



Submitted By: Alishba Asif

Sap Id: 60089

Section: BSCS-4

Subject: Problem Solving

Submitted To: Mr. Haroon

JavaScript DOM Lab Manual

Practice:

A screenshot of a code editor window titled "lab6(DOM)". The window shows an Explorer sidebar on the left with files: index.html, example.html, script.js, app.js, and style.css. The main editor area displays the content of example.html. The code uses the DOM API to manipulate an HTML document. It includes a heading "Favourite Animated Movies", a list of four items, a paragraph, and a button. Another section below it has a heading "Favourite Movies" and a list of four items. The code also includes a script tag pointing to "app.js".

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=, initial-scale=1.0">
<link rel="stylesheet" href="style.css">
<title>Document</title>
</head>
<body>
<div class="container">
<h1 id="main-heading">Favourite Animated Movies</h1>
<ul>
<li class="List-items">Kung Fu Panda 4</li>
<li class="List-items">Tom and Jerry</li>
<li class="List-items">Wish Dragon</li>
<li class="List-items">Leo</li>
</ul>
<p>This is my paragraph </p>
<button id="btn"> Button </button>
</div>
<!-- <div class="container">
<h1 id="main-heading">Favourite Movies</h1>
<ul>
<li class="List-items">Kung Fu Panda 4</li>
<li class="List-items">Tom and Jerry</li>
<li class="List-items">Wish Dragon</li>
<li class="List-items">Leo</li>
</ul>
</div> -->
<script src="app.js"></script>
</body>
</html>
```

The status bar at the bottom of the editor shows: Ln 33, Col 1, Spaces: 4, UTF-8, CRLF, HTML, Port: 5500, Prettier, 25°C, 2:30 pm, 08/11/2025.

The screenshot shows a code editor interface with the title bar "lab6(DOM)". The left sidebar has sections for "EXPLORER", "OUTLINE", and "TIMELINE". The main area displays a file named "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";
33
34
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 //innerContent
47 let text = document.getElementById("main-heading").innerContent;
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 screenshot shows a code editor interface with the title bar "lab6(DOM)". The left sidebar has sections for "EXPLORER", "OUTLINE", and "TIMELINE". The main area displays a file named "JS app.js" with the following code:

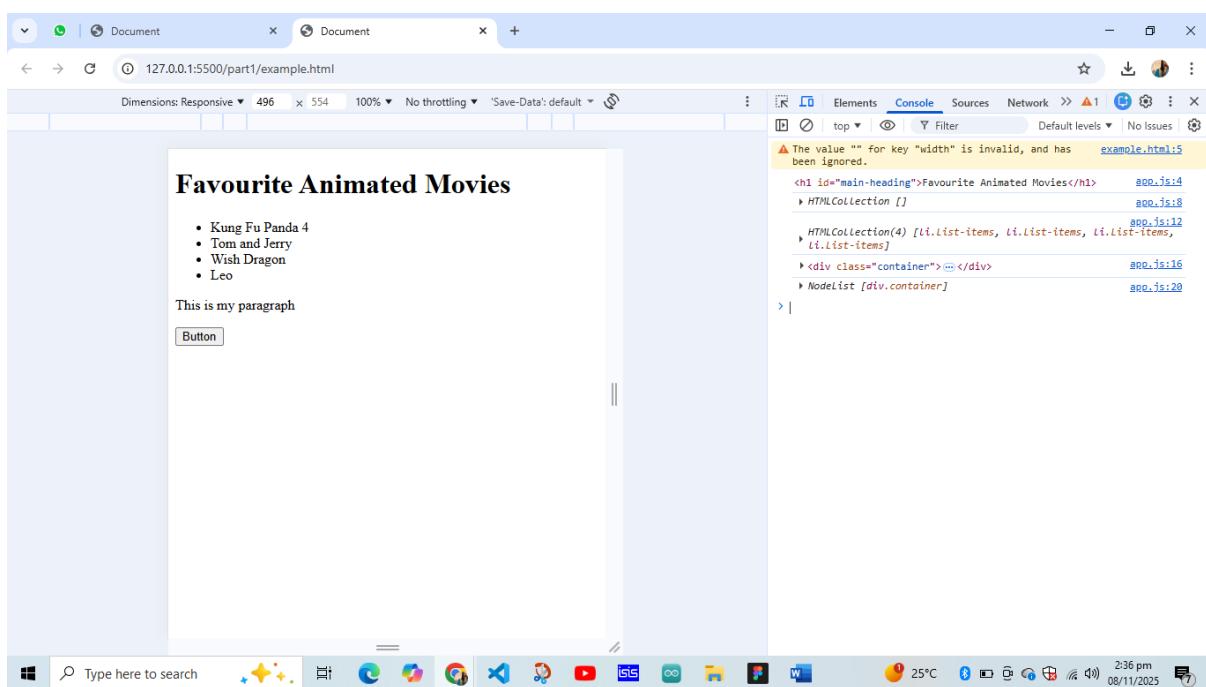
```
part1 > JS app.js > ...
33 item.style.color = "darkgreen";
34 item.style.fontWeight = "bold";
35 );
36 console.log(container);
37
38 //innerHTML
39 const h1 = document.getElementById("main-heading").innerHTML = "<b> New HTML Added! </b>";
40 console.log(h1);
41
42 //innerText
43 let text = document.getElementById("main-heading").innerText;
44 console.log(text);
45
46 //innerContent
47 let text = document.getElementById("main-heading").innerContent;
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!";
```

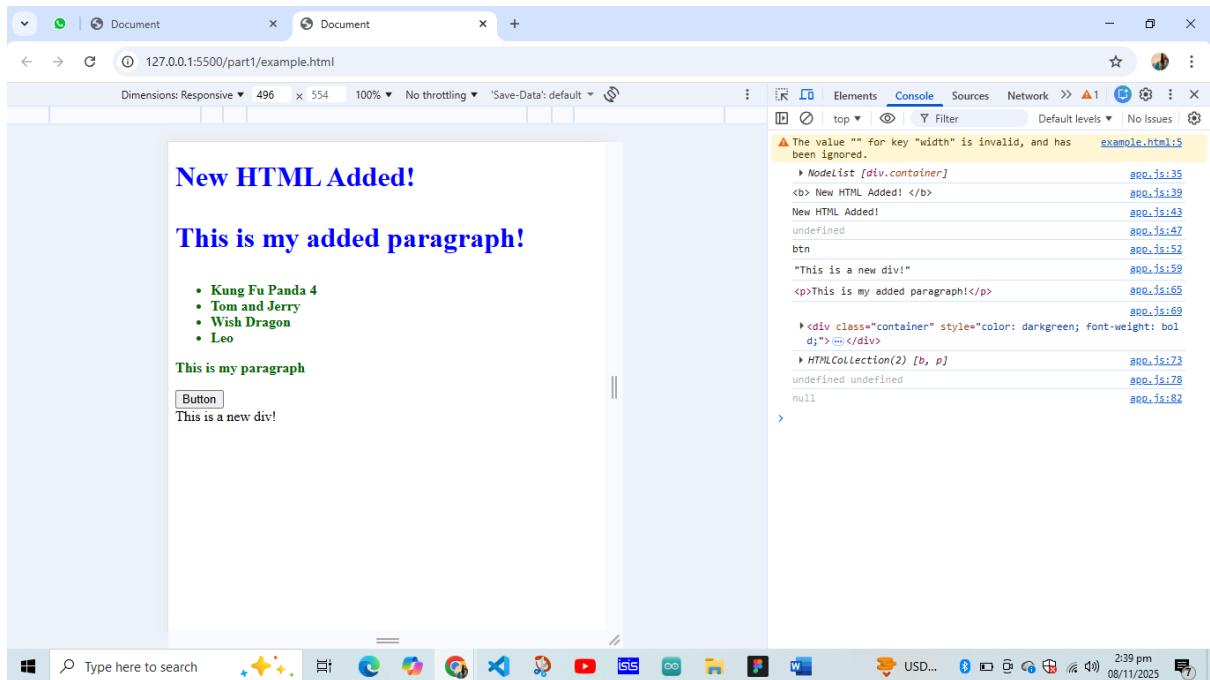
The screenshot shows a code editor interface with the title bar "lab6(DOM)". The left sidebar includes an "EXPLORER" view showing files like "index.html", "example.html", and "app.js". The main editor area displays the following JavaScript code:

```
part1 > JS app.js > ...
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("hhhhh");
89
90 //DOM Events (onclick, addEventListener())
91 document.getElementById("btn").addEventListener("click", function(){
92 | alert("Button clicked!");
93 });

```

The status bar at the bottom shows "In 33, Col 4 (1024 selected) Spaces: 4 UTF-8 CRLF JavaScript Port: 5500 Prettier".





Lab Exercises

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
  </head>
  <body>
    <div id="container">
      <h1 id="main-heading">Favourite Animated Movies</h1>
      <p id="p1">this is my paragraph</p>
      <p id="p2">Lorem ipsum dolor sit, amet consectetur adipisicing elit. Numquam quam nisi, quidem obcaecati repellendu
      <span>Some Span Text</span>
      <ul id="movie-list">
        <li class="List-items">Kung Fu Panda 4</li>
        <li class="List-items">Tom and Jerry</li>
        <li class="List-items">Wish Dragon</li>
        <li class="List-items">Leo</li>
      </ul>
    </div>
    <button id="btn"> Button</button>
    <br><br>
    <label for="text">Add Movie:</label>
    <input type="text" id="movie-input" placeholder="Add a new movie">
    <button id="add-btn">Add</button>
    <br>
    <script src="script.js"></script>
  </body>
</html>
```

The code editor interface shows the project structure under the Explorer tab, with files like index.html, example.html, app.js, script.js, and style.css. The status bar at the bottom indicates the time is 2:40 pm on 08/11/2025.

File Edit Selection View Go Run ... labExercise > JS script.js U

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

Ln 57, Col 1 Spaces: 4 UTF-8 CRLF ↵ JavaScript ⚡ Port: 5500 ✅ Prettier

File Edit Selection View Go Run ... labExercise > JS script.js U

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

Ln 32, Col 25 Spaces: 4 UTF-8 CRLF ↵ JavaScript ⚡ Port: 5500 ✅ Prettier

The screenshot shows a browser window with two tabs open, both titled "Document". The active tab displays a web page with the following content:

Favourite Animated Movies

This is the new paragraph replacing the original one.

Some Span Text

- Kung Fu Panda 4
- Tom and Jerry
- Wish Dragon
- Leo

Button

Add Movie:

This is my new Div

The browser's developer tools are open, specifically the Console tab, which shows the following logs:

```
<h1 id="main-heading">Favourite Animated Movies</h1> script.js:20
<p id="p1"> this is my paragraph </p> script.js:28
><p id="p2"> ...</p>
<span>Some Span Text</span> script.js:20
><ul id="movie-list">...</ul> script.js:20
>
```

The screenshot shows the same browser setup as the first one, but with a noticeable difference: the paragraph element with id="p2" has been removed from the DOM. This change is reflected in the browser's developer tools Console tab, which now includes a log entry indicating the removal of the p2 element:

```
p2 has been removed after 1 second. script.js:33
>
```

Advanced JavaScript DOM Lab Manual

Practice:

The screenshot shows a code editor window with the title bar "lab6(DOM)". The left sidebar has an "EXPLORER" tab open, showing a file tree with "LAB6(DOM)" expanded, containing "advanceDOM", "index.html", "JS script.js", "part1", "app.js", "example.html", "# style.css", and "DOMLab6.docx". The main editor area displays the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>DOM Manipulation Demo</title>
    <style>
        .completed { text-decoration: line-through; color: gray; }
        .active { background-color: lightgreen; padding: 5px; }
        .highlight { border: 2px solid orange; }
        .hidden { display: none; }
        ul li { cursor: pointer; padding: 3px; }
    </style>
</head>
<body>
    <h1>Favourite Animated Movies</h1>
    <ul id="movie-list">
        <li>Kung Fu Panda</li>
        <li>Wish Dragon</li>
        <li>Tom and Jerry</li>
    </ul>
    <input type="text" id="movie-input" placeholder="Add a new movie">
    <button id="add-btn">Add</button>
    <button id="clone-btn">Clone First Movie</button>
    <div id="box" class="hidden">I am a box</div>
    <button id="toggle-box">Toggle Box Classes</button>
    <p id="welcome"></p>
    <script src="script.js"></script>
</body>
</html>
```

The status bar at the bottom shows "Ln 33, Col 1" and "Port: 5500".

The screenshot shows a code editor window with the title bar "lab6(DOM)". The left sidebar has an "EXPLORER" tab open, showing a file tree with "LAB6(DOM)" expanded, containing "advanceDOM", "index.html", "JS script.js", "part1", "app.js", "example.html", "# style.css", and "DOMLab6.docx". The main editor area displays the following JavaScript code:

```
document.getElementById("movie-list").addEventListener("click", function(e) {
    if (e.target.tagName === "LI") {
        e.target.classList.toggle("completed");
    }
});

let ul = document.getElementById("movie-list");
console.log("Parent of ul:", ul.parentNode);
console.log("First movie:", ul.firstElementChild.textContent);
console.log("Last movie:", ul.lastElementChild.textContent);
console.log("All children:", ul.children);

document.getElementById("clone-btn").addEventListener("click", function() {
    let firstItem = ul.querySelector("li");
    let copy = firstItem.cloneNode(true);
    ul.appendChild(copy);
});

localStorage.setItem("username", "Haroon");
let name = localStorage.getItem("username");
document.getElementById("welcome").textContent = "Welcome " + name;

let box = document.getElementById("box");
document.getElementById("toggle-box").addEventListener("click", function() {
    box.classList.toggle("active");
    box.classList.toggle("highlight");
    box.classList.toggle("hidden");
});

