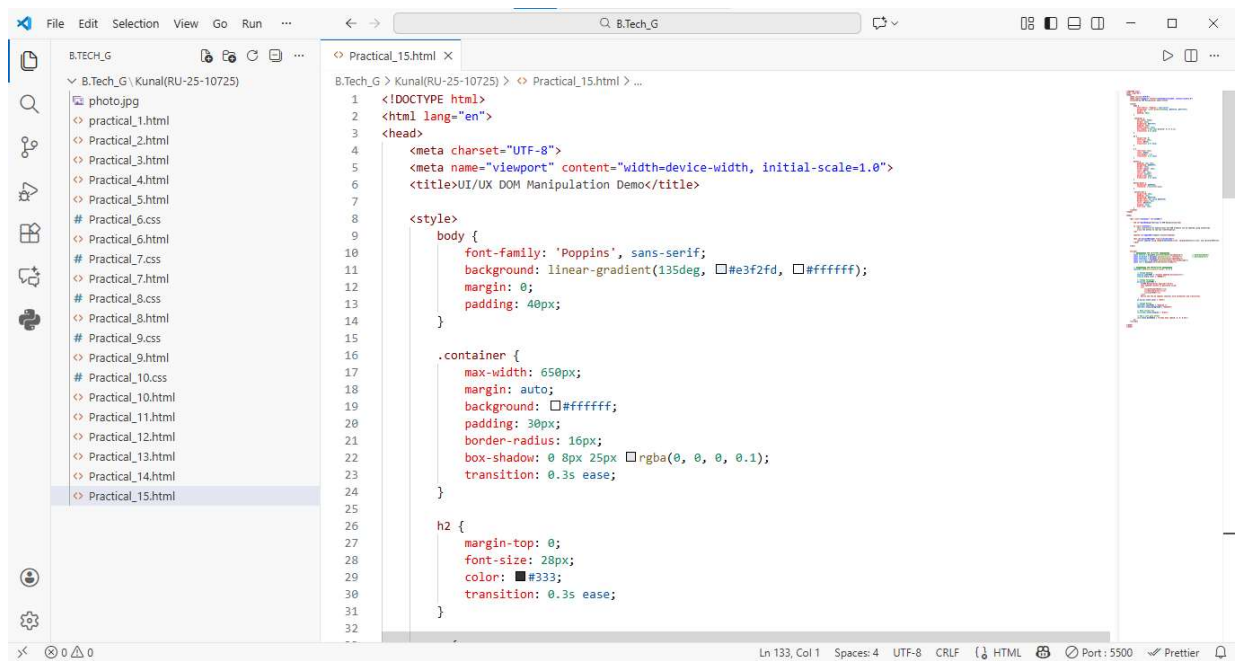
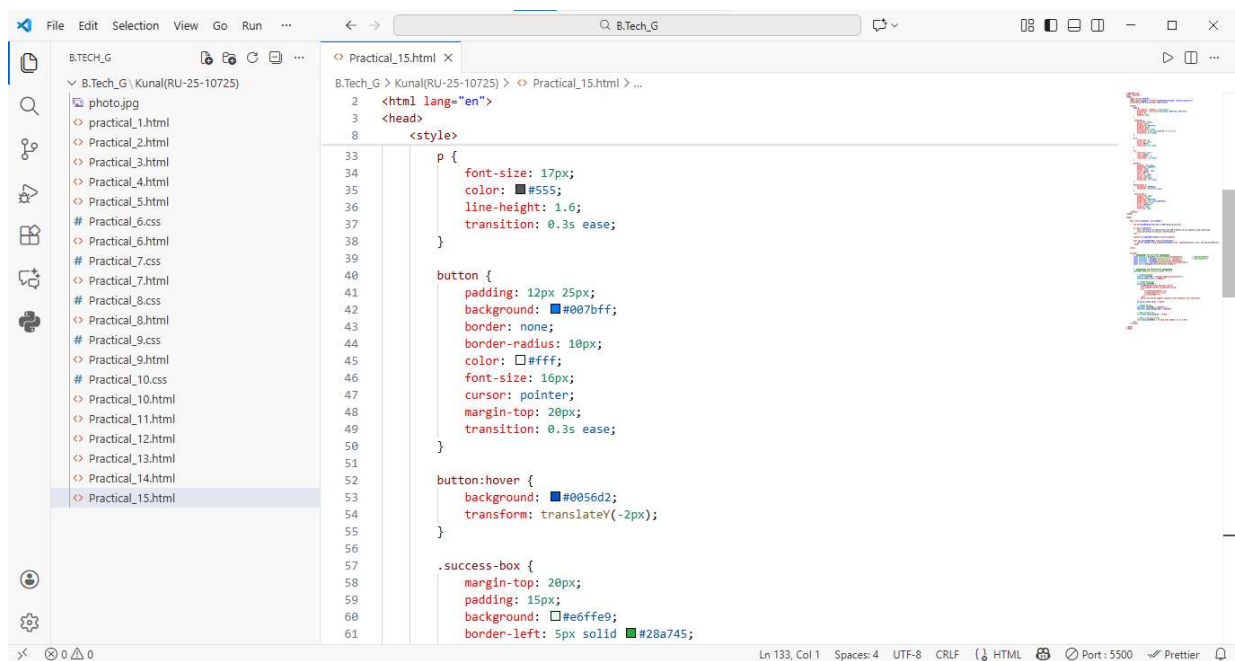


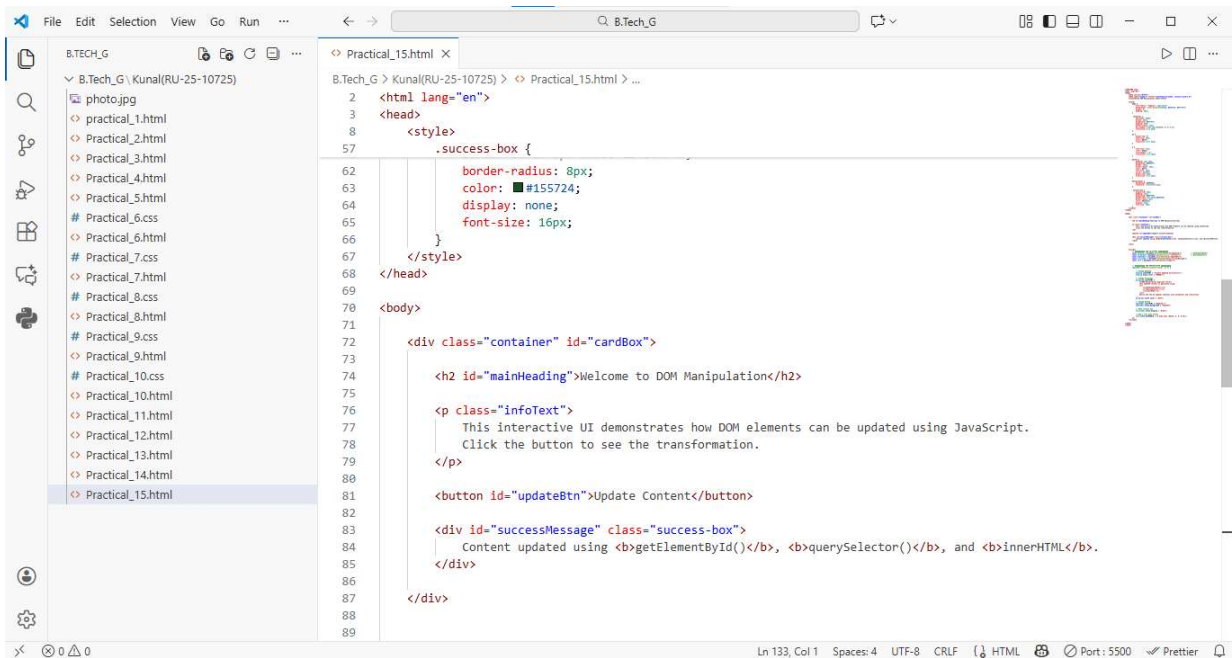
15. Demonstrate DOM manipulation using getElementById(), querySelector(), and innerHTML.



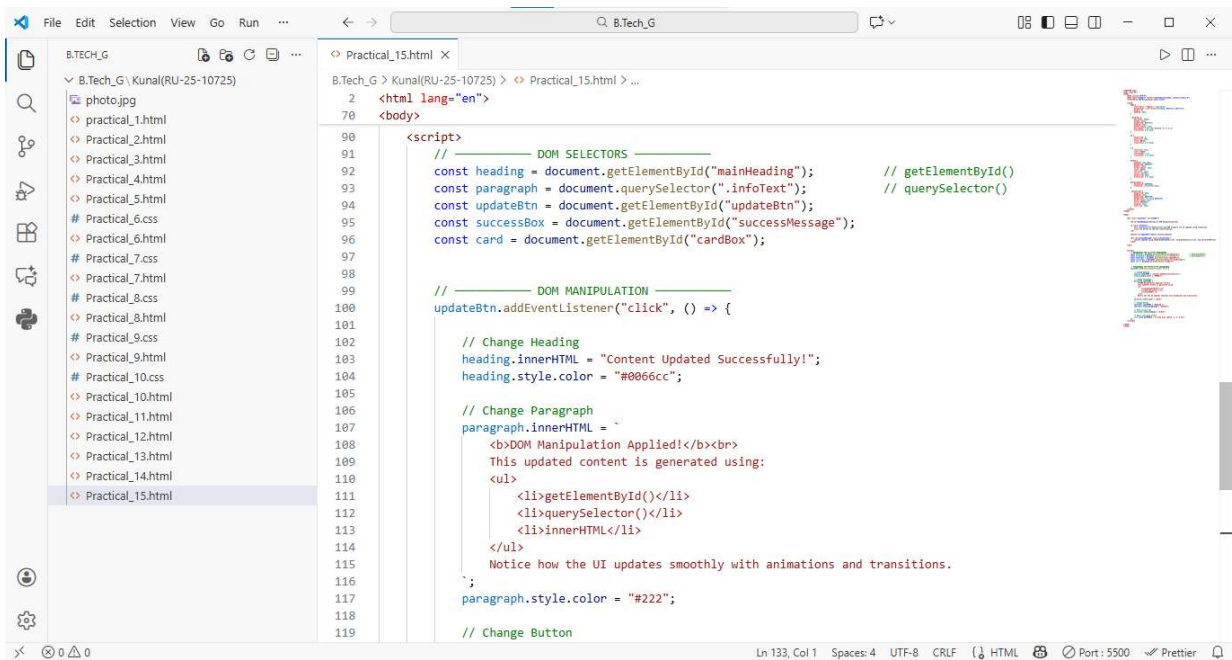
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>UI/UX DOM Manipulation Demo</title>
7
8   <style>
9     body {
10       font-family: 'Poppins', sans-serif;
11       background: linear-gradient(135deg, #e3f2fd, #ffffff);
12       margin: 0;
13       padding: 40px;
14     }
15
16     .container {
17       max-width: 650px;
18       margin: auto;
19       background: #ffffff;
20       padding: 30px;
21       border-radius: 16px;
22       box-shadow: 0 8px 25px #0000000a;
23       transition: 0.3s ease;
24     }
25
26     h2 {
27       margin-top: 0;
28       font-size: 28px;
29       color: #333;
30       transition: 0.3s ease;
31     }
32
```



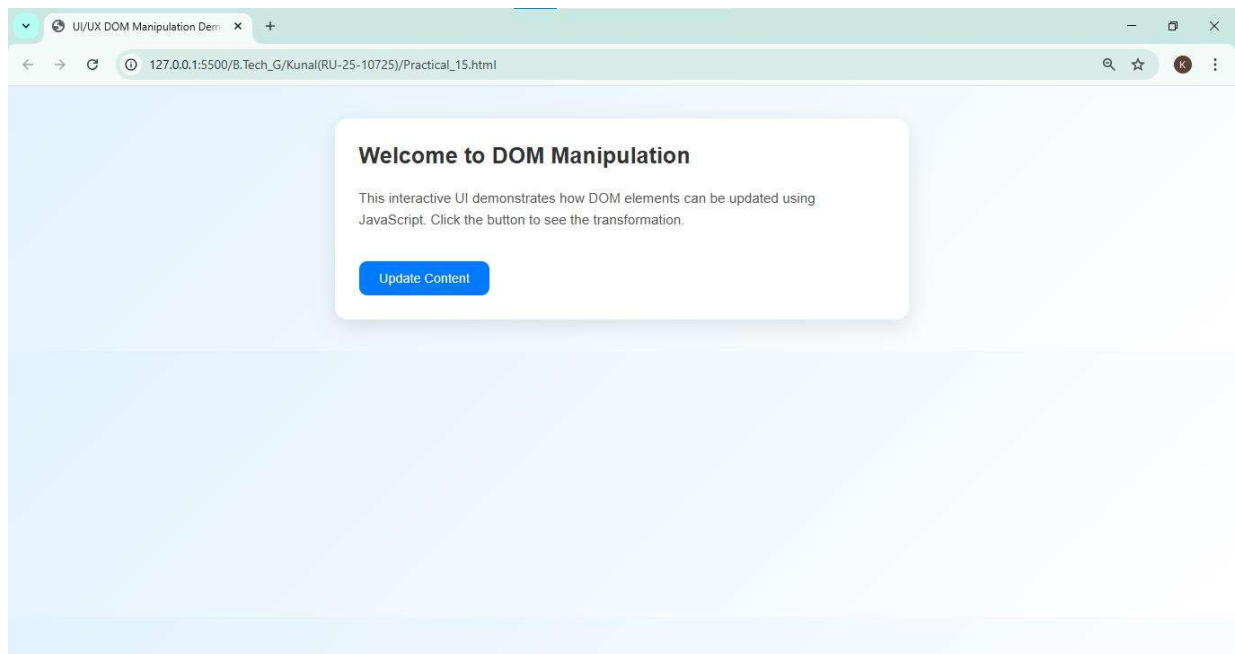
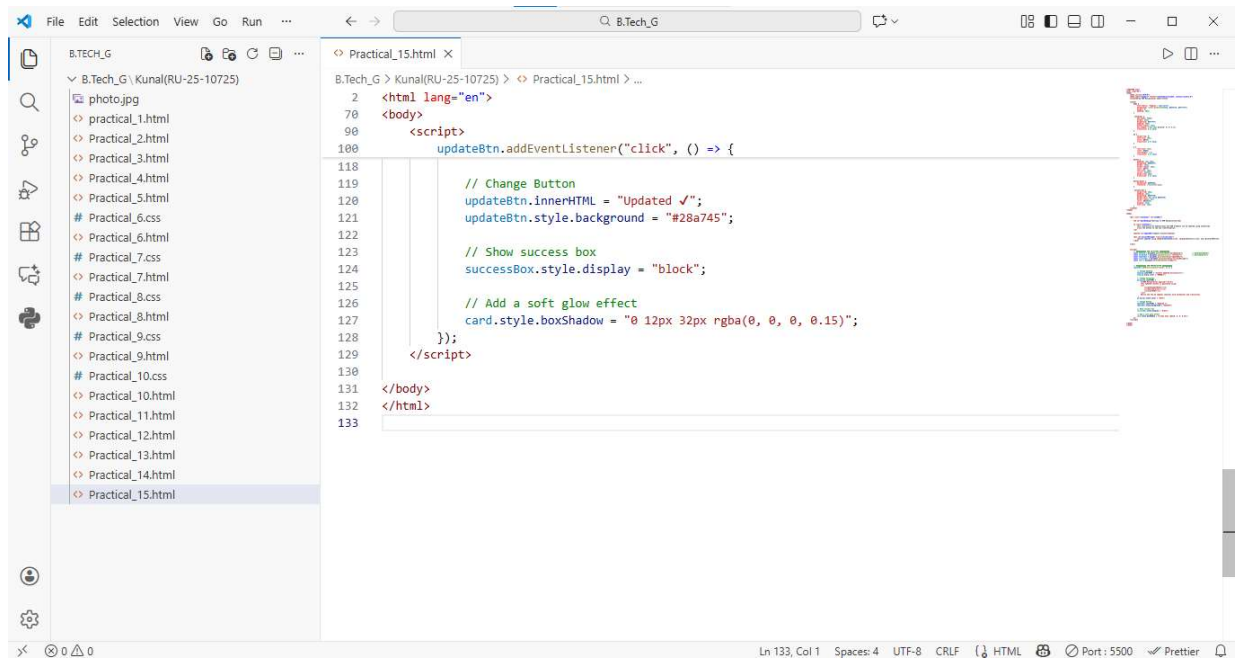
```
33 p {
34   font-size: 17px;
35   color: #555;
36   line-height: 1.6;
37   transition: 0.3s ease;
38 }
39
40 button {
41   padding: 12px 25px;
42   background: #007bff;
43   border: none;
44   border-radius: 10px;
45   color: #fff;
46   font-size: 16px;
47   cursor: pointer;
48   margin-top: 20px;
49   transition: 0.3s ease;
50 }
51
52 button:hover {
53   background: #0056b3;
54   transform: translateY(-2px);
55 }
56
57 .success-box {
58   margin-top: 20px;
59   padding: 15px;
60   background: #e6ffe9;
61   border-left: 5px solid #28a745;
```

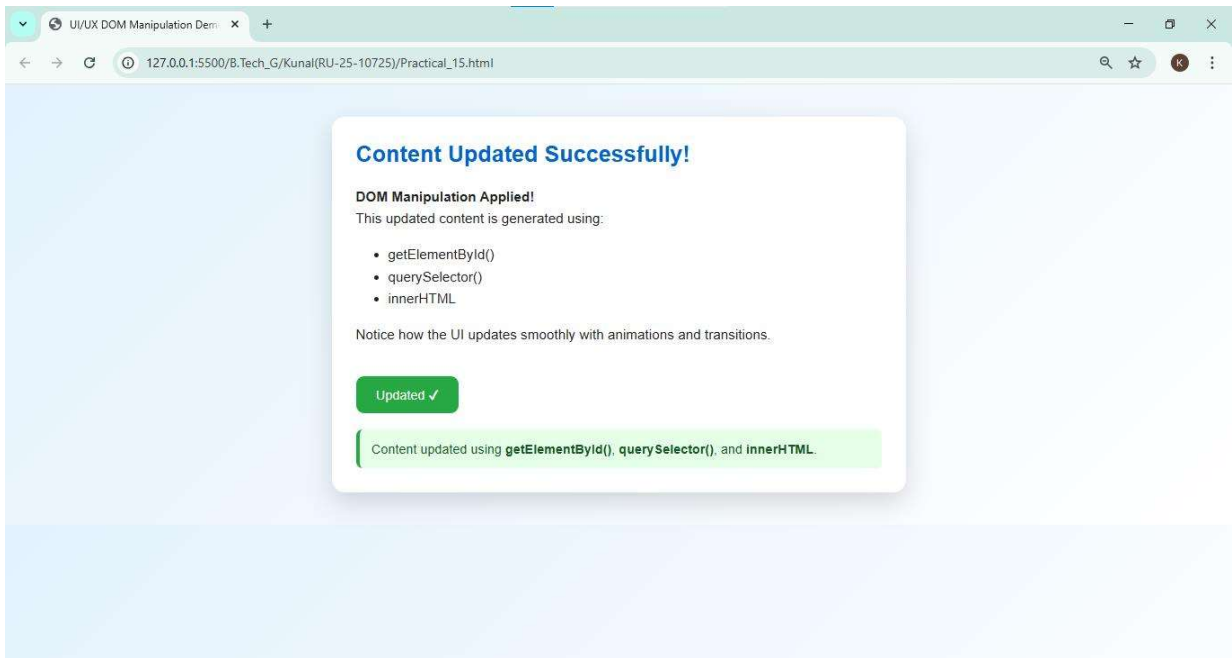


```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>DOM Manipulation</title>
6 <link rel="stylesheet" href="Practical_6.css">
7
8 <style>
9
10 .success-box {
11     border-radius: 8px;
12     color: #155724;
13     display: none;
14     font-size: 16px;
15 }
16
17 </style>
18 </head>
19
20 <body>
21
22 <div class="container" id="cardBox">
23
24 <h2 id="mainHeading">Welcome to DOM Manipulation</h2>
25
26 <p class="infoText">
27     This interactive UI demonstrates how DOM elements can be updated using JavaScript.
28     Click the button to see the transformation.
29 </p>
30
31 <button id="updateBtn">Update Content</button>
32
33 <div id="successMessage" class="success-box">
34     Content updated using <b>getElementById()</b>, <b>querySelector()</b>, and <b>innerHTML</b>.
35 </div>
36
37 </div>
38
39 </body>
40 </html>
```

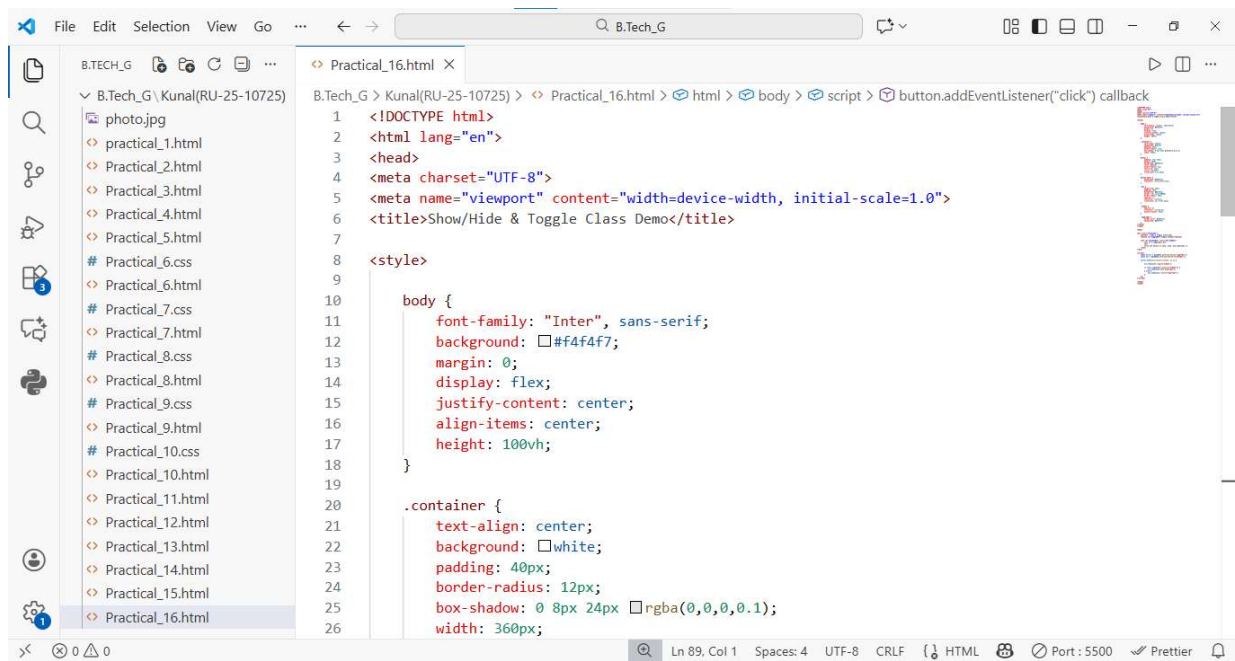


```
1 <script>
2
3 // ----- DOM SELECTORS -----
4 const heading = document.getElementById("mainHeading"); // getElementById()
5 const paragraph = document.querySelector(".infoText"); // querySelector()
6 const updateBtn = document.getElementById("updateBtn");
7 const successBox = document.getElementById("successMessage");
8 const card = document.getElementById("cardBox");
9
10 // ----- DOM MANIPULATION -----
11 updateBtn.addEventListener("click", () => {
12
13     // Change Heading
14     heading.innerHTML = "Content Updated Successfully!";
15     heading.style.color = "#0066cc";
16
17     // Change Paragraph
18     paragraph.innerHTML = `
19         <b>DOM Manipulation Applied!</b><br>
20         This updated content is generated using:
21         <ul>
22             <li>getElementById()</li>
23             <li>querySelector()</li>
24             <li>innerHTML</li>
25         </ul>
26         Notice how the UI updates smoothly with animations and transitions.
27     `;
28     paragraph.style.color = "#222";
29
30     // Change Button
31
32 </script>
```



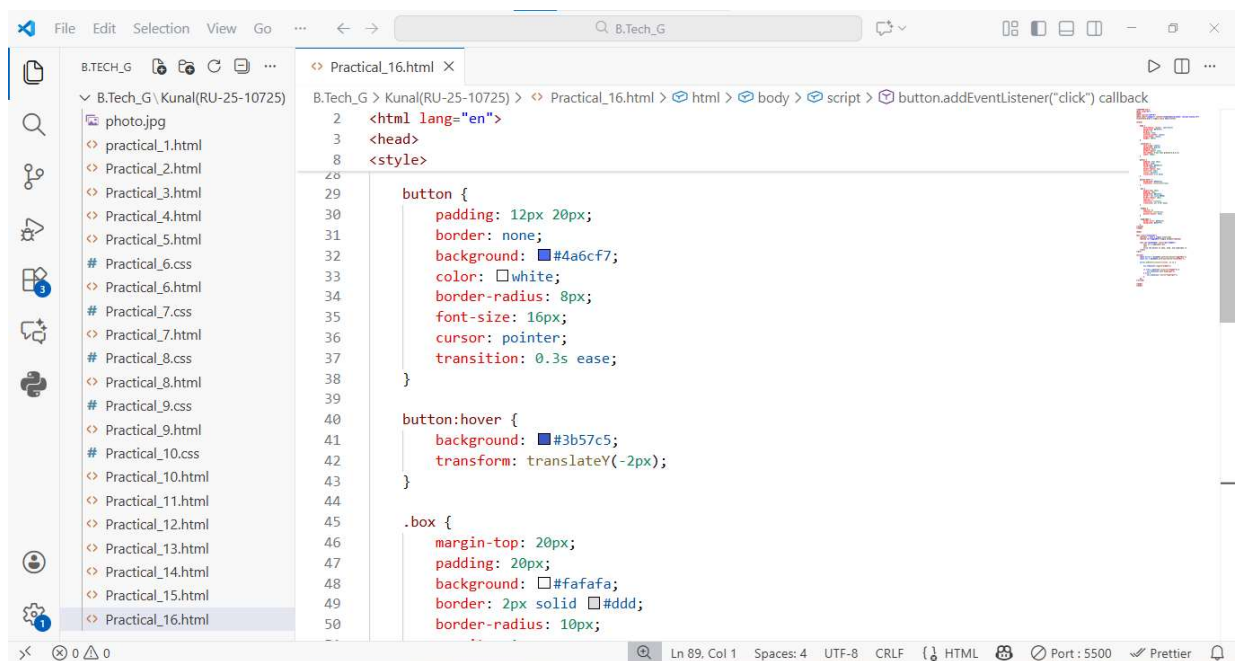


16. Show or hide HTML elements using JavaScript and toggle CSS classes dynamically.



The screenshot shows the VS Code editor with the file 'Practical_16.html' open. The left sidebar displays a file explorer with a project structure including 'photo.jpg' and various 'Practical_1.html' through 'Practical_16.html' files, along with corresponding CSS files. The main editor area shows the following code:

```
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>Show/Hide & Toggle Class Demo</title>
7
8 <style>
9
10   body {
11     font-family: "Inter", sans-serif;
12     background: #f4f4f7;
13     margin: 0;
14     display: flex;
15     justify-content: center;
16     align-items: center;
17     height: 100vh;
18   }
19
20   .container {
21     text-align: center;
22     background: white;
23     padding: 40px;
24     border-radius: 12px;
25     box-shadow: 0 8px 24px rgba(0,0,0,0.1);
26     width: 360px;
```

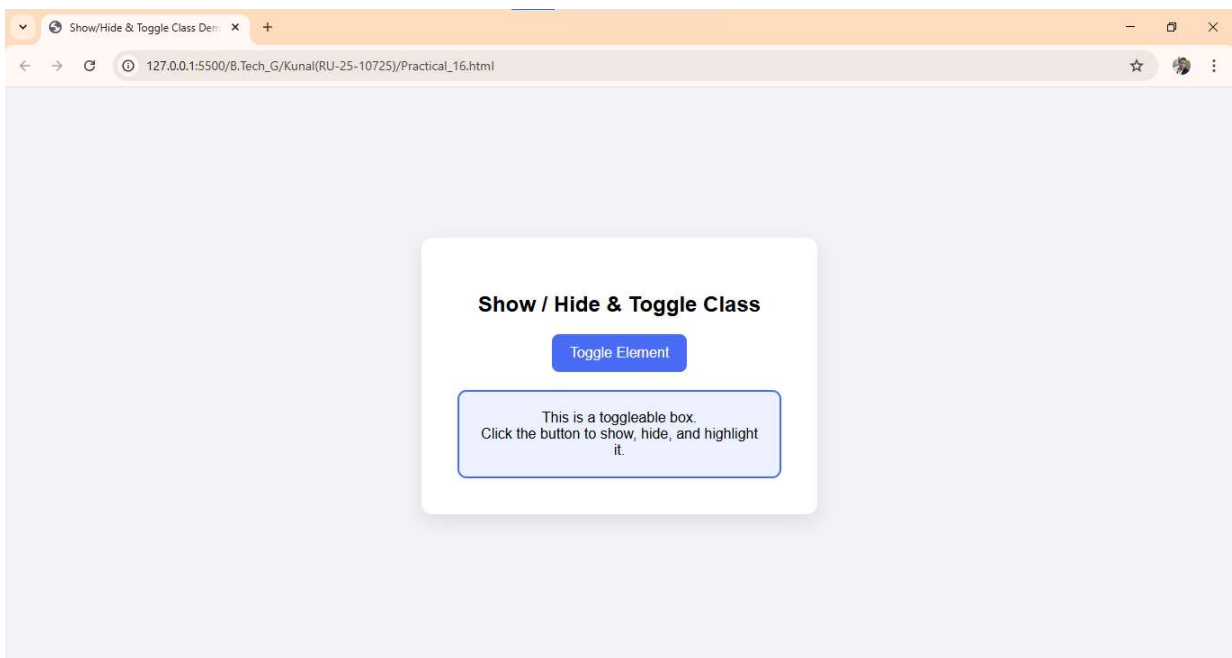
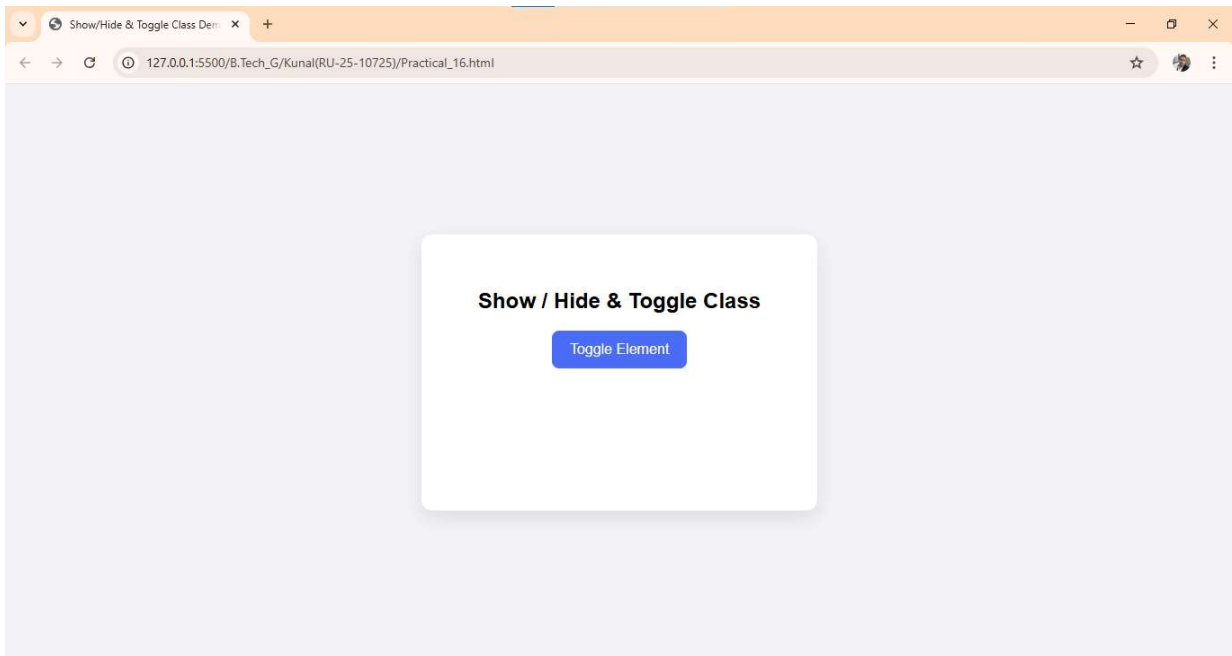


The screenshot shows the continuation of the CSS code in the VS Code editor for 'Practical_16.html'. The code defines styles for a button and a box:

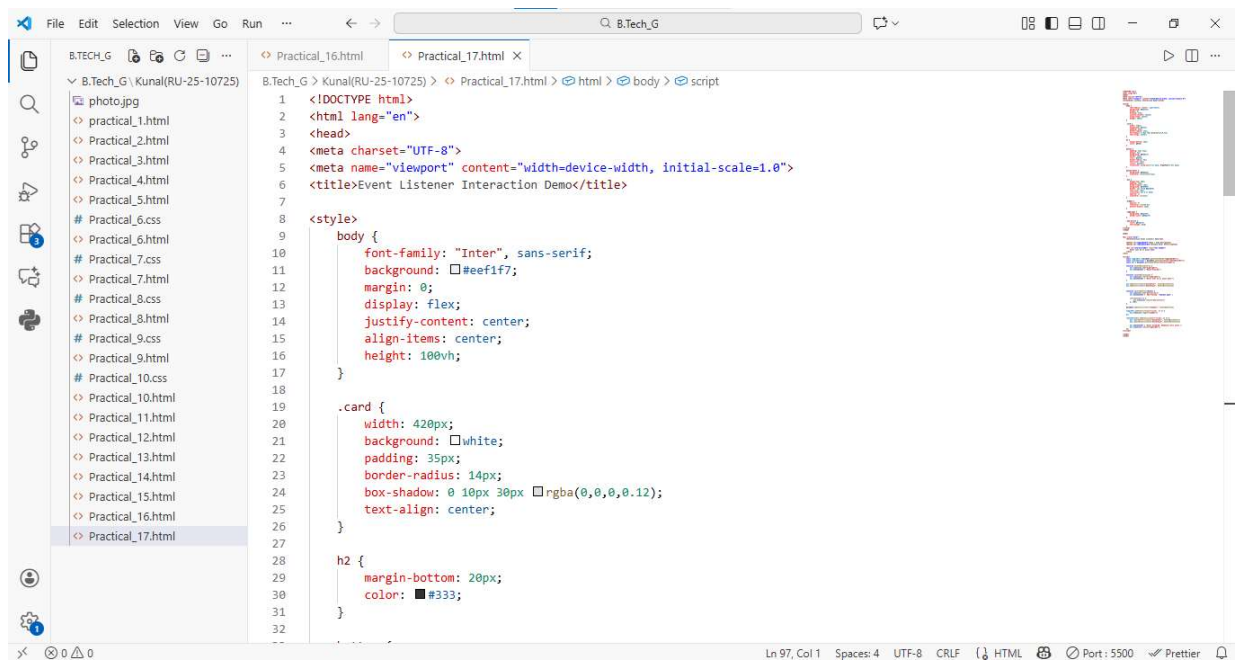
```
29   button {
30     padding: 12px 20px;
31     border: none;
32     background: #4a6cf7;
33     color: white;
34     border-radius: 8px;
35     font-size: 16px;
36     cursor: pointer;
37     transition: 0.3s ease;
38   }
39
40   button:hover {
41     background: #3b57c5;
42     transform: translateY(-2px);
43   }
44
45   .box {
46     margin-top: 20px;
47     padding: 20px;
48     background: #fafafa;
49     border: 2px solid #ddd;
50     border-radius: 10px;
```

```
2 <html lang="en">
3 <head>
8 <style>
55
56   .hidden {
57     opacity: 0;
58     transform: scale(0.9);
59     pointer-events: none;
60   }
61
62   .highlight {
63     border-color: #4a6cf7;
64     background: #eef1ff;
65   }
66 </style>
67 </head>
68
69 <body>
70
71 <div class="container">
72   <h2>Show / Hide & Toggle Class</h2>
73   <button id="toggleBtn">Toggle Element</button>
74
75   <div id="contentBox" class="box hidden">
76     This is a toggleable box.
77   <br>
```

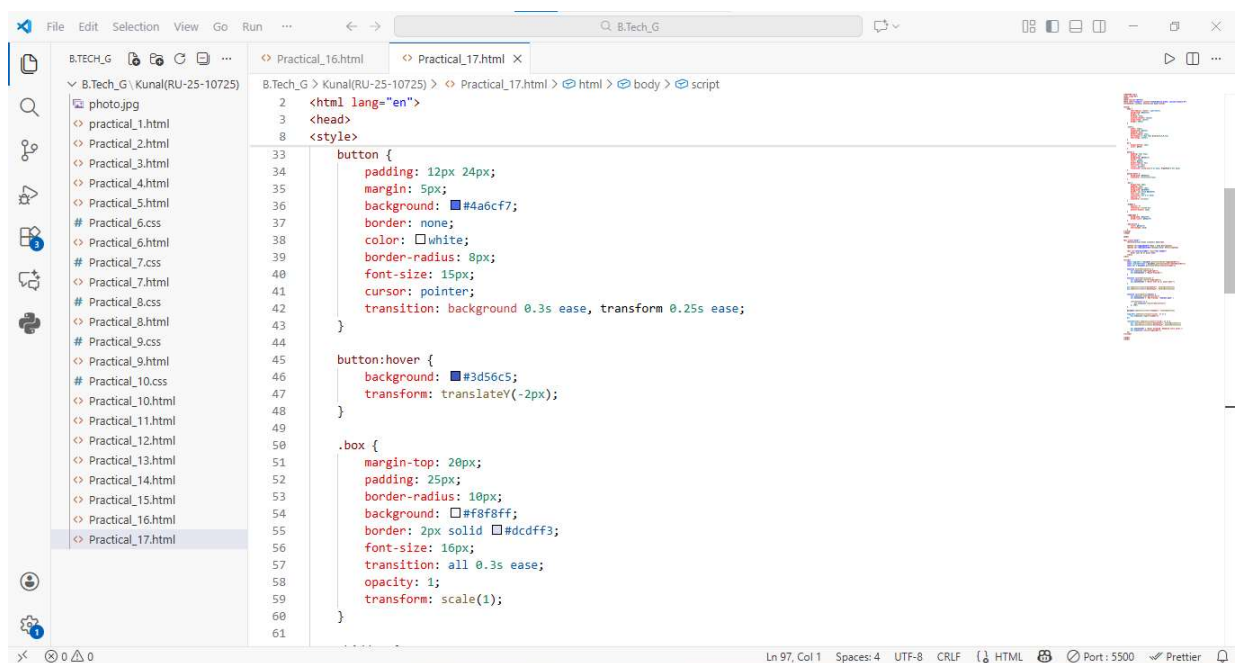
```
69 <body>
71 <div class="container">
79   </div>
80 </div>
81
82 <script>
83   const button = document.getElementById("toggleBtn");
84   const box = document.getElementById("contentBox");
85
86   button.addEventListener("click", () => {
87
88     box.classList.toggle("hidden");
89
90     if (!box.classList.contains("hidden")) {
91       box.classList.add("highlight");
92     } else {
93       box.classList.remove("highlight");
94     }
95   });
96 </script>
97
98 </body>
99 </html>
100
```



17. Add interactivity using event listeners (addEventListener, removeEventListener) for mouse/keyboard events.



```
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>Event Listener Interaction Demo</title>
7
8 <style>
9   body {
10     font-family: "Inter", sans-serif;
11     background: #eef1f7;
12     margin: 0;
13     display: flex;
14     justify-content: center;
15     align-items: center;
16     height: 100vh;
17   }
18
19   .card {
20     width: 420px;
21     background: #fff;
22     padding: 35px;
23     border-radius: 14px;
24     box-shadow: 0 10px 30px #0000000a;
25     text-align: center;
26   }
27
28   h2 {
29     margin-bottom: 20px;
30     color: #333;
31   }
32
```



```
33   button {
34     padding: 12px 24px;
35     margin: 5px;
36     background: #4a6cf7;
37     border: none;
38     color: #fff;
39     border-radius: 8px;
40     font-size: 15px;
41     cursor: pointer;
42     transition: background 0.3s ease, transform 0.25s ease;
43   }
44
45   button:hover {
46     background: #3d56c5;
47     transform: translateY(-2px);
48   }
49
50   .box {
51     margin-top: 20px;
52     padding: 25px;
53     border-radius: 10px;
54     background: #f8f8ff;
55     border: 2px solid #dcdff3;
56     font-size: 16px;
57     transition: all 0.3s ease;
58     opacity: 1;
59     transform: scale(1);
60   }
61
```


File Edit Selection View Go Run ... B.Tech_G

photo.jpg
practical_1.html
practical_2.html
practical_3.html
practical_4.html
practical_5.html
Practical_6.css
practical_6.html
Practical_7.css
practical_7.html
Practical_8.css
practical_8.html
Practical_9.css
practical_9.html
Practical_10.css
practical_10.html
practical_11.html
practical_12.html
practical_13.html
practical_14.html
practical_15.html
practical_16.html
practical_17.html

Practical_16.html Practical_17.html X

B.Tech_G > Kunal(RU-25-10725) > Practical_17.html > html > body > script

```
2 <html lang="en">
3 <head>
8 <style>
62 .hidden {
63   opacity: 0;
64   transform: scale(0.9);
65   pointer-events: none;
66 }
67
68 .highlight {
69   background: #eef2ff;
70   border-color: #4a6cf7;
71 }
72
73 .key-active {
74   color: #4a6cf7;
75   font-weight: bold;
76 }
77 </style>
78 </head>
79
80 <body>
81
82 <div class="card">
83   <h2>Interactive Event Listeners Demo</h2>
84
85   <button id="toggleBoxBtn">Show / Hide Box</button>
86   <button id="removeHoverBtn">Disable Hover Effect</button>
87
88   <div id="interactiveBox" class="box hidden">
89     Hover over me or press keys!
90   </div>
```

Ln 97, Col 1 Spaces: 4 UTF-8 CRLF HTML Port: 5500 Prettier

File Edit Selection View Go Run ... B.Tech_G

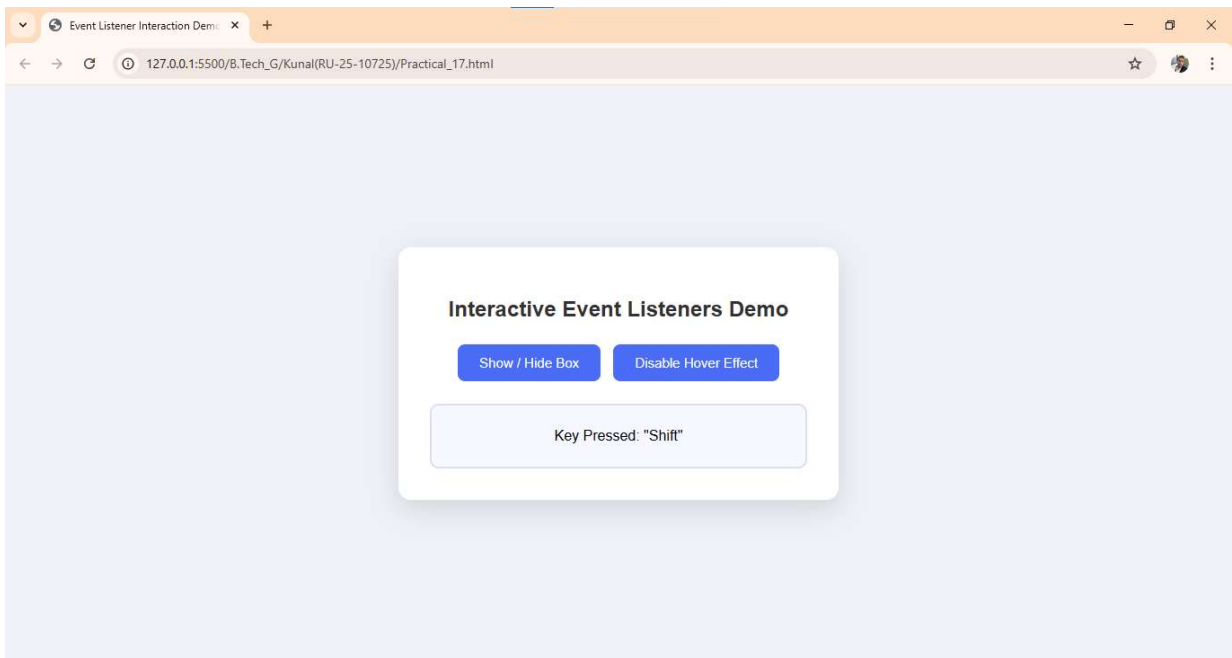
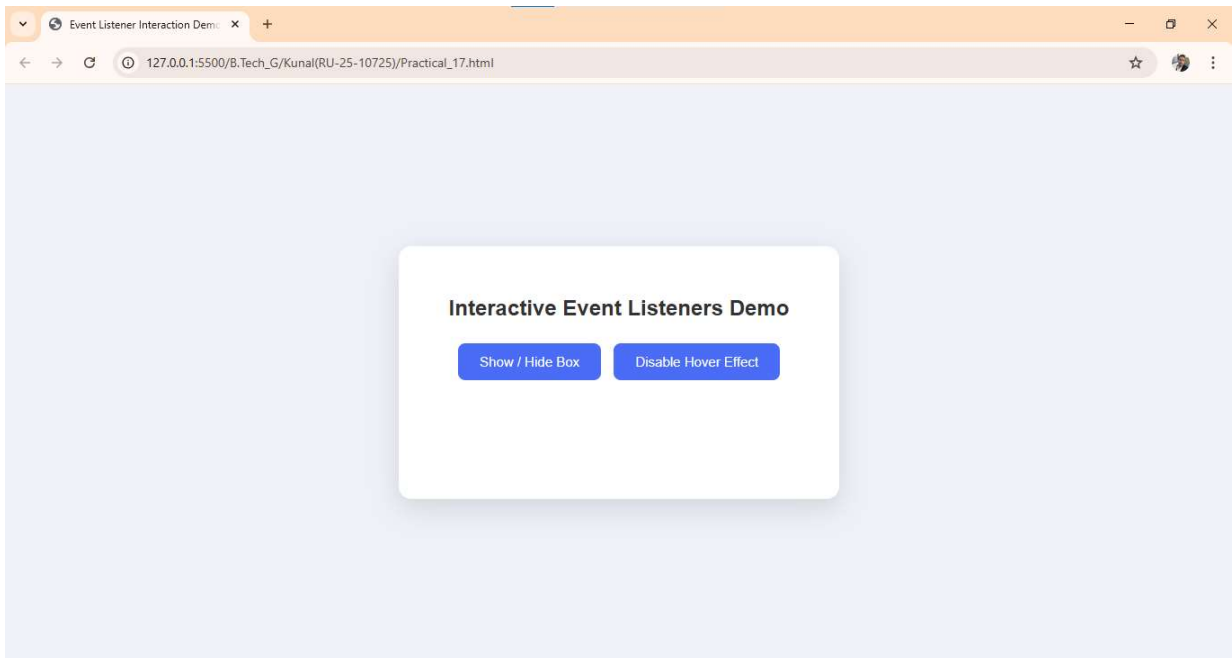
photo.jpg
practical_1.html
practical_2.html
practical_3.html
practical_4.html
practical_5.html
Practical_6.css
practical_6.html
Practical_7.css
practical_7.html
Practical_8.css
practical_8.html
Practical_9.css
practical_9.html
Practical_10.css
practical_10.html
practical_11.html
practical_12.html
practical_13.html
practical_14.html
practical_15.html
practical_16.html
practical_17.html

Practical_16.html Practical_17.html X

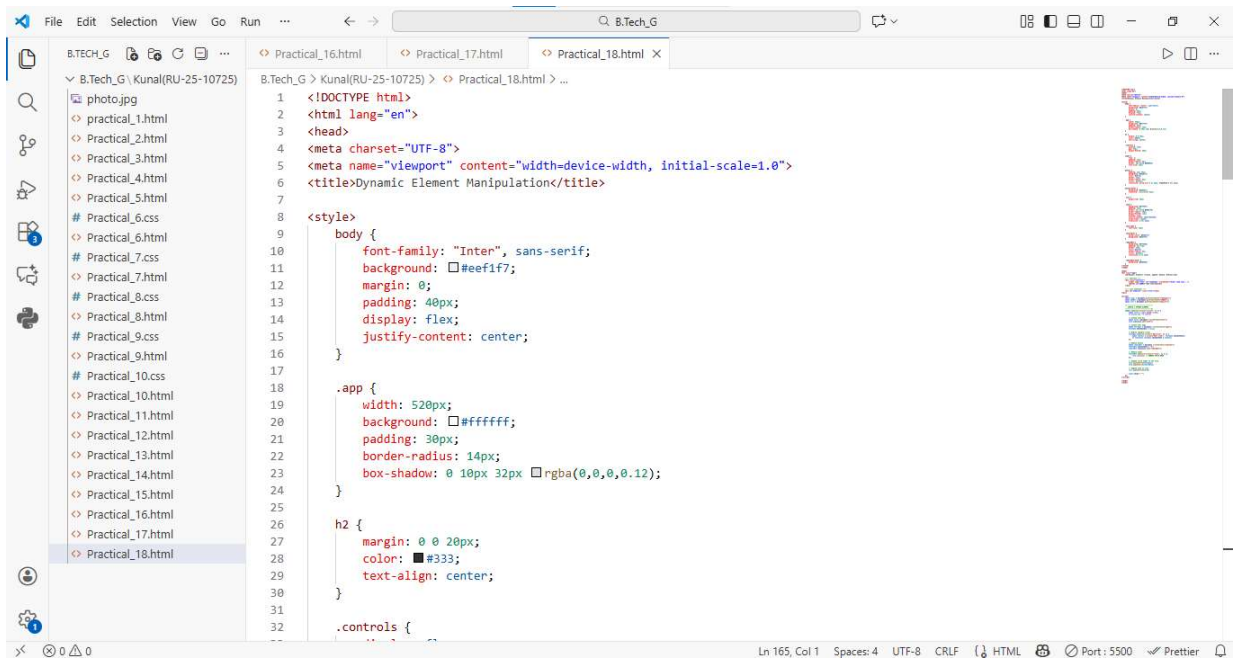
B.Tech_G > Kunal(RU-25-10725) > Practical_17.html > html > body > script

```
2 <html lang="en">
88 <body>
93 <script>
112 function handleKeyPress(event) {
113   box.classList.add("key-active");
114   box.textContent = "Key Pressed: "${event.key}";
115
116   setTimeout(() => {
117     box.classList.remove("key-active");
118   }, 300);
119
120 document.addEventListener("keydown", handleKeyPress);
121
122 toggleBtn.addEventListener("click", () => {
123   box.classList.toggle("hidden");
124 });
125
126 removeHoverBtn.addEventListener("click", () => {
127   box.removeEventListener("mouseenter", handleMouseEnter);
128   box.removeEventListener("mouseleave", handleMouseLeave);
129
130   box.textContent = "Hover disabled. Keyboard still works.";
131   box.classList.remove("highlight");
132 });
133 </script>
134
135 </body>
136 </html>
```

Ln 97, Col 1 Spaces: 4 UTF-8 CRLF HTML Port: 5500 Prettier



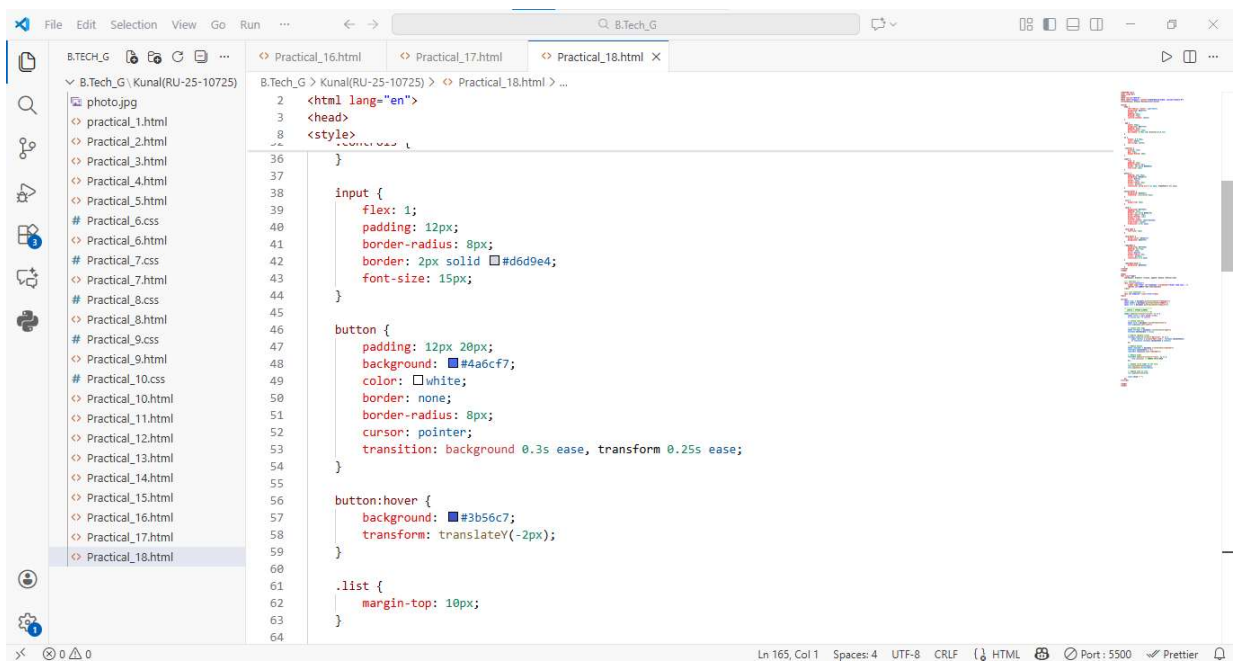
18. Create and manipulate elements using JavaScript: append, remove, or modify child nodes.



The screenshot shows the VS Code editor with the file 'Practical_18.html' open. The editor displays the following code:

```
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>Dynamic Element Manipulation</title>
7
8 <style>
9   body {
10     font-family: "Inter", sans-serif;
11     background: #eef1f7;
12     margin: 0;
13     padding: 40px;
14     display: flex;
15     justify-content: center;
16   }
17
18   .app {
19     width: 520px;
20     background: #ffffff;
21     padding: 30px;
22     border-radius: 14px;
23     box-shadow: 0 10px 32px rgba(0,0,0,0.12);
24   }
25
26   h2 {
27     margin: 0 0 20px;
28     color: #333;
29     text-align: center;
30   }
31
32   .controls {
```

The status bar at the bottom indicates 'Ln 165, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', 'HTML', 'Port: 5500', and 'Prettier'.



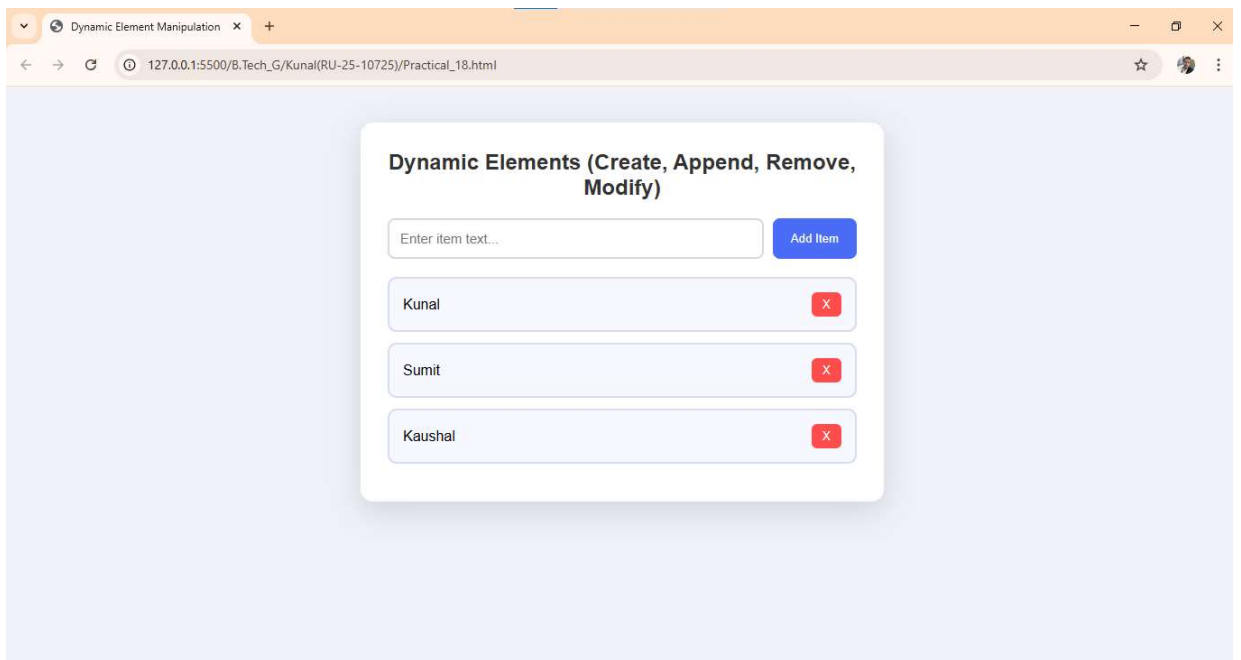
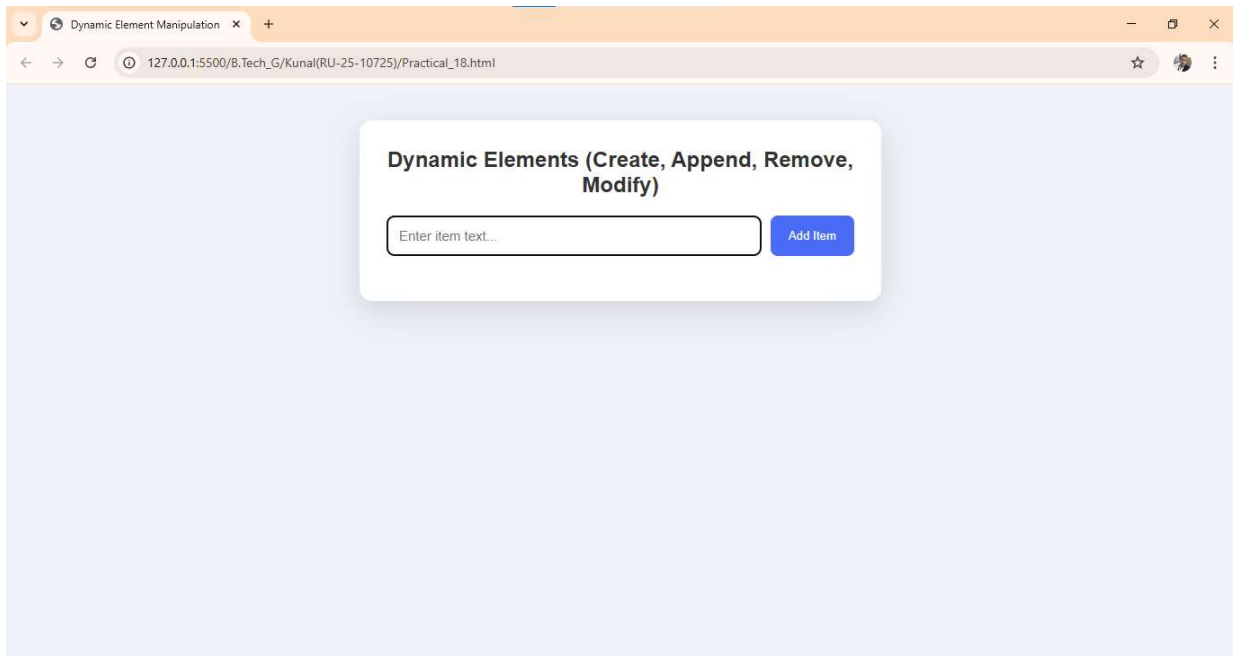
The screenshot shows the VS Code editor with the file 'Practical_18.html' open. The editor displays the following code:

```
2 <html lang="en">
3 <head>
4 <style>
5   body {
6     font-family: "Inter", sans-serif;
7     background: #eef1f7;
8     margin: 0;
9     padding: 40px;
10    display: flex;
11    justify-content: center;
12  }
13
14  .app {
15    width: 520px;
16    background: #ffffff;
17    padding: 30px;
18    border-radius: 14px;
19    box-shadow: 0 10px 32px rgba(0,0,0,0.12);
20  }
21
22  h2 {
23    margin: 0 0 20px;
24    color: #333;
25    text-align: center;
26  }
27
28  .controls {
29    display: flex;
30    gap: 10px;
31    margin-top: 20px;
32  }
33
34  input {
35    flex: 1;
36    padding: 12px;
37    border-radius: 8px;
38    border: 2px solid #d6d9e4;
39    font-size: 15px;
40  }
41
42  button {
43    padding: 12px 20px;
44    background: #4a6cf7;
45    color: white;
46    border: none;
47    border-radius: 8px;
48    cursor: pointer;
49    transition: background 0.3s ease, transform 0.25s ease;
50  }
51
52  button:hover {
53    background: #3b56c7;
54    transform: translateY(-2px);
55  }
56
57  .list {
58    margin-top: 10px;
59  }
60
61  .list-item {
62    padding: 10px;
63    border: 1px solid #d6d9e4;
64    margin-bottom: 5px;
65  }
```

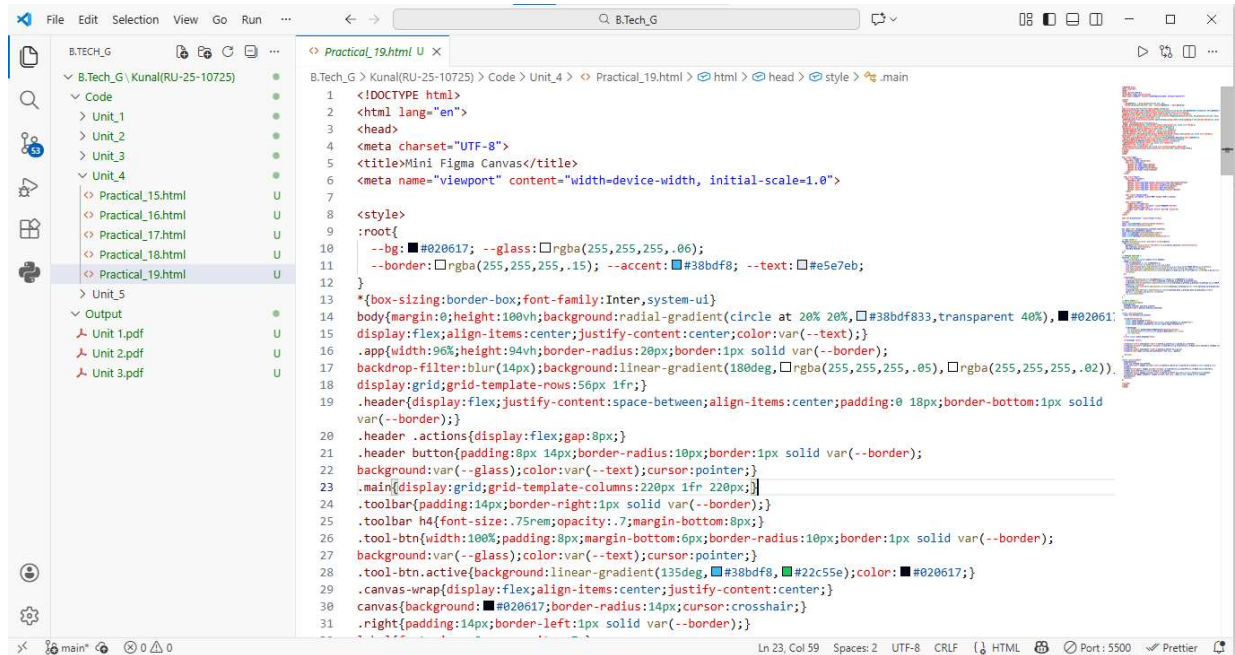
The status bar at the bottom indicates 'Ln 165, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', 'HTML', 'Port: 5500', and 'Prettier'.

```
File Edit Selection View Go Run ... B.Tech_G photo.jpg practical_1.html practical_2.html practical_3.html practical_4.html practical_5.html # Practical_6.css practical_6.html # Practical_7.css practical_7.html # Practical_8.css practical_8.html # Practical_9.css practical_9.html # Practical_10.css practical_10.html practical_11.html practical_12.html practical_13.html practical_14.html practical_15.html practical_16.html practical_17.html practical_18.html B.Tech_G > Kunal(RU-25-10725) > Practical_18.html > ... 2 <html lang="en"> 3 <head> 8 <style> 86 .removeBtn { 87 background: #ff4d4d; 88 padding: 6px 12px; 89 border: none; 90 color: white; 91 border-radius: 6px; 92 cursor: pointer; 93 transition: 0.2s ease; 94 } 95 96 .removeBtn:hover { 97 background: #d63838; 98 } 99 </style> 100 </head> 101 102 <body> 103 <div class="app"> 104 <h2>Dynamic Elements (Create, Append, Remove, Modify)</h2> 105 106 <!-- Controls --> 107 <div class="controls"> 108 <input type="text" id="itemInput" placeholder="Enter item text..."> 109 <button id="addBtn">Add Item</button> 110 </div> 111 112 <!-- List Container --> 113 <div id="itemList" class="list"></div> 114 </div> Ln 165, Col 1 Spaces: 4 UTF-8 CRLF HTML Port: 5500 Prettier
```

```
File Edit Selection View Go Run ... B.Tech_G photo.jpg practical_1.html practical_2.html practical_3.html practical_4.html practical_5.html # Practical_6.css practical_6.html # Practical_7.css practical_7.html # Practical_8.css practical_8.html # Practical_9.css practical_9.html # Practical_10.css practical_10.html practical_11.html practical_12.html practical_13.html practical_14.html practical_15.html practical_16.html practical_17.html practical_18.html B.Tech_G > Kunal(RU-25-10725) > Practical_18.html > ... 2 <html lang="en"> 102 <body> 116 <script> 124 addBtn.addEventListener("click", () => { 137 textSpan.addEventListener("dblclick", () => { 140 }); 141 142 // Remove button 143 const removeBtn = document.createElement("button"); 144 removeBtn.textContent = "X"; 145 removeBtn.classList.add("removeBtn"); 146 147 // Remove event 148 removeBtn.addEventListener("click", () => { 149 card.remove(); // REMOVE CHILD NODE 150 }); 151 152 // Append child nodes to the card 153 card.appendChild(textSpan); 154 card.appendChild(removeBtn); 155 156 // Append card to list 157 list.appendChild(card); 158 159 input.value = ""; 160 }); 161 </script> 162 163 </body> 164 </html> 165 Ln 165, Col 1 Spaces: 4 UTF-8 CRLF HTML Port: 5500 Prettier
```

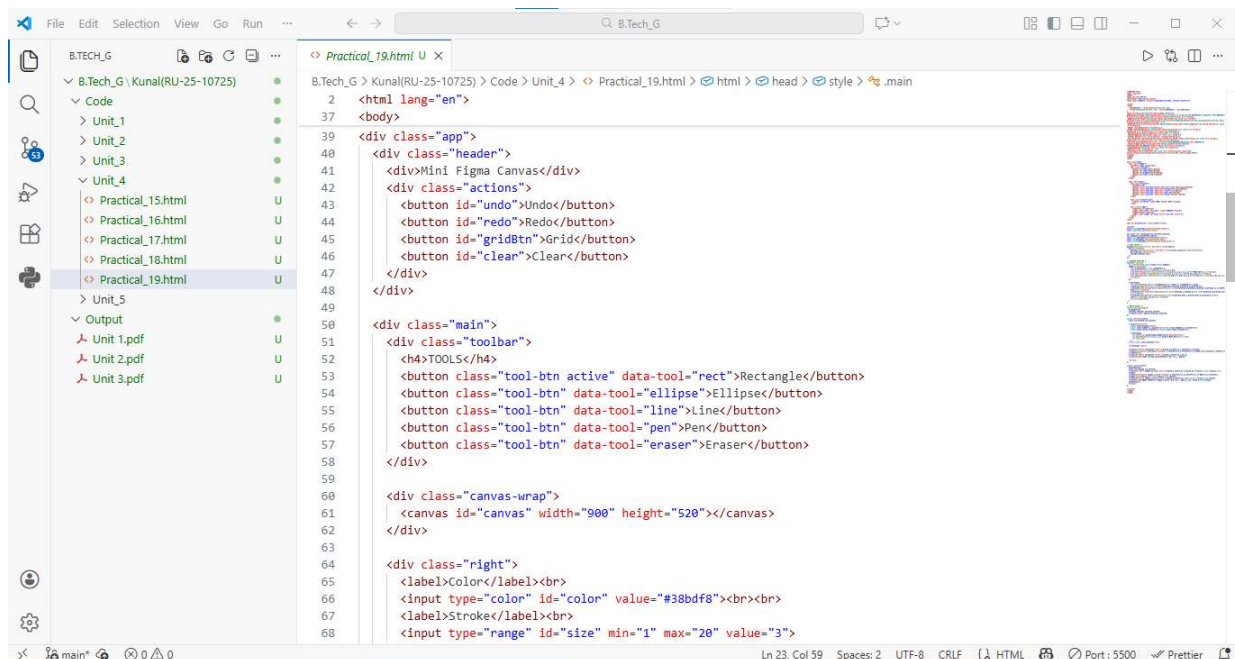


19. Create a simple drawing using the element: draw shapes and fill colors.



This screenshot shows the CSS portion of the code for a drawing application. The code is written in a modern, utility-first style using Tailwind CSS. It defines a root container with a glass-like background and a border. The main content area is a grid with a toolbar on the left and a canvas on the right. The toolbar contains buttons for undo, redo, grid, and clear. The canvas is a large area for drawing. The right sidebar contains a color picker and a size selector.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Mini Figma Canvas</title>
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8 <style>
9 :root{
10 --bg: #020617; --glass: rgba(255,255,255,.06);
11 --border: rgba(255,255,255,.15); --accent: #38bdf8; --text: #e5e7eb;
12 }
13 *{box-sizing:border-box;font-family:Inter,system-ui}
14 body{margin:0;height:100vh;background:radial-gradient(circle at 20% 20%, rgba(38,183,255,.4) 0%, transparent 0%), #020617;
15 display:flex;align-items:center;justify-content:center;color:var(--text);}
16 .app{width:90%;height:94vh;border-radius:20px;border:1px solid var(--border);
17 backdrop-filter:blur(14px);background:linear-gradient(180deg, rgba(255,255,255,.05), transparent 0%);
18 display:grid;grid-template-rows:56px 1fr;}
19 .header{display:flex;justify-content:space-between;align-items:center;padding:0 18px;border-bottom:1px solid
20 var(--border);}
21 .header .actions{display:flex;gap:8px;}
22 .header button{padding:8px 14px;border-radius:10px;border:1px solid var(--border);
23 background:var(--glass);color:var(--text);cursor:pointer;}
24 .main{display:grid;grid-template-columns:220px 1fr 220px;}
25 .toolbar{padding:14px;border-right:1px solid var(--border);}
26 .tool-h4{font-size:.75rem;opacity:.7;margin-bottom:8px;}
27 .tool-btn{width:100%;padding:8px;margin-bottom:6px;border-radius:10px;border:1px solid var(--border);
28 background:var(--glass);color:var(--text);cursor:pointer;}
29 .tool-btn.active{background:linear-gradient(135deg, #38bdf8, #2c5e5e);color:#020617;}
30 .canvas-wrap{display:flex;align-items:center;justify-content:center;}
31 canvas{background:#020617;border-radius:14px;cursor:crosshair;}
32 .right{padding:14px;border-left:1px solid var(--border);}
```



This screenshot shows the HTML portion of the code for a drawing application. The code uses a simple, semantic structure with classes and IDs for styling and scripting. It includes a header with a title and a main content area with a toolbar and a canvas. The toolbar contains buttons for undo, redo, grid, and clear. The canvas is a large area for drawing. The right sidebar contains a color picker and a size selector.

```
2 <html lang="en">
37 <body>
39 <div class="app">
40 <div class="header">
41 <div>Mini Figma Canvas</div>
42 <div class="actions">
43 <button id="undo">Undo</button>
44 <button id="redo">Redo</button>
45 <button id="gridBtn">Grid</button>
46 <button id="clear">Clear</button>
47 </div>
48 </div>
49
50 <div class="main">
51 <div class="toolbar">
52 <h4>TOOLS</h4>
53 <button class="tool-btn active" data-tool="rect">Rectangle</button>
54 <button class="tool-btn" data-tool="ellipse">Ellipse</button>
55 <button class="tool-btn" data-tool="line">Line</button>
56 <button class="tool-btn" data-tool="pen">Pen</button>
57 <button class="tool-btn" data-tool="eraser">Eraser</button>
58 </div>
59
60 <div class="canvas-wrap">
61 <canvas id="canvas" width="900" height="520"></canvas>
62 </div>
63
64 <div class="right">
65 <label>Color</label><br>
66 <input type="color" id="color" value="#38bdf8"><br>
67 <label>Stroke</label><br>
68 <input type="range" id="size" min="1" max="20" value="3">
```



```
File Edit Selection View Go Run ... B.Tech_G Practical_19.html U x
B.Tech_G > Kunal(RU-25-10725) > Code > Unit_4 > Practical_19.html > html > head > style > .main
2 <html lang="en">
37 <body>
73 <div id="eraserCursor" class="eraser"></div>
74
75 <script>
76 const canvas=document.getElementById("canvas");
77 const ctx=canvas.getContext("2d");
78
79 let tool="rect",drawing=false,startX=0,startY=0;
80 let shapes=[],preview=null,pen=[];
81 const color=document.getElementById("color");
82 const size=document.getElementById("size");
83 const eraser=document.getElementById("eraserCursor");
84
85 /* TOOL SELECT */
86 document.querySelectorAll(".tool-btn").forEach(btn=>{
87   btn.onclick=>{
88     document.querySelectorAll(".tool-btn").forEach(b=>b.classList.remove("active"));
89     btn.classList.add("active");
90     tool=btn.dataset.tool;
91   };
92 });
93
94 /* DRAWING FUNCTION */
95 function redraw(){
96   ctx.clearRect(0,0,canvas.width,canvas.height);
97   shapes.forEach(s=>{
98     ctx.strokeStyle=s.c; ctx.lineWidth=s.w;
99     if(s.t=="rect")ctx.strokeRect(s.x,s.y,s.wi,s.hi);
100     if(s.t=="ellipse"){ctx.beginPath();ctx.ellipse(s.x,s.y,s.rx,s.ry,0,0,Math.PI*2);ctx.stroke();}
101     if(s.t=="line"){ctx.beginPath();ctx.moveTo(s.x,s.y);ctx.lineTo(s.x2,s.y2);ctx.stroke();}
102     if(s.t=="pen"){ctx.beginPath();ctx.moveTo(s.p[0].x,s.p[0].y);s.p.forEach(pt=>ctx.lineTo(pt.x,pt.y));ctx.
```

```
File Edit Selection View Go Run ... B.Tech_G Practical_19.html U x
B.Tech_G > Kunal(RU-25-10725) > Code > Unit_4 > Practical_19.html > html > head > style > .main
115 /* MOUSE EVENTS */
116 canvas.onmousedown=e=>{
117   drawing=true;
118   startX=e.offsetX; startY=e.offsetY;
119   if(tool=="pen" pen=[{x:startX,y:startY}];
120 };
121
122 canvas.onmousemove=e=>{
123   const x=e.offsetX,y=e.offsetY;
124
125   if(tool=="eraser"){
126     eraser.style.display="block";
127     eraser.style.width=size.value*2+"px";eraser.style.height=size.value*2+"px";
128     eraser.style.left=e.clientX+"px";eraser.style.top=e.clientY+"px";
129
130     if(drawing){
131       ctx.save();ctx.globalCompositeOperation="destination-out";
132       ctx.beginPath();ctx.arc(x,y,size.value,0,Math.PI*2);ctx.fill();
133       ctx.restore();
134     }
135     else eraser.style.display="none";
136
137     if(!drawing) return;
138
139     if(tool=="rect") preview={t:"rect",x:startX,y:startY,wi:x-startX,hi:y-startY};
140     if(tool=="ellipse") preview={t:"ellipse",x:(startX+x)/2,y:(startY+y)/2,rx:Math.abs(x-startX)/2,ry:Math.abs(y-startY)/2};
141     if(tool=="line") preview={t:"line",x:startX,y:startY,x2:x,y2:y};
142     if(tool=="pen"){pen.push({x,y});preview={t:"pen",p:[...pen]};}
```

