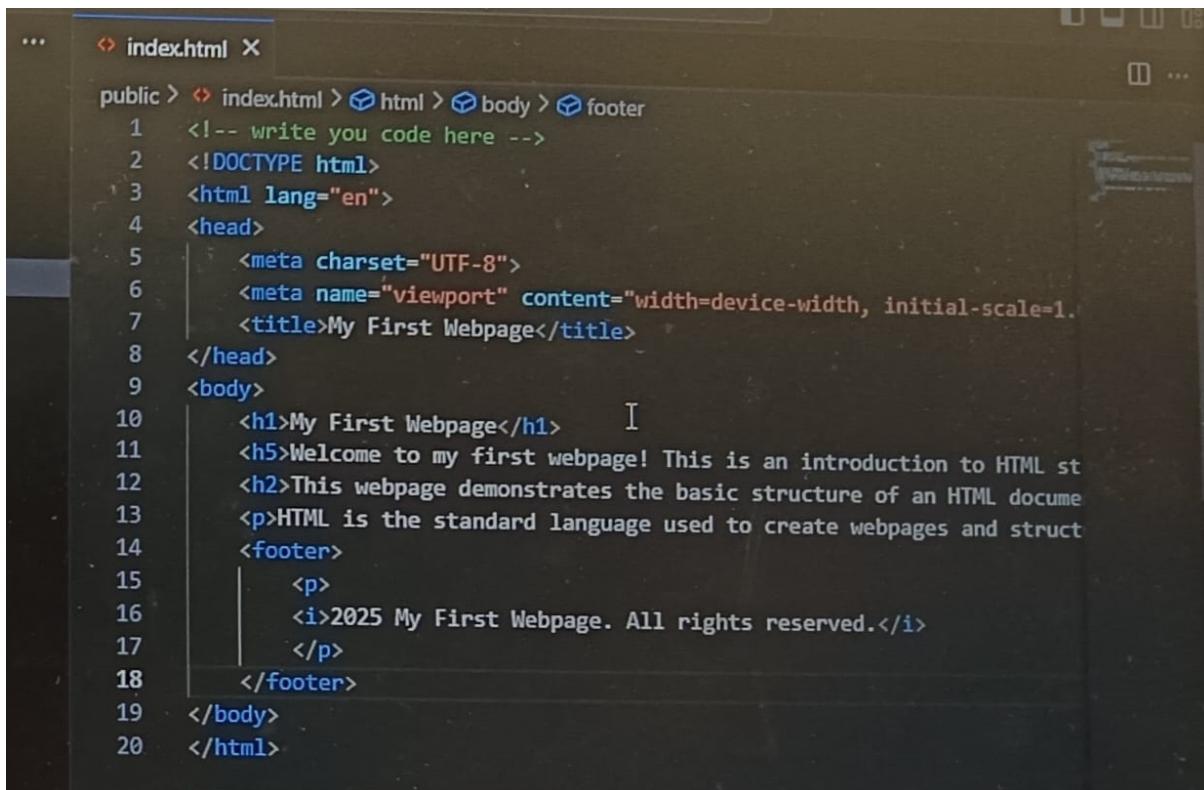


Web Programming

Week 1

Type the Code:



```
... ④ index.html X
public > ④ index.html > ⏺ html > ⏺ body > ⏺ footer
1   <!-- write your code here -->
2   <!DOCTYPE html>
3   <html lang="en">
4     <head>
5       <meta charset="UTF-8">
6       <meta name="viewport" content="width=device-width, initial-scale=1.>
7       <title>My First Webpage</title>
8     </head>
9     <body>
10    <h1>My First Webpage</h1>      I
11    <h2>Welcome to my first webpage! This is an introduction to HTML st
12    <h2>This webpage demonstrates the basic structure of an HTML docume
13    <p>HTML is the standard language used to create webpages and struct
14    <footer>
15      <p>
16        <i>2025 My First Webpage. All rights reserved.</i>
17      </p>
18    </footer>
19  </body>
20 </html>
```

Practice at home 1:

```
index.html x MongoDB
public > index.html > html > body
1  <!-- write your code here -->
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Tech Shop Online</title>
8  </head>
9  <body>
10     <h1>Tech Shop Online</h1>
11     <h2>Our Mission</h2>
12     <p>Our mission is to provide affordable, high-quality electronics to customers around the world.</p>
13     <h3>Contact Information</h3>
14     <p>We are committed to providing excellent customer service. Reach us at contact@techshop.com or call 1-800-123-4567.</p>
15     <footer>
16         <p>2025 Tech Shop Online. All rights reserved.</p>
17     </footer>
18     <h2>Why Choose Us?</h2>
19     <p>At Tech Shop Online, we offer the best deals on electronics, fast shipping, and excellent customer support. We make</p>
20     <p>We offer a range of premium laptops, including the latest models with high-speed processors and long-lasting batteries.</p>
21
22 </body>
23 </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>Tech Conference 2025</title>
</head>
<body>
    <header>
        <h1>Tech Conference 2025</h1>
        <h3><b><em><i>Innovating the Future of Technology</i></em></b></h3>
    </header>
    <h2>Conference Introduction</h2>
    <p>The <b>exciting</b> and <b>innovative</b> Tech Conference 2025 is a gathering of</p>
    <h3>Event Details</h3>
    <p>This year's conference will feature <strong>global speakers</strong> and</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>Innovations in Cybersecurity</b></p>
    <h2>Speakers</h2>
    <h5>Dr. John Doe</h5>
    <span><b>CEO, Tech Innovations Inc.</b></span>

```

```
<h2>Conference Introduction</h2>
<p>The <b>exciting</b> and <b>innovative</b> Tech Conference 2025 is a gathering of global experts from across the industry. Join us for two days of insightful sessions, networking opportunities, and exciting exhibits.
<h3>Event Details</h3>
<p>This year's conference will feature <strong>global speakers</strong> and <strong>interactive workshops</strong>. The event will take place at the <b>Tech Expo Center</b> from <b>Day 1</b> to <b>Day 2</b>.
<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>Innovations in Cybersecurity</b></p>
<h2>Speakers</h2>
<h5>Dr. John Doe</h5>
<span><b>CEO, Tech Innovations Inc.</b></span>
<h5>Jane Smith</h5>
<span><b>i>Lead Engineer, AI Solutions</i></b></span>
<footer>
    <h6>Contact Us</h6>
    <p>For more information, email us at <a href="mailto:contact@techconference.com">contact@techconference.com</a></p>
</footer>
</body>
</html>
```

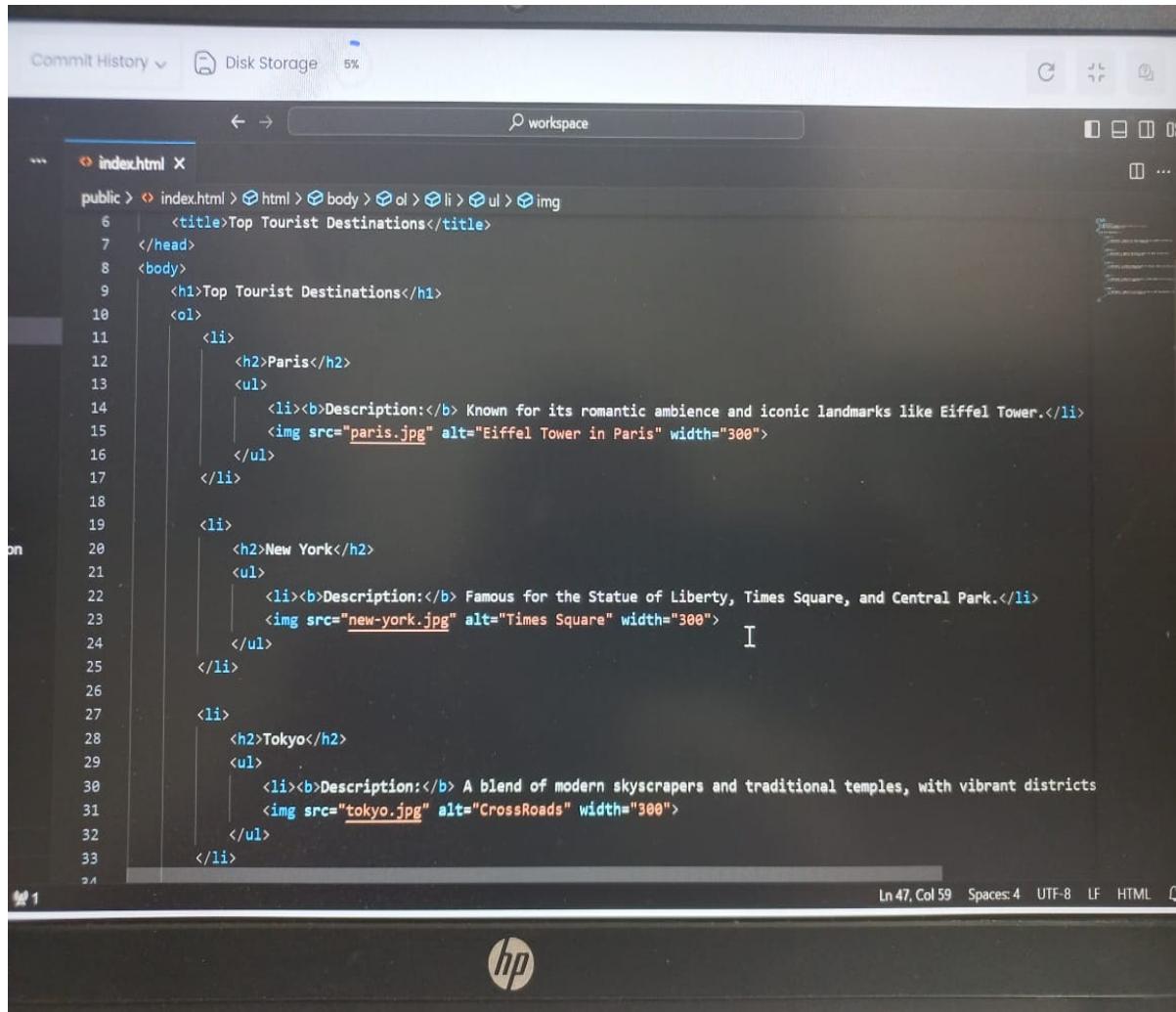
Challenge Yourself:

```
index.html X
public > index.html > html > body > h2
    -- 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>The Reading Nook - Your Favorite Bookstore</title>
        /head>
        body>
            header>
                h1>Welcome to The Reading Nook</h1>
                h3><b><em>Where every book has a story</b></em></h3>
            /header>
            h2>Our Mission</h2>
            p>At The Reading Nook, we are dedicated to providing <b>quality books</b> and <u><i>literature</i></u> to all readers.
            h3>New Arrivals</h3>
            p><i>The Silent Sea</i> - A must-read for all 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 recommend <span style="text-decoration: underline"><a href="#">Buy Now</a></span>.
            footer>
                h6>Contact Us</h6>
                p>Email: <a href="mailto:info@readingnook.com">info@readingnook.com</a> | Phone: <span><u>123-456-7890</u></span>
                h2>Why Choose Us?</h2>
                p>We offer a <strong>wide selection</strong> of books at <strong><em>affordable prices.</em></strong>.
            /footer>
        /body>
    /html>
```

Web Programming

Week 2

Type the code:



```
index.html
public > index.html > index.html > body > ol > li > ul > img
  6   <title>Top Tourist Destinations</title>
  7 </head>
  8 <body>
  9   <h1>Top Tourist Destinations</h1>
10   <ol>
11     <li>
12       <h2>Paris</h2>
13       <ul>
14         <li><b>Description:</b> Known for its romantic ambience and iconic landmarks like Eiffel Tower.</li>
15         
16       </ul>
17     </li>
18
19     <li>
20       <h2>New York</h2>
21       <ul>
22         <li><b>Description:</b> Famous for the Statue of Liberty, Times Square, and Central Park.</li>
23         
24       </ul>
25     </li>
26
27     <li>
28       <h2>Tokyo</h2>
29       <ul>
30         <li><b>Description:</b> A blend of modern skyscrapers and traditional temples, with vibrant districts
31         
32       </ul>
33     </li>
```

Commit History ▾ Disk Storage 5% workspace

index.html

```
public > index.html > html > body > ol > li > ul > img
27     <li>
28         <h2>Tokyo</h2>
29         <ul>
30             <li><b>Description:</b> A blend of modern skyscrapers and traditional temples, with vibrant districts
31                 
32             </ul>
33     </li>
34
35     <li>
36         <h2>Rome</h2>
37         <ul>
38             <li><b>Description:</b> Known for its ancient history, including landmarks like the Colosseum and Vat
39                 
40             </ul>
41     </li>
42
43     <li>
44         <h2>London</h2>
45         <ul>
46             <li><b>Description:</b> Famous for its rich history, the Tower of London, Buckingham Palace, and the
47                 
48             </ul>
49     </li>
50 </ol>
51 </body>
52 </html>
```

Line 47, Col 59 Spaces: 4 UTF-8 LF HTML

Fix the Code:

```
<!-- index.html > </html> </body> </header>
<!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.</p>
        <strong>Best stay ever!</strong>
    </section>

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

```
32
33    <section id="rooms">
34        <h2>Rooms and Pricing</h2>
35        <table border="1">
36            <thead>
37                <tr>
38                    <th>Room Type</th>
39                    <th>Price per Night</th>
40                    <th>Description</th>
41                </tr>
42            </thead>
43            <tbody>
44                <tr>
45                    <td>Single Room</td>
46                    <td>$100</td>
47                    <td>A comfortable room with a single bed and basic amenities.</td>
48                </tr>
49                <tr>
50                    <td>Double Room</td>
51                    <td>$150</td>
52                    <td>A spacious room with a king-size bed and a beautiful view.</td>
53                </tr>
54                <tr>
55                    <td>Suite</td>
56                    <td>$250</td>
57                    <td>A luxurious suite with a living area, jacuzzi, and premium amenities.</td>
58                </tr>
59                <tr>
60                    <td>Presidential Suite</td>
61                    <td>$500</td>
62                    <td>An extravagant suite with multiple rooms, a private balcony, and butler service.</td>
63                </tr>
64            </tbody>
65        </table>
66    </section>
67
```

```
index.html M test.js
public > index.html > html > body > footer
68     <section id="services">
69         <h2>Hotel Services</h2>
70         <ul>
71             <li>Free Wi-Fi</li>
72             <li>24/7 Room Service</li>
73             <li>Swimming Pool</li>
74             <li>Gym</li>
75             <li>Airport Shuttle</li>
76             <li>Exclusive Spa</li>
77         </ul>
78     </section>
79
80     <section id="gallery">
81         <h2>Gallery</h2>
82         
83         
84         
85     </section>
86
87     <aside>
88         <h3>Customer Reviews</h3>
89         <h4>John Doe</h4>
90         <p>"The stay was amazing. The service was exception, and the room was perfect!"</p>
91         <h4>Jane Smith</h4>
92         <p>"A wonderful experience. Highly recommend the suite for a luxury stay!"</p>
93         <h4>Mark Johnson</h4>
94         <p>"Great location, fantastic amenities, and friendly staff. Will definitely return!"</p>
95     </aside>
96
97     <footer>
98         <h2>Contact Us</h2>
99         <p>For booking inquiries, contact us at: info@luxurystays.com</p>
100        <p>&copy; 2025 Luxury Stays. All rights reserved.</p>
101    </footer>
102 </body>
103 </html>
```

Practice at home 1:

The screenshot shows a dual-pane interface. On the left is a code editor for an `index.html` file, and on the right is a MongoDB database browser.

Code Editor (index.html):

```
public > indexhtml > indexhtml > html > body > table > tbody > tr > td
1  <!-- write your code here -->
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Product Catalog</title>
8  </head>
9  <body>
10     <h1>Welcome to Our Product Catalog</h1>
11     <p>Welcome to our product catalog. Here you will find a selection of the latest and most popular products available in our store.</p>
12     <h2>Our Products</h2>
13     <ul>
14         <li>Electronics</li>
15         <li>Accessories</li>
16         <li>Home Appliances</li>
17     </ul>
18     <table border="1">
19         <thead>
20             <th>Product Name</th>
21             <th>Price</th>
22             <th>Description</th>
23         </thead>
24         <tbody>
25             <tr>
26                 <td>laptop</td>
27                 <td>$1200</td>
28                 <td>A high-performance laptop</td>
29             </tr>
30             <tr>
31                 <td>Smartphone</td>
32                 <td>$800</td>
33                 <td>A latest model smartphone</td>
34             </tr>
35             <tr>
36                 <td>Headphones</td>
```

MongoDB Browser:

- PORT: 5031, 8080
- Commit History
- Disk Storage
- workspace

A screenshot of a code editor window titled "index.html". The code is an HTML document with a table containing five rows of product information. The table has three columns: Product Name, Price, and Description.

```
public > index.html > index.html > body > table > tbody > tr > td
24   <tbody>
25     <tr>
26       <td>Laptop</td>
27       <td>$1200</td>
28       <td>A high-performance laptop</td>
29     </tr>
30   <tr>
31     <td>Smartphone</td>
32     <td>$800</td>
33     <td>A latest model smartphone</td>
34   </tr>
35   <tr>
36     <td>Headphones</td>
37     <td>$150</td>
38     <td>Noise-cancelling headphones</td>
39   </tr>
40   <tr>
41     <td>Smartwatch</td>
42     <td>$250</td>
43     <td>Fitness tracking smartwatch</td>
44   </tr>
45   <tr>
46     <td>Tablet</td>
47     <td>$500</td>
48     <td>A lightweight tablet with great features</td>
49   </tr>
50 </tbody>
51 </table>
52 </body>
53 </html>
```

Practice at home 2:

PORT: 8081, 8080 Commit History Disk Storage workspace

index.html × 25 test.js

```
public > index.html > HTML > body
1 <!-- write your code here -->
2 <!DOCTYPE html>
3 <html lang="en">
4   <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>E-commerce Product Catalog</title>
8   </head>
9   <body>
10    <header>
11      <h1>Welcome to Our E-commerce Store!</h1>
12      <p>We offer a wide variety of products to suit all your needs.</p>
13    </header>
14
15    <section>
16      <h2>Product Categories</h2>
17      <ul>
18        <li>Electronics</li>
19        <li>Clothing</li>
20        <li>Home Appliances</li>
21      </ul>
22    </section>
23
24    <section>
25      <h2>Product Catalog</h2>
26      <p>Browse through our catalog to find the latest products and deals.</p>
27      <table border="1">
28        <thead>
29          <tr>
30            <th>Product Name</th>
31            <th>Price</th>
32            <th>Description</th>
33            <th>Image</th>
34        </thead>
35        <tbody>
36          <tr>
37            <td>Laptop</td>
```

PORT: 8081, 8080 Commit History Disk Storage workspace

index.html x JS test.js

```
public > index.html > html > body
  34   <tbody>
  35     <tr>
  36       <td>laptop</td>
  37       <td>$1200</td>
  38       <td>A high-performance laptop</td>
  39       <td></td>
  40     </tr>
  41     <tr>
  42       <td>Smartphone</td>
  43       <td>$800</td>
  44       <td>Latest model smartphone</td>
  45       <td></td>
  46     </tr>
  47     <tr>
  48       <td>Headphones</td>
  49       <td>$150</td>
  50       <td>Noise-cancelling headphones</td>
  51       <td></td>
  52     </tr>
  53     <tr>
  54       <td>Smartwatch</td>
  55       <td>$250</td>
  56       <td>Fitness tracking smartwatch</td>
  57       <td></td>
  58     </tr>
  59     <tr>
  60       <td>Tablet</td>
  61       <td>$500</td>
  62       <td>Lightweight tablet with great features</td>
  63       <td></td>
  64     </tr>
  65   </tbody>
  66 </table>
  67 </section>
  68
  69 <section>
  70
  71 main @ 0△0 1
```

```
public > index.html > test.js
```

```
68
69     <section>
70         <h2>Best selling Products</h2>
71         <article>
72             <h3>Laptop</h3>
73             <p>a high-performance laptop perfect for work and entertainment.</p>
74             
75             <ul>
76                 <li>Fast processor</li>
77                 <li>Large screen</li>
78                 <li>Long battery life</li>
79             </ul>
80         </article>
81         <article>
82             <h3>Smartphone</h3>
83             <p>Latest model smartphone with advanced features.</p>
84             
85             <ul>
86                 <li>High-resolution camera</li>
87                 <li>Fast performance</li>
88                 <li>Long-lasting battery</li>
89             </ul>
90         </article>
91         <article>
92             <h3>Smartwatch</h3>
93             <p>Fitness tracking smartwatch for a healthy lifestyle.</p>
94             
95             <ul>
96                 <li>Tracks steps and heart rate</li>
97                 <li>Water-resistant</li>
98                 <li>Long battery life</li>
99             </ul>
100        </article>
101    </section>
102
103    <section>
```

```
PORT:8081,8080 Commit History Disk Storage workspace
```

```
public > index.html > test.js
```

```
88             <ul>
89                 <li>Long battery life</li>
90         </article>
91         <article>
92             <h3>Smartwatch</h3>
93             <p>Fitness tracking smartwatch for a healthy lifestyle.</p>
94             
95             <ul>
96                 <li>Tracks steps and heart rate</li>
97                 <li>Water-resistant</li>
98                 <li>Long battery life</li>
99             </ul>
100        </article>
101    </section>
102
103    <section>
104        <h2>Customer Testimonials</h2>
105        <h3>John Doe</h3>
106        <p>"I am thrilled with my new laptop from this store. The delivery was fast, and the product is amazing!"</p>
107        <h3>Jane Smith</h3>
108        <p>"The smartphone I bought exceeded my expectations. Great performance and value for money!"</p>
109        <h3>Mark Johnson</h3>
110        <p>"The smartwatch I purchased is exactly what I needed to stay fit. Highly recommend!"</p>
111    </section>
112
113
114 </body>
115 </html>
```

Challenge yourself:

```
index.html M X
public > index.html > html > body > section > table > tbody > tr > td
1   <!-- write your code here -->
2   <!DOCTYPE html>
3   <html lang="en">
4   <head>
5       <meta charset="UTF-8">
6       <meta name="viewport" content="width=device-width, initial-scale=1.0">
7       <title>Tech Innovators 2025 - Event Page</title>
8   </head>
9   <body>
10      <header>
11          <h1>Tech Innovators 2025</h1>
12          <h2>Innovating the Future of Technology</h2>
13          
14      </header>
15      <section>
16          <h2>About the Event</h2>
17          <p>Tech Innovators 2025 is a gathering of the brightest minds in technology to discuss the latest trends and
18      </section>
19      <section>
20          <h2>Event Schedule</h2>
21          <table style="border: 1px;">
22              <thead>
23                  <tr>
24                      <th>Time</th>
25                      <th>Session</th>
26                      <th>Speaker</th>
27                  </tr>
28              </thead>
29          <tbody>
30              <tr>
31                  <td>9.00 AM</td>
32                  <td>Keynote: The Future of AI</td>
33                  <td>Dr. John Doe</td>
34              </tr>
35              <tr>
36                  <td>11.00 AM</td>
37                  <td>Blockchain for Business</td>
38                  <td>Jane Smith</td>
39              </tr>
40              <tr>
41                  <td>1.00 PM</td>
42                  <td>Innovations in Cloud Computing</td>
43                  <td>Mark Johnson</td>
44              </tr>
45              <tr>
46                  <td>3.00 PM</td>
47                  <td>Tech Startups: How to Disrupt the Market</td>
48                  <td>Alice Brown</td>
49              </tr>
50              <tr>
51                  <td>5.00 PM</td>
52                  <td>Closing Remarks & Networking</td>
53                  <td>Event Pictures</td>
54              </tr>
55          </tbody>
56      </table>
```

```
index.html M X
public > index.html > html > body > section > table > tbody > tr > td
1   <tbody>
2       <tr>
3           <td>9.00 AM</td>
4           <td>Keynote: The Future of AI</td>
5           <td>Dr. John Doe</td>
6       </tr>
7       <tr>
8           <td>11.00 AM</td>
9           <td>Blockchain for Business</td>
10          <td>Jane Smith</td>
11      </tr>
12      <tr>
13          <td>1.00 PM</td>
14          <td>Innovations in Cloud Computing</td>
15          <td>Mark Johnson</td>
16      </tr>
17      <tr>
18          <td>3.00 PM</td>
19          <td>Tech Startups: How to Disrupt the Market</td>
20          <td>Alice Brown</td>
21      </tr>
22      <tr>
23          <td>5.00 PM</td>
24          <td>Closing Remarks & Networking</td>
25          <td>Event Pictures</td>
26      </tr>
27  </tbody>
```

index.html M X

```
public > <> index.html > html > body > section > table > tbody > tr > td
57     </section>
58     <section>
59         <h2>Speakers</h2>
60         <article>
61             <h3>Dr. John Doe</h3>
62             <p>Dr. John Doe is a renowned expert in AI with over 20 years of experience in the field. He will be discussi
63             
64         </article>
65         <article>
66             <h3>Jane Smith</h3>
67             <p>Jane Smith is a leading blockchain expert and a thought leader in decentralized technologies. She will
68             
69         </article>
70         <article>
71             <h3>Mark Johnson</h3>
72             <p>Mark Johnson is a cloud computing visionary and CEO of Cloud Tech Innovations. He will share has insig
73             
74         </article>
75     </section>
76     <section>
77         <h2>Venue</h2>
78         <p>The event will be held at the Grant Conference Center in San Fransisco, CA.</p>
79         
80     </section>
81     <aside>
82         <h3>Our Sponsors</h3>
83         <ul>
84             <li>TechCorp</li>
85         </ul>
86     </aside>
```

Ln 53, Col 39 Spaces: 4 UTF-8 LF HI

```
public > <> index.html > html > body > section > table > tbody > tr > td
76     <section>
77         <h2>Venue</h2>
78         <p>The event will be held at the Grant Conference Center in San Fransisco, CA.</p>
79         
80     </section>
81     <aside>
82         <h3>Our Sponsors</h3>
83         <ul>
84             <li>TechCorp</li>
85             <li>Innovative Solutions</li>
86             <li>Global Technologies</li>
87         </ul>
88     </aside>
89     <footer>
90         <h2>Contact Us</h2>
91         <p>For more information, email us at <u>info@techinnovators2025.com</u></p>
92         <p>Copyright, 2025 Tech Innovators 2025. All rights reserved.</p>
93     </footer>
94     <section>
95         <h2>Register Now</h2>
96         <ul>
97             <li>General: $100</li>
98             <li>VIP: $250</li>
99             <li>Student: $50</li>
100        </ul>
101    </section>
102 </body>
103 </html>
```

Ln 53, Col 39 Spaces: 4 UTF-8

Web Programming

Week 3

Type the Code:

```
index.html X
public > index.html ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <style>
7          /* Custom styles here */
8          .profile-card{
9              box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
10             border-radius: 10px;
11             background-color: #rgb(255, 255, 255);
12             width: 300px;
13             text-align: center;
14             margin: auto;
15         }
16         h2{
17             background-color: #rgb(76, 174, 79);
18             color:#white;
19         }
20         .button{
21             background-color: #rgb(76, 174, 79);
22             color:#white;
23         }
24         .button:hover{
25             background-color: #rgb(69, 170, 63);
26         }
27     </style>
28 </head>
29 <body>
30     <div class="profile-card">
31         <header>
32             <h2>John Doe</h2>
33         </header>
34         <p>Web Developer at XYZ Company. Loves coding and coffee.</p>
35         <button class="button">Contact</button>
36     </div>
37 </body>
38 </html>
39 |
```

Fix the Code:

```
public > index.html > html > body > div.login-form > form > span
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>SkillUp Login</title>
7      <link rel="stylesheet" href="styles.css">
8  </head>
9  <body>
10     <div class="login-form">
11         <h1>Welcome to SkillUp - Learn, Grow, Succeed!</h1>
12         <h2>LOGIN FORM</h2>
13         <form>
14             <username>
15                 Username
16             </username>
17             <input type="text" name="username" id="username">
18             <br>
19             <password>
20                 Password
21             </password>
22             <input type="password" name="password" id="password">
23             <br>
24             <button type="submit" id="submitButton">Submit</button>
25             <span>Error</span>
26         </form>
27     </div>
28     <script> src="script.js" </script>
29 </body>
30 </html>
31
```

```
public > # styles.css > input[type="text"]
1 .login-form {
2     background-color: □rgb(255, 255, 255);
3     border-radius: 5px;
4     margin-top: 50px;
5     padding: 30px;
6 }
7
8 h1{
9     color: ■rgba(255, 0, 0, 0.5);
10    font-size: 100px;
11    text-align: center;
12 }
13
14 input[type="text"], input[type="password"] {
15     width: 250px;
16     height: 50px;
17     font-size: 16px;
18     padding: 20px;
19     margin: 10px 0;
20     border: 1px solid;
21     background-color: transparent;
22 }
23
24 button[type="submit"] {
25     background-color: ■rgb(0,0,255);
26     padding: 15px 30px;
27     font-size: 16px;
28     cursor: non-resizable;
29 }
30
31 input[type="text"]:focus, input[type="password"]:focus {
32     outline: none;
33 }
34
35 label {
36     font-size: 18px;
37
38     > label {
39         font-size: 18px;
40         font-weight: bold;
41         display: block;
42         text-align: left;
43         color: ■red;
44     }
45     > span {
46         display: none;
47     }
48 }
```

Practice at home 1:

```
public > index.html > html > head > style > .contact-form button
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Simple OOP in JavaScript</title>
7      <style>
8          header{
9              color: white;
10             background-color: #rgb(0, 115, 230);
11             text-align: center;
12             padding: 16px;
13         }
14         nav,a{
15             color: #rgb(255, 255, 255);
16             text-transform: uppercase;
17             background-color: #rgb(0, 115, 230);
18             text-align: center;
19         }
20         .container{
21             max-width: 1200px;
22             padding-left: 500px;
23         }
24         h3{
25             padding-left: 120px;
26         }
27         .contact-form{
28             align-items: center;
29         }
30         .contact-form button{
31             background-color: #rgb(0,115,230);
32         }
33         footer{
34             background-color: #rgb(0, 115, 230);
35             color: white;
```

```
32 |         rooter{
33 |             background-color: #rgb(0, 115, 230);
34 |             color: white;
35 |             text-align: center;
36 |         }
37 |
38 |         input{
39 |             padding: 10px;
40 |             width: 340px;
41 |         }
42 |         textarea{
43 |             padding: 15px;
44 |             width: 330px;
45 |         }
46 |     
```

46 | </style>

47 | </head>

48 | <body>

49 | <!-- write your code here -->

50 | <header>

51 | <h1>Contact Us</h1>

52 | </header>

53 | <nav>

54 |

55 | <a>Home

56 | <a>Products

57 | <a>About

58 | <a>Contact

59 |

60 | </nav>

61 | <div class="container">

62 | <h3>Get in Touch</h3>

63 | <p>Feel free to reach out to us for any inquiries or feedback.</p>

64 | <form class="contact-form">

65 | <input type="text" placeholder="Your Name">

66 | <input type="text" placeholder="Your Email">

67 | <textarea type="text" placeholder="Your Message"></textarea>

68 | </form>

69 | </div>

70 | <nav>

71 | <div class="container">

72 | <h3>Get in Touch</h3>

73 | <p>Feel free to reach out to us for any inquiries or feedback.</p>

74 | <form class="contact-form">

75 | <input type="text" placeholder="Your Name">

76 | <input type="text" placeholder="Your Email">

77 | <textarea type="text" placeholder="Your Message"></textarea>

78 | <button class="contact-form button">Send Message</button>

79 | </form>

80 | </div>

81 | </nav>

82 | <footer>

83 | <p>© 2024 Online Store. All Rights Reserved.</p>

84 | </footer>

85 | </body>

86 | </html>

Practice at home 2:

```
public > index.html > html > head > style > .section-title h2
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Simple OOP in JavaScript</title>
7      <style>
8          h1{
9              color: white;
10             background-color: crimson;
11             text-align: center;
12             padding: 10px;
13             margin: 0;
14         }
15         nav{
16             background-color: darkred;
17             text-align: center;
18             padding: 10px;
19         }
20         nav,a{
21             color: white;
22             margin: 0;
23             font-weight: bold;
24         }
25         .container{
26             width: 80%;
27             margin: 20px auto;
28         }
29         .section-title h2{
30             color: crimson;
31             margin-top: 30px;
32             border-bottom: 2px solid #ddd;
33             padding-bottom: 5px;
34             text-align: center;
35         }
36         .item{
```

public > index.html > html > head > style > section-title h2

```
36     .item{
37         background-color: #f9f9f9;
38         padding: 15px;
39         margin: 10px 0;
40         border: 1px solid #eee;
41         border-radius: 5px;
42     }
43     .item h3{
44         margin: 0;
45         font-size: 18px;
46     }
47     .item p{
48         margin: 5px 0;
49         color: #555;
50     }
51     .price{
52         color: crimson;
53         font-size: 16px;
54         font-weight: bold;
55     }
56     footer{
57         background-color: crimson;
58         color: white;
59         text-align: center;
60         padding: 10px;
61     }
62     </style>
63 </head>
64 <body>
65     <!-- write your code here -->
66     <header>
67         <h1>Delicious Restaurant</h1>
68     </header>
69     <nav>
70         <a>Home</a>
71         <a>Menu</a>
```

```
public > index.html > html > head > style .section-title h2
70      <a>Home</a>
71      <a>Menu</a>
72      <a>About</a>
73      <a>Contact</a>
74  </nav>
75  <div class="container">
76    <div class="section-title">
77      <h2>Starters</h2>
78    </div>
79    <div class="item">
80      <h3>Garlic Bread</h3>
81      <p>Crispy Bread wtih garlic and herbs</p>
82      <div class="price">$5.99</div>
83    </div>
84    <div class="item">
85      <h3>Tomato Soup</h3>
86      <p>Fresh Tomato Soup with basil.</p>
87      <div class="price">$4.99</div>
88    </div>
89
90    <div class="section-title">
91      <h2>Main Course</h2>
92    </div>
93    <div class="item">
94      <h3>Grilled Chicken</h3>
95      <p>Served with vegetables and fries.</p>
96      <div class="price">$12.99</div>
97    </div>
98    <div class="item">
99      <h3>Veggie Pasta</h3>
100     <p>Pasta with fresh vegetables and herbs.</p>
101     <div class="price">$10.99</div>
102   </div>
103
```

```
103
104     <div class="section-title">
105         <h2>Desserts</h2>
106     </div>
107     <div class="item">
108         <h3>Chocolate Cake</h3>
109         <p>Rich and moist chocolate cake.</p>
110         <div class="price">$6.99</div>
111     </div>
112     <div class="item">
113         <h3>Ice Cream</h3>
114         <p>Choice of vanilla, chocolate, or strawberry.</p>
115         <div class="price">$3.99</div>
116     </div>
117 </div>
118 <footer>
119     <p>&copy; 2024 Delicious Restaurant. All rights reserved</p>
120 </footer>
121 </body>
122 </html>
123
```

Challenge Yourself:

```
public > index.html ...  
1  <!DOCTYPE html>  
2  <html lang="en">  
3  <head>  
4      <meta charset="UTF-8">  
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">  
6      <link rel="stylesheet" href="style.css">  
7  </head>  
8  <body>  
9      <!-- Write your HTML here -->  
10     <nav class="navbar">  
11         <a>Introduction</a>  
12         <a>CSS Syntax</a>  
13         <a>CSS Properties</a>  
14     </nav>  
15     <main>  
16         <section class="card">  
17             <h2 class="heading">Introduction to CSS</h2>  
18             <p>CSS stands for <span style="color: #gold;">Cascading Style Sheets.</span> It is used to style and layout web page. CSS allows you to apply styles to web pages. More importantly,  
19                 CSS enables you to do this independently of the HTML that makes up each web page.</p>  
20             <h3 class="subheading">CSS Syntax</h3>  
21             <p>The syntax of CSS consists of a selector and a declaration block:</p>  
22             <div class="code-block">  
23                 selector{  
24                     property: value;  
25                 }  
26             </div>  
27             <h3 class="subheading">CSS Properties</h3>  
28             <p>There are numerous CSS properties that can be used to style elements. For example, the <span style="color:#crimson">color</span> property defines the text color,  
29                 and the <span style="color:#crimson">font-size</span> properly defines the size of the text.</p>  
30         </section>  
31         <footer class="footer">  
32             <p>&copy;2024 CSS Introduction Project</p>  
33         </footer>  
34     </main>  
35 </body>  
36 </html>
```

```
public > # style.css >  .code-block
1  body{
2      background-color: ■rgb(52, 73, 94);
3  }
4  .navbar{
5      background-color: ■rgb(44,62,80);
6      color: □white;
7      text-align: center;
8      padding: 10px;
9      margin: 0;
10 }
11 .card{
12     background-color: □rgb(255,255,255);
13     width: 55%;
14     margin: auto;
15     padding: 5px 10px;
16     border-radius: 5px;
17 }
18 .code-block{
19     background-color: □#f9f9f9;
20     border-left: 2px solid □#ddd;
21 }
22 .footer{
23     background-color: ■rgb(44,62,80);
24     color: □white;
25     text-align: center;
26     padding: 10px;
27 }
28
29 .heading{
30     color: ■rgb(192,57,43);
31 }
32 .subheading{
33     color: ■rgb(41,128,185);
34 }
```

Web Programming

Week 4

Type the Code:

```
public > index.html > html > body > main.container > section.card > div.table-wrap
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Building Material Inventory Management System</title>
8      <style>
9          #addMaterialBtn{
10             background-color: #rgb(76, 175, 80);
11         }
12     </style>
13 </head>
14 <body>
15     <main class="container">
16         <section class="card">
17             <h3>Building Material Inventory</h3>
18             <form id="materialForm" novalidate>
19                 <div class="field">
20                     <label for="materialName">Material Name:</label>
21                     <input type="text" id="materialName" placeholder="Enter material name"/>
22                 </div>
23                 <div class="field">
24                     <label for="category">Material Category:</label>
25                     <select id="category">
26                         <option value="Wood">Wood</option>
27                         <option value="Cement">Cement</option>
28                         <option value="Steel">Steel</option>
29                         <option value="Bricks">Bricks</option>
30                         <option value="Glass">Glass</option>
31                     </select>
32                 </div>
33                 <div class="field">
34                     <label for="quantity">Quantity:</label>
35                     <input type="number" id="quantity" placeholder="Enter quantity"/>
36                 </div>
37                 <div class="field">
38                     <label for="supplierName">Supplier Name:</label>
39                     <input type="text" id="supplierName" placeholder="Enter supplier name"/>
40                 </div>
41                 <div class="field">
42                     <label for="costPerUnit">Cost Per Unit (RS):</label>
43                     <input type="number" id="costPerUnit" placeholder="Enter cost per unit" step="0.01"/>
44                 </div>
45             <button type="button" id="addMaterialBtn" style="background-color: #rgb(76,175,80);">Submit Material</button><br>
```

```
46     |     |     |     <span id="error-message" class="error"></span>
47     |     |     |   </form>
48     |     |     <h4 class="subheading">Material List:</h4>
49     |     |     <div class="table-wrap">
50     |     |     |     <table id="materialTable">
51     |     |     |     |     <thead>
52     |     |     |     |     |     <tr>
53     |     |     |     |     |     |     <th>Material Name</th>
54     |     |     |     |     |     |     <th>Category</th>
55     |     |     |     |     |     |     <th>Quantity</th>
56     |     |     |     |     |     |     <th>Supplier</th>
57     |     |     |     |     |     |     <th>Cost Per Unit (RS)</th>
58     |     |     |     |     |     </tr>
59     |     |     |     |     </thead>
60     |     |     |     <tbody>
61     |     |     |     |     </tbody>
62     |     |     |     </table>
63     |     |     </div>
64     |     |     <h4>Inventory Summary:</h4><br>
65     |     |     <p>Total Products: <span id="totalProducts">0</span></p><br>
66     |     |     <p>Most Expensive Product: <span id="mostExpensive">None</span></p>
67     |     |     <script src="script.js"></script>
68     |     </body>
69     </html>
```

```
1  const materialName=document.getElementById("materialName");
2  const categoryInput=document.getElementById("category");
3  const quantityInput=document.getElementById("quantity");
4  const supplierNameInput=document.getElementById("supplierName");
5  const costPerUnitInput=document.getElementById("costPerUnit");
6  const addMaterialBtn=document.getElementById("addMaterialBtn");
7  const errorMessage=document.getElementById("error-message");
8  const materialTableBody=document.getElementById("materialTable").querySelector("tbody");
9  const totalProductsSpan=document.getElementById("totalProducts");
10 const mostExpensiveSpan=document.getElementById("mostExpensive");
11
12 let materials=[];
13 addMaterialBtn.addEventListener("click",()=>{
14     const name=materialName.value.trim();
15     const category=categoryInput.value;
16     const quantity=quantityInput.value.trim();
17     const supplier=supplierNameInput.value.trim();
18     const cost=costPerUnitInput.value.trim();
19     if(!name||!category||!quantity||!supplier||!cost)
20     {
21         errorMessage.textContent="Please fill out all fields!";
22         return;
23     }
24     if(Number(quantity)<=0 || Number(cost)<=0)
25     {
26         errorMessage.textContent="Quantity and cost must be positive numbers!";
27         return;
28     }
29     errorMessage.textContent="";
30     const material={
31         name,
32         category,
33         quantity: Number(quantity),
34         supplier,
35         cost: Number(cost)
36     };
37     materials.push(material);
38     renderTable();
39     updateSummary();
```

```

38     renderTable();
39     updateSummary();
40     materialName.value="";
41     categoryInput.value="";
42     quantityInput.value="";
43     supplierNameInput.value="";
44     costPerUnitInput.value="";
45   });
46   function renderTable(){
47     materialTableBody.innerHTML="";
48     materials.forEach(mat=>{
49       const row=document.createElement("tr");
50       row.innerHTML=` ${mat.name}</td> 51         <td>${mat.category}</td> 52         <td>${mat.quantity}</td> 53         <td>${mat.supplier}</td> 54         <td>RS ${mat.cost.toFixed(2)}</td>`; 55       materialTableBody.appendChild(row); 56     }) 57   } 58 } 59 function updateSummary(){ 60   totalProductsSpan.textContent=materials.length; 61   if(materials.length>0) 62   { 63     let expensiveMaterial=materials.reduce((prev,current)=> 64       current.cost>prev.cost?current:prev 65     ); 66     mostExpensiveSpan.textContent=`${expensiveMaterial.name} (RS ${expensiveMaterial.cost.toFixed(2)})`; 67   } 68   else{ 69     mostExpensiveSpan.textContent="None"; 70   } 71 } |
```

Fix the Code:

```
public > index.html > html > body > script
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Music Album Collection Manager</title>
7      <link rel="stylesheet" href="style.css">
8  </head>
9  <body>
10     <header>
11         <h3>Music Album Collection</h3>
12     </header>
13     <form>
14         <label for="albumName">Album Name</label>
15         <input id="albumName" type="text" placeholder="Enter album name">
16         <br><br>
17         <label for="releaseYear">Release Year</label>
18         <input id="releaseYear" type="number" placeholder="Enter release year">
19         <br><br>
20         <label for="genre">Genre</label>
21         <select id="genre">
22             <option value="Pop">Pop</option>
23             <option value="Rock">Rock</option>
24             <option value="Jazz">Jazz</option>
25             <option value="Clasical">Clasical</option>
26         </select>
27         <br><br>
28         <label for="favorite">Mark as Favorite</label>
29         <input type="checkbox" id="favorite">
30         <br><br>
31         <button id="addAlbumBtn" type="button">Add Album</button>
32         <span id="error-message"></span>
33     </form>
34     <h2>Album List</h2>
35     <ul id="albumList"></ul>
36     <h2>Favorites Album</h2>
37     <ul id="favoriteAlbumList"></ul>
38     <script src=".//script.js"></script>
39 </body>
40 </html>
..
```

```
public > # style.css > button
1  body{
2      font: 18px Arial, sans-serif;
3      background: #lightblue
4      padding: 10px;
5  }
6
7  h1 {
8      font-family: 'Times New Roman';
9      font-size: 40px;
10 }
11
12 #albumList {
13     color: #blue;
14     background-color: #lightgray;
15     text-align: center;
16     font-size: 14px;
17 }
18
19 ul {
20     margin-left: 10px;
21     padding-left: 20px;
22     list-style-type: square;
23 }
24
25 input[type="text"] {
26     padding: 5px 10px;
27     width: 180px;
28 }
29
30 button {
31     background-color: #rgb(76, 175, 80);
32     color: #red;
33     padding: 10px 20px;
34     border: none;
35 }
36
37 #error-message {
38     color: #ff0000;
39     font-weight: normal;
40     font-size: 14px;
41     display: block;
42     text-align: left;
43 }
44
45 #favoriteAlbumList li {
46     color: #green;
47     font-size: 12px;
48     font-weight: light;
49 }
50
51 input[type="number"] {
52     width: 120px;
53     padding: 8px;
54     margin-top: 5px;
55     background-color: #yellow;
56 }
```

```
public > JS script.js > addEventListener("click") callback
1  document.getElementById("addAlbumBtn").addEventListener("click", ()=>{
2
3  const albumnameinput=document.getElementById("albumName").value.trim();
4  const releaseyearinput=document.getElementById("releaseYear").value.trim();
5  const genreinput=document.getElementById("genre").value;
6  const favoriteinput=document.getElementById("favorite").checked;
7  const errorMessage=document.getElementById("error-message");
8  const albumlistinput=document.getElementById("albumList");
9  const favoritealbumlistinput=document.getElementById("FavoriteAlbumList");
10
11 if(!albumnameinput||!releaseyearinput||!genreinput)
12 {
13     errorMessage.textContent="Please fill out all fields!";
14     return;
15 }
16 errorMessage.textContent="";
17 const li=document.createElement("li");
18 li.innerHTML=`<strong>${albumnameinput}</strong><br>
19 | | | | <em>Released:</em> ${releaseyearinput} | <em>Genre:</em> ${genreinput}`;
20 if(favoriteinput)
21 {
22     favoritealbumlistinput.appendChild(li);
23 }
24 else{
25     albumlistinput.appendChild(li);
26 }
27 document.getElementById("albumName").value="";
28 document.getElementById("releaseYear").value="";
29 document.getElementById("genre").value="";
30 document.getElementById("favorite").checked=false;
31
32});
```

Practice at home 1:

```
public > index.html > html > head > style #form
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <style>
8     body{
9       background-color: lightgray;
10    }
11   #form{
12     background-color: white;
13     width: 300px;
14     margin: auto;
15     margin-top:150px;
16     border-radius: 5px;
17     padding-left: 100px;
18     padding-right: 10px;
19   }
20   #mark-progress{
21     background-color: rgb(41,168,71);
22     color: white;
23     padding: 10px 10px 10px 10px;
24     border-radius:5px;
25   }
26   #reset-progress{
27     background-color: crimson;
28     color: white;
29     padding: 10px 10px 10px 10px;
30     border-radius:5px;
31   }
32   </style>
33 </head>
```

```
34  <body>
35    <div id="form">
36      <h1>Habit Tracker</h1>
37      <input type="checkbox" id="chkWater">
38      <label>Drink Water</label><br>
39      <input type="checkbox" id="chkExercise">
40      <label>Exercise</label><br>
41      <input type="checkbox" id="chkReading">
42      <label>Read a Book</label><br><br>
43      <button id="mark-progress" type="submit" onclick="progress()">Mark Progress</button>
44      <button id="reset-progress" type="reset" onclick="resetAll()">Reset</button>
45      <p id="summary">You have completed 0 habits today.</p>
46    </div>
47    <script src="script.js"></script>
48  </body>
49 </html>
```

```
public > JS scriptjs > ⚡ progress
1  var box1=document.getElementById('chkWater');
2  var box2=document.getElementById('chkExercise');
3  var box3=document.getElementById('chkReading');
4  var count=0;
5  function progress(){
6    if(box1.checked)
7    {
8      count++;
9    }
10   if(box2.checked)
11   {
12     count++;
13   }
14   if(box3.checked)
15   {
16     count++;
17   }
18   document.getElementById('summary').textContent="You have completed "+count+" habits today.";
19 }
20 }
21 function resetAll(){
22   box1.checked=false;
23   box2.checked=false;
24   box3.checked=false;
25   document.getElementById('summary').textContent="You have completed 0 habits today.";
26 }
```

Practice at home 2:

```
public > index.html > html > body > div#container > button#generateBtn
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="UTF-8">
5       <meta http-equiv="X-UA-Compatible" content="IE=edge">
6       <meta name="viewport" content="width=device-width, initial-scale=1.0">
7       <title>Multiplication Table Generator</title>
8       <style>
9           body{
10               background-color: #lightgray;
11           }
12           #container
13           {
14               background-color: #white;
15               padding: 10px 10px 10px 10px;
16               width: 300px;
17               border-radius: 5px;
18               margin: auto;
19               margin-top: 100px;
20           }
21           input{
22               width: 280px;
23               height:35px;
24               border-radius: 5px;
25               border-color: #rgba(0, 0, 0, 0.467);
26           }
27           #generateBtn{
28               background-color: #rgb(77,176,80);
29               color: #white;
30               text-align: center;
31               width: 290px;
32               height: 30px;
33               border-radius: 5px;
34               border-style: none;
35           }
36       </style>
37   </head>
```

```
39 <body>
40   <div id="container">
41     <h3>Multiplication Table Generator</h3>
42     <label for="number">Enter a number: </label><br><br>
43     <input type="number" id="number" placeholder="Enter a number"><br><br>
44     <label for="start">Start Value: </label><br><br>
45     <input type="number" id="start" placeholder="Start of range"><br><br>
46     <label for="end">End Value: </label><br><br>
47     <input type="number" id="end" placeholder="End of range"><br><br>
48     <button id="generateBtn" type="button" onclick="generatetable()">Generate Table</button><br><br>
49     <div id="errormessage"></div><br><br>
50     <div id="table"></div><br><br>
51   </div>
52   <script src="script.js"></script>
53 </body>
54 </html>
```

```
public > JS script.js > ↯ generatetable
1  function generatetable()
2  {
3    let number=document.getElementById('number').value;
4    let start=document.getElementById('start').value;
5    let end=document.getElementById('end').value;
6
7    let errorDiv=document.getElementById('errormessage');
8    let tableDiv=document.getElementById('table');
9
10   errorDiv.textContent='';
11   tableDiv.innerHTML='';
12
13   if(number===""||start===""||end=="")
14   {
15     errorDiv.textContent="All fields are required!";
16     return;
17   }
18
19   number=parseInt(number);
20   start=parseInt(start);
21   end=parseInt(end);
22
23   let output=`Multiplication table of ${number}<br>`;
24   for(let i=start;i<=end;i++)
25   {
26     |   output+=`${number} ✖ ${i} = ${number*i}<br>`  

27   }
28   tableDiv.innerHTML=output;
29 }
```

Challenge yourself:

```
public > index.html > html > body > label#marksLabel
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9      <!-- Write your code -->
10     <h1>Grade Calculator</h1>
11     <label id="marksLabel">Enter your marks (0-100):</p>
12     <input type="number" id="marks">
13     <button id="getGradeButton" type="button" onclick="calculate()">Convert</button><br><br>
14     <p id="gradeResult"></p>
15     <script src="script.js"></script>
16 </body>
17 </html>
```

```
public > scriptjs > ...
● 1  function calculate(){
2      var mark=document.getElementById('marks').value;
3      mark=mark.trim();
4
5      if(mark=="")
6      {
7          document.getElementById('gradeResult').textContent="Enter any marks to get the grades.";
8          return;
9      }
10     mark=Number(mark);
11     if(mark>=90 && mark<=100)
12     {
13         document.getElementById('gradeResult').textContent="For marks " + mark + ", your grade is: A";
14     }
15     else if(mark>=80 && mark<=89)
16     {
17         document.getElementById('gradeResult').textContent="For marks " + mark + ", your grade is: B";
18     }
19     else if(mark>=70 && mark<=79)
20     {
21         document.getElementById('gradeResult').textContent="For marks " + mark + ", your grade is: C";
22     }
23     else if(mark>=60 && mark<=69)
24     {
25         document.getElementById('gradeResult').textContent="For marks " + mark + ", your grade is: D";
26     }
27     else if(mark<60 && mark>=0)
28     {
29         document.getElementById('gradeResult').textContent="For marks " + mark + ", your grade is: F";
30     }
31     else
32     {
33         document.getElementById('gradeResult').textContent="The marks should be between 0-100.";
34     }
35 }
```

Web Programming

Week 5

Type the code:

```
index.html X

public > index.html > html > body > button#highlightBtn
1  <!-- write your code here -->
2  <html>
3      <head>
4          <title>Button Toggle</title>
5          <style>
6              .highlight{
7                  background-color: yellow;
8              }
9          </style>
10         </head>
11         <body>
12             <p id="textPara">This is a sample paragraph to be highlighted.</p>
13             <button id="highlightBtn">Highlight</button>
14
15             <script>
16                 document.addEventListener("DOMContentLoaded", ()=>{
17                     const para=document.getElementById("textPara");
18                     const button=document.getElementById("highlightBtn");
19                     button.addEventListener("click", ()=>{
20                         para.classList.toggle("highlight");
21                     });
22                 });
23             </script>
24         </body>
25     </html>
```

Fix the code:

```
public > index.html > html > body.p-4 > div.container > form#userForm > div.mb-3 > div#ageError.invalid-feedback
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8" />
5    <title>Form Validation</title>
6    <meta name="viewport" content="width=device-width, initial-scale=1" />
7
8    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" />
9  </head>
10 <body class="p-4">
11
12   <div class="container">
13     <h3 class="mb-4">User Information Form</h3>
14
15     <form id="userForm" action="">
16       <div class="mb-3">
17         <label for="name" class="form-label">Name</label>
18         <input type="text" class="form-control" id="name" required />
19         <div id="nameError" class="invalid-feedback">Please enter your name.</div>
20       </div>
21
22       <div class="mb-3">
23         <label for="email" class="form-label">Email</label>
24         <input type="email" class="form-control" id="email" required />
25         <div id="emailError" class="invalid-feedback">Please enter a valid email address.</div>
26       </div>
27
28       <div class="mb-3">
29         <label for="age" class="form-label">Age</label>
30         <input type="number" class="form-control" id="age" min="18" max="120" required />
31         <div id="ageError" class="invalid-feedback">Age must be between 18 and 120.</div>
32       </div>
33
34       <button type="submit" class="btn btn-primary">Submit</button>
35     </form>
36
37     <div id="successMessage" class="mt-3 alert alert-success d-none">
38       <p>Success! Here is the data:</p>
39       <p id="successData"></p>
40     </div>
41   </div>
42
43   <script src="script.js"></script>
44
45 </body>
46 </html>
```

```
public > js script.js > document.addEventListener("DOMContentLoaded") callback > form.addEventListener("submit") callback

---

1  document.addEventListener("DOMContentLoaded",function(){  
2      const Form=document.getElementById("userForm");  
3      const name=document.getElementById("name");  
4      const email=document.getElementById("email");  
5      const age=document.getElementById("age");  
6      const emailError=document.getElementById("emailError");  
7  
8      const successmsg=document.getElementById("successMessage");  
9  
10     const successdata=document.getElementById("successData");  
11  
12  
13     form.addEventListener("submit",function(e){  
14         if(!/^[^]+@[^\.]+\.\com$/.test(email))  
15         {  
16             email.classList.add("is-invalid");  
17             emailError.classList.remove("d-none");  
18         }  
19         e.preventDefault();  
20         successmsg.classList.remove("d-none");  
21         successdata.textContent='Name: ${name.value}, Email: ${email.value}, Age: ${age.value}';  
22         form.reset();  
23     })  
24 })
```

Practice at home 1:

```
public > index.html > html > head > style > .layered-img
1  <!-- write your code here -->
2  <html>
3      <head>
4          <title>Images</title>
5          <style>
6              .layered-img{
7                  position: absolute;
8                  width:100%;
9                  height:100%;
10             }
11             .image-container{
12                 position: relative;
13                 width: 400px;
14                 height: 250px;
15                 margin: 20px auto;
16             }
17         }
18     </style>
19 </head>
20 <body>
21     <div class="image-container">
22         
23         
24         
25     </div>
26     <div class="button-container">
27         <button id="btn1" type="button">Image 1</button>
28         <button id="btn2" type="button">Image 2</button>
29         <button id="btn3" type="button">Image 3</button>
30     </div>
31     <script>
32         document.addEventListener("DOMContentLoaded", ()=>{
33             const imgs={img1:document.getElementById("img1"),img2:document.getElementById("img2"),img3:document.getElementById("img3")};
34
35
36
37             function bringFront(selectedimg){
38                 for(let id in imgs)
39                 {
40                     imgs[id].style.zIndex=(id==selectedimg)? "10": "1";
41                 }
42             }
43             document.getElementById("btn1").addEventListener("click", ()=>bringFront("img1"));
44             document.getElementById("btn2").addEventListener("click", ()=>bringFront("img2"));
45             document.getElementById("btn3").addEventListener("click", ()=>bringFront("img3"));
46
47         });
48     </script>
49     </body>
50 </html>
```

Practice at home 2:

```
public > index.html > html > body > div.col-md-3 > select#fontColorSelect.form-select > option
1  <!-- write your code here -->
2  <html>
3      <head>
4          <title>Live text editor</title>
5          <script src="script.js"></script>
6      </head>
7      <body>
8          <h3>Live Text Editor</h3>
9          <div class="col-md-3">
10             <label for="FontSizeSelect">Font Size</label>
11             <select id="FontSizeSelect" class="form-select">
12                 <option selected>12px</option>
13                 <option>16px</option>
14                 <option>20px</option>
15             </select>
16         </div>
17
18         <div class="col-md-3">
19             <label for="FontColorSelect">Font Color</label>
20             <select id="FontColorSelect" class="form-select">
21                 <option value="black" selected>Black</option>
22                 <option value="blue">Blue</option>
23                 <option value="red">Red</option>
24             </select>
25         </div>
26
27         <div class="btn-grp">
28             <button id="boldBtn" class="btn btn-outline-dark">Bold</button>
29             <button id="italicBtn" class="btn btn-outline-dark">Italic</button>
30             <button id="resetBtn">Reset</button>
31         </div>
32         <textarea id="editor" placeholder="Type something here... "></textarea>
33     </body>
34 </html>
```

```
public > JS script.js > ⏪ document.addEventListener("DOMContentLoaded") callback
1 // write your Scripts here
2 document.addEventListener("DOMContentLoaded",function(){
3     const editor=document.getElementById("editor");
4     const fontSizeSelect=document.getElementById("fontSizeSelect");
5     const fontColorSelect=document.getElementById("fontColorSelect");
6     const boldBtn=document.getElementById("boldBtn");
7     const italicBtn=document.getElementById("italicBtn");
8     const resetBtn=document.getElementById("resetBtn");
9
10    editor.style.fontSize="12px";
11    editor.style.color="black";
12    editor.style.fontWeight="Normal"
13    editor.style.fontStyle="Normal";
14
15    fontSizeSelect.addEventListener("change",function(){
16        editor.style.fontSize=fontSizeSelect.value;
17    });
18    fontColorSelect.addEventListener("change",function(){
19        editor.style.color=fontColorSelect.value;
20    });
21    boldBtn.addEventListener("click",function(){
22        editor.style.fontWeight=editor.style.fontWeight==="bold"?"normal":"bold";
23        boldBtn.classList.toggle("active");
24    })
25    italicBtn.addEventListener("click",function(){
26        editor.style.fontStyle=editor.style.fontStyle==="italic"?"normal":"italic";
27        italicBtn.classList.toggle("active");
28    })
29    resetBtn.addEventListener("click",function(){
30        editor.style.fontSize="12px";
31        editor.style.color="black";
32        editor.style.fontWeight="Normal"
33        editor.style.fontStyle="Normal";
34
35        fontColorSelect.value="black";
36        fontSizeSelect.value="12px";
37
38        boldBtn.classList.remove("active");
39        italicBtn.classList.remove("active");
40    })
41
42});
```

Challenge yourself:

```
public > index.html > html > body.text-center.p-4 > script
1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <meta charset="UTF-8">
5          <meta name="viewport" content="width=device-width, initial-scale=1.0">
6          <title>Suresh's Kids' Sketch Pad</title>
7          <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
8          <link rel="stylesheet" href="style.css">
9      </head>
10     <body class="text-center p-4">
11         <h3>Suresh's Kids' Sketch Pad</h3>
12         <canvas id="sketchPad" width="500" height="300"></canvas>
13         <div class="mt-3">
14             <button id="defaultBtn" class="btn btn-dark">Default</button>
15             <button id="redBtn" class="btn btn-danger">Red</button>
16             <button id="blueBtn" class="btn btn-primary">Blue</button>
17             <button id="greenBtn" class="btn btn-success">Green</button>
18             <button id="eraserBtn" class="btn btn-secondary">Eraser</button>
19             <button id="clearBtn" class="btn btn-warning">Clear Canvas</button>
20         </div>
21         <p>Current Drawing Color: <span id="currentColor">Black</span></p>
22         <script src="script.js"></script>
23     </body>
24 </html>
25
26
```

```
public > # style.css > ...
1  /* write your stylings here */
2  body{
3      background-color: #f8f9fa;
4  }
5  #sketchPad{
6      border: 2px solid #000;
7      background-color: #fff;
8      cursor: crosshair;
9  }
10
11
```

```
public > js scriptjs > ...
1 // write your Scripts here
2 const canvas=document.getElementById('sketchPad');
3 const ctx=canvas.getContext('2d');
4 let drawing=false, prevX=0, prevY=0, color="black";
5
6 ctx.strokeStyle=color;
7 ctx.lineWidth=3;
8 ctx.lineCap="round";
9 const setColor=(color)=>{
10     ctx.strokeStyle=color;
11     document.getElementById('currentColor').textContent=color=="white"?`Eraser`:color.charAt(0).toUpperCase()+color.slice(1);
12
13 };
14 canvas.addEventListener('mousedown',e=>{
15     drawing=true;
16     const r=canvas.getBoundingClientRect();
17     prevX=e.clientX - r.left;
18     prevY=e.clientY - r.top;
19 });
20 canvas.addEventListener('mouseup',()=>drawing=false);
21 canvas.addEventListener('mouseout',()=>drawing=false);
22 canvas.addEventListener('mousemove',e=>{
23     if(!drawing) return;
24     const r = canvas.getBoundingClientRect();
25     const x = e.clientX - r.left, y = e.clientY - r.top;
26     ctx.beginPath();
27     ctx.moveTo(prevX,prevY);
28     ctx.lineTo(x,y);
29     ctx.stroke();
30     prevX=x, prevY=y;
31 });
32
33 document.getElementById('defaultBtn').addEventListener('click', ()=>{ctx.strokeStyle="black"; setColor("black")});
34 document.getElementById('redBtn').addEventListener('click', ()=>{ctx.strokeStyle="red"; setColor("red")});
35 document.getElementById('blueBtn').addEventListener('click', ()=>{ctx.strokeStyle="blue"; setColor("blue")});
36 document.getElementById('greenBtn').addEventListener('click', ()=>{ctx.strokeStyle="green"; setColor("green")});
37 document.getElementById('eraserBtn').addEventListener('click', ()=>{ctx.strokeStyle="white"; setColor("white")});
38 document.getElementById('clearBtn').addEventListener('click', ()=>{ctx.fillStyle="rgb(255,255,255)"; ctx.fillRect(0,0,canvas.width, canvas.height); })
39
40 
```

Web programming – week 6

TYPE THE CODE

```
nodeapp > controllers > JS taskController.js > ...
1  /*
2   * WEEK 6 - TYPE THE CODE
3   * controllers > taskController.js
4  */
5
6 const Task = require("../models/taskModel");
7
8 exports.createTask = async(req, res) => {
9   try {
10     const {title, completed} = req.body;
11     const task = new Task({title, completed});
12     const savedTask = await task.save();
13     res.status(201).json(savedTask);
14   } catch(err) {
15     res.status(500).json({message: err.message});
16   }
17 };
18
19 exports.getAllTasks = async(req, res) => {
20   try {
21     res.status(200).json(await Task.find());
22   } catch(err) {
23     res.status(500).json({message: err.message});
24   }
25 };
26
27 exports.getTaskById = async(req, res) => {
28   try {
29     const task = await Task.findById(req.params.id);
30     if(!task) return res.status(404).json({message: "Task not found"});
31     res.status(200).json(task);
32   } catch(err) {
33     res.status(500).json({message: err.message});
34   }
35 }
```

MongoDB JS index.js JS taskRoutes.js M X JS taskController.js M JS taskModel.js M

```
nodeapp > routers > JS taskRoutes.js > ...
1  /*
2   * WEEK 6 - TYPE THE CODE
3   * routers > taskRoutes.js
4  */
5
6 const express = require('express');
7 const {createTask, getAllTasks, getTaskById} = require('../controllers/taskController');
8
9 const router = express.Router();
10 router.post("/", createTask);
11 router.get("/", getAllTasks);
12 router.get("/:id", getTaskById);
13
14 module.exports = router;
```

MongoDB JS index.js JS taskRoutes.js JS taskController.js M J

nodeapp > models > JS taskModel.js > ...

```
1  /*
2   |   WEEK 6 - TYPE THE CODE
3   |   models > taskModel.js
4  */
5
6  const mongoose = require('mongoose');
7
8  const taskSchema = new mongoose.Schema({
9      title: {
10          type: String,
11          required: true
12      },
13      completed: {
14          type: Boolean,
15          default: false
16      }
17 });
18
19 module.exports = mongoose.model("Task", taskSchema);
```

MongoDB JS index.js M X JS taskRoutes.js M JS taskController.js M JS taskModel.js M

nodeapp > JS index.js > ...

```
1  /*
2   |   WEEK 6 - TYPE THE CODE
3   |   index.js
4  */
5
6  const express = require('express');
7  const mongoose = require('mongoose')
8  const taskRoutes = require('./routers/taskRoutes');
9
10 const app = express();
11 app.use(express.json());
12 app.use("/tasks",taskRoutes);
13
14 mongoose.connect("mongodb://127.0.0.1:27017/taskdb")
15     .then(() => app.listen(8080, () => console.log("Server running on 8080")))
16     .catch(err => console.error(err))|
```

FIX THE CODE

```
nodeapp > JS index.js > ...
1 // WEEK 6 - Fix the Code
2
3 const express = require('express');
4 const mongoose = require('mongoose');
5 const itemRoutes = require('./routers/itemRouter');
6
7 const app = express();
8
9 app.use(express.json());
10
11 mongoose.connect('mongodb://127.0.0.1:27017/itemDB', {
12   useNewUrlParser: true,
13   useUnifiedTopology: true
14 })
15   .then(() => console.log('Connected to MongoDB'))
16   .catch(err => console.log(err));
17
18 app.use('/items', itemRoutes);
19
20 const port = 8080;
21 app.listen(port, () => {
22   console.log('Server is running on port ' + port);
23 });
24
```

```
nodeapp > routers > JS itemRouter.js > ...
2 // WEEK 6 - Fix the Code
3
4 const { createItem, getAllItems } = require('../controllers/itemController');
5
6 const router = express.Router();
7
8 router.post('/', createItem);
9
10 router.get('/', getAllItems);
11
12 module.exports = router;
13
```

```
nodeapp > controllers > itemController.js > ...
1 // WEEK 6 - Fix the Code
2
3 const Item = require('../models/itemModel');
4 const createItem = async (req, res) => {
5   const { name, quantity } = req.body;
6
7   if (!name || quantity == null) {
8     return res.status(400).json({ message: 'Name and quantity is required' });
9   }
10
11  try {
12    const newItem = new Item({
13      name, quantity
14    });
15
16    const savedItem = await newItem.save();
17    res.status(201).json(savedItem);
18  } catch (err) {
19    res.status(500).json({ message: err.message });
20  }
21};
22
23 const getAllItems = async (req, res) => {
24  try {
25    const items = await Item.find();
26    res.status(200).json(items);
27  } catch (err) {
28    res.status(500).json({ message: err.message });
29  }
30};
31
32 module.exports = {
33   createItem,
34   getAllItems
35 };
36
```

```
nodeapp > models > itemModel.js > ...
```

```
1 // WEEK 6 - Fix the Code
2
3 const mongoose = require('mongoose');
4
5 const itemSchema = new mongoose.Schema({
6   name: {
7     type: String,
8     required: true
9   },
10  quantity: {
11    type: Number,
12    required: true
13  }
14});
15
16 const Item = mongoose.model('Item', itemSchema);
17
18 module.exports = Item;
```

PRACTICE AT HOME 1

```
nodeapp > JS index.js > ...
1  /*
2   |   WEEK 6 - PRACTICE AT HOME 1
3   |   index.js
4   */
5
6  const express = require('express');
7  const mongoose = require('mongoose');
8  const productRoutes = require('./routers/productRoutes');
9  const errorHandler = require('./middlewares/errorHandler');
10
11 const app = express();
12 app.use(express.json());
13
14 mongoose.connect("mongodb://127.0.0.1:27017/productsDB", {
15   |   useNewUrlParser: true,
16   |   useUnifiedTopology: true
17 })
18 .then(() => console.log('MongoDB connected successfully'))
19 .catch((err)=> console.error(err));
20
21 app.use('./products', productRoutes);
22 app.use(errorHandler);
23
24 app.listen(8080, ()=>{
25   |   console.log('Server running on port 8080');
26 });


```

```
nodeapp > routers > JS productRoutes.js > ...
1  /*
2   |   WEEK 6 - PRACTICE AT HOME 1
3   |   routers > productRoutes.js
4   */
5
6  const express = require('express');
7  const router = express.Router();
8  const {createProduct, getAllProducts} = require('../controllers/productController');
9
10 router.post('/', createProduct);
11 router.get('/', getAllProducts);
12
13 module.exports = router;
14 |
```

```
nodeapp > models > JS productModel.js > ...
1  /*
2   |   WEEK 6 - PRACTICE AT HOME 1
3   |   models > productModel.js
4   */
5
6  const mongoose = require('mongoose');
7  const productSchema = new mongoose.Schema({
8      name: {
9          type: String,
10         required: [true, 'Product name is required']
11     },
12     price: {
13         type: Number,
14         required: [true, 'Product price is required']
15     },
16     category: {
17         type: String,
18         required: [true, 'Product category is required']
19     },
20     inStock: {
21         type: Boolean,
22         default: true
23     }
24 }, {timestamps: true});
25
26 module.exports = mongoose.model('Product', productschema);
```

```
nodeapp > middlewares > JS errorHandler.js > ...
1  /*
2   |   WEEK 6 - PRACTICE AT HOME 1
3   |   middlewares > errorHandler.js
4   */
5
6  const errorHandler = (err, req, res, next) => {
7      console.error(err.stack);
8      const statusCode = 500;
9      res.status(statusCode).json({
10         error: {
11             message: err.message,
12             stack: process.env.NODE_ENV === 'production' ? null : err.stack
13         }
14     });
15 };
16
17 module.exports = errorHandler;
18 |
```

```

nodeapp > controllers > JS productController.js > ...
1  /*
2   * WEEK 6 - PRACTICE AT HOME 1
3   * controllers > productController.js
4   */
5
6 const Product = require('../models/productModel');
7
8 const createProduct = async(req, res, next) => {
9     try {
10         const {name, price, category, inStock} = req.body;
11         if(!name || !price || !category) {
12             throw new Error('Database Error');
13         }
14         const product = new Product({name, price, category, inStock});
15         const savedProduct = await product.save();
16         res.status(201).json(savedProduct);
17     } catch(err) {
18         next(err);
19     }
20 };
21
22 const getAllProducts = async(req, res, next) => {
23     try {
24         const products = await Product.find();
25         res.status(200).json(products);
26     } catch(err) {
27         next(err);
28     }
29 };
30
31 module.exports = {createProduct, getAllProducts};

```

PRACTICE AT HOME 2

JS movieController.js M JS movieRoutes.js M X JS movieModel.js M JS index.js M

```

nodeapp > routers > JS movieRoutes.js > ...
1  /*
2   * WEEK 6 - PRACTICE AT HOME 2
3   * routers > movieRoutes.js
4   */
5
6 const ex = require('express');
7 const r = ex.Router();
8 const {cm, gm, gmi, um, dm} = require('../controllers/movieController');
9
10 r.post('/movies', cm);
11 r.get('/movies', gm);
12 r.get('/movies/:id', gmi);
13 r.put('/movies/:id', um);
14 r.delete('/movies/:id', dm);
15
16 module.exports = r;
17
18

```

nodeapp > **JS** index.js M **JS** movieRoutes.js M **JS** movieModel.js M **JS** index.js M X

```
1  /*
2   *      WEEK 6 - PRACTICE AT HOME 2
3   *      index.js
4   */
5
6  const ex = require('express');
7  const mg = require('mongoose');
8  const mr = require('./routers/movieRoutes');
9
10 const app = ex();
11 app.use(ex.json());
12
13 mg.connect('mongodb://127.0.0.1:27017/movieDB')
14     .catch(e => console.log(e));
15
16 app.use('/movies', mr);
17 app.listen(8080);
```

nodeapp > models > **JS** movieModel.js > ...

```
1  /*
2   *      WEEK 6 - PRACTICE AT HOME 2
3   *      models > movieModels.js
4   */
5
6  const mg = require('mongoose');
7  const ms = new mg.Schema({
8      title: {
9          type: String,
10         required: true
11     },
12     director: {
13         type: String,
14         required: true
15     },
16     releaseYear: {
17         type: Number,
18         required: true
19     },
20     genre: {
21         type: String,
22         required: true
23     }
24 });
25
26 const m = mg.model('Movie', ms);
27 module.exports = m;
```

```
JS movieController.js M X JS movieRoutes.js M JS movieModel.js M JS index.js M
nodeapp > controllers > JS movieController.js > [e] getMovieById
1  /*
2   * WEEK 6 - PRACTICE AT HOME 2
3   * controller > movieController.js
4   */
5
6  const Movie = require('../models/movieModel');
7  const createMovie = async(req, res) => {
8    const {title, director, releaseYear, genre} = req.body;
9    try {
10      const newMovie = new Movie({
11        title, director, releaseYear, genre
12      });
13
14      const savedMovie = await newMovie.save();
15      res.status(201).json(savedMovie);
16    } catch (err){ res.status(500).json({message: err.message}); }
17  };
18
19  const getAllMovies = async(req, res) => {
20    try {
21      const movies = await Movie.find();
22      res.status(200).json(movies);
23    } catch(err) { res.status(500).json({message: err.message}); }
24  };
25
26  const getMovieById = async(req, res) => {
27    try {
28      const movie = await Movie.findById(req.params.id);
29      if(!movie) { return res.status(404).json({message: "Movie not found"}); }
30      res.status(200).json(movie);
31    } catch(err) { res.status(500).json({message: err.message}); }
32  };
33
34  const updateMovie = async(req, res) => {
35    try {
36      const movie = await Movie.findByIdAndUpdate(req.params.id, req.body, {new: true, runValidators: true});
37      if(!movie) {
38        return res.status(404).json({message: "Movie not found"});
39      }
40
41      res.status(200).json(movie);
42    } catch(err) {
43      res.status(500).json({message: err.message});
44    }
45  };
46
47  const deleteMovie = async(req, res) => {
48    try {
49      const movie = await Movie.findByIdAndDelete(req.params.id);
50      if(!movie) {
51        return res.status(404).json({message: "Movie not found"});
52      }
53
54      res.status(200).json({message: "Movie deleted successfully"});
55    } catch(err) {
56      res.status(500).json({message: err.message});
57    }
58  };
59
60  module.exports = {createMovie, getAllMovies, getMovieById, updateMovie, deleteMovie};
```

CHALLENGE YOURSELF

```
nodeapp > controllers > JS customerController.js > ...
1  /*
2   | WEEK 6 - CHALLENGE YOURSELF
3   | controllers > customerController.js
4  */
5
6  const Cust = require('../models/customerModel');
7  const createCustomer = async(req, res, next) => {
8    try{
9      const {name, email, phone, isActive} = req.body;
10     if(!name || !email || !phone) return res.status(404).json({message: "Name, email, and phone are required"});
11     const c = new Cust({name, email, phone, isActive});
12     const s = await c.save();
13     res.status(201).json(s);
14   } catch(err) { next(err); }
15 };
16
17 const getAllCustomers = async(req, res, next) => {
18   try { res.status(200).json(await Cust.find()); } catch(err) { next(err); }
19 };
20
21 const updateCustomer = async(req, res, next) => {
22   try {
23     const u = await Cust.findByIdAndUpdate(req.params.id, req.body, {new: true});
24     if(!req.body.name || !req.body.email || !req.body.phone)
25       return res.status(404).json({message: "Name, email, and phone are required to update the customer"});
26     if(!u) return res.status(404).json({message: "Customer not found"});
27     res.status(200).json(u);
28   } catch(err) { next(err); }
29 };
30
31 module.exports =[createCustomer, getAllCustomers, updateCustomer];

nodeapp > middlewares > JS errorHandler.js > ...
1  /*
2   | WEEK 6 - CHALLENGE YOURSELF
3   | middlewares > errorHandler.js
4  */
5
6  const eh = (err, req, res, next) => {
7    console.error(err.message);
8    res.status(500).json({message: "Server Error"});
9  };
10
11 module.exports = eh;
```

```
nodeapp > models > JS customerModel.js > ...
```

```
1  /*
2   |     WEEK 6 - CHALLENGE YOURSELF
3   |     models > customerModel.js
4  */
5
6  const mg = require("mongoose");
7  const cs = new mg.Schema({
8      name: {
9          type: String,
10         required: true
11     },
12     email: {
13         type: String,
14         required: true
15     },
16     phone: {
17         type: String,
18         required: true
19     },
20     isActive: {
21         type: Boolean,
22         default: true
23     }
24 });
25
26 module.exports = mg.model("Customer", cs);
27 |
```

```
nodeapp > routers > JS customerRoutes.js > ...
```

```
1  /*
2   |     WEEK 6 - CHALLENGE YOURSELF
3   |     routers > customerRoutes.js
4  */
5
6  const ex = require('express')
7  const {cc, gc, uc} = require("../controllers/customerController");
8  const r = ex.Router()
9
10 r.post('/customers', cc);
11 r.get('/customers', gc);
12 r.put('/customer/:id', uc);
13
14 module.exports = r;
15
16 |
```

```
nodeapp > JS index.js > ...
1  /*
2   |   WEEK 6 - CHALLENGE YOURSELF
3   |   index.js
4   */
5
6  const ex = require('express');
7  const mg = require('mongoose');
8  const r = require('./routers/customerRoutes');
9  const eh = require('./middlewares/errorHandler');
10
11 const app = ex();
12 app.use(ex.json());
13
14 mg.connect("mongodb://127.0.0.1:27017/customerDB")
15   .then(() => console.log("DB connected"))
16   .catch(err => console.error(err));
17
18 app.use('/api', r);
19 app.use(eh);
20
21 app.listen(8080, () => console.log("Server running on 8080"));|
```

Web Programming – Week 7

Type the Code:

```
MedicalShop.jsx X JS App.js
reactapp > src > components > MedicalShop.jsx > handleAddMedicine
1 import React, {useState} from 'react';
2 const MedicalShop = ()=>{
3     const [inventory, setInventory]=useState([
4         {name: 'Paracetamol', quantity: 50, price: 0.5},
5         {name: 'Aspirin', quantity: 30, price: 0.8},
6         {name: 'Band-Aid', quantity: 100, price: 0.2}
7     ]);
8     const [formData, setFormData]=useState({
9         name:'',
10        quantity:'',
11        price:''
12    });
13    const [search, setSearch]=useState('');
14    const handleChange=(e)=>{
15        const {placeholder,value}=e.target;
16        setFormData((prev)=>({
17            ...prev,
18            [placeholder.toLowerCase()]:value,
19        }));
20    };
21    const handleAddMedicine=()=>{
22        const {name,quantity,price}=formData;
23        if(!name||!quantity||!price){
24            alert('Please fill in all fields');
25            return;
26        }
27        const newMedicine={name,quantity,price};
28        setInventory((prev)=>[...prev,newMedicine]);
29        setFormData({name:'',quantity:'',price:''});
30    };
31    const filteredInventory=inventory.filter((item)=>item.name.toLowerCase().includes(search.toLowerCase()));
32    return(
33        <div className="container my-4">
34            <h2 className='text-center mb-4'>Medical Shop Inventory</h2>
35            {/*Search*/}
36            <input type='text' placeholder='Search Medicine' className='form-control mb-3' value={search} onChange={(e)=>setSearch(e.target.value)}/>
37            {/*Form*/}
38            <div className='row mb-3'>
39                <div className='col-md-4'>
40                    <input type='text' placeholder='Name' className='form-control mb-2' value={formData.name} onChange={handleChange}/>
41                </div>
42                <div className='col-md-4'>
43                    <input type='number' placeholder='Quantity' className='form-control mb-2' value={formData.quantity} onChange={handleChange}/>
44                </div>
45                <div className='col-md-4'>
```

MedicalShop.jsx X JS App.js

reactapp > src > components > MedicalShop.jsx > handleAddMedicine

```
40     |         <input type='text' placeholder='Name' className='form-control mb-2' value={formData.name} onChange={handleChange}/>
41     |     </div>
42     |     <div className='col-md-4'>
43     |         <input type='number' placeholder='Quantity' className='form-control mb-2' value={formData.quantity} onChange={handleChange}/>
44     |     </div>
45     |     <div className='col-md-4'>
46     |         <input type='number' placeholder='Price' step='0.01' className='form-control mb-2' value={formData.price} onChange={handleChange}/>
47     |     </div>
48   </div>
49   <button className='btn btn-success w-100 mb-4' onClick={handleAddMedicine}>Add Medicine</button>
50   {/*Table*/}
51   <table className='table table-striped table-bordered'>
52     <thead className='table-dark'>
53       <tr>
54         <th>Name</th>
55         <th>Quantity</th>
56         <th>Price</th>
57       </tr>
58     </thead>
59     <tbody>
60       {filteredInventory.length>0?(
61         filteredInventory.map((item,index)=>(
62           <tr key={index}>
63             <td>{item.name}</td>
64             <td>{item.quantity}</td>
65             <td>{item.price}</td>
66           </tr>
67         ))
68       ):(
69         <tr>
70           <td colspan='3' className='text-center'>No medicines found</td>
71         </tr>
72       )}
73     </tbody>
74   </table>
75 </div>
76 );
77 }
78 export default MedicalShop;
```

MedicalShop.jsx JS App.js X

reactapp > src > JS App.js > default

```
1 import React from "react";
2 import MedicalShop from "./components/MedicalShop";
3 import './App.css';
4 function App(){
5   return(
6     <div><MedicalShop/></div>
7   );
8 }
9 export default App;
```

Fix the Code:

```
⌘ FoodOrderjsx × JS App.jsx
reactapp > src > components > ⌘ FoodOrderjsx > ⌘ FoodOrder
1 import React, { useState } from 'react';
2 import 'bootstrap/dist/css/bootstrap.min.css';
3
4 const FoodOrder = () => {
5   const [orders, setOrders] = useState([
6     { id: 1, item: 'Burger', quantity: 2, price: 150 },
7     { id: 2, item: 'Pizza', quantity: 1, price: 300 },
8     { id: 3, item: 'Pasta', quantity: 3, price: 200 },
9   ]);
10
11 const [newOrder, setNewOrder] = useState({
12   item: '',
13   quantity: '',
14   price: '',
15 });
16
17 const handleNewOrderChange = (e) => {
18   const { name, value } = e.target;
19   setNewOrder({ ...newOrder, [name]: value });
20 };
21
22 const handleAddOrder = () => {
23   const {item,quantity,price}=newOrder;
24   if (item.trim()=='' || quantity=='' || price=='') {
25     alert('Please fill in all fields');
26     return;
27   }
28
29   const nextId = orders.length==0?1:Math.max(...orders.map((o)=>o.id)) + 1;
30   const toAdd = {
31     id: nextId,
32     item:item.trim(),
33     quantity: Number(quantity),
34     price: Number(price),
35   };

```

FoodOrder.jsx X JS App.js

reactapp > src > components > FoodOrder.jsx > FoodOrder

```
37 |     setOrders((prev)=>[...prev,toAdd]);
38 |     setNewOrder({
39 |         item: '',
40 |         quantity: '',
41 |         price: '',
42 |     });
43 |
44 |
45 |     return (
46 |         <div className="container my-4">
47 |             <h1 className="text-primary text-center mb-4">Food Order Management</h1>
48 |             <div className='table-responsive'>
49 |                 <table className="table table-striped table-bordered">
50 |                     <thead>
51 |                         <tr>
52 |                             <th>ID</th>
53 |                             <th>Food Item</th>
54 |                             <th>Quantity</th>
55 |                             <th>Price ($)</th>
56 |                         </tr>
57 |                     </thead>
58 |                     <tbody>
59 |                         {orders.map((o) => (
60 |                             <tr key={o.id}>
61 |                                 <td>{o.id}</td>
62 |                                 <td>{o.item}</td>
63 |                                 <td>{o.quantity}</td>
64 |                                 <td>{o.price}</td>
65 |                             </tr>
66 |                         ))}
67 |                     </tbody>
68 |                 </table>
69 |             </div>
70 |             <h2 className="text-success text-center mt-5 mb-3">Add New Food Order</h2>
71 |             <div className="row g-3">
72 |                 <div className="col-md-4">
73 |                     <input
74 |                         type="text"
75 |                         className="form-control"
```

FoodOrder.jsx × JS App.js

reactapp > src > components > FoodOrder.jsx > FoodOrder

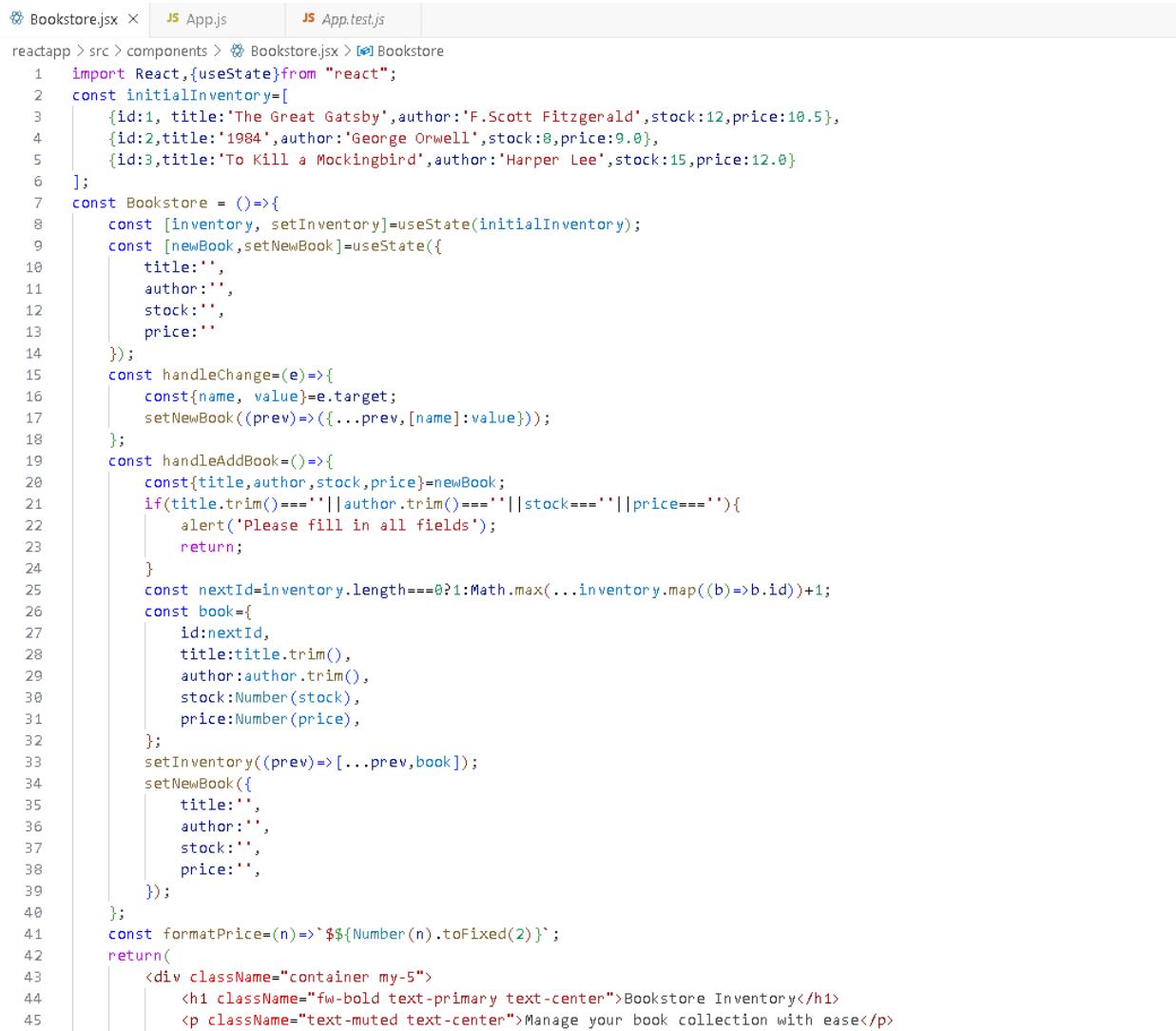
```
76     placeholder="Food Item"
77     name="item"
78     value={newOrder.item}
79     onChange={handleNewOrderChange}
80   
81 
```

FoodOrder.jsx × JS App.js

reactapp > src > App.js > ...

```
1 import React from 'react';
2 import FoodOrder from './components/FoodOrder';
3
4 function App() {
5   return (
6     <div className="App">
7       <FoodOrder />
8     </div>
9   );
10 }
11 export default App;
```

Practice at home:



```
reactapp > src > components > Bookstore.jsx > Bookstore
  1 import React,{useState}from "react";
  2 const initialInventory=[
  3   {id:1,title:'The Great Gatsby',author:'F.Scott Fitzgerald',stock:12,price:10.5},
  4   {id:2,title:'1984',author:'George Orwell',stock:8,price:9.0},
  5   {id:3,title:'To Kill a Mockingbird',author:'Harper Lee',stock:15,price:12.0}
  6 ];
  7 const Bookstore = ()=>{
  8   const [inventory, setInventory]=useState(initialInventory);
  9   const [newBook, setNewBook]=useState({
 10     title:'',
 11     author:'',
 12     stock:'',
 13     price:''
 14   });
 15   const handleChange=(e)=>{
 16     const{name, value}=e.target;
 17     setNewBook((prev)=>({...prev,[name]:value}));
 18   };
 19   const handleAddBook=()=>{
 20     const{title,author,stock,price}=newBook;
 21     if(title.trim()===''||author.trim()===''||stock===''||price===''){
 22       alert('Please fill in all fields');
 23       return;
 24     }
 25     const nextId=inventory.length==0?1:Math.max(...inventory.map((b)=>b.id))+1;
 26     const book={
 27       id:nextId,
 28       title:title.trim(),
 29       author:author.trim(),
 30       stock:Number(stock),
 31       price:Number(price),
 32     };
 33     setInventory((prev)=>[...prev,book]);
 34     setNewBook({
 35       title:'',
 36       author:'',
 37       stock:'',
 38       price:''
 39     });
 40   };
 41   const formatPrice=(n)=>` ${Number(n).toFixed(2)} `;
 42   return(
 43     <div className="container my-5">
 44       <h1 className="fw-bold text-primary text-center">Bookstore Inventory</h1>
 45       <p className="text-muted text-center">Manage your book collection with ease</p>
```

```
Bookstore.jsx X JS App.js JS App.test.js
reactapp > src > components > Bookstore.jsx > Bookstore
46   <h2 className="mt-4 mb-3">Available Books</h2>
47   <div className="row g-3">
48     {inventory.map((book)=>(
49       <div className="col-md-4" key={book.id}>
50         <div className="card h-100 shadow-sm">
51           <div className="card-body">
52             <h5 className="card-title text-success">{book.title}</h5>
53             <p className="card-subtitle mb-2 text-muted">{book.author}</p>
54             <p className="mb-1"><span className="fw-bold">Stock:</span>{book.stock}</p>
55             <p className="mb-0"><span className="fw-bold">Price:</span>{formatPrice(book.price)}</p>
56           </div>
57           <div className="card-footer bg-transparent border-0">
58             <span className="badge bg-info">ID:{book.id}</span>
59           </div>
60         </div>
61       </div>
62     )));
63   </div>
64   <h2 className="mt-5 mb-3">Add a New Book</h2>
65   <div className="card shadow-sm">
66     <div className="card-body">
67       <div className="row g-3">
68         <div className="col-md-3">
69           <input className="form-control" type="text" name="title" placeholder="Book Title" value={newBook.title} onChange={handleChange}/>
70         </div>
71         <div className="col-md-2">
72           <input className="form-control" type="text" name="author" placeholder="Author" value={newBook.author} onChange={handleChange}/>
73         </div>
74         <div className="col-md-2">
75           <input className="form-control" type="number" name="stock" placeholder="Stock" min="0" value={newBook.stock} onChange={handleChange}/>
76         </div>
77         <div className="col-md-2">
78           <input className="form-control" type="number" name="price" placeholder="Price ($)" step=".01" value={newBook.price} onChange={handleChange}/>
79         </div>
80         <div className="col-md-2 d-grid">
81           <button className="btn btn-success w-100" onClick={handleAddBook}>Add Book</button>
82         </div>
83       </div>
84     </div>
85   </div>
86 );
87 </div>;
88 }
89 export default Bookstore;
```

```
Bookstore.jsx X JS App.js JS App.test.js
reactapp > src > JS App.js > default
1 import React from "react";
2 import Bookstore from "./components/Bookstore";
3 import './App.css';
4 function App(){
5   return(
6     <div><Bookstore/></div>
7   );
8 }
8 export default App;
```

Challenge Yourself:

```
❖ MusicStore.jsx ✘ js App.test.js js App.js
reactapp > src > components > ❖ MusicStore.jsx > initialItems > name
1 import React,{useState} from "react";
2 const initialItems=[
3   {id:1,name:'Acoustic Guitar',quantity:10,price:150},
4   {id:2,name:'Electric Keyboard',quantity:5,price:300},
5   {id:3,name:'Drum Set',quantity:3,price:500},
6 ];
7 export default function MusicStore(){
8   const[inventory,setInventory]=useState(initialItems);
9   const[newItem,setNewItem]=useState({name:'',quantity:'',price:''});
10  const onChange=(e)=>{
11    const{name,value}=e.target;
12    setNewItem((prev)=>({...prev,[name]:value}));
13  };
14  const addItem=()=>{
15    const {name,quantity,price}=newItem;
16    if(name.trim()==''||quantity==''||price==''){
17      alert('Please fill in all fields');
18      return;
19    }
20    const nextId=inventory.length==0?1:Math.max(...inventory.map((i)=>i.id))+1;
21    setInventory((prev)=>[...prev,{
22      id:nextId,
23      name:name.trim(),
24      quantity:Number(quantity),
25      price:Number(price),
26    }]);
27    setNewItem({name:'',quantity:'',price:''});
28  };
29  const money=(n)=>`$${Number(n).toFixed(2)}`;
30  return(
31    <div className="container my-4">
32      /*Header bar styled to resemble screenshot*/
33      <div className="card shadow-sm mb-3">
34        <div className="card-body bg-dark text-warning text-center">
35          <h1 className="m-0">Music Store Inventory</h1>
36        </div>
37        <div className="card h-100 shadow-sm">
38          <div className="card-body">
39            <div className="table-responsive">
40              <table className="table table-bordered table-hover shadow-sm m-0">
41                <thead className="table-warning">
42                  <tr>
43                    <th scope="col">ID</th>
44                    <th scope="col">Instrument</th>
45                    <th scope="col">Quantity</th>
46                    <th scope="col">Price ($)</th>
47                  </tr>
48                </thead>
```

MusicStore.jsx X App.test.js App.js

reactapp > src > components > MusicStore.jsx > initialItems > name

```
50      <tbody>
51        {inventory.map((item)=>(
52          <tr key={item.id}>
53            <td>{item.id}</td>
54            <td>{item.name}</td>
55            <td>{item.quantity}</td>
56            <td>{money(item.price)}</td>
57          </tr>
58        )));
59      </tbody>
60    </table>
61  </div>
62  /*Add form*/
63 <h2 className="text-info text-center mt-4 mb-3">Add New Instrument</h2>
64 <div className="row g-2 align-items-stretch">
65   <div className="col-md-5">
66     <input className="form-control border-info" type="text" name="name" placeholder="Instrument / Album Name" value={newItem.name} onChange={onChange}/>
67   </div>
68   <div className="col-md-2">
69     <input className="form-control border-info" type="number" name="quantity" min="0" placeholder="Quantity" value={newItem.quantity} onChange={onChange}/>
70   </div>
71   <div className="col-md-5">
72     <input className="form-control border-info" type="number" name="price" placeholder="Price ($)" step="0.01" min="0" value={newItem.price} onChange={onChange}/>
73   </div>
74   <div className="col-md-2 d-grid">
75     <button className="btn btn-warning w-100 fw-bold" onClick={addItem}>Add Item</button>
76   </div>
77 </div>
78 </div>
79 </div>
80 );
81 }
82 }
```

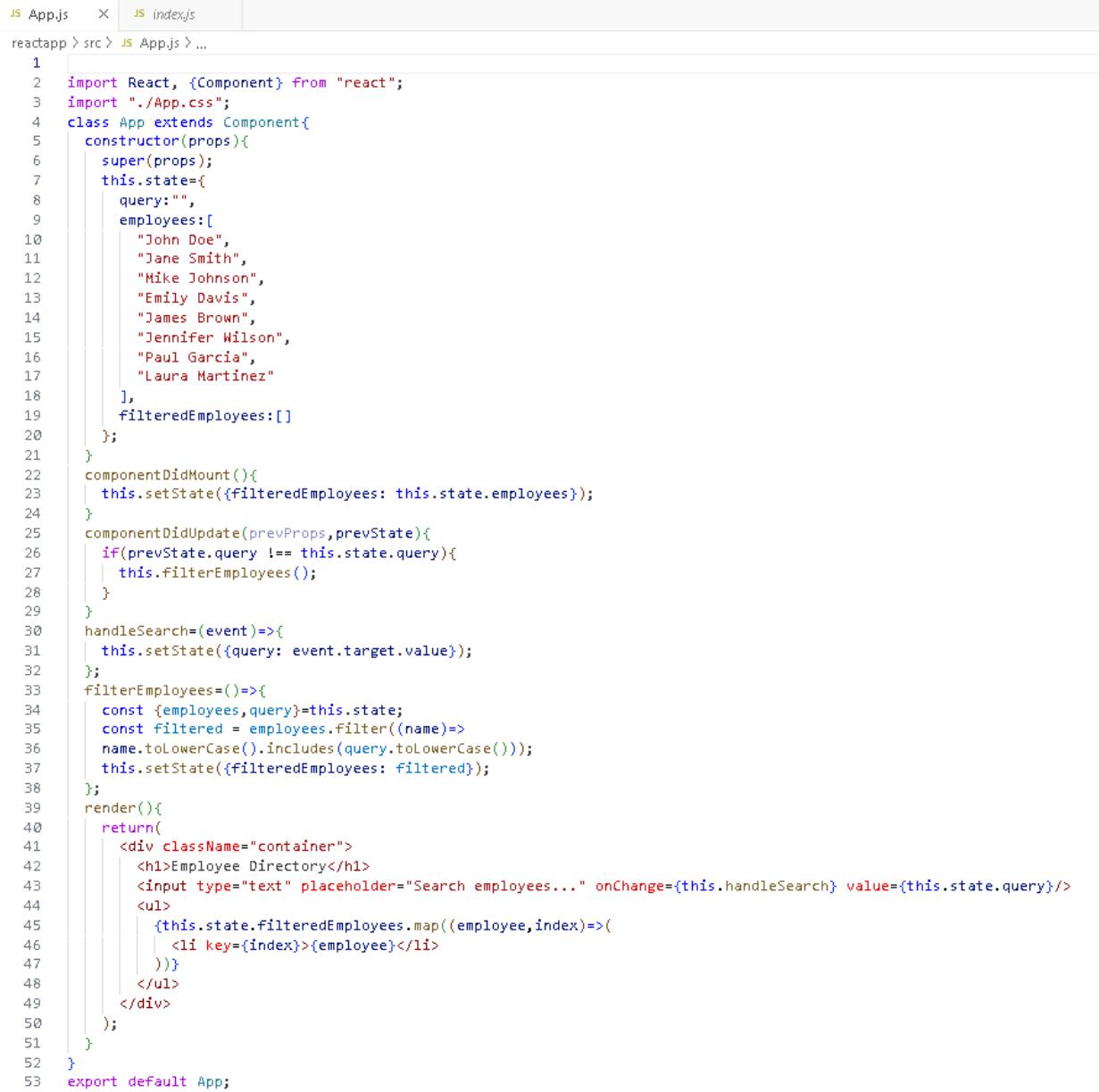
MusicStore.jsx App.test.js App.js X

reactapp > src > App.js > App

```
1 import React from "react";
2 import MusicStore from "./components/MusicStore";
3 export default function App(){
4   return <MusicStore/>;
5 }
```

Web Programming - Week 8

Type the Code:



The image shows a code editor interface with two tabs: "JS App.js" and "JS index.js". The "JS App.js" tab is active, displaying the following code:

```
1 import React, {Component} from "react";
2 import "./App.css";
3 class App extends Component{
4   constructor(props){
5     super(props);
6     this.state={
7       query:"",
8       employees:[
9         "John Doe",
10        "Jane Smith",
11        "Mike Johnson",
12        "Emily Davis",
13        "James Brown",
14        "Jennifer Wilson",
15        "Paul Garcia",
16        "Laura Martinez"
17      ],
18      filteredEmployees:[]
19    };
20  }
21  componentDidMount(){
22   this.setState({filteredEmployees: this.state.employees});
23 }
24  componentDidUpdate(prevProps,prevState){
25   if(prevState.query !== this.state.query){
26     this.filterEmployees();
27   }
28 }
29  handleSearch=(event)=>{
30   this.setState({query: event.target.value});
31 };
32  filterEmployees=()=>{
33   const {employees,query}=this.state;
34   const filtered = employees.filter((name)=>
35     name.toLowerCase().includes(query.toLowerCase()));
36   this.setState({filteredEmployees: filtered});
37 };
38  render(){
39   return(
40     <div className="container">
41       <h1>Employee Directory</h1>
42       <input type="text" placeholder="Search employees..." onChange={this.handleSearch} value={this.state.query}/>
43       <ul>
44         {this.state.filteredEmployees.map((employee,index)=>(
45           <li key={index}>{employee}</li>
46         )))
47       </ul>
48     </div>
49   );
50 }
51 }
52 }
53 export default App;
```

```
JS App.js # App.css X
reactapp > src > # App.css > 📁 .App
1  .App {
2    text-align: center;
3  }
4
5  .App-logo {
6    height: 40vmin;
7    pointer-events: none;
8  }
9
10 @media (prefers-reduced-motion: no-preference) {
11   .App-logo {
12     animation: App-logo-spin infinite 20s linear;
13   }
14 }
15
16 .App-header {
17   background-color: #282c34;
18   min-height: 100vh;
19   display: flex;
20   flex-direction: column;
21   align-items: center;
22   justify-content: center;
23   font-size: calc(10px + 2vmin);
24   color: white;
25 }
26
27 .App-link {
28   color: #61dafb;
29 }
30
31 @keyframes App-logo-spin {
32   from {
33     transform: rotate(0deg);
34   }
35   to {
36     transform: rotate(360deg);
37   }
38 }
39
```

```
JS App.js      JS index.js  X
reactapp > src > JS index.js > ...
1 import React from 'react';
2 import ReactDOM from 'react-dom/client';
3 import './index.css';
4 import App from './App';
5 import reportWebVitals from './reportWebVitals';
6
7 const root = ReactDOM.createRoot(document.getElementById('root'));
8 root.render(
9   <React.StrictMode>
10  |   <App />
11  | </React.StrictMode>
12 );
13
14 // If you want to start measuring performance in your app, pass a function
15 // to log results (for example: reportWebVitals(console.log))
16 // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
17 reportWebVitals();
18
```

```
JS App.js      # index.css  X
reactapp > src > # index.css > body
1 body {
2   margin: 0;
3   font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
4   |   'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
5   |   sans-serif;
6   -webkit-font-smoothing: antialiased;
7   -moz-osx-font-smoothing: grayscale;
8 }
9
10 code {
11   font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
12   |   monospace;
13 }
14
```

Fix the code:

Stopwatch.jsx × JS App.test.js JS App.js

```
reactapp > src > components > Stopwatch.jsx > Stopwatch > formatTime
```

```
1 import React from "react";
2 export default class Stopwatch extends React.Component{
3     constructor(props){
4         super(props);
5         this.state={
6             elapsedMs:0,
7             isRunning:false,
8             laps:[],
9         };
10        this.intervalId=null;
11    }
12    componentWillUnmount(){
13        if(this.intervalId) clearInterval(this.intervalId);
14    }
15    formatTime=(ms)=>{
16        const totalCs=Math.floor(ms/10);
17        const minutes=Math.floor(totalCs/600);
18        const seconds=Math.floor((totalCs%600)/100);
19        const cs=totalCs%100;
20        const mm=String(minutes).padStart(2,'0');
21        const ss=String(seconds).padStart(2,'0');
22        const cc=String(cs).padStart(2,'0');
23        return `${mm}:${ss}.${cc}`;
24    };
25    start=()=>{
26        if(this.state.isRunning) return;
27        this.setState({isRunning: true});
28        const startAt=performance.now()-this.state.elapsedMs;
29        this.intervalId=setInterval(()=>{
30            const now=performance.now();
31            this.setState({elapsedMs:now-startAt});
32        },10);
33    };
34    stop=()=>{
35        if(!this.state.isRunning) return;
36        this.setState({isRunning:false});
37        if(this.intervalId){
38            clearInterval(this.intervalId);
39            this.intervalId=null;
40        }
41    };
42    reset=()=>{
43        this.stop();
44        this.setState({elapsedMs:0,laps:[]});
45    };
}
```

```

46     addLap={()=>{
47       if(!this.state.isRunning) return;
48       this.setState((prev)=>({laps:[...prev.laps, prev.elapsedMs]}));
49     };
50   }
51   render(){
52     const {elapsedMs, isRunning, laps}=this.state;
53     return (
54       <div className="d-flex justify-content-center align-items-start py-5">
55         <div className="border rounded p-4 text-center" style={{width:320}}>
56           <h3 className="mb-3">Stopwatch</h3>
57           <div className="display-6 mb-3" aria-label='time'>
58             {this.formatTime(elapsedMs)}
59           </div>
60           <div className="d-flex gap-2 justify-content-center mb-3">
61             <button className="btn btn-success" onClick={this.start} disabled={isRunning}>Start</button>
62             <button className="btn btn-danger" onClick={this.stop} disabled={!isRunning}>Stop</button>
63             <button className="btn btn-info text-white" onClick={this.reset} disabled={isRunning}>Reset</button>
64             <button className="btn btn-warning fw-semibold mb-3" onClick={this.addLap} disabled={!isRunning}>Add Lap</button>
65             <ul className="list-unstyled m-0">
66               {
67                 laps.map((ms,idx)=>(
68                   <li key={idx} className="text-muted small">{this.formatTime(ms)}</li>
69                 ))
70               </ul>
71             </div>
72           </div>
73         );
74       );
75     }
76   }

```

Stopwatch.jsx JS App.test.js JS App.js X

reactapp > src > JS App.js > App

