



RIPHAH
INTERNATIONAL UNIVERSITY

Submitted By: Alishba Asif

Sap Id: 60089

Section: BSCS-4

Subject: Problem Solving

Submitted To: Mr. Haroon

JavaScript DOM Lab Manual

Practice:

```
part1 > example.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=, initial-scale=1.0">
6   <link rel="stylesheet" href="style.css">
7   <title>Document</title>
8 </head>
9 <body>
10   <div class="container">
11     <h1 id="main-heading">Favourite Animated Movies</h1>
12     <ul>
13       <li class="List-items">Kung Fu Panda 4</li>
14       <li class="List-items">Tom and Jerry</li>
15       <li class="List-items">Wish Dragon</li>
16       <li class="List-items">Leo</li>
17     </ul>
18     <p>This is my paragraph </p>
19     <button id="btn"> Button </button>
20   </div>
21   <!-- <div class="container">
22     <h1 id="main-heading">Favourite Movies</h1>
23     <ul>
24       <li class="List-items">Kung Fu Panda 4</li>
25       <li class="List-items">Tom and Jerry</li>
26       <li class="List-items">Wish Dragon</li>
27       <li class="List-items">Leo</li>
28     </ul>
29   </div> -->
30   <script src="app.js"></script>
31 </body>
32 </html>
```

This screenshot shows the VS Code editor with a file explorer on the left containing 'lab6(DOM)', 'part1', 'JS app.js', 'example.html', and 'style.css'. The main editor displays the content of 'JS app.js' with the following code:

```
part1 > JS app.js > ...
1 //DOM Manipulation
2 //getElementById
3 const title = document.getElementById('main-heading');
4 console.log(title);
5
6 //getElementsByClassName
7 const listItems = document.getElementsByClassName('list-items');
8 console.log(listItems);
9
10 //getElementsByTagName
11 const items = document.getElementsByTagName('li');
12 console.log(items);
13
14 //querySelector
15 const container = document.querySelector('div');
16 console.log(container);
17
18 //querySelectorAll
19 const container2 = document.querySelectorAll('div');
20 console.log(container2);
21
22
23 // Styling with the help of DOM
24 // style using querySelector(only select 1st element)
25 // const heading = document.querySelector("#main-heading");
26 // heading.style.color = 'blue';
27
28 // style with querySelectorAll(select All match)
29 // const container = document.querySelectorAll('div');
30 // loop for each list item & styling
31 // container.forEach(item => {
32 //   item.style.backgroundColor = "light-gray";
```

The status bar at the bottom indicates 'Ln 36, Col 27', 'Spaces: 4', 'UTF-8', 'CRLF', 'JavaScript', 'Port: 5500', and 'Prettier'.

This screenshot shows the continuation of the 'JS app.js' file in the VS Code editor. The code continues from line 32:

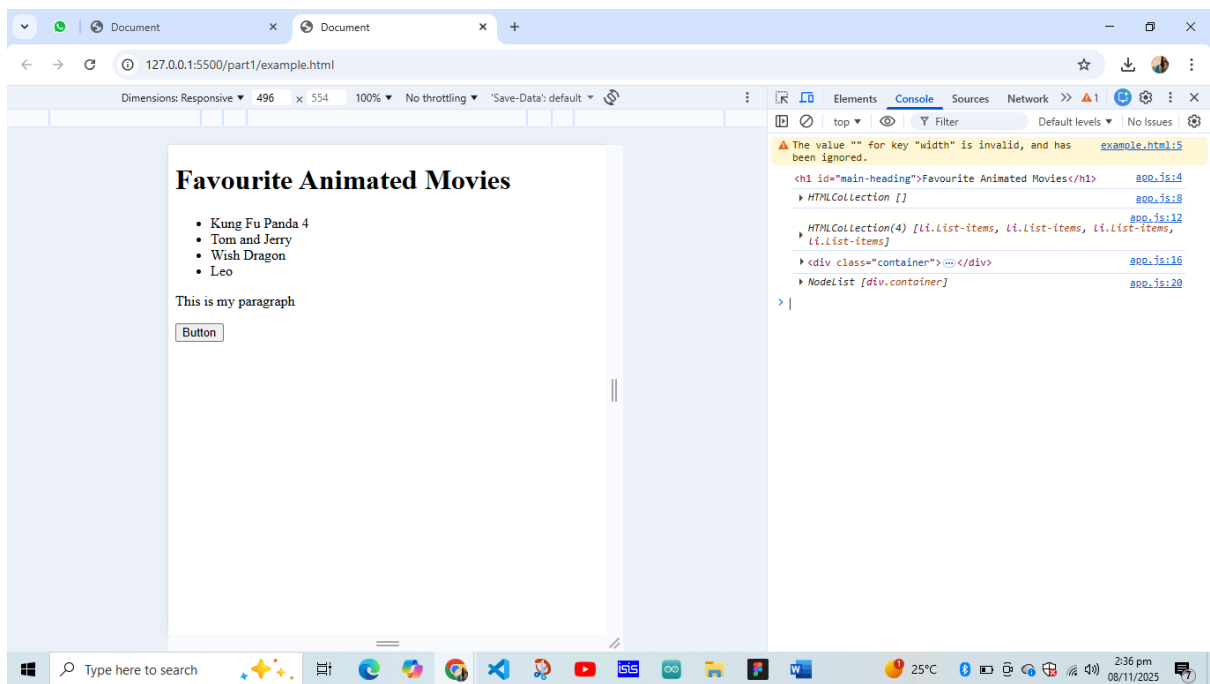
```
32 //   item.style.backgroundColor = "light-gray";
33   item.style.color = "darkgreen";
34   item.style.fontWeight = "bold";
35 });
36 console.log(container);
37
38 //innerHTML
39 const h1 = document.getElementById("main-heading").innerHTML = "<b> New HTML Added! </b>";
40 console.log(h1);
41
42 //innerText
43 let text = document.getElementById("main-heading").innerText;
44 console.log(text);
45
46 //innerHTML
47 let text = document.getElementById("main-heading").innerHTML;
48 console.log(text);
49
50 //setAttribute/getAttribute
51 let btn = document.getElementById("btn");
52 btn.setAttribute("title", "Click this button");
53 console.log(btn.getAttribute("id"));
54
55 //createElement() and createTextNode()
56 let newDiv = document.createElement("div");
57 let textNode = document.createTextNode("This is a new div!");
58 newDiv.appendChild(textNode);
59 document.body.appendChild(newDiv);
60 console.log(textNode);
61
62 //appendChild()
63 let newpara = document.createElement("p");
64 newpara.textContent = "This is my added paragraph!"
```

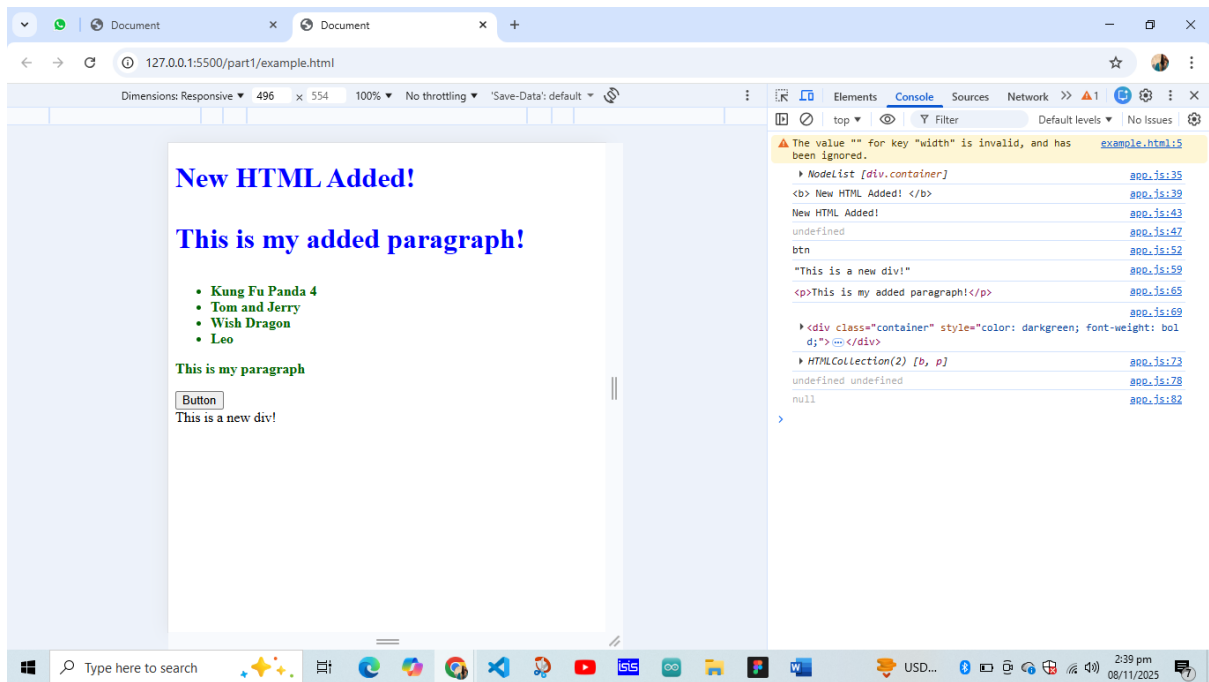
The status bar at the bottom indicates 'Ln 64, Col 53', 'Spaces: 4', 'UTF-8', 'CRLF', 'JavaScript', 'Port: 5500', and 'Prettier'.

