D. Y. Patil college of Engineering & Technology, Kolhapur

Department: Computer Science & Engg.

Subject: – Scripting Concepts **Class: TY (A, B) (C.S.E.)-II**

Assignment No.: 3

Title - Implementation of event handling using JavaScript.

Event handling in JavaScript is used to perform an action when a user interacts with a web page.User actions such as clicking a button, moving the mouse, typing on the keyboard, loading a page etc. are called events. JavaScript provides event handlers to respond to these events and execute specific code.

Event

An event is an action performed by the user or browser.

Examples of Events:

- Clicking a button
- Moving mouse over an element
- Pressing a key
- Submitting a form
- Loading a webpage

Event Handling

Event handling is the process of capturing an event and responding to it using JavaScript code.

Types of Event Handling in JavaScript

1. Inline Event Handling
2. Internal Event Handling
3. External Event Handling

Common JavaScript Events

Event	Description
onclick	Mouse click
onmouseover	Mouse over element
onmouseout	Mouse leaves element
onkeydown	Key pressed
onload	Page load
onchange	Value change
onsubmit	Form submission

1. Inline Event Handling

JavaScript code is written directly inside the HