FSD LAB

1. a. Write a script that Logs "Hello, World!" to the console. Create a script that calculates the sum of two numbers and displays the result in an alert box.

b. Create an array of 5 cities and perform the following operations:

Log the total number of cities. Add a new city at the end. Remove the first city. Find and log the index of a specific city.

1. **Install Node.js**:
    - **Download**: Go to the Node.js official website and download the latest version suitable for Windows.
    - **Install**: Run the downloaded installer and follow the on-screen instructions to install Node.js. This will also install npm (Node Package Manager), which is useful for managing JavaScript packages.
2. **Set Up Your Environment**:
    - **Text Editor**: Use a text editor like Visual Studio Code or any other text editor of your choice.
    - **Command Line**: Use the Command Prompt or PowerShell to run your JavaScript files.
3. **Running the Script**:
    - **Save Your Script**: Save the above JavaScript code in a file with a .js extension, e.g., script.js.
    - **Execute the Script**:
        - Open the Command Prompt or PowerShell.
        - Navigate to the directory where your script.js file is located.
        - Run the script using Node.js by typing:

shell

node script.js

- This will execute the JavaScript code and you will see the output in the console.

By following these steps, you can run and test the JavaScript programs on your Windows desktop.

To know the node.js version
==========

node -v

experiment1a.js

```
// Script to log "Hello, World!" to the console
console.log("Hello, World!");

// Script to calculate the sum of two numbers and display the result in an alert box
function calculateSum(a, b) {
   return sum = a + b;
}

// Example usage
if (typeof alert === "function") {
   alert("The sum of 5 and 7 is: " + calculateSum(5, 7));
} else {
   console.log("The sum of 5 and 7 is: " + calculateSum(5, 7));
}
```

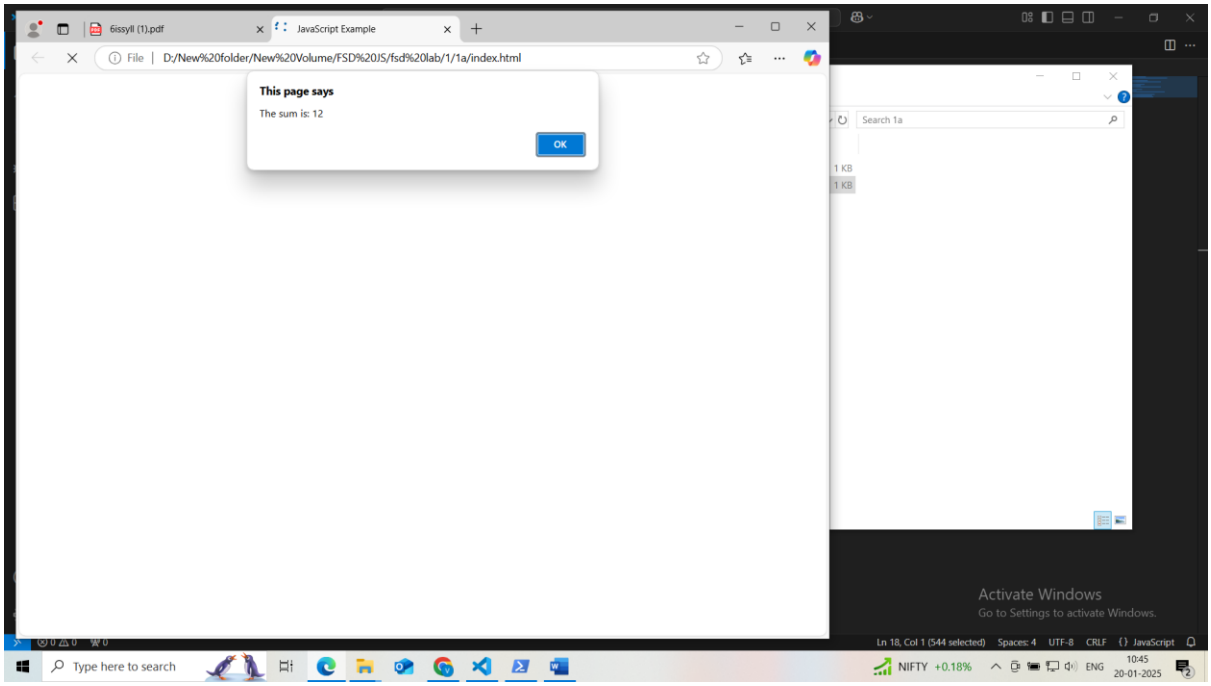
index.html to make working of alert function

```
<!DOCTYPE html>
<html>
```

```
<head>
   <title>JavaScript Example</title>
</head>
<body>
   <script src="experiment1a.js"></script>
</body>
</html>
```

o/p:

just click on index.html icon to see the result.



Some other way like whole thing implemented in single html file

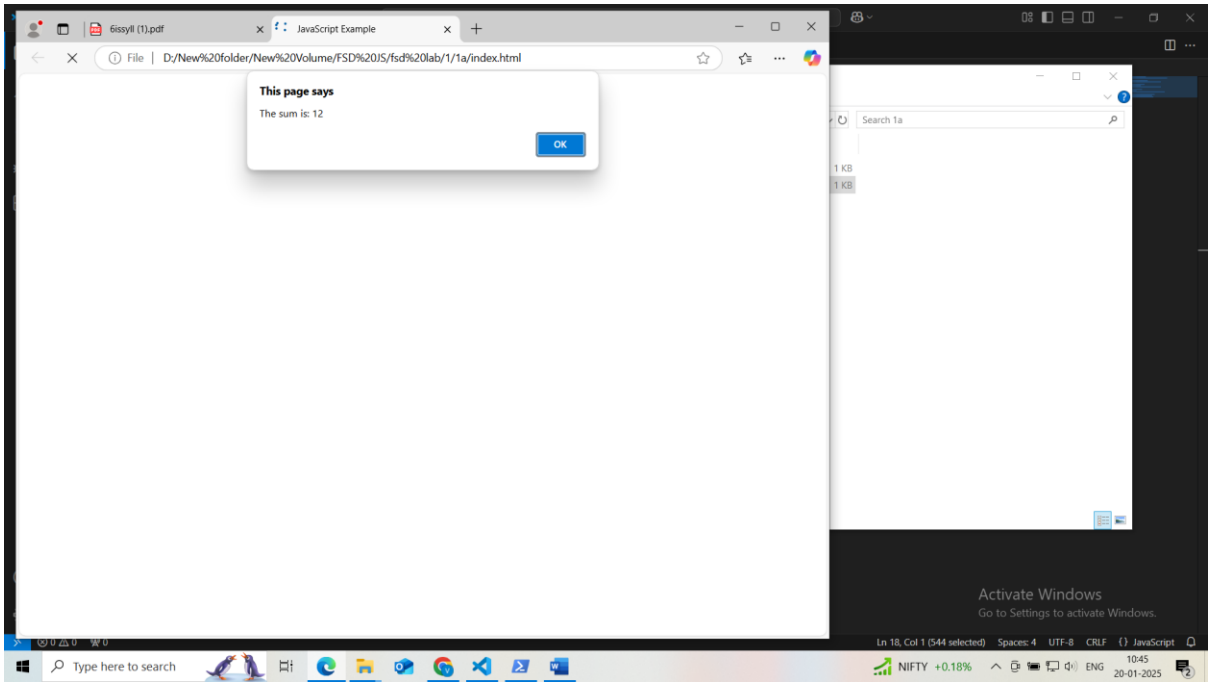
1a.html

==========

```
<!DOCTYPE html>
<head>
```

```
    <title>Dynamic Input Example</title>
    <script>
        // Script to log "Hello, World!" to the console
        console.log("Hello, World!");

        // Script to calculate the sum of two numbers and display the result in an alert box
            // Read numbers from user input
            let num1 = parseFloat(prompt("Enter the first number:"));
            let num2 = parseFloat(prompt("Enter the second number:"));

            // Validate input
                let sum = num1 + num2;
                alert("The sum is: " + sum);

        // Example usage

    </script>
</head>
</html>
```

o/p:



1b.js

```
const prompt = require('prompt-sync')();

// Prompt the user to enter 5 city names, separated by commas
let input = prompt("Enter 5 cities separated by commas: ");
let cities = input.split(',').map(city => city.trim());
console.log("Initial cities:", cities);

// Log the total number of cities
console.log("Total number of cities:", cities.length);

// Add a new city at the end
let newCity = prompt("Enter a city to add to the end: ");
cities.push(newCity);
console.log("Cities after adding a new one:", cities);
```

```
// Remove the first city
console.log("Removing the first city:", cities[0]); // Log the first city before removal
cities.shift();
console.log("Cities after removing the first one:", cities);

// Find and log the index of a specific city
let searchCity = prompt("Enter a city to find its index: ");
let cityIndex = cities.indexOf(searchCity);
console.log("Index of", searchCity + ":", cityIndex !== -1 ? cityIndex : "City not found");
```

1bb.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dynamic Cities Array</title>
</head>
<body>
    <h1>Dynamic Cities Array</h1>
    <script src="1b.js"></script>
</body>
</html>
```

Execution

First install node js

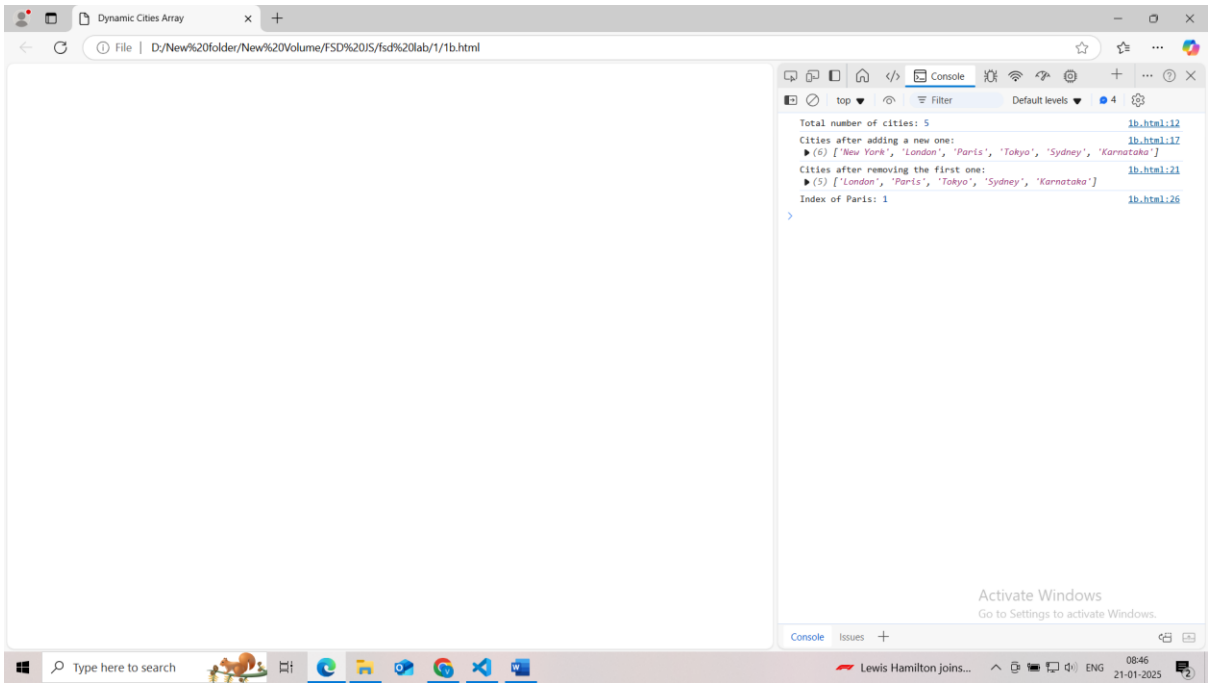
Next run

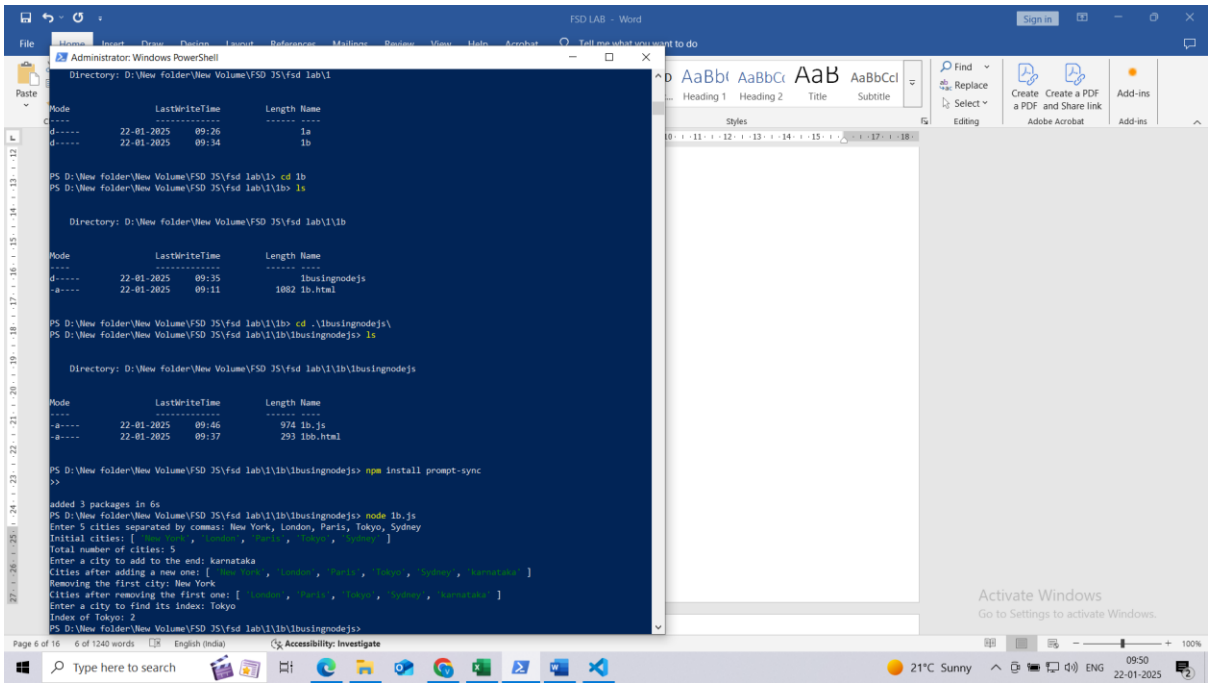
npm install prompt-sync

node 1b.js

o/p: node 1b.js

i/p: New York, London, Paris, Tokyo, Sydney





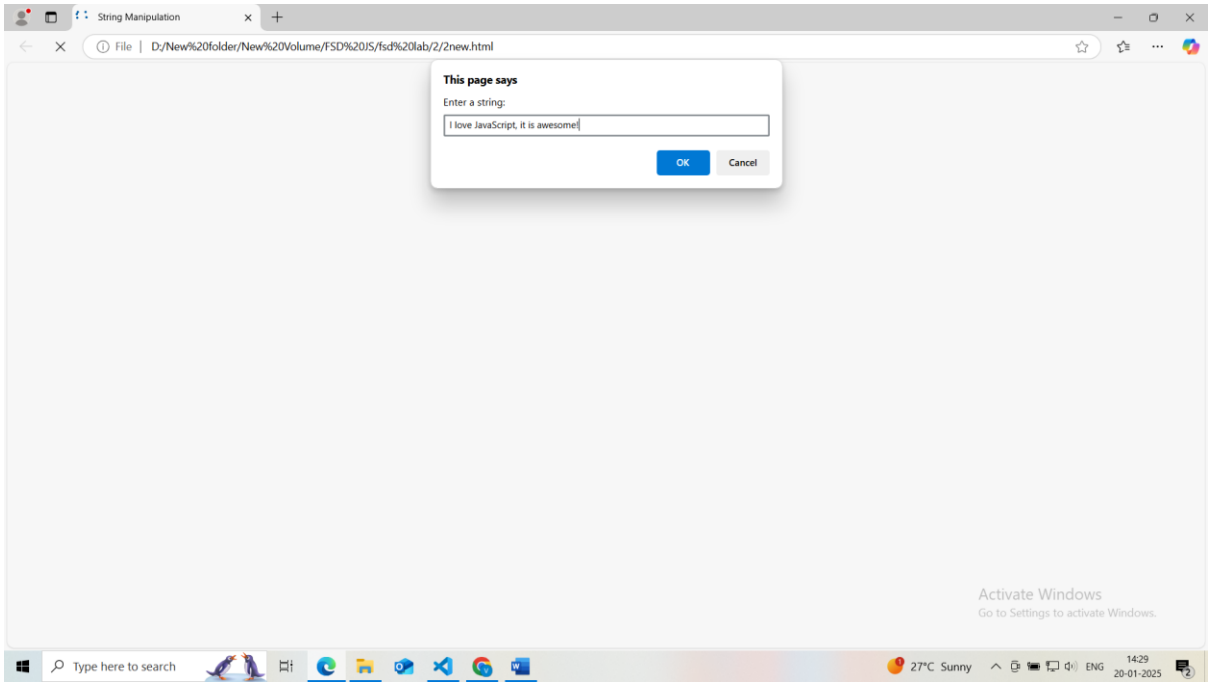
2. Read a string from the user, Find its length. Extract the word "JavaScript" using substring() or slice(). Replace one word with another word and log the new string. Write a function isPalindrome(str) that checks if a given string is a palindrome (reads the same backward).

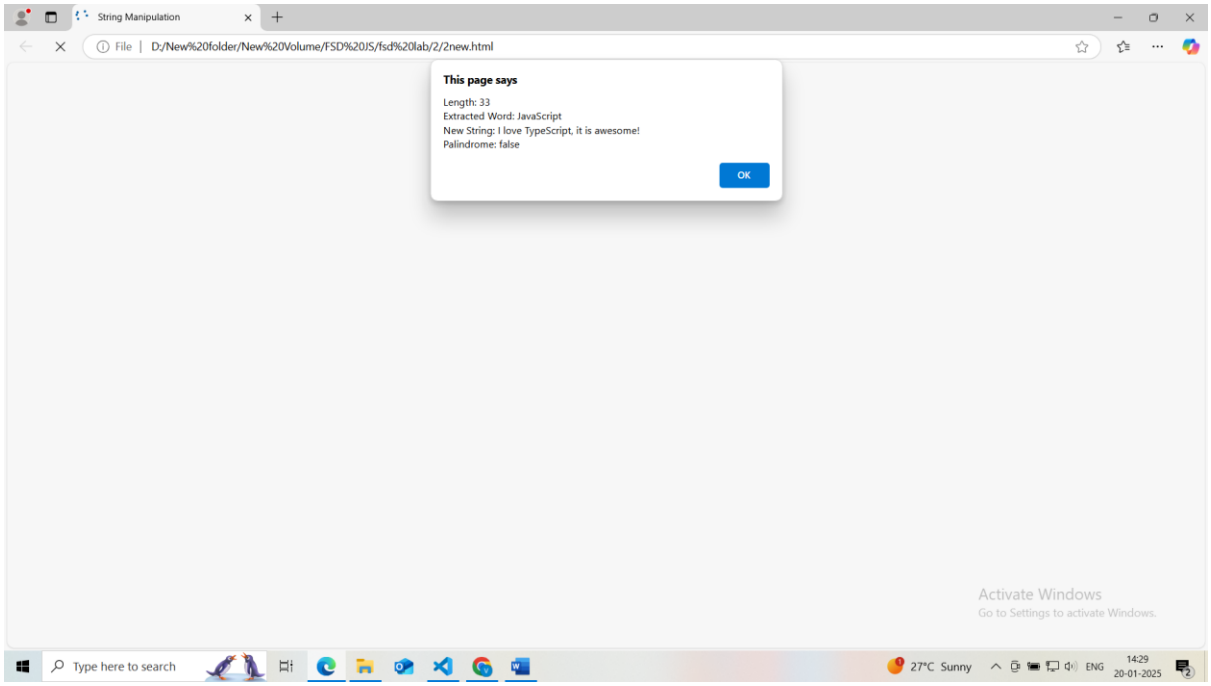
```
<!DOCTYPE html>
<head>

  <title>String Manipulation</title>
  <script>
    // Function to check if a string is a palindrome
    function isPalindrome1(str) {
      for (let i = 0; i < str.length / 2; i++) {
        if (str.charAt(i) != str.charAt(str.length - 1 - i)) {
          return false;
        }
      }
      return true;
    }

    function isPalindrome(str) {
      let cleanedStr = str.replace(/[^A-Za-z0-9]/g, '').toLowerCase();
      return cleanedStr === cleanedStr.split('').reverse().join('');
    }

    // Read string from user and perform operations
    let userInput = prompt("Enter a string:");
    let lengthOfString = userInput.length;

    // Extract 'JavaScript' and replace if present
```

```
    let extractedWord = userInput.includes("JavaScript") ? userInput.substring(userInput.indexOf("JavaScript"), userInput.indexOf("JavaScript") + "JavaScript".length) : "JavaScript not found";

    let newString = userInput.replace("JavaScript", "TypeScript");

    let palindromeCheck = isPalindrome(userInput);


    // Log results

    console.log(`New String after Replacement: ${newString}`);


    // Display alerts

    alert(`Length: ${lengthOfString}\nExtracted Word: ${extractedWord}\nNew String: ${newString}\nPalindrome: ${palindromeCheck}`);

  </script>


</head>

</html>
```

o/p:

give the input string as ----------I love JavaScript, it is awesome!





3. Create an object student with properties: name (string), grade (number), subjects (array), displayInfo() (method to log the student's details) Write a script to dynamically add a passed property to the student object, with a value of true or false based on their grade. Create a loop to log all keys and values of the student object.

<u>One student</u>

<!DOCTYPE html>

```
<head>

    <title>Dynamic Student Object</title>
    <script>
        // Function to create a student object
        function createStudent() {
            // Collect inputs for the student object
            let student = {
                name: prompt("Enter the student's name:"),
                grade: parseInt(prompt("Enter the student's grade:")),
                subjects: prompt("Enter the student's subjects separated by
commas:").split(',').map(subject => subject.trim()),
            };
            return student;
        }

        // Function to display student details
        function displayInfo(student) {
            console.log("Student Details:");

            // Loop through the properties of the student object and log the keys and values
            for (let key in student) {
                if (typeof student[key] !== 'function') {
                    console.log(`${key}: ${student[key]}`);
                }
            }
```

```
        // Log whether the student passed based on the 'passed' function
        console.log("Passed:", student.passed() ? "Yes" : "No");
        console.log("------------------------------");
    }

    let student = createStudent();  // Collect details and return student object
    // Dynamically add 'passed' property based on grade
    student.passed = student.grade >= 40;
    displayInfo(student);
  </script>
</head>

</html>
```

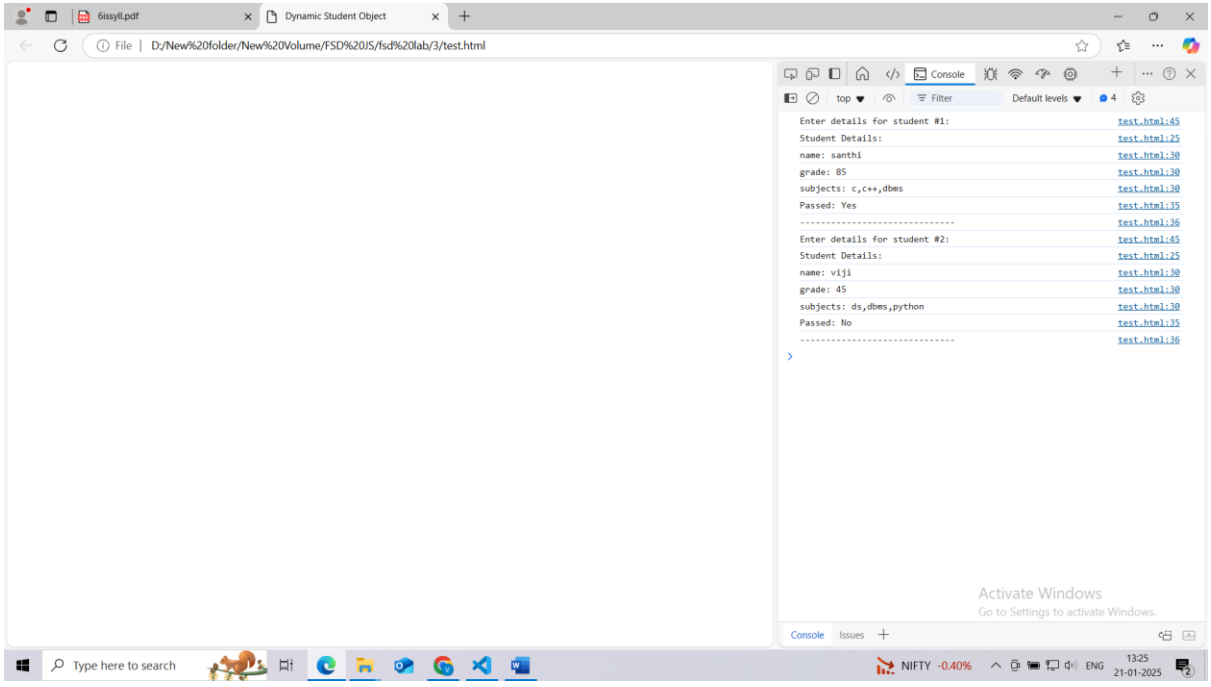
Many students

```
<!DOCTYPE html>

<head>

  <title>Dynamic Student Object</title>
  <script>
    // Function to create a student object
    function createStudent() {
      // Collect dynamic inputs for the student object
      let student = {
        name: prompt("Enter the student's name:"),
        grade: parseFloat(prompt("Enter the student's grade:")),
```

```
        subjects: prompt("Enter the student's subjects separated by commas:").split(',').map(subject => subject.trim()),

        // Dynamically add 'passed' property based on grade
        passed: function() {
            return this.grade >= 50;
        }
    };
    return student;
}

// Function to display student details
function displayInfo(student) {
    console.log("Student Details:");

    // Loop through the properties of the student object and log the keys and values
    for (let key in student) {
        if (typeof student[key] !== 'function') {
            console.log(`${key}: ${student[key]}`);
        }
    }

    // Log whether the student passed based on the 'passed' function
    console.log("Passed:", student.passed() ? "Yes" : "No");
    console.log("------------------------------");
}

// Collect details for multiple students
```

```
    let numStudents = parseInt(prompt("How many students' details would you like to enter?"));


    // Loop to collect details for each student
    for (let i = 0; i < numStudents; i++) {
      console.log(`Enter details for student #${i + 1}:`);
      let student = createStudent();  // Collect details and return student object
      displayInfo(student);
    }