let input = document.getElementById("movie-input");
let addBtn = document.getElementById("add-btn");
```

The status bar at the bottom shows "Ln 38, Col 6" and "Port: 5500".

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

- File Explorer:** Shows a project structure for "LAB6(DOM)" containing files like "index.html", "script.js", "app.js", and "example.html".
- Code Editor:** Displays the content of "script.js". The code is as follows:

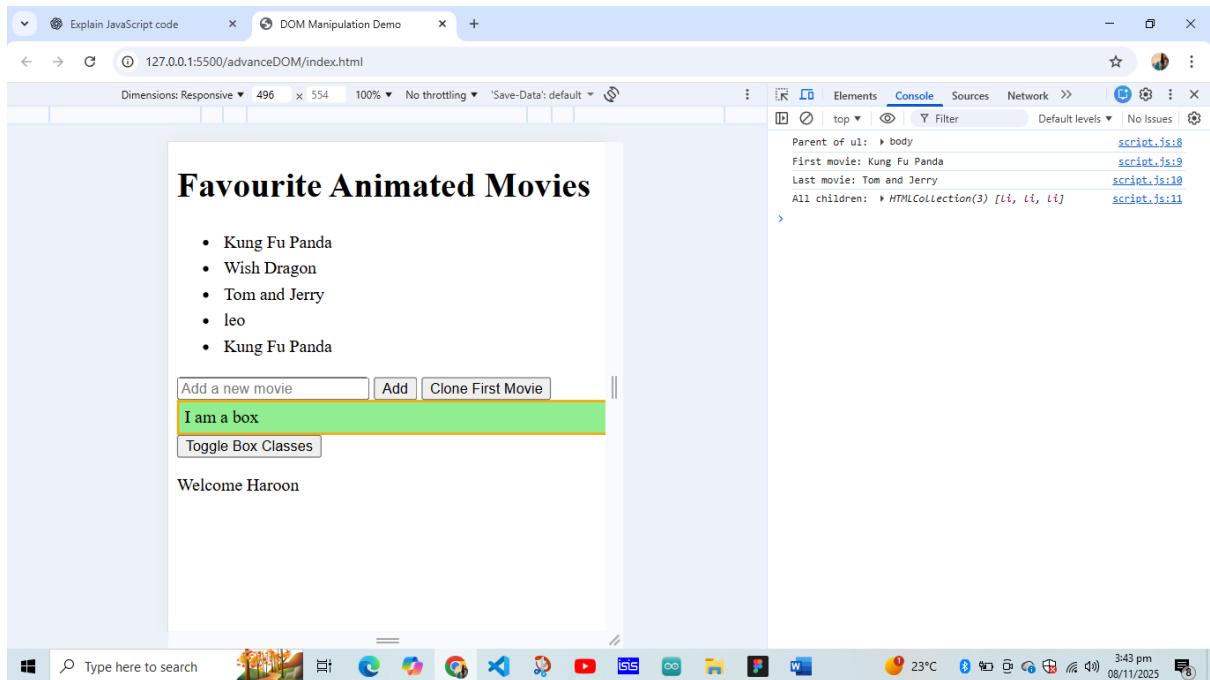
```
32 33  function addMovie() { 34      let movieName = input.value.trim(); 35      if (movieName === "") { 36          alert("Please enter a movie name!"); 37          return; 38      } 39      let template = `<li class="item">${movieName}</li>`;  
40      ul.insertAdjacentHTML("beforeend", template);  
41      input.value = "";  
42      input.focus();  
43  }  
44  
45  addBtn.addEventListener("click", addMovie);  
46  input.addEventListener("keydown", function(e) {  
47      if (e.key === "Enter") addMovie();  
48  });  
49  
50  
```

Below the code editor, the status bar shows: Ln 50, Col 1 | Spaces: 4 | UTF-8 | CRLF | JavaScript | Port: 5500 | Prettier.

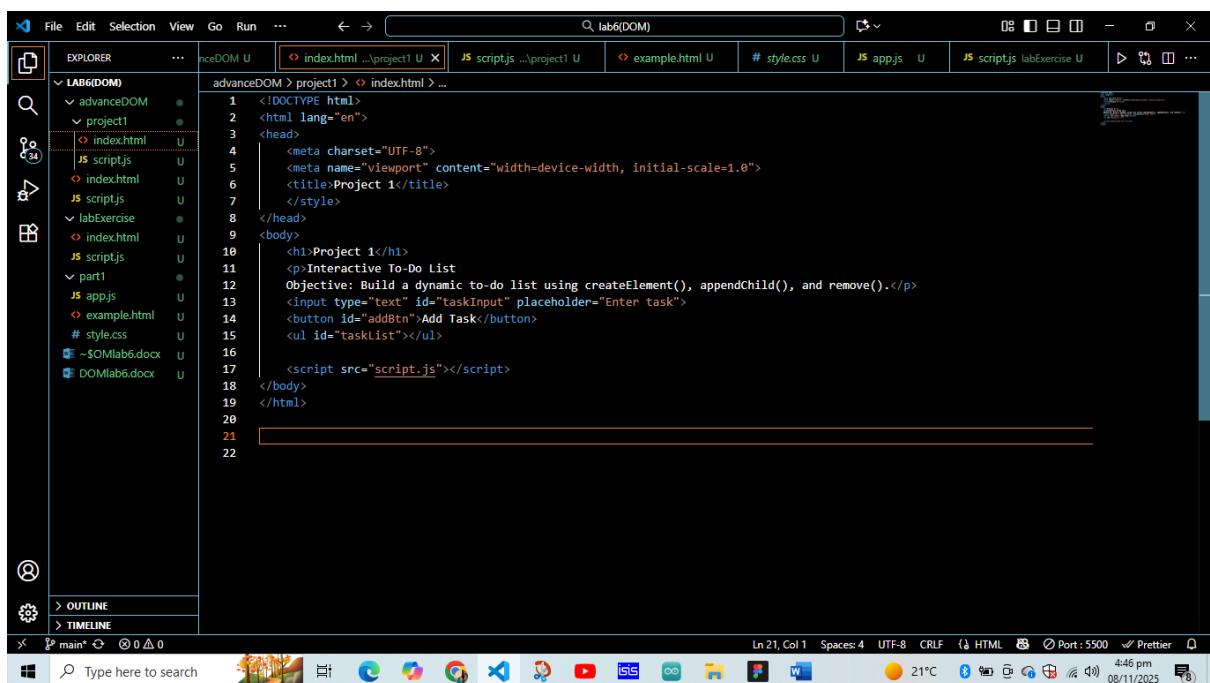
The screenshot shows a browser window with the following details:

- Title Bar:** Explain JavaScript code - DOM Manipulation Demo
- Address Bar:** 127.0.0.1:5500/advanceDOM/index.html
- Page Content:** A heading "Favourite Animated Movies" followed by a list
 - Kung Fu Panda
 - Wish Dragon
 - Tom and JerryA form with fields "Add a new movie", "Add", and "Clone First Movie", and a button "Toggle Box Classes". Below the form is the text "Welcome Haroon".
- Developer Tools:** The "Console" tab is selected, showing the following output:

```
Parent of ul: > body  
First movie: Kung Fu Panda  
script.js:8  
Last movie: Tom and Jerry  
script.js:10  
All children: > HTMLCollection(3) [li, li, li]  
script.js:11
```
- Bottom Bar:** Windows taskbar with various pinned icons.



Project 1:



The screenshot shows a code editor window titled "lab6(DOM)". The left sidebar displays a file tree for a project named "project1". The "JS script.js" file is open in the main editor area, containing the following JavaScript code:

```
1 const input = document.getElementById("taskInput");
2 const list = document.getElementById("taskList");
3 document.getElementById("addBtn").addEventListener("click", () => {
4   if (input.value.trim() !== "") {
5     let li = document.createElement("li");
6     li.textContent = input.value;
7     li.addEventListener("click", () => li.remove());
8     list.appendChild(li);
9     input.value = "";
10  }
11 });
12 
```

The status bar at the bottom indicates "Ln 13, Col 1" and "Port: 5500".

The screenshot shows a browser window with the address bar set to "127.0.0.1:5500/advanceDOM/project1/index.html". The browser toolbar includes icons for various applications like File Explorer, Task View, and Edge. The developer tools are open, with the "Elements" tab selected. The left panel shows the DOM structure of the page, which includes an

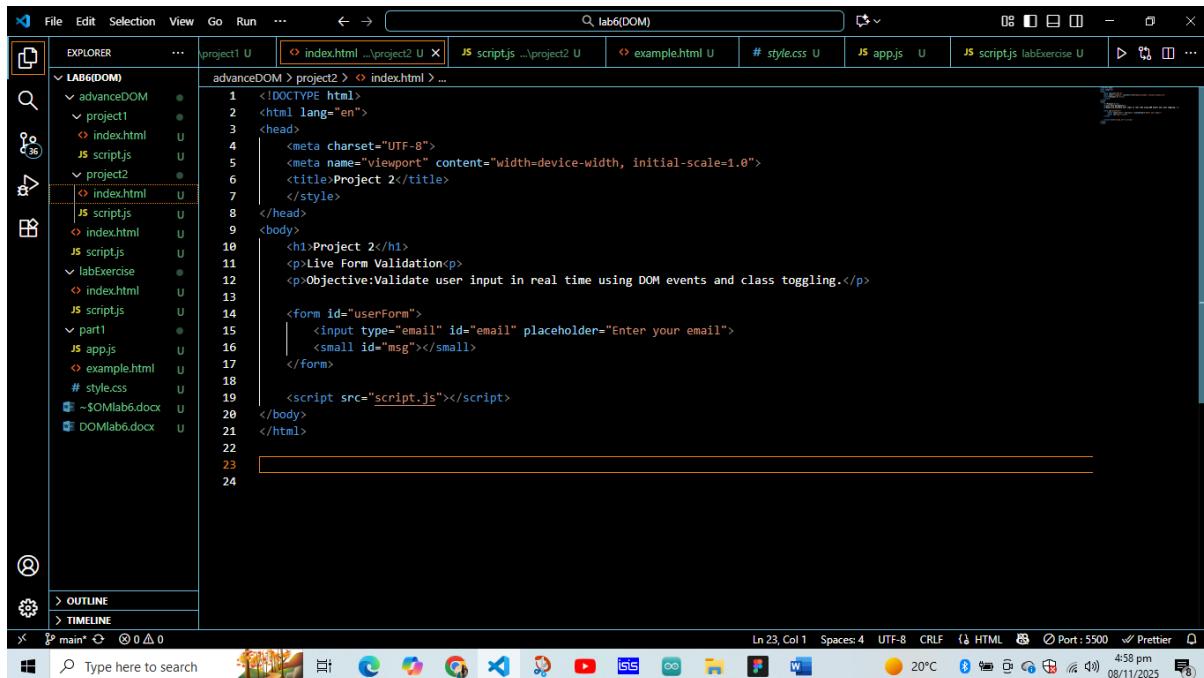
Project 1

 heading and a form with an input field and a button. The right panel shows the "html" tab of the element inspector, displaying the raw HTML code:

```
<!DOCTYPE html>
<html lang="en"> == $0
  > <head> ... </head>
  > <body>
    <h1>Project 1</h1>
    <p>...</p>
    <input type="text" id="taskInput" placeholder="Enter task">
    <button id="addBtn">Add Task</button>
    <ul id="taskList">...</ul>
    <script src="script.js"></script>
    <!-- Code injected by live-server -->
    <script>...</script>
  </body>
</html>
```

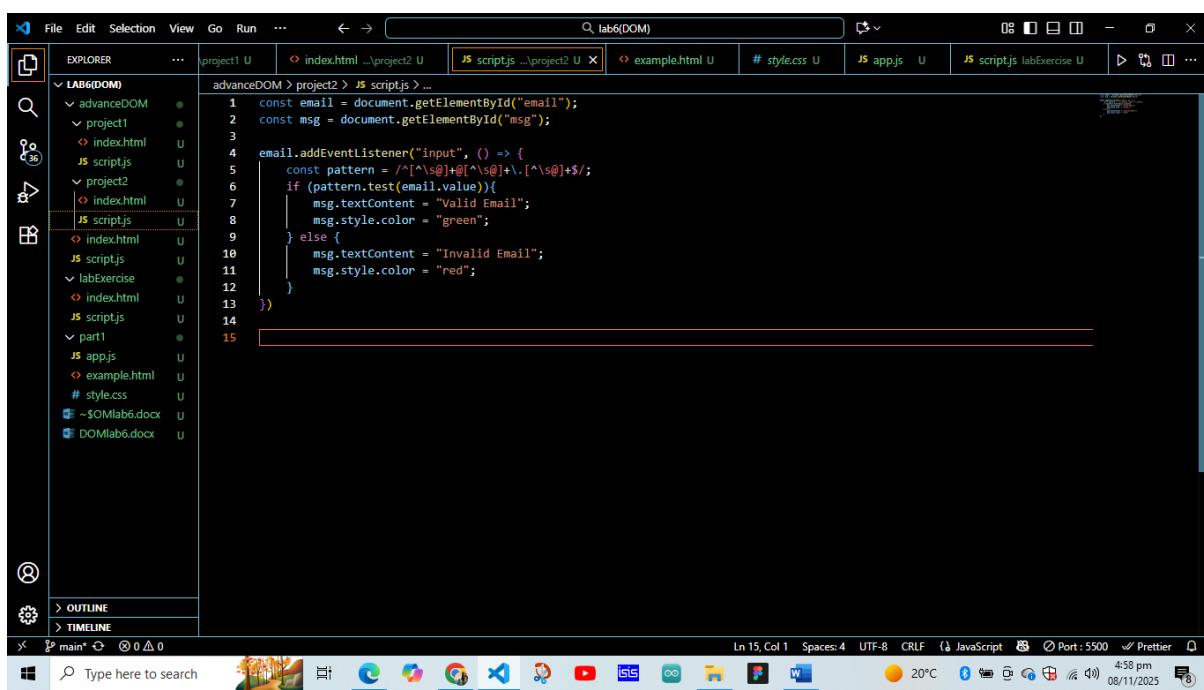
The status bar at the bottom indicates "21°C" and "08/11/2025".

Project 2:



The screenshot shows a Windows desktop environment with a code editor window titled "lab6(DOM)". The file being edited is "index.html" located in the "project2" folder. The code uses JavaScript to validate an email input field in real-time.

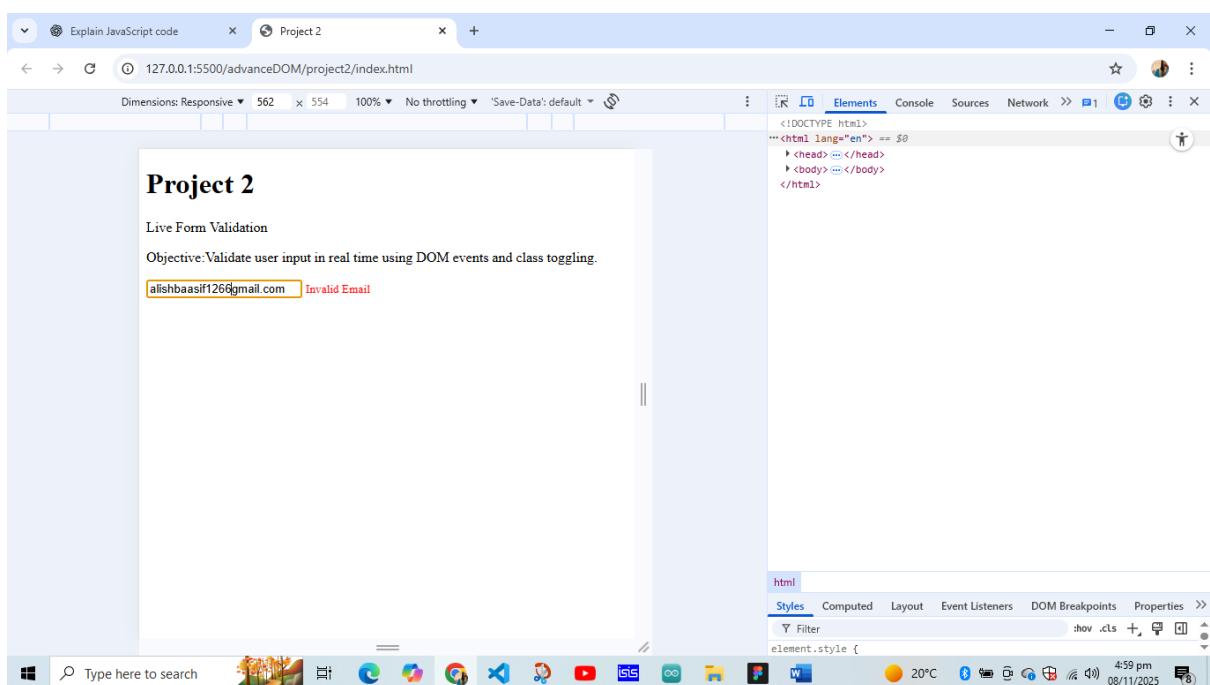
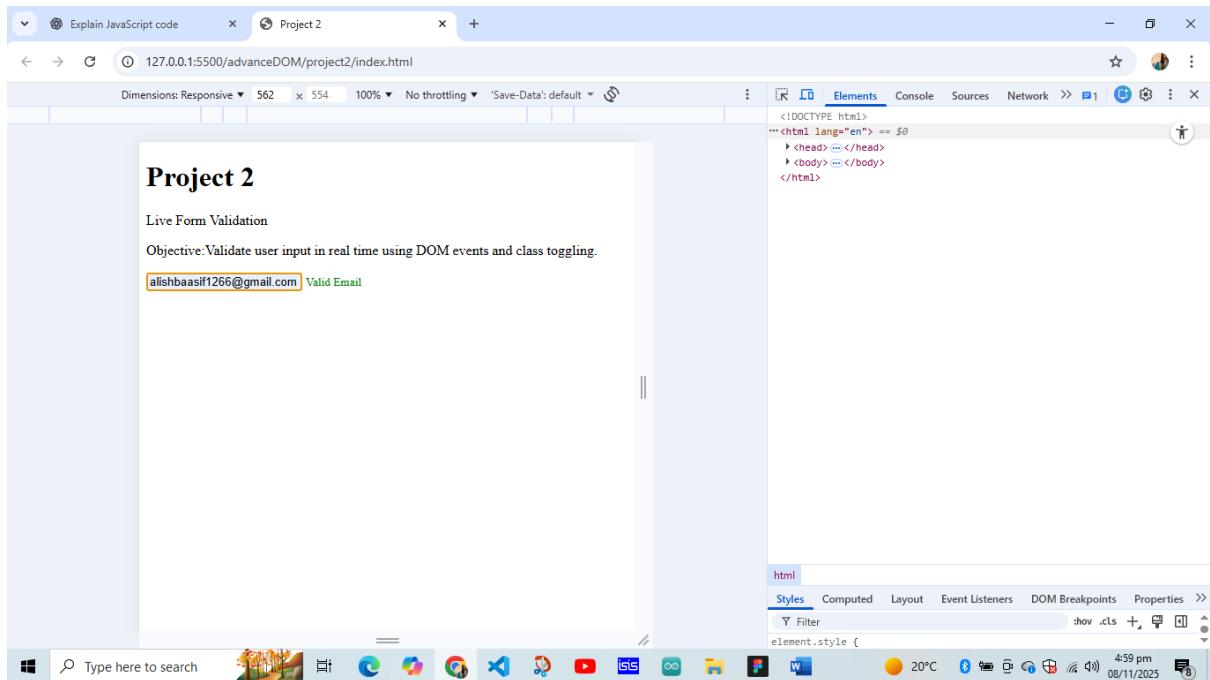
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Project 2</title>
</head>
<body>
<h1>Project 2</h1>
<p>Live Form Validation</p>
<p>Objective: Validate user input in real time using DOM events and class toggling.</p>
<form id="userForm">
<input type="email" id="email" placeholder="Enter your email">
<small id="msg"></small>
</form>
<script src="script.js"></script>
</body>
</html>
```



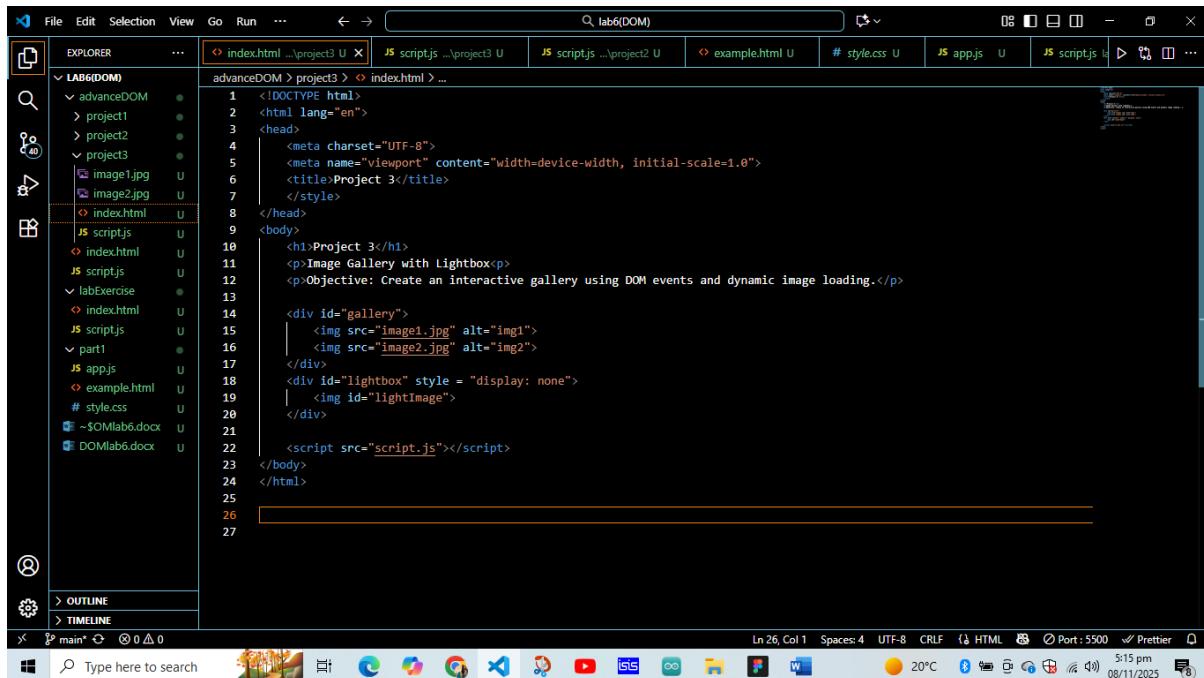
The screenshot shows a Windows desktop environment with a code editor window titled "lab6(DOM)". The file being edited is "script.js" located in the "project2" folder. The code adds an event listener to an input field to validate email addresses.

```
const email = document.getElementById("email");
const msg = document.getElementById("msg");

email.addEventListener("input", () => {
  const pattern = /^[^@\s]+@[^\s]+\.[^\s]+$/;
  if (pattern.test(email.value)) {
    msg.textContent = "Valid Email";
    msg.style.color = "green";
  } else {
    msg.textContent = "Invalid Email";
    msg.style.color = "red";
  }
});
```