The screenshot shows the Visual Studio Code editor with a file explorer on the left, a central code editor, and a right-hand sidebar with a console and a DOM tree. The code in the editor is as follows:

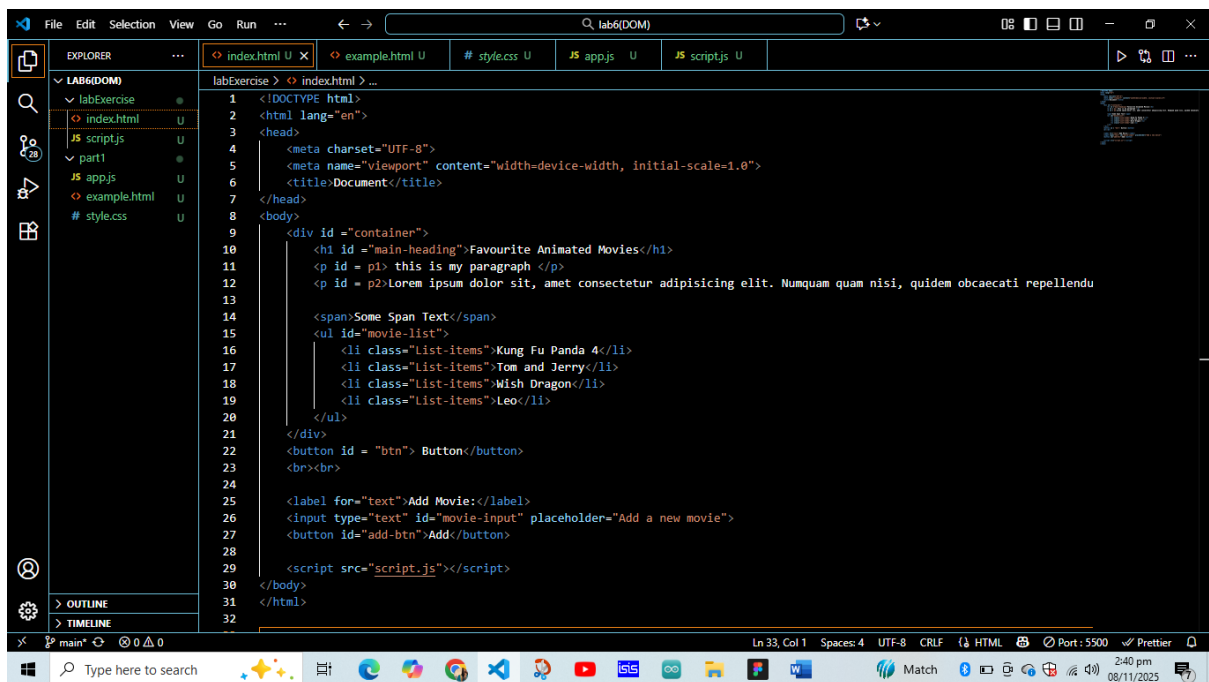
```
part1 > JS appjs > ...
65 document.getElementById("main-heading").appendChild(newpara);
66 console.log(newpara);
67
68 // parentNode / parentElement
69 let button = document.getElementById("btn");
70 console.log(button.parentNode);
71
72 //childNodes / children
73 let container = document.getElementById("main-heading");
74 console.log(container.children);
75
76 //firstChild / lastChild
77 let first = container.firstChild;
78 let last = container.lastChild;
79 console.log(first,last);
80
81 // nextSibling / previousSibling
82 let para1 = document.querySelector("p");
83 console.log(para1.nextSibling);
84
85 //classList
86 let heading= document.getElementById("main-heading");
87 heading.classList.add("highlight");
88 heading.classList.toggle("hhhhhh");
89
90 //DOM Events (onclick, addEventListener())
91 document.getElementById("btn").addEventListener("click", function(){
92 |   alert("Button clicked!");
93 | });
```

The status bar at the bottom indicates the file is 'main*' at line 33, column 4, with 1024 characters selected. The workspace is 'lab6(DOM)'.



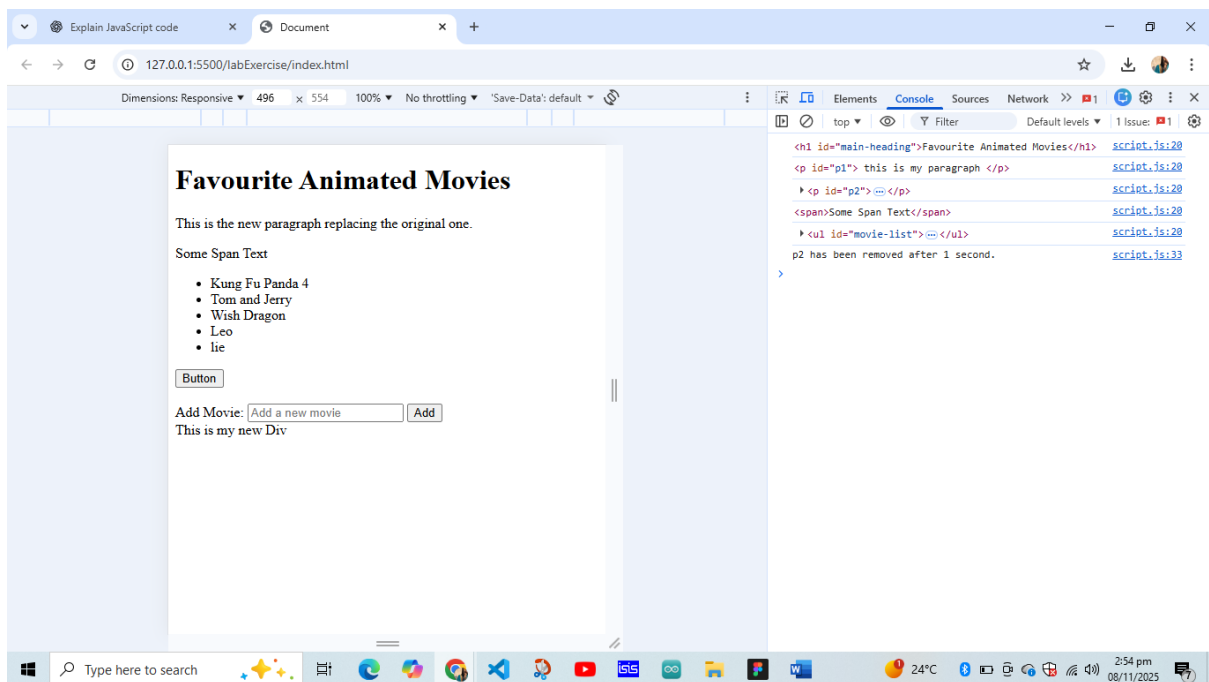
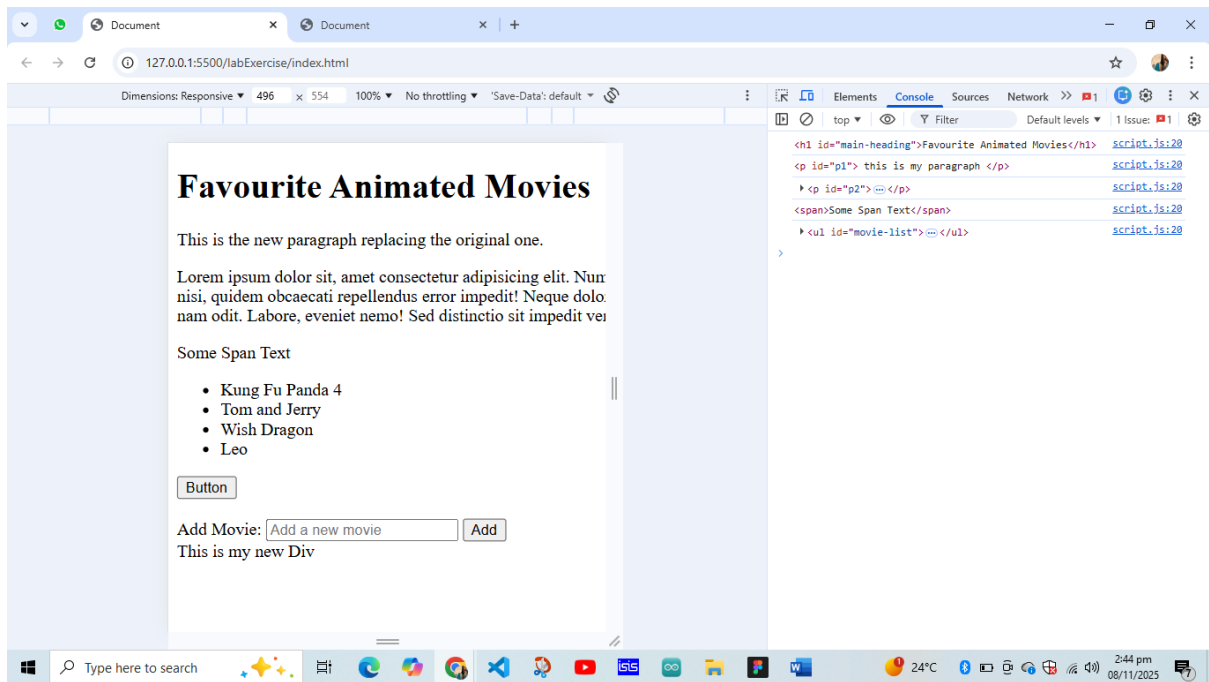


Lab Exercises



```
labExercise > JS scripts > ...
1  /*Create a new <div> dynamically.
2  Add some text to it.
3  Append it to the body.*/
4  const newDiv = document.createElement("div");
5  newDiv.textContent = "This is my new Div";
6  document.body.appendChild(newDiv);
7
8  /*Modify CSS Using JavaScript
9  Change the color and font size of an element when a button is clicked.*/
10 document.getElementById("btn").addEventListener("click", function(){
11     btn.style.color = "blue";
12     btn.style.fontSize = "2rem";
13 });
14
15 /* DOM Traversal
16 Print all child elements of a specific container in the console.*/
17 let container = document.getElementById("container");
18 let children = container.children;
19 for (let i=0; i<children.length; i++){
20     console.log(children[i]);
21 }
22
23 /*Replace and Remove Elements
24 Replace one paragraph with another dynamically and remove a node after 3 seconds.*/
25 let para1 = document.getElementById("p1");
26 let newpara = document.createElement("p");
27 newpara.textContent = "This is the new paragraph replacing the original one.";
28 para1.replaceWith(newpara);
29
30 let removepara = document.getElementById("p2");
31 setTimeout(function(){
32     removepara.remove();
```

```
labExercise > JS scripts > setTimeout() callback
31 setTimeout(function(){
32     console.log("p2 has been removed after 1 second.");
33 }, 1000);
34
35 /*Interactive To-Do List
36 Add input and "Add" button. Add new list items dynamically using appendChild().*/
37 let input = document.getElementById("movie-input");
38 let addBtn = document.getElementById("add-btn");
39 let list = document.getElementById("movie-list");
40
41 addBtn.addEventListener("click", function() {
42     let movieName = input.value.trim();
43     if (movieName === "") {
44         alert("Please enter a movie name!");
45         return;
46     }
47     let newItem = document.createElement("li");
48     newItem.textContent = movieName;
49     newItem.classList.add("list-items");
50     list.appendChild(newItem);
51     input.value = "";
52 });
53
54
55
56
57
```

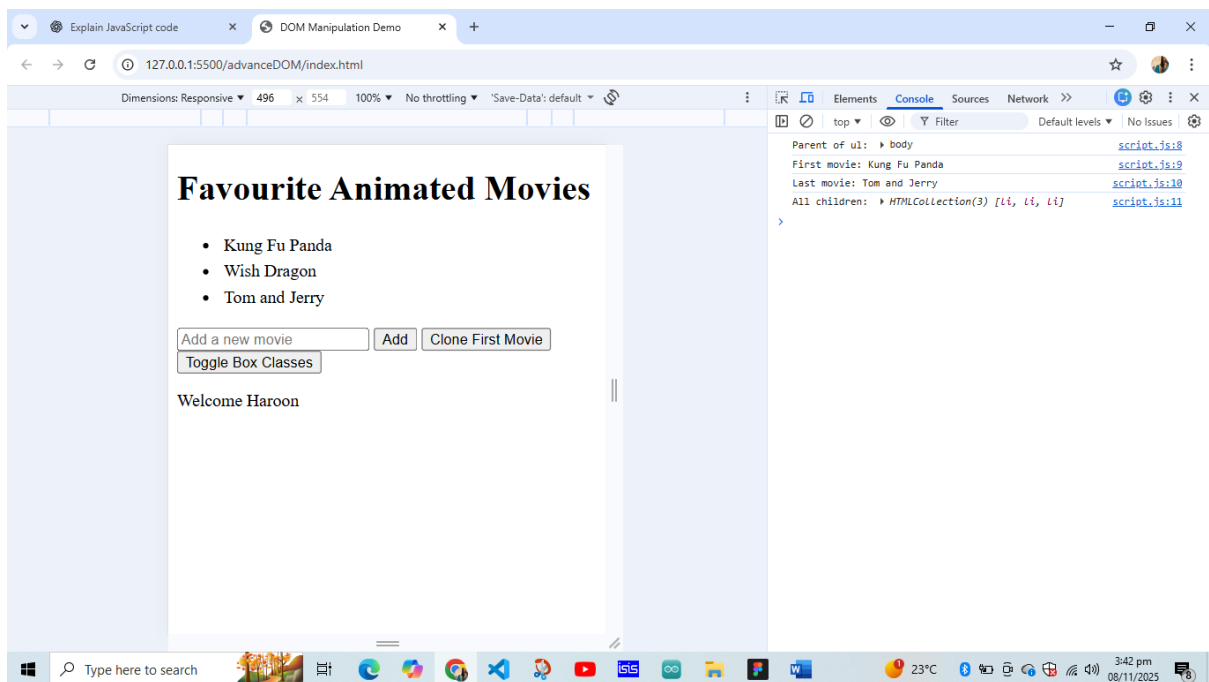
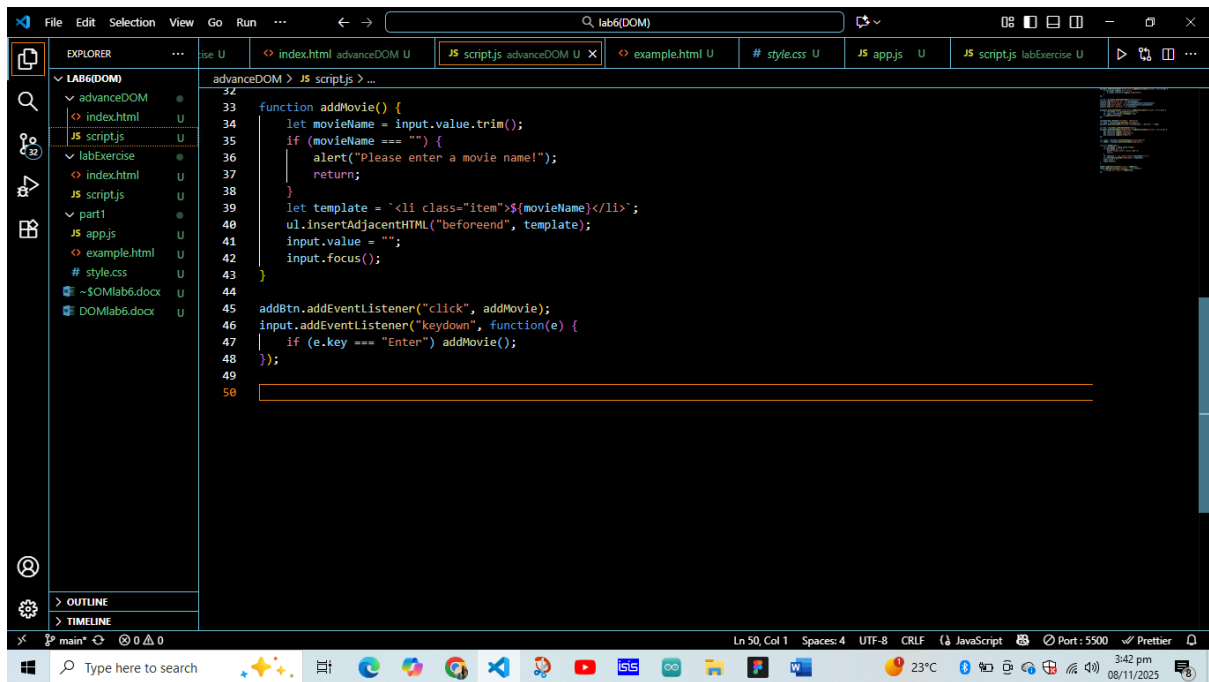


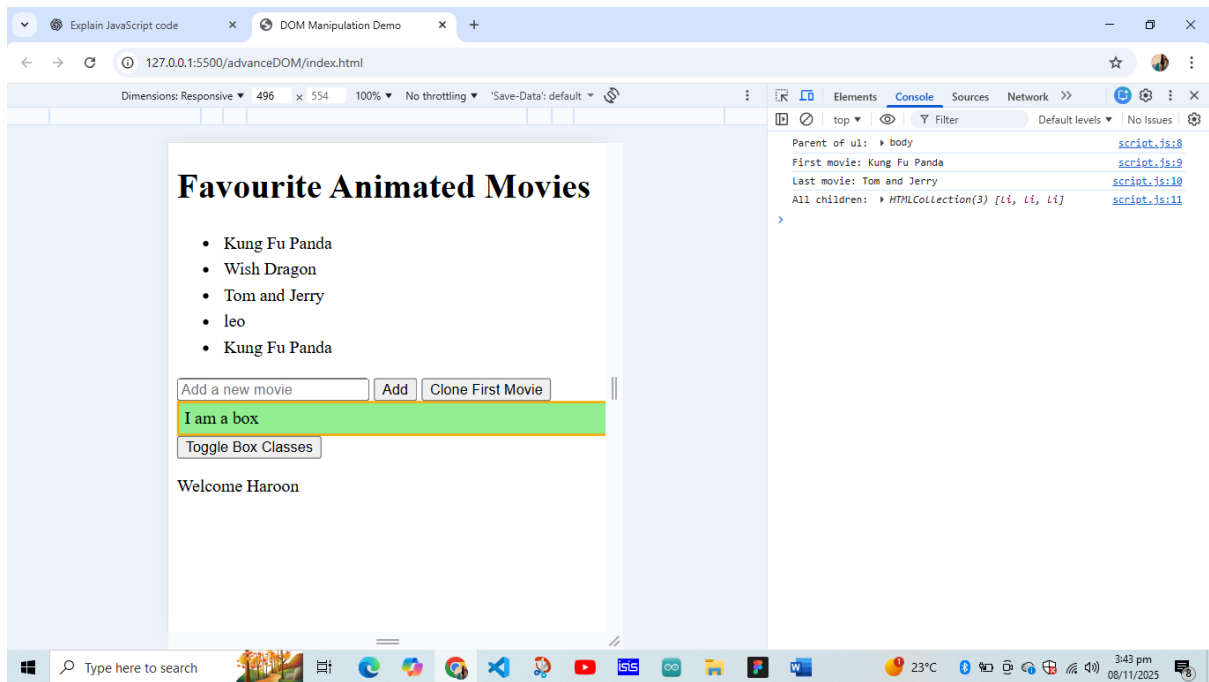
Advanced JavaScript DOM Lab Manual

Practice:

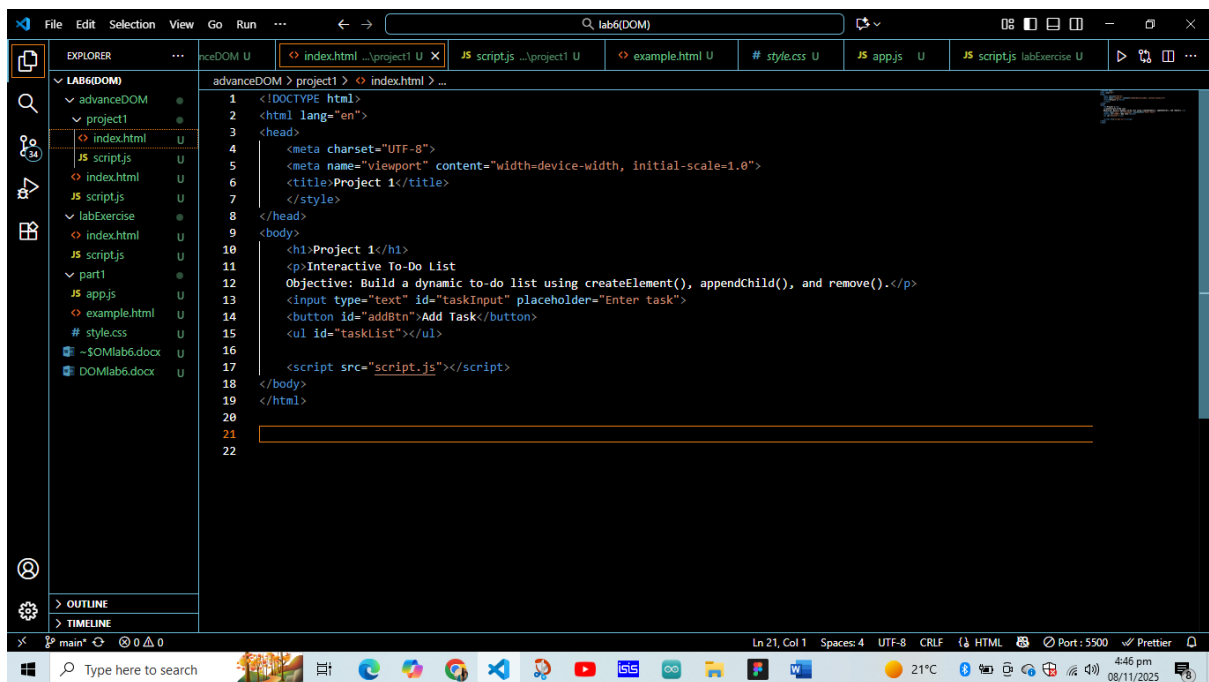
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>DOM Manipulation Demo</title>
7   <style>
8     .completed { text-decoration: line-through; color: gray; }
9     .active { background-color: lightgreen; padding: 5px; }
10    .highlight { border: 2px solid orange; }
11    .hidden { display: none; }
12    ul li { cursor: pointer; padding: 3px; }
13  </style>
14 </head>
15 <body>
16   <h1>Favourite Animated Movies</h1>
17   <ul id="movie-list">
18     <li>Kung Fu Panda</li>
19     <li>Wish Dragon</li>
20     <li>Tom and Jerry</li>
21   </ul>
22   <input type="text" id="movie-input" placeholder="Add a new movie">
23   <button id="add-btn">Add</button>
24   <button id="clone-btn">Clone First Movie</button>
25   <div id="box" class="hidden">I am a box</div>
26   <button id="toggle-box">Toggle Box Classes</button>
27   <p id="welcome"></p>
28   <script src="script.js"></script>
29 </body>
30 </html>
31
32
```

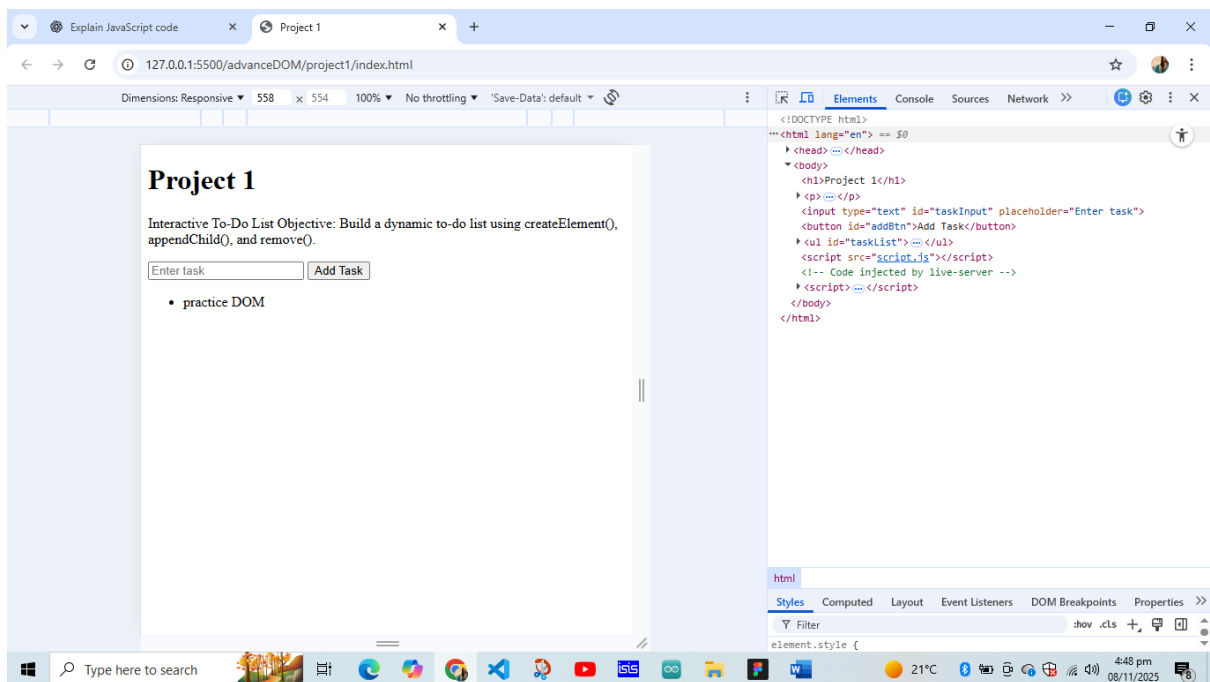
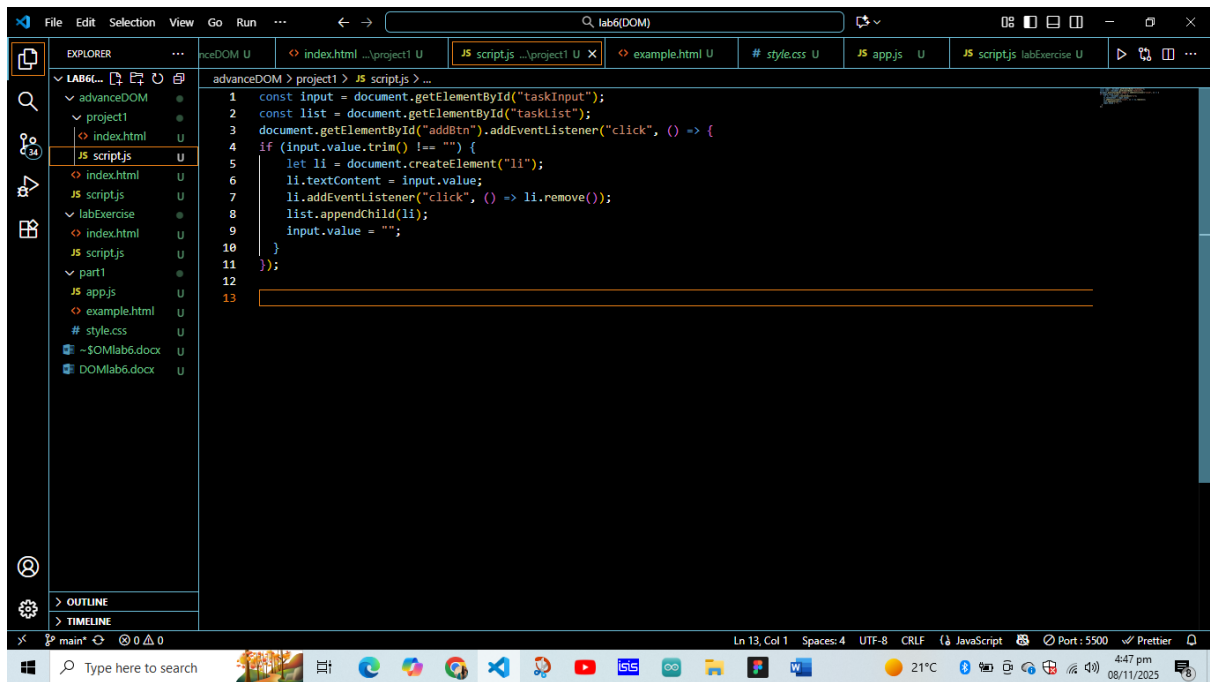
```
1 document.getElementById("movie-list").addEventListener("click", function(e) {
2   if (e.target.tagName === "LI") {
3     e.target.classList.toggle("completed");
4   }
5 });
6
7 let ul = document.getElementById("movie-list");
8 console.log("Parent of ul:", ul.parentNode);
9 console.log("First movie:", ul.firstChild.textContent);
10 console.log("Last movie:", ul.lastElementChild.textContent);
11 console.log("All children:", ul.children);
12
13 document.getElementById("clone-btn").addEventListener("click", function() {
14   let firstItem = ul.querySelector("li");
15   let copy = firstItem.cloneNode(true);
16   ul.appendChild(copy);
17 });
18
19 localStorage.setItem("username", "Haroon");
20 let name = localStorage.getItem("username");
21 document.getElementById("welcome").textContent = "Welcome " + name;
22
23 let box = document.getElementById("box");
24 document.getElementById("toggle-box").addEventListener("click", function() {
25   box.classList.toggle("active");
26   box.classList.toggle("highlight");
27   box.classList.toggle("hidden");
28 });
29
30 let input = document.getElementById("movie-input");
31 let addBtn = document.getElementById("add-btn");
32
```



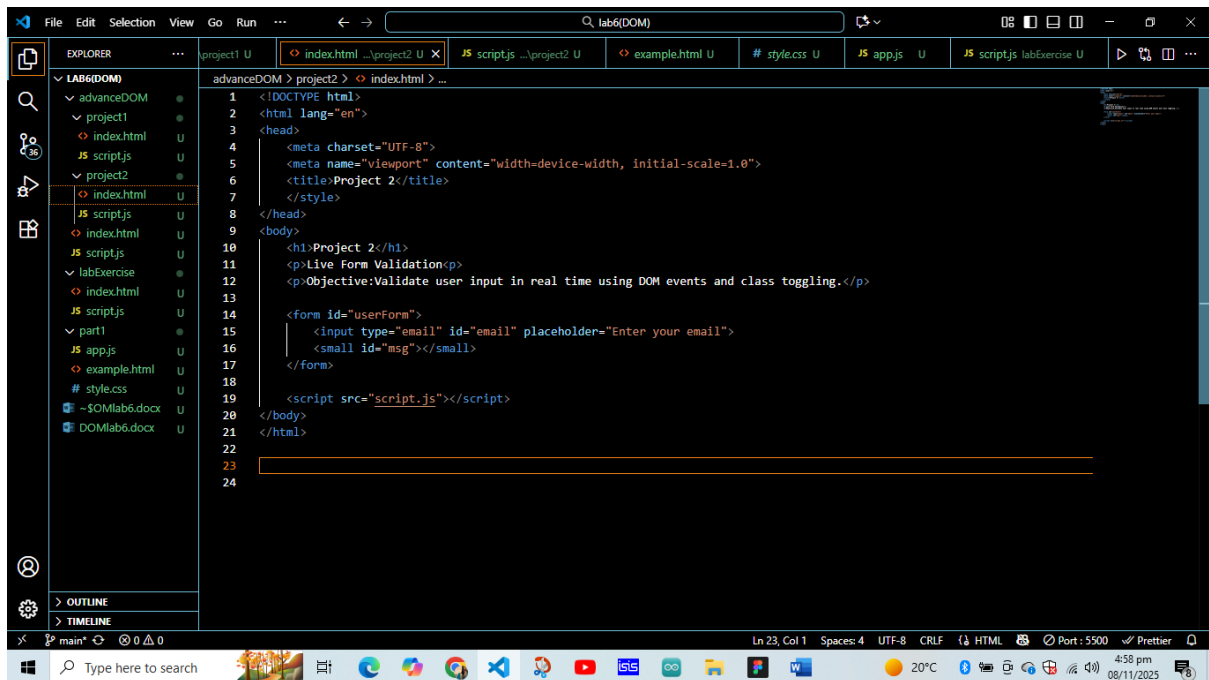


Project 1:



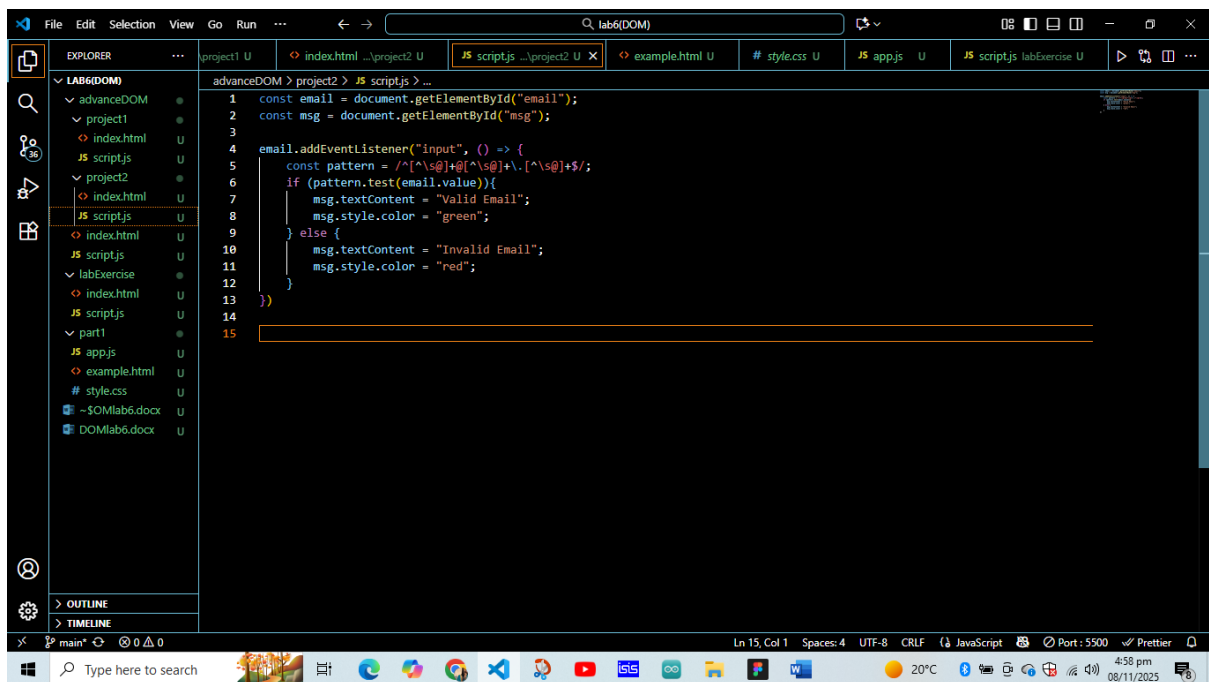


Project 2:



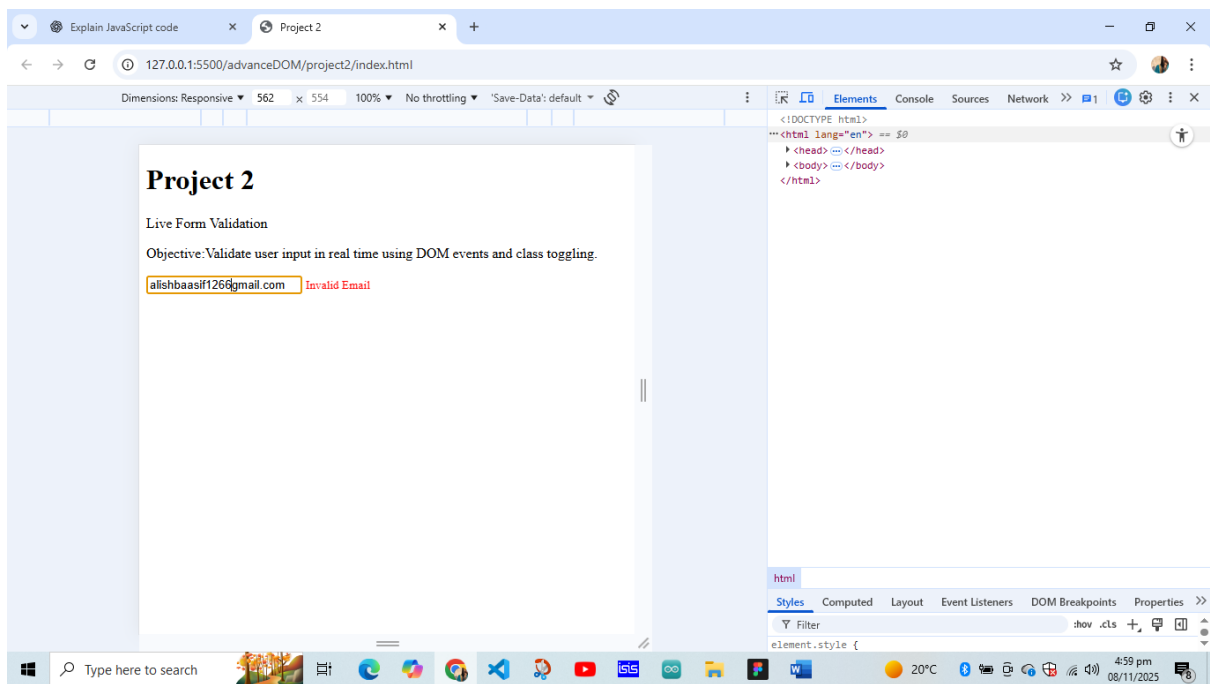
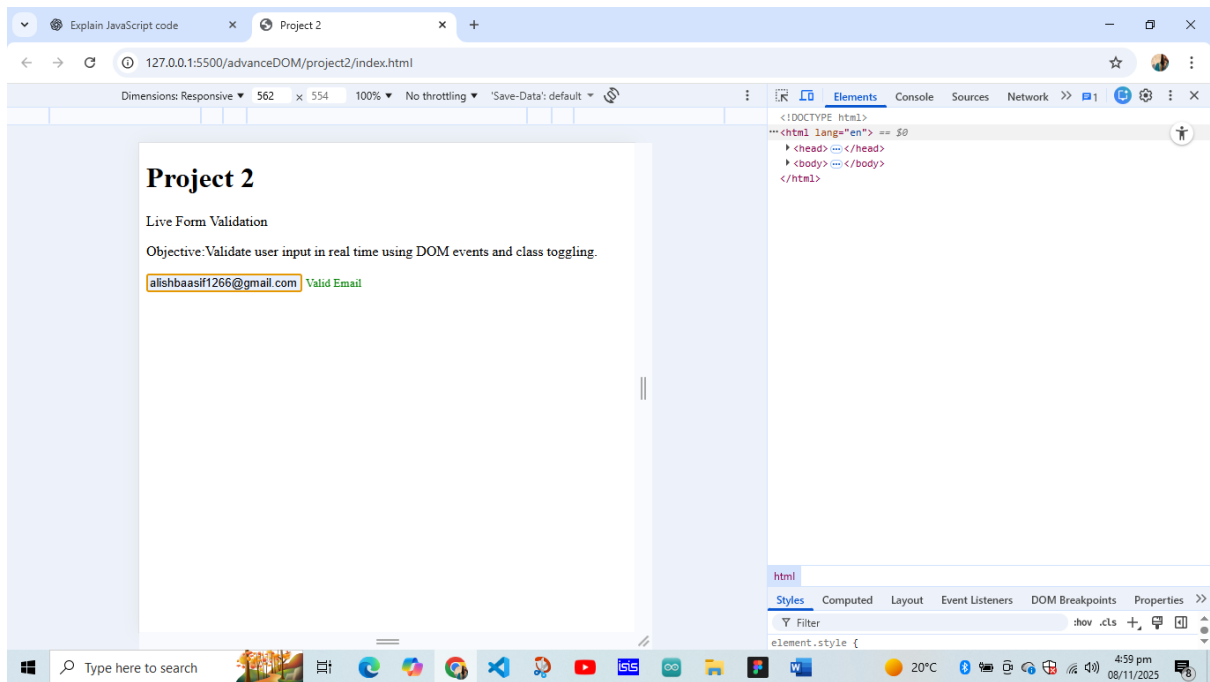
This screenshot shows the VS Code editor with the HTML file open. The Explorer sidebar on the left shows the project structure. The main editor area displays the HTML code for 'index.html'.

```
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   <title>Project 2</title>
7 </head>
8 <body>
9
10  <h1>Project 2</h1>
11  <p>Live Form Validation<p>
12  <p>Objective: Validate user input in real time using DOM events and class toggling.</p>
13
14  <form id="userForm">
15    <input type="email" id="email" placeholder="Enter your email">
16    <small id="msg"></small>
17  </form>
18
19  <script src="script.js"></script>
20 </body>
21 </html>
22
23
24
```

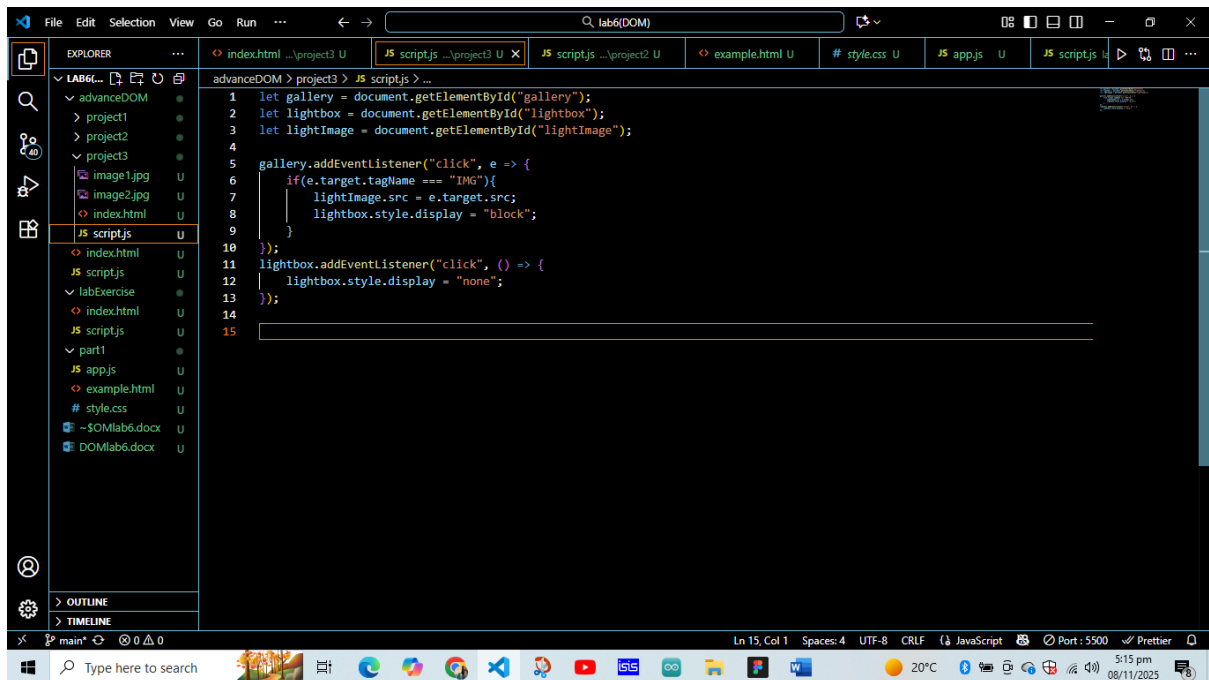
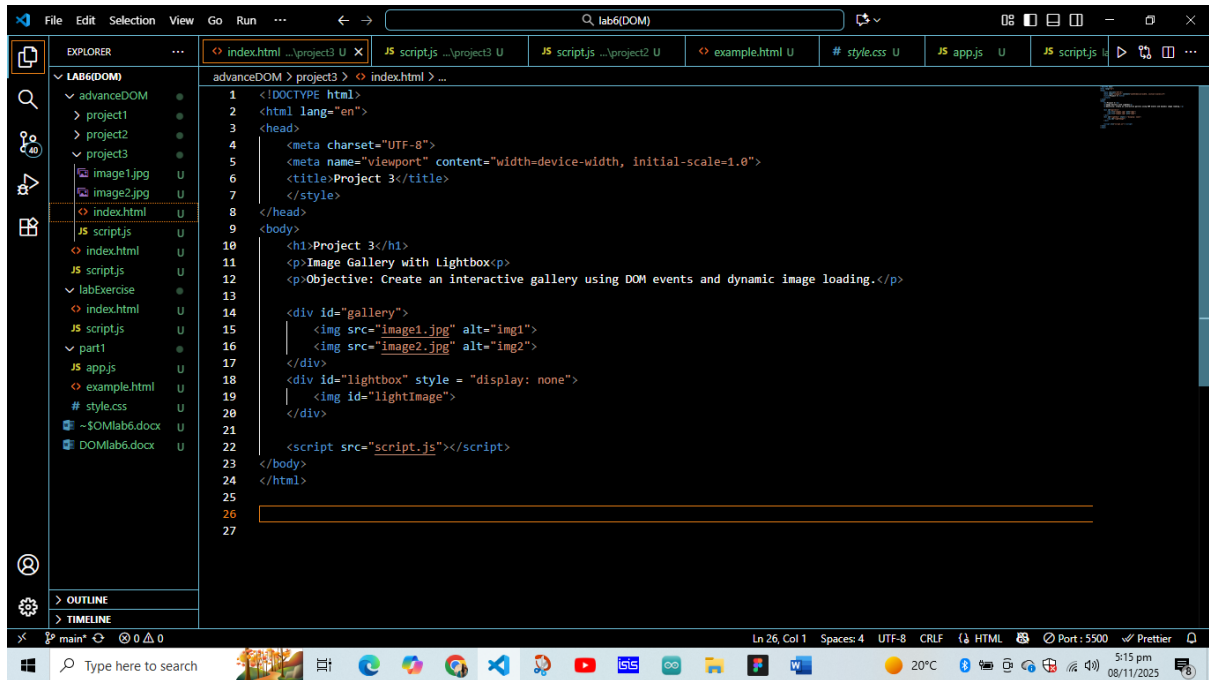


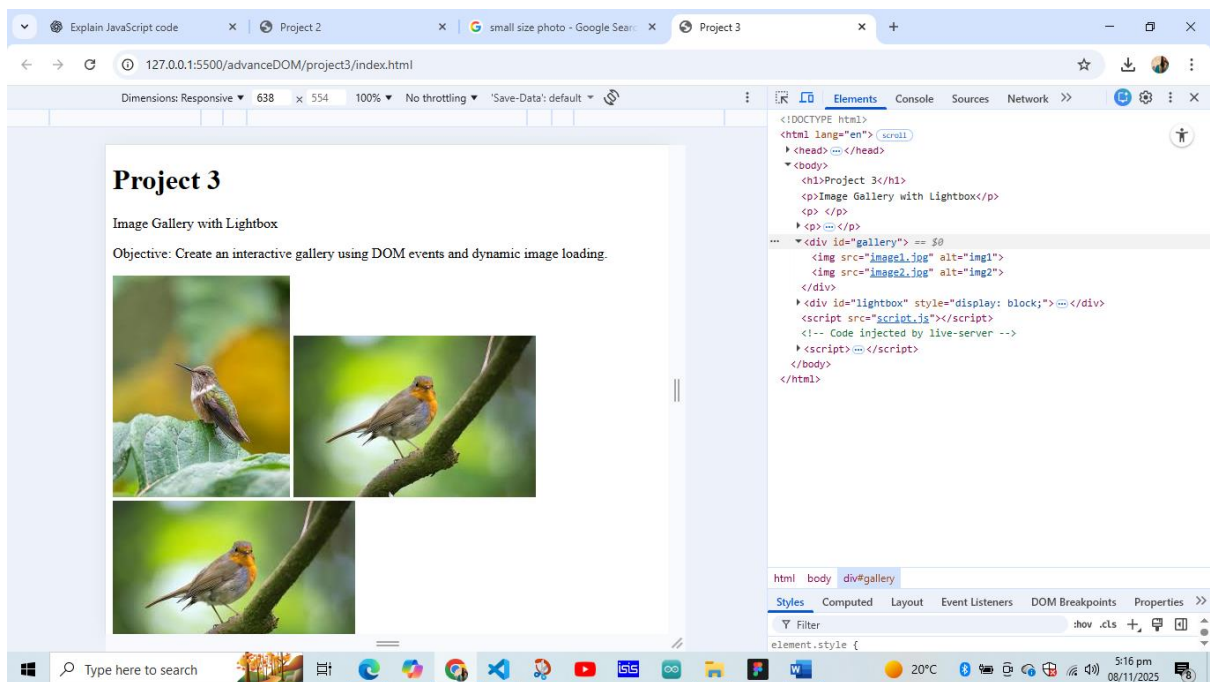
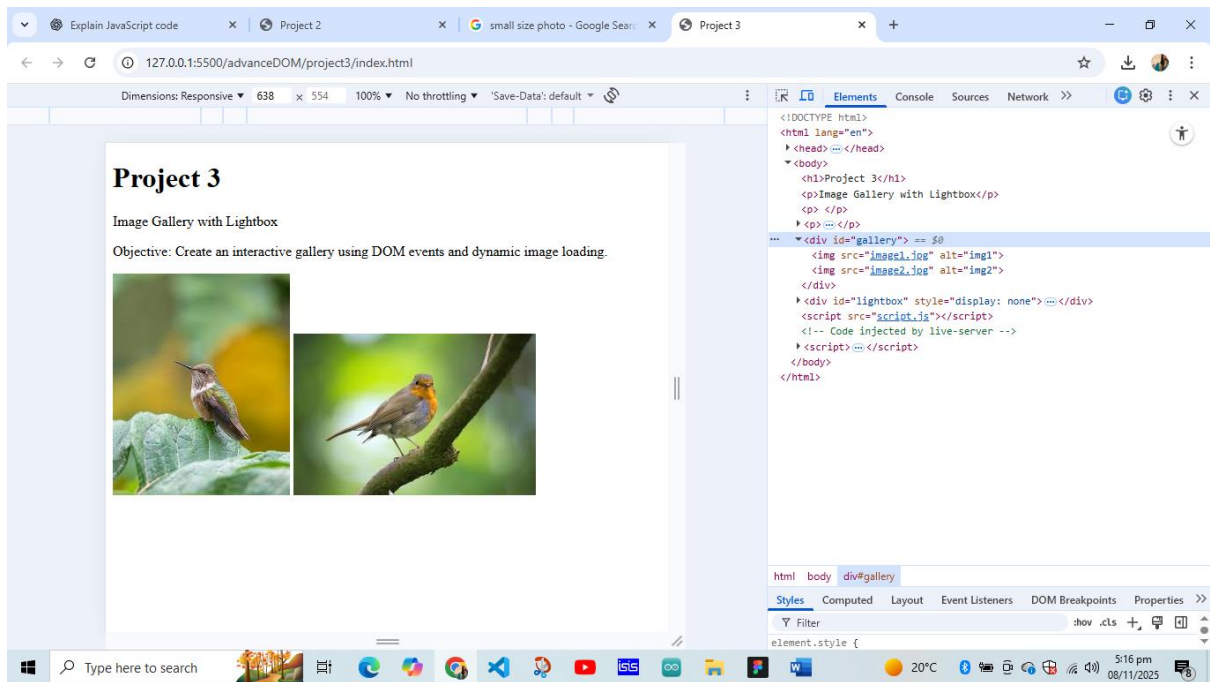
This screenshot shows the VS Code editor with the JavaScript file open. The Explorer sidebar on the left shows the project structure. The main editor area displays the JavaScript code for 'script.js'.

```
1 const email = document.getElementById("email");
2 const msg = document.getElementById("msg");
3
4 email.addEventListener("input", () => {
5   const pattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
6   if (pattern.test(email.value)){
7     msg.textContent = "Valid Email";
8     msg.style.color = "green";
9   } else {
10    msg.textContent = "Invalid Email";
11    msg.style.color = "red";
12  }
13 })
14
15
```

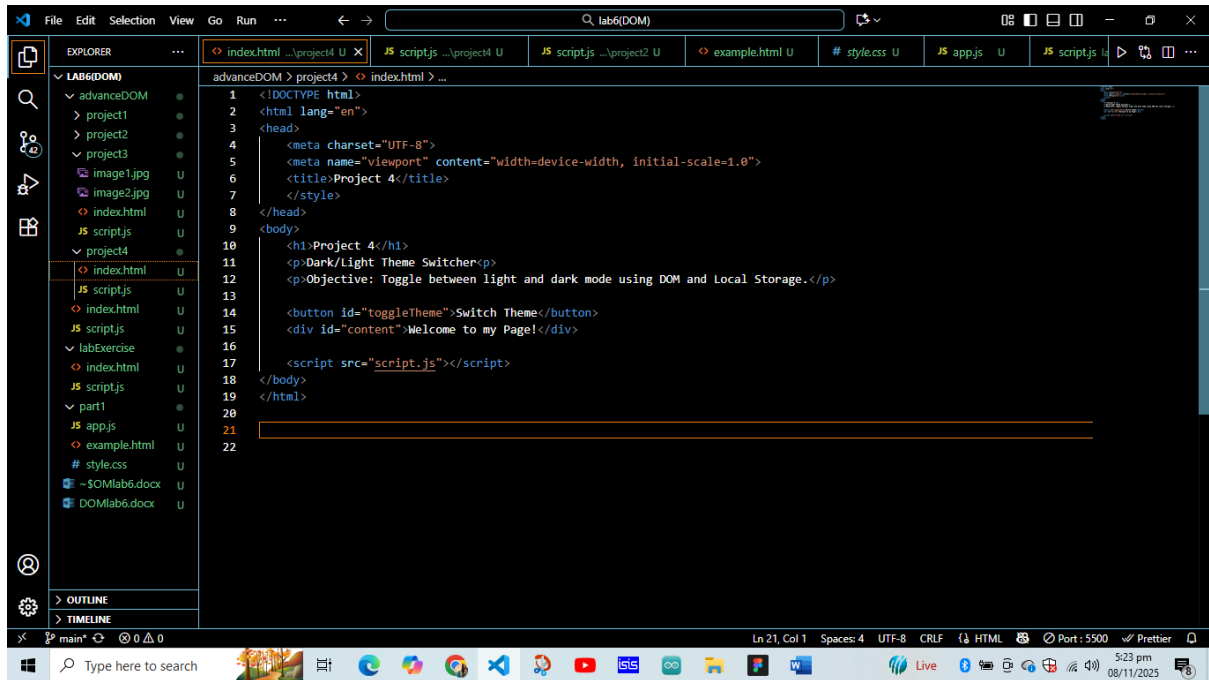


Project 3:



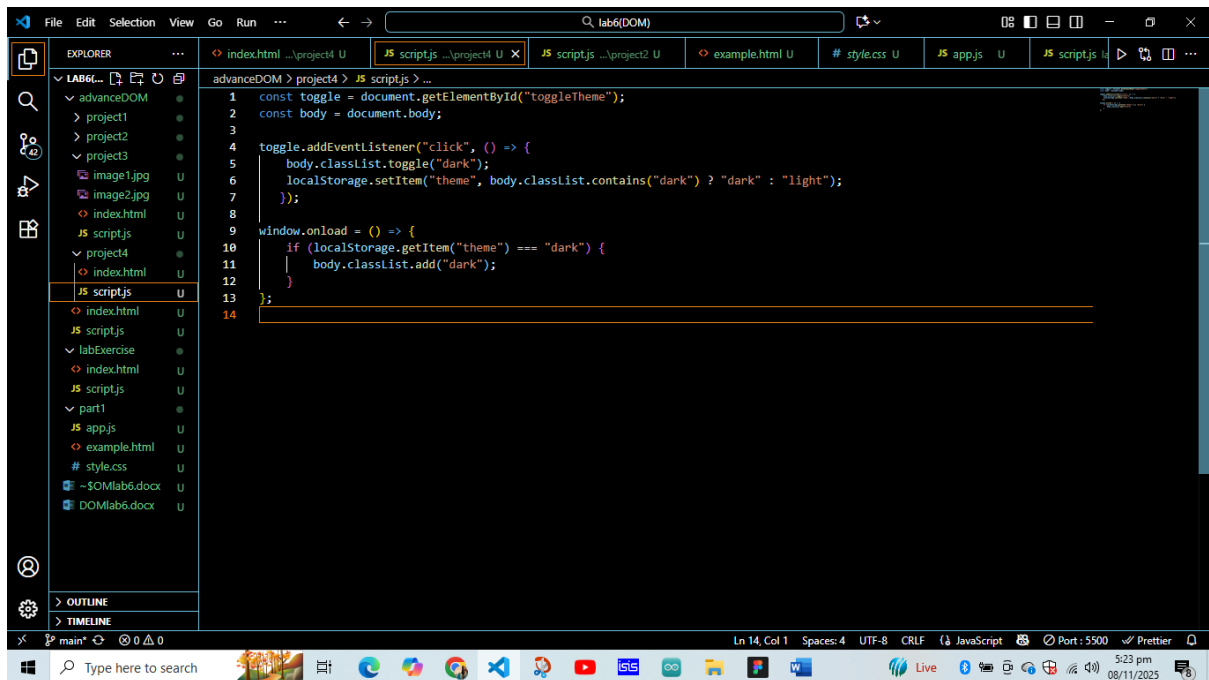


Project 4:



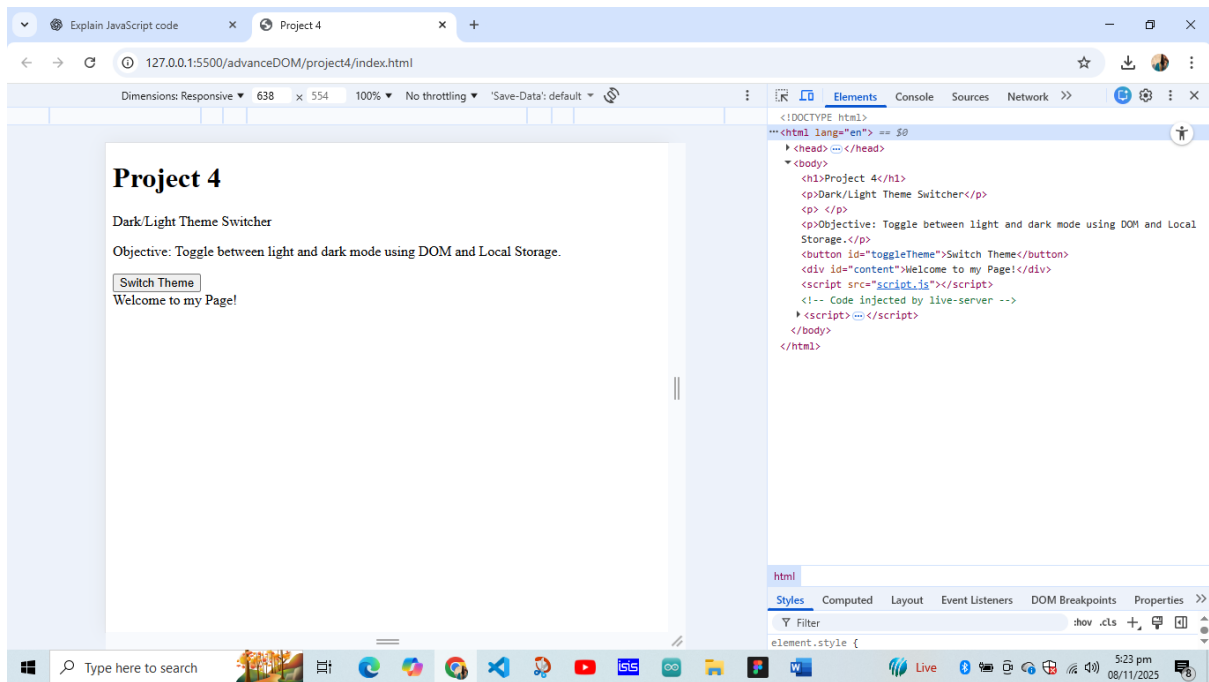
This screenshot shows the HTML structure of Project 4 in VS Code. The Explorer panel on the left lists the project files, including 'index.html' and 'script.js'. The main editor displays the content of 'index.html', which includes a DOCTYPE declaration, HTML and head tags with meta information, a body tag with a heading, a paragraph, a button, a div, and a script tag.