```

1 import logo from './logo.svg';
2 import './App.css';
3 import Stopwatch from './components/Stopwatch';
4 export default function App(){
5   return <Stopwatch/>;
6 }

```

Practice at home:

js App.js X js App.test.js

reactapp > src > js App.js > App

```
1 import React,{useState, useEffect} from 'react';
2
3 function App(){
4     const[l,setL]=useState(true);
5     const[t,setT]=useState([
6         {id:1,t:'Complete React Assignment',p:'High',c:false},
7         {id:2,t:'Review Bootstrap Documentation',p:'Medium',c:true},
8         {id:3,t:'Debug Lifecycle Methods',p:'High',c:false},
9         {id:4,t:'Setup React Dev Tools',p:'Low',c:false},
10    ]);
11    const[f,setF]=useState('All');
12    const[nt,setNT]=useState('');
13    const[np,setNP]=useState('High');
14    const[sm,setSM]=useState(false);
15
16    useEffect(()=>{
17        const task=setTimeout(()=>setL(false),2000);
18        return ()=>clearTimeout(task);
19    },[]);
20    if(l) return <div className='spinner-border'></div>;
21    let fd=t;
22    if(f!=='All'){
23        if(f==='Completed') fd=t.filter(task=>task.c);
24        else fd=t.filter(task=>task.p==f);
25    }
26    const addTask=>{
27        if(!nt) return;
28        setT([...t,{id:Date.now(),t:nt,p:np,c:false}]);
29        setNT('');
30        setNP('High');
31        setSM(false);
32    }
33    const deleteTask=id=>setT(t.filter(task=>task.id!=id));
34    const toggleComplete=id=>setT(t.map(task=>(task.id==id?{...task,c:!task.c}:t)));
35
36    return (
37        <div>
38            <h1>Task Manager</h1>
39            <nav className='navbar'>Navbar</nav>
40            <div>
41                {'All','High','Medium','Low','Completed'].map(fr=>(
42                    <button key={fr} onClick={()=>setF(fr)}>{fr}</button>
43                )));
44            </div>
45            <button onClick={()=>setSM(true)}>Add New Task</button>
46            {sm && (
47                <div>
48                    <input placeholder='Enter task title...' value={nt} onChange={ev=>setNT(ev.target.value)}/>
```

```
49
50     <label>Priority
51         <select value={np} onChange={(ev)=>setNP(ev.target.value)}>
52             <option>High</option>
53             <option>Medium</option>
54             <option>Low</option>
55         </select>
56     </label>
57     <button onClick={addTask}>Add Task</button>
58 </div>
59 )
60 <ul>
61     {fd.map(task=>(
62         <li key={task.id}>
63             <input type='checkbox' aria-label={task.t} checked={task.c} onChange={()=>toggleComplete(task.id)}/>
64             {task.t}
65             <span className={`badge ${task.p==='High'?`bg-primary`:task.p==='Medium'?`bg-warning`:`bg-secondary`} `}>{task.p}</span>
66             <button onClick={()=>deleteTask(task.id)}>Delete</button>
67         </li>
68     )));
69 </ul>
70 </div>
71 )
72 export default App;
```

Challenge yourself:

The screenshot shows a code editor with two tabs open: 'CropDashboard.jsx' and 'App.js'. The 'CropDashboard.jsx' tab is active, displaying the following code:

```
1 import React,{useState,useMemo} from 'react';
2 import CropList from './CropList';
3 export default function CropDashboard({crops=[]}){
4   const [query,setQuery]=useState('');
5   const filtered=useMemo(()=>{
6     const q=query.trim().toLowerCase();
7     if(!q) return crops;
8     return crops.filter((c)=>c.name.toLowerCase().includes(q));
9   },[query,crops]);
10  return (
11    <div className='container py-5 d-flex justify-content-center'>
12      <div className='card shadow-sm p-4' style={{maxWidth:720,width:'100%'}}>
13        <h2 className='fw-bold text-uppercase mb-2'>Crop Management System</h2>
14        <hr className='border-2 border-success opacity-100 mt-0' />
15        <div className='mb-3'>
16          <label className='form-label fw-semibold'>Search by Crop Name</label>
17          <input type='text' className='form-control border-success' placeholder='Search crops...' value={query} onChange={(e)=>setQuery(e.target.value)} />
18        </div>
19        <CropList crops={filtered}/>
20      </div>
21    </div>
22  );
23}
```

The screenshot shows a code editor with two tabs open: 'CropDashboard.jsx' and 'App.js'. The 'CropList.jsx' tab is active, displaying the following code:

```
reactapp > src > Components > CropList.jsx > CropList
1 import React from 'react';
2 export default function CropList({crops=[]}){
3   if(crops.length==0){
4     return <div className='text-muted'>No crops found.</div>;
5   }
6   return(
7     <ul className='List-unstyled m-0'>
8       {crops.map((c)=>(
9         <li key={c.name} className='d-flex justify-content-between py-2 border-bottom'>
10           <div className='d-flex align-items-center'>
11             <span className='text-success me-2' aria-hidden='true'>{'{'}</span>
12             <span className='fw-semibold'>{c.name}</span>
13           </div>
14           <div className='text-muted small'>-Type: {c.type} - Status: {c.status}</div>
15         </li>
16       )));
17     </ul>
18   );
19}
```

CropDashboard.jsx × JS App.js × CropList.jsx

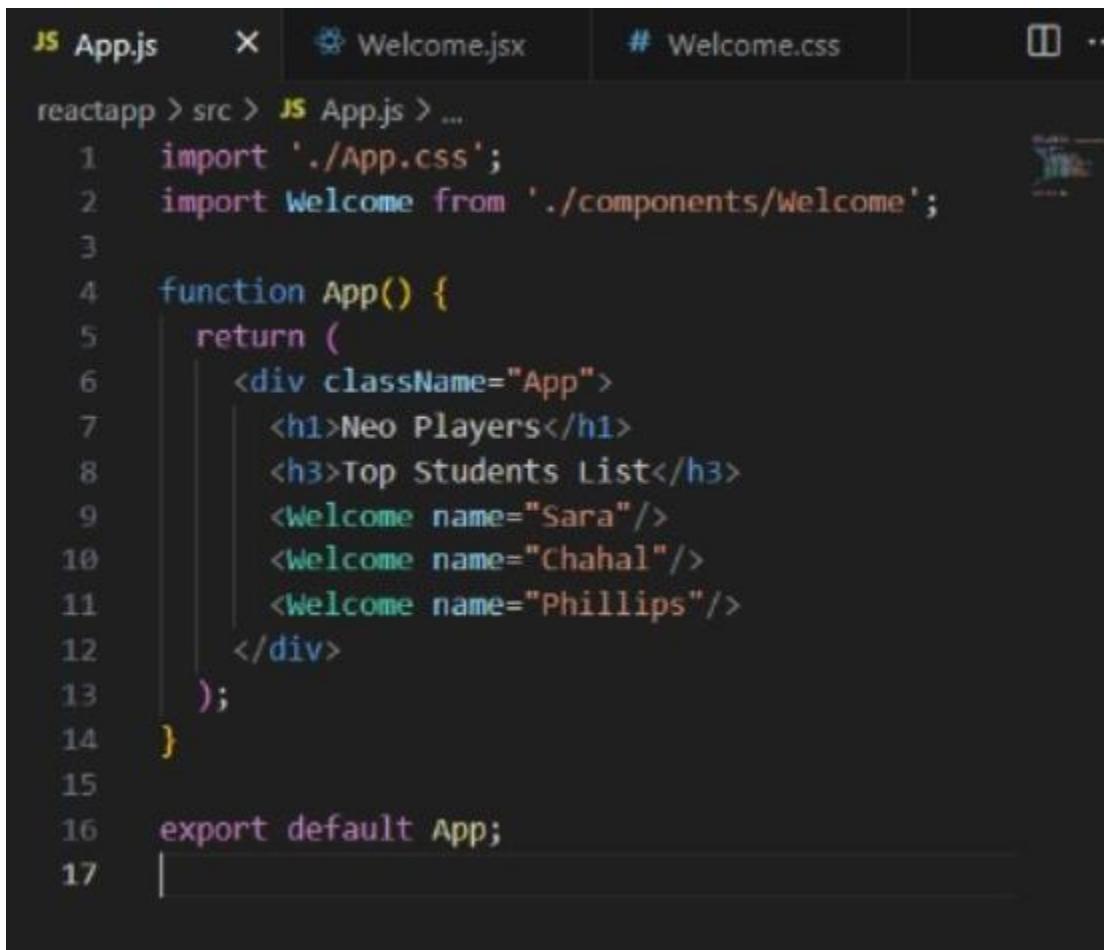
reactapp > src > JS App.js > ...

```
1 import './App.css';
2 import React from 'react';
3 import CropDashboard from './Components/CropDashboard';
4 const initialCrops=[
5   {name:'Tomato', type:'Vegetables',status:'Growing'},
6   {name:'Apple',type:'Fruits',status:'Planted'},
7   {name:'Wheat',type:'Grains',status:'Harvested'},
8   {name:'Carrot',type:'Vegetables',status:'Growing'},
9   {name:'Banana',type:'Fruits',status:'Growing'}
10 ];
11 export default function App(){
12   return <CropDashboard crops={initialCrops}/>;
13 }
14 
```

Web Programming

Week 9

Skill builder



The screenshot shows a code editor window with the following details:

- File tabs:** App.js (active), Welcome.jsx, Welcome.css.
- Project structure:** reactapp > src > App.js > ...
- Code content:**

```
reactapp > src > JS App.js > ...
1 import './App.css';
2 import Welcome from './components/Welcome';
3
4 function App() {
5     return (
6         <div className="App">
7             <h1>Neo Players</h1>
8             <h3>Top Students List</h3>
9             <Welcome name="Sara"/>
10            <Welcome name="Chahal"/>
11            <Welcome name="Phillips"/>
12        </div>
13    );
14 }
15
16 export default App;
17 |
```

⚙️ Welcome.jsx ✎

reactapp > src > components > ⚙️ Welcome.jsx > [e] default

```
1 import React from 'react';
2 import './Welcome.css';
3
4 function Welcome({name}) {
5     return(
6         <div className='welcome'>
7             <h2>{name}</h2>
8         </div>
9     );
10}
11
12 export default Welcome;
```

Welcome.css ✎

reactapp > src > components > # Welcome.css > 📁 .welcome h2

```
1 .welcome h2 {
2     margin: 5px 0;
3     color: □#333;
4 }
```

Challenge yourself

```
reactapp > src > components > JS LifecycleLogs.js > ...
1  import React from 'react';
2
3  const LifecycleLogs = ({logs,CL}) => {
4      return(
5          <div style={{marginTop:'30px'}}>
6              <h2>Lifecycle Logs</h2>
7              <button onClick={CL}>Clear Log</button>
8              <div style={{
9                  border:'1px solid #ccc',
10                 marginTop:'10px',
11                 padding: '10px',
12                 minHeight: '100px'
13             }}>
14                 {logs.length === 0? (
15                     <p>No logs yet</p>
16                 ):(
17                     logs.map((l,ind) => <p key={ind}>{l}</p>)
18                 )}
19                 </div>
20             </div>
21         );
22     };
23
24     export default LifecycleLogs;
```

```
JS CityFilter.js X JS LifecycleLogs.js JS WeatherCard.js JS App.js M
reactapp > src > components > JS CityFilter.js > [e] default
1 import React,{useEffect} from 'react';
2
3 const CityFilter = ({ currentFilter, onFilterChange, addLog, renderCount }) => {
4   useEffect(() => { addLog('CityFilter Mounted') }, [addLog]);
5   return (
6     <div style={{marginBottom:'20px'}}>
7       <h2>Filter by Condition</h2>
8       <select role='combobox' value={currentFilter} onChange={(e) => onFilterChange(e.target.value)}>
9         <option value='all'>All Conditions</option>
10        <option value='Sunny'>Sunny</option>
11        <option value='Cloudy'>Cloudy</option>
12        <option value='Rainy'>Rainy</option>
13        <option value='Snowy'>Snowy</option>
14      </select>
15      <p>Render Count: {renderCount}</p>
16    </div>
17  );
18};
19
20 export default CityFilter;
```

```
JS CityFilter.js JS LifecycleLogs.js JS WeatherCard.js X JS App.js
reactapp > src > components > JS WeatherCard.js > [e] WeatherCard
1 import React, {useEffect} from 'react';
2
3 const WeatherCard = ({ city, onToggle, addLog }) => {
4   useEffect(() => { addLog(`WeatherCard Mounted: ${city.name}`), [addLog, city] });
5   return (
6     <div style={
7       {
8         border:'1px solid #ddd',
9         marginBottom:'10px',
10        padding:'10px',
11        borderRadius:'8px',
12        backgroundColor:'#f9f9f9'
13       }
14     }>
15       <h3>{city.name}</h3>
16       <p>{city.temp}°</p>
17       <span style={
18         {
19           background:'#ddd',
20           padding:'4px 8px',
21           borderRadius:'6px'
22         }
23       }>{city.condition}</span>
24       <div style={{marginTop:'10px'}}>
25         <button onClick={() => onToggle(city.id)}>{city.active==true?'Remove City':'Add City'}</button>
26       </div>
27       <div>
28         <p>{city.condition}</p>
29         <p>{city.condition}</p>
30       </div>
31     </div>
32   );
33 };
34
35 export default WeatherCard;
36
```

reactapp > src > **JS** App.js > App

```
1 import React, { useState, useEffect, useCallback } from 'react';
2 import CityFilter from './components/CityFilter';
3 import WeatherCard from './components/WeatherCard';
4 import LifecycleLogs from './components/LifecycleLogs';
5
6 function App() {
7   const [c, setC] = useState([
8     {id: 1, name:'New York', condition:'Sunny', temp:22, active:true}, {id: 2, name:'London', condition:'Cloudy', temp:15, active:true},
9     {id: 3, name:'Tokyo', condition:'Rainy', temp:18, active:true}, {id: 4, name:'Paris', condition:'Snowy', temp:20, active:true}
10   ]);
11
12   const [l, setL] = useState([]);
13   const [f, setF] = useState('all');
14   const [rc, setRC] = useState(1);
15   const AL = useCallback((m) => { setL((p) => [...p, `${new Date().toLocaleTimeString()} - ${m}`]); }, []);
16   const FC = (v) => { setF(v); AL(`Filter changed to: ${v}`) };
17   const TC = (id) => { setC((p) => p.map((ct)=>ct.id==id?...ct, active:!ct.active:ct)); AL(`City toggled: ${id}`) };
18   const CL = () => { setL(['Logs cleared']) };
19   useEffect(()=>{AL('App Mounted')}, [AL]);
20   useEffect(()=>{setRC((cnt)=>cnt+1)}, []);
21   const fc = f==='all'? c.filter((ct)=>ct.active): c.filter((ct)=>ct.active && ct.condition === f);
22   return (
23     <div style={{padding:'20px', fontFamily:'Arial'}}>
24       <h1>Weather Dashboard</h1>
25       <CityFilter currentFilter={f} onFilterChange={FC} addLog={AL} renderCount={rc}/>
26
27       <h2>Cities</h2>
28       {
29         fc.length > 0 ? [
30           fc.map((ct) => (
31             | <WeatherCard key={ct.id} city={ct} onToggle={TC} addLog={AL}/>
32           ))
33         ]:(
34           | <p>No cities to display</p>
35         )
36       }
37       <LifecycleLogs logs={l} CL={CL}/>
38     </div>
39   )
40 }
41 export default App;
```