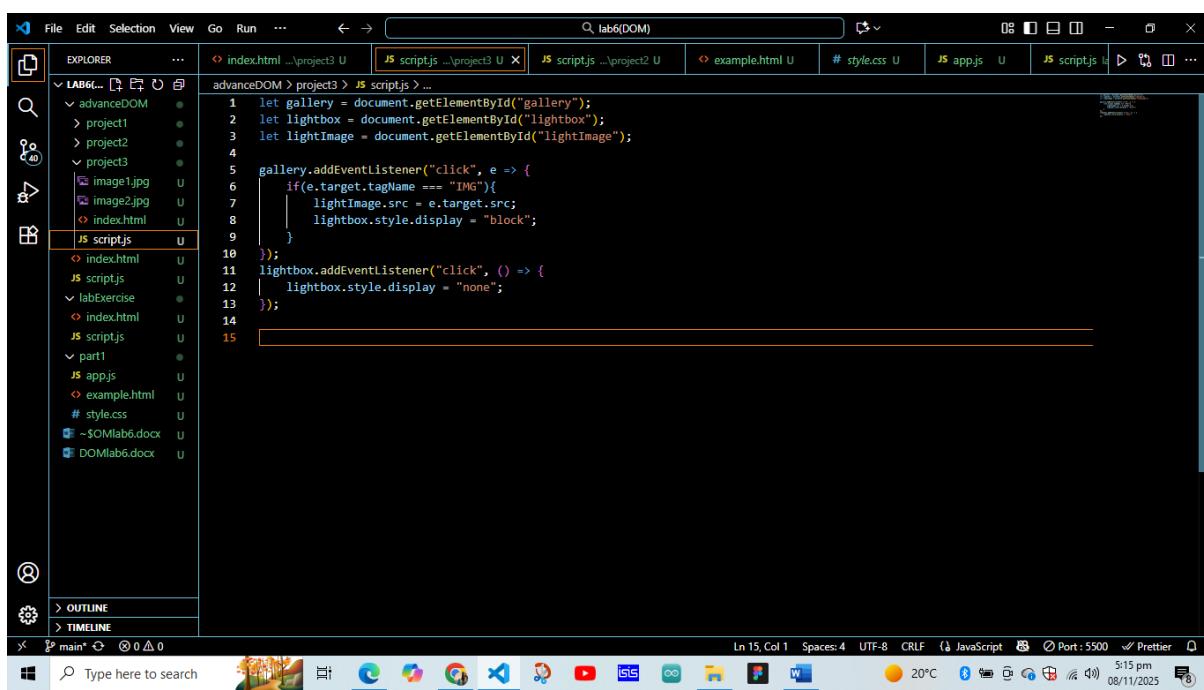
Project 3:



The screenshot shows a code editor window titled "lab6(DOM)". The left sidebar displays a file tree under "LAB6(DOM)" with files like advanceDOM, project1, project2, project3, image1.jpg, image2.jpg, index.html, JS script.js, labExercise, index.html, JS script.js, part1, JS app.js, example.html, # style.css, and two .docx files. The main editor area shows the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Project 3</title>
</head>
<body>
    <h1>Project 3</h1>
    <p>Image Gallery with Lightbox</p>
    <p>Objective: Create an interactive gallery using DOM events and dynamic image loading.</p>
    <div id="gallery">
        
        
    </div>
    <div id="lightbox" style = "display: none">
        <img id="lightImage">
    </div>
<script src="script.js"></script>
</body>
</html>
```

The status bar at the bottom indicates "Ln 26, Col 1" and "Port: 5500".

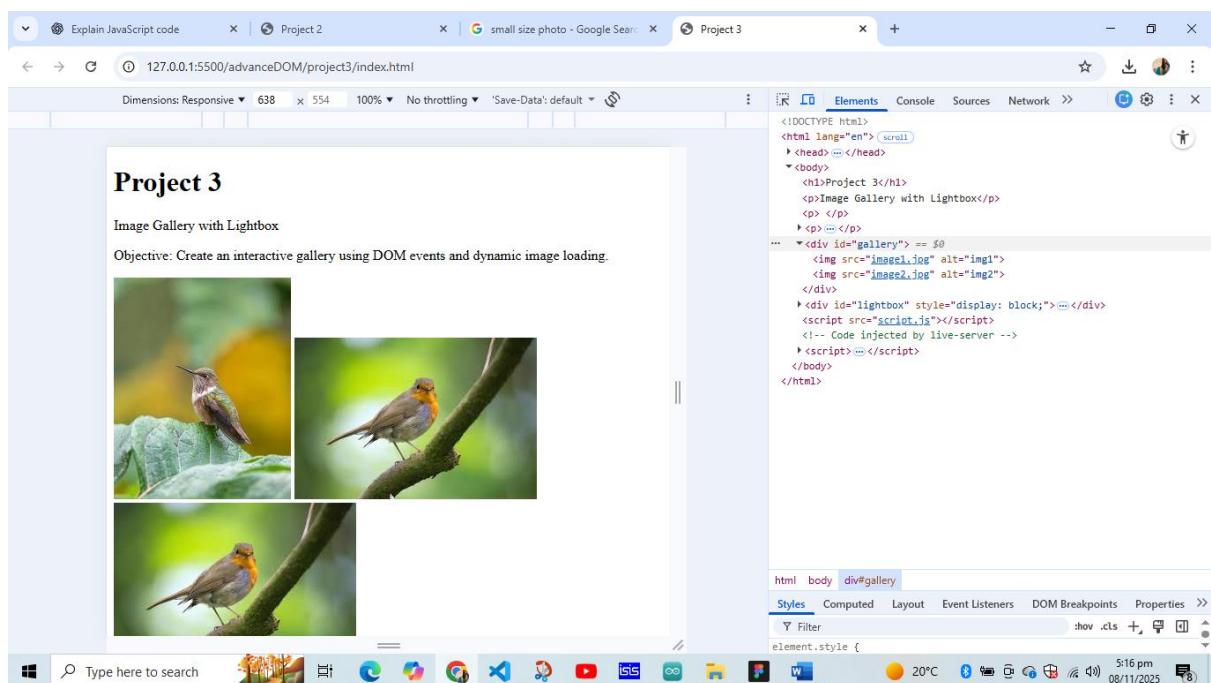
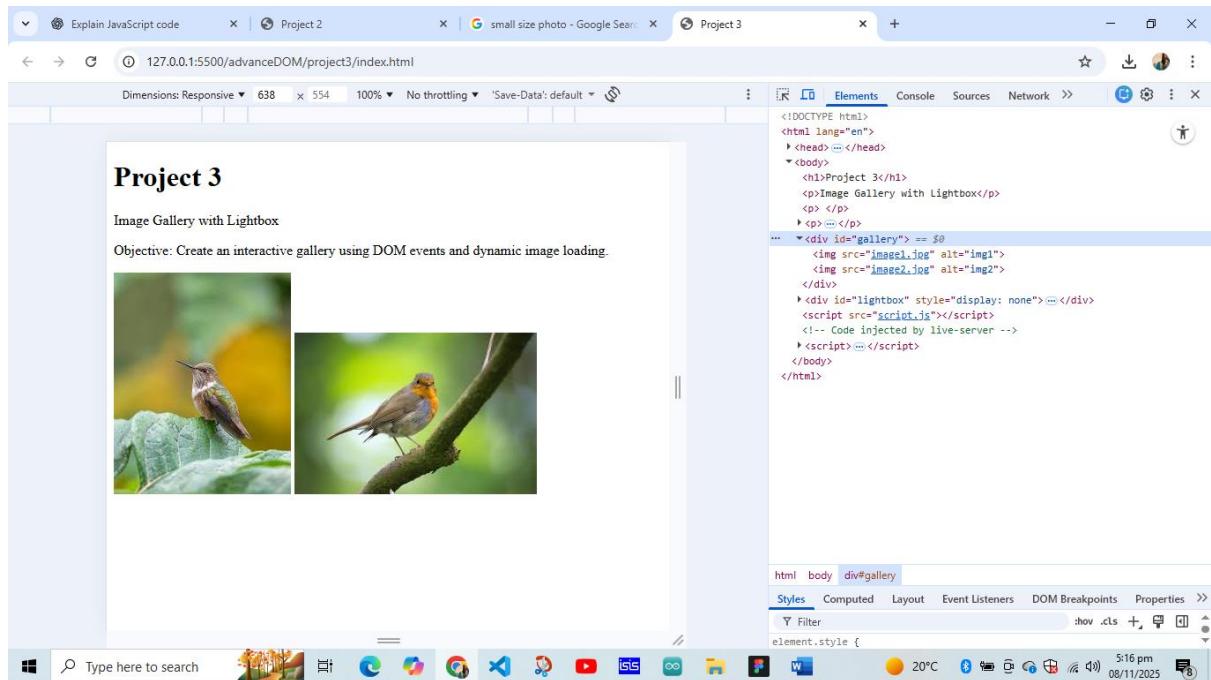


The screenshot shows a code editor window titled "lab6(DOM)". The left sidebar displays a file tree under "LAB6(DOM)" with files like advanceDOM, project1, project2, project3, image1.jpg, image2.jpg, index.html, JS script.js, labExercise, index.html, JS script.js, part1, JS app.js, example.html, # style.css, and two .docx files. The main editor area shows the following JavaScript code:

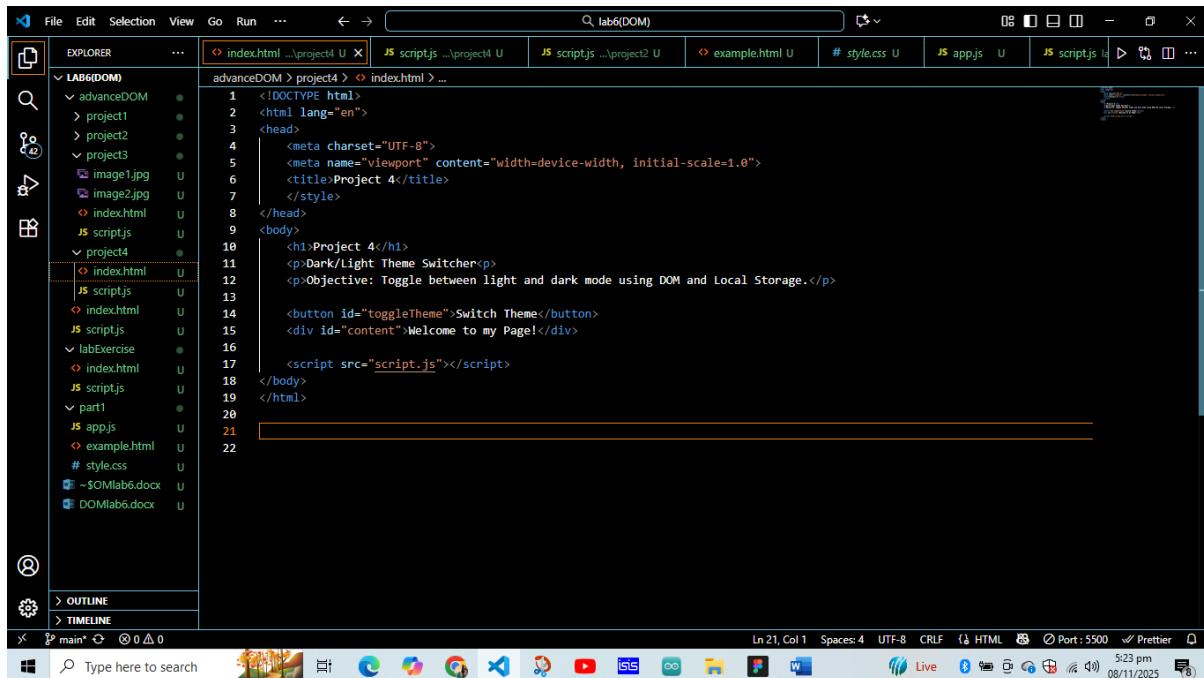
```
let gallery = document.getElementById("gallery");
let lightbox = document.getElementById("lightbox");
let lightImage = document.getElementById("lightImage");