  </script>
</head>

</html>
```

o/p:

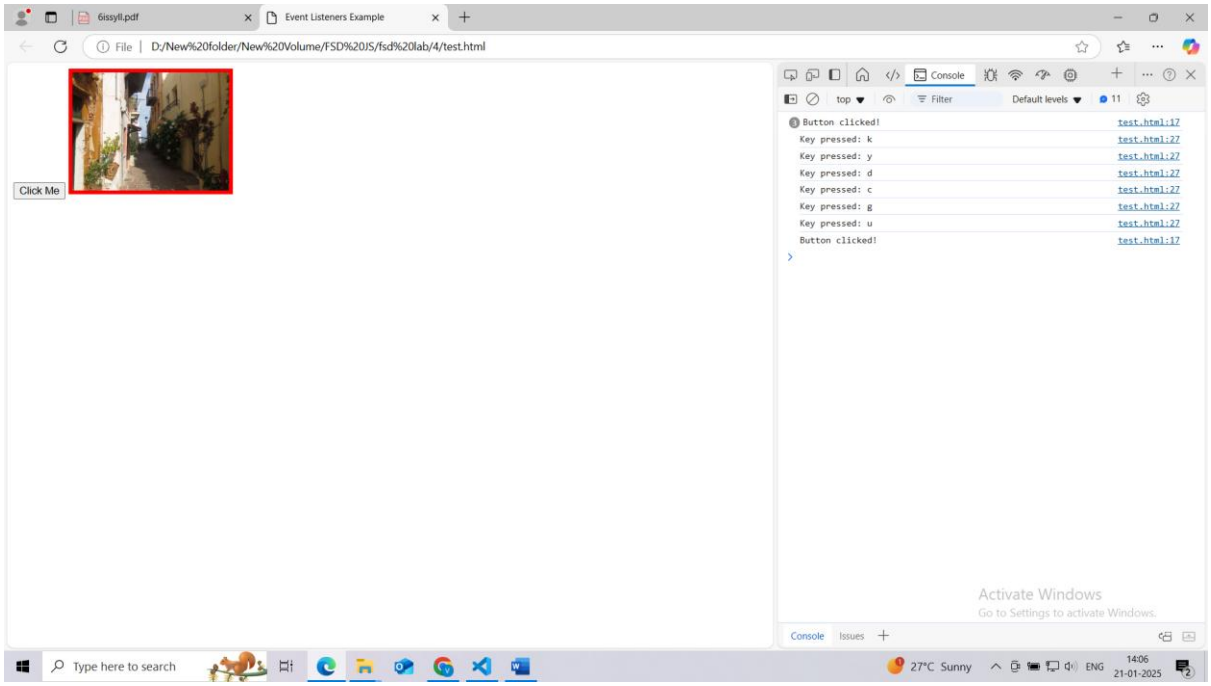


4. Create a button in your HTML with the text "Click Me". Add an event listener to log "Button clicked!" to the console when the button is clicked. Select an image and add a mouseover event listener to change its border color. Add an event listener to the document that logs the key pressed by the user.

```
<!DOCTYPE html>
<head>

  <title>Event Listeners Example</title>
</head>
<body>

  <!-- Button to click -->
  <button id="clickButton">Click Me</button>

  <!-- Example image with a reliable URL -->
  <img id="exampleImage" src="https://www.w3schools.com/html/img_chania.jpg" alt="Example Image" width="200">

  <script>
    // Event listener for the button click
    document.getElementById('clickButton').addEventListener('click', function() {
      console.log("Button clicked!");
    });

    // Event listener for mouseover on the image to change its border color
    document.getElementById('exampleImage').addEventListener('mouseover', function() {
      this.style.border = '5px solid red';  // Change border color to red
```

```
        });

        // Event listener to log the key pressed by the user
        document.addEventListener('keydown', function(event) {
            console.log("Key pressed: " + event.key);
        });

        // Event listener for mouseout on the image to change its border color
        document.getElementById('exampleImage').addEventListener('mouseout', function() {
            this.style.border = 'none';  // remove the border
        });

    </script>

</body>
</html>
```

o/p:


6issyll.pdf
Event Listeners Example
File | D:/New%20folder/New%20Volume/FSD%20JS/fsd%20lab/4/test.html
Click Me
Console
top
Filter
Default levels
11
Button clicked! test.html:17
Key pressed: k test.html:27
Key pressed: y test.html:27
Key pressed: d test.html:27
Key pressed: c test.html:27
Key pressed: g test.html:27
Key pressed: u test.html:27
Button clicked! test.html:17
Activate Windows
Go to Settings to activate Windows.
Console
Issues
Type here to search
27°C Sunny
ENG
14:06
21-01-2025