```
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>Project 4</title>
7 </head>
8 <body>
9   <h1>Project 4</h1>
10  <p>Dark/Light Theme Switcher</p>
11  <p>Objective: Toggle between light and dark mode using DOM and Local Storage.</p>
12  <button id="toggleTheme">Switch Theme</button>
13  <div id="content">Welcome to my Page!</div>
14  <script src="script.js"></script>
15 </body>
16 </html>
```

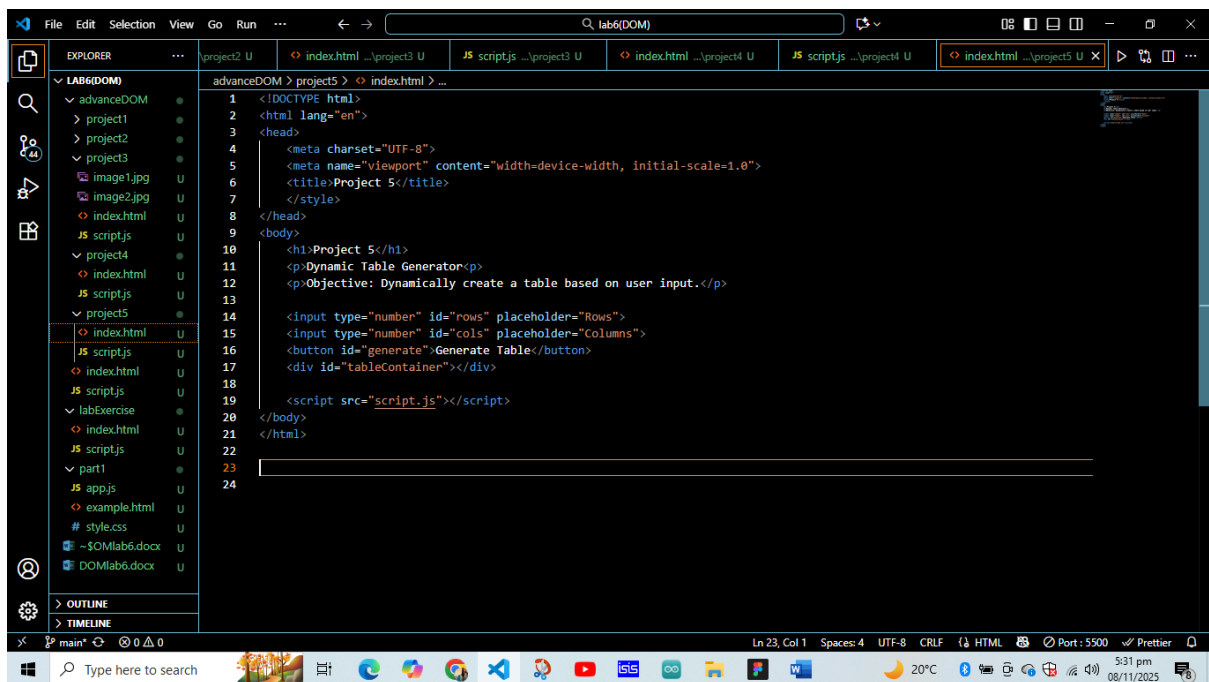


This screenshot shows the JavaScript logic for Project 4 in VS Code. The Explorer panel on the left lists the project files, including 'index.html' and 'script.js'. The main editor displays the content of 'script.js', which includes a function to toggle the theme and a window.onload event to initialize the theme.