gallery.addEventListener("click", e => {
    if(e.target.tagName === "IMG"){
        lightImage.src = e.target.src;
        lightbox.style.display = "block";
    }
});
lightbox.addEventListener("click", () => {
    lightbox.style.display = "none";
});
```

The status bar at the bottom indicates "Ln 15, Col 1" and "Port: 5500".



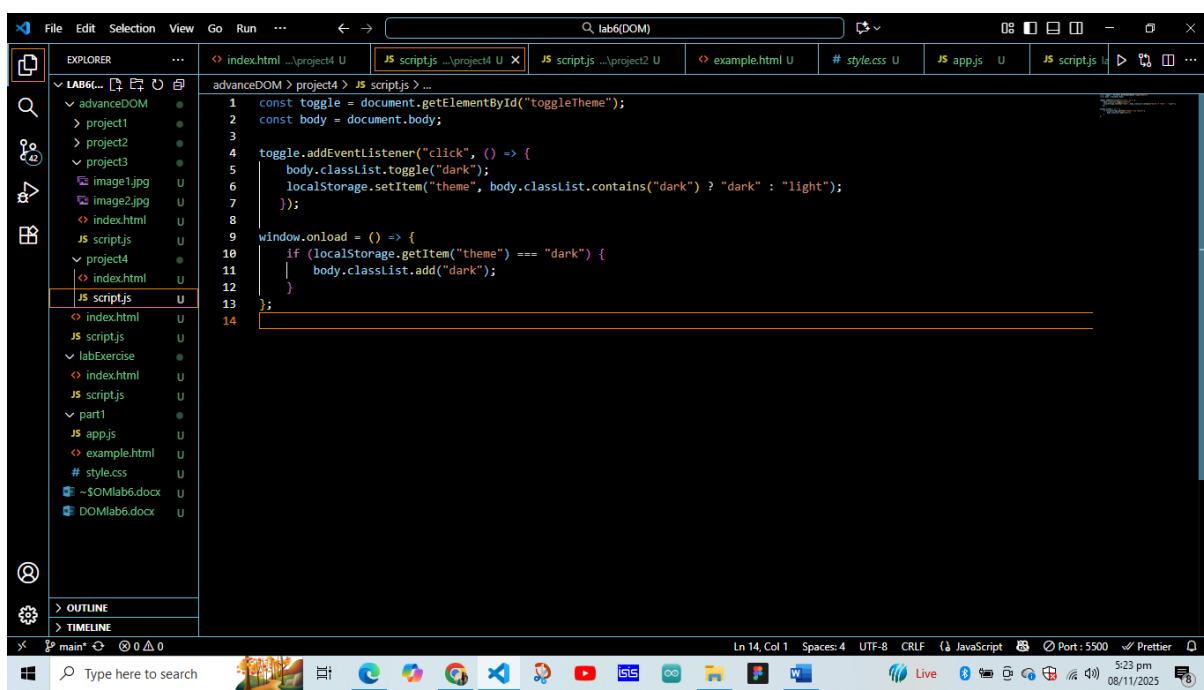
Project 4:



The screenshot shows a code editor interface with the title bar "lab6(DOM)". The left sidebar is titled "EXPLORER" and lists project files: advanceDOM, project1, project2, project3, image1.jpg, image2.jpg, index.html, JS script.js, project4, labExercise, index.html, JS script.js, part1, JS app.js, example.html, # style.css, and two Microsoft Word documents. The main editor area displays the content of index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Project 4</title>
</head>
<body>
    <h1>Project 4</h1>
    <p>Dark/Light Theme Switcher</p>
    <p>Objective: Toggle between light and dark mode using DOM and Local Storage.</p>
    <button id="toggleTheme">Switch Theme</button>
    <div id="content">Welcome to my Page!</div>
    <script src="script.js"></script>
</body>
</html>
```

The status bar at the bottom shows "Ln 21, Col 1" and "Port: 5500".



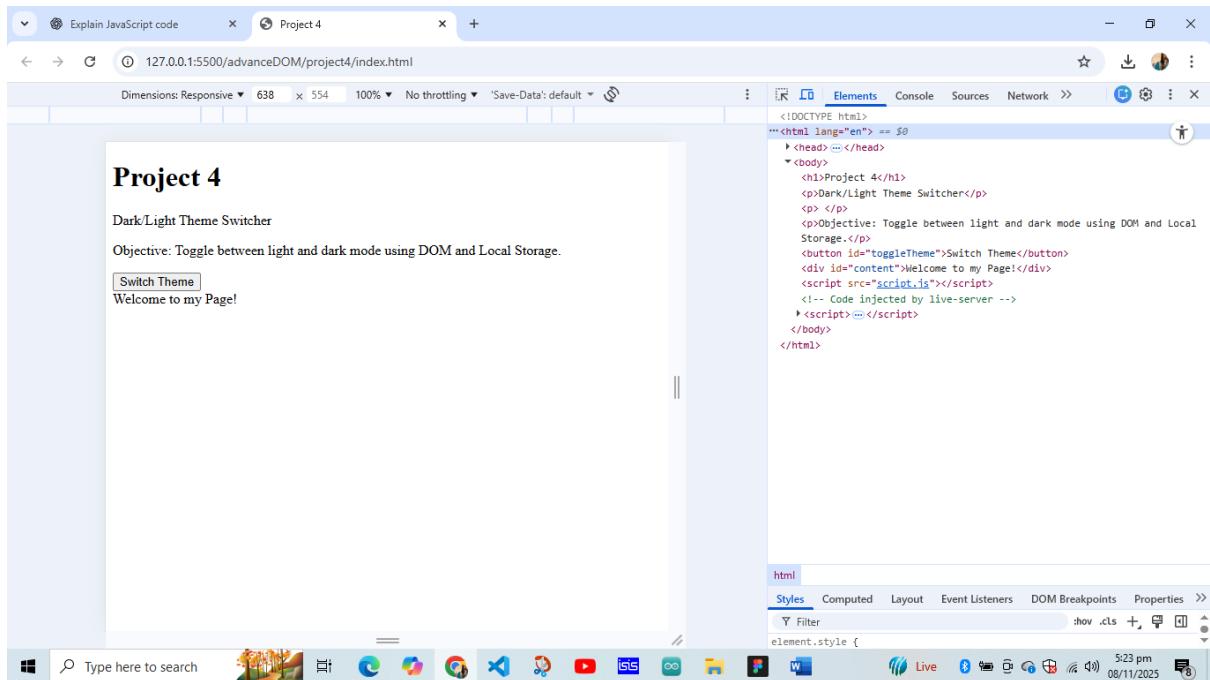
The screenshot shows a code editor interface with the title bar "lab6(DOM)". The left sidebar is titled "EXPLORER" and lists project files: advanceDOM, project1, project2, project3, image1.jpg, image2.jpg, index.html, JS script.js, project4, labExercise, index.html, JS script.js, part1, JS app.js, example.html, # style.css, and two Microsoft Word documents. The main editor area displays the content of JS script.js:

```
const toggle = document.getElementById("toggleTheme");
const body = document.body;

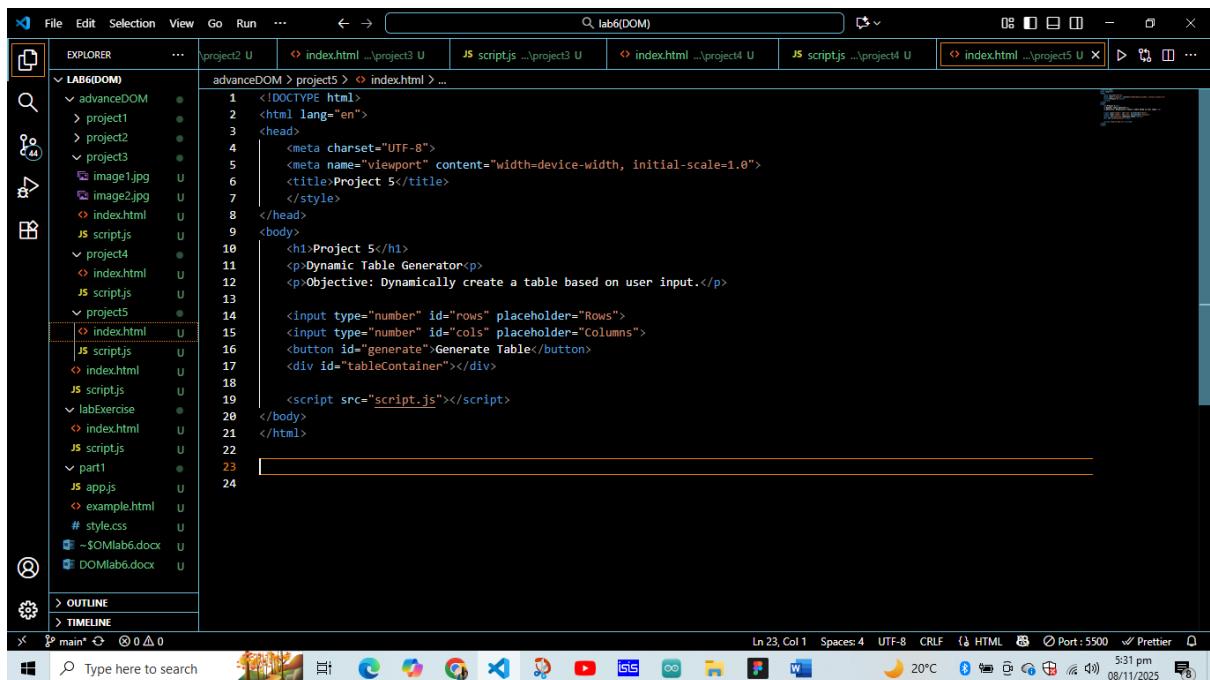
toggle.addEventListener("click", () => {
    body.classList.toggle("dark");
    localStorage.setItem("theme", body.classList.contains("dark") ? "dark" : "light");
});

window.onload = () => {
    if (localStorage.getItem("theme") === "dark") {
        body.classList.add("dark");
    }
};
```

The status bar at the bottom shows "Ln 14, Col 1" and "Port: 5500".



Project 5:



File Edit Selection View Go Run ...

LAB6... JS scriptjs ...\\project3 U JS scriptjs ...\\project3 U index.html ...\\project4 U JS scriptjs ...\\project4 U index.html ...\\project5 U JS scriptjs ...\\project5 U

EXPLORER

```

LAB6... JS scriptjs ...\\project3 U
  advanceDOM > project5 > JS scriptjs ...
    > project1
    > project2
    > project3
      image1.jpg
      image2.jpg
      index.html
      JS scriptjs
    > project4
      index.html
      JS scriptjs
    > project5
      index.html
      JS scriptjs
      JS scriptjs
      labExercise
        index.html
        JS scriptjs
      part1
        app.js
        example.html
        # style.css
      ~$OMlab6.docx
      DOMLab6.docx
  > OUTLINE
  > TIMELINE

```

Ln 22, Col 1 Spaces: 4 UTF-8 CRLF JavaScript Port: 5500 Prettier

Type here to search

Windows Taskbar: File Explorer, Edge, File Manager, Google Chrome, FileZilla, YouTube, Microsoft Edge, File Manager, Microsoft Word, 20°C, 5:31 pm, 08/11/2025

Explain JavaScript code Project 5

Dimensions: Responsive 688 x 554 100% No throttling Save-Data: default

Project 5

Dynamic Table Generator

Objective: Dynamically create a table based on user input.

5	5	Generate Table		
R1C1	R1C2	R1C3	R1C4	R1C5
R2C1	R2C2	R2C3	R2C4	R2C5
R3C1	R3C2	R3C3	R3C4	R3C5
R4C1	R4C2	R4C3	R4C4	R4C5
R5C1	R5C2	R5C3	R5C4	R5C5

Elements Console Sources Network

<!DOCTYPE html>
...<html lang="en"> == \$0
 > <head> ...</head>
 > <body> ...</body>
</html>

html Styles Computed Layout Event Listeners DOM Breakpoints Properties

Type here to search

Windows Taskbar: File Explorer, Edge, File Manager, Google Chrome, FileZilla, YouTube, Microsoft Edge, File Manager, Microsoft Word, 20°C, 5:31 pm, 08/11/2025