```
1 const toggle = document.getElementById("toggleTheme");
2 const body = document.body;
3
4 toggle.addEventListener("click", () => {
5   body.classList.toggle("dark");
6   localStorage.setItem("theme", body.classList.contains("dark") ? "dark" : "light");
7 });
8
9 window.onload = () => {
10   if (localStorage.getItem("theme") === "dark") {
11     body.classList.add("dark");
12   }
13 };
14
```



Project 5:



The screenshot shows the VS Code editor with a project named 'advanceDOM'. The Explorer sidebar on the left shows a file tree with folders 'project1' through 'project5', each containing 'index.html' and 'script.js'. The active file is 'script.js' in 'project5', which contains the following JavaScript code:

```
1 document.getElementById("generate").addEventListener("click", () => {
2   let rows = document.getElementById("rows").value;
3   let cols = document.getElementById("cols").value;
4   let table = document.createElement("table");
5   table.border = "1";
6
7   for (let i = 0; i < rows; i++) {
8     let tr = document.createElement("tr");
9     for (let j = 0; j < cols; j++) {
10      let td = document.createElement("td");
11      td.textContent = `R${i+1}C${j+1}`;
12      tr.appendChild(td);
13    }
14    table.appendChild(tr);
15  }
16
17  let container = document.getElementById("tableContainer");
18  container.innerHTML = "";
19  container.appendChild(table);
20 });
21
22
```

The status bar at the bottom indicates 'Ln 22, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', 'JavaScript', 'Port: 5500', and 'Prettier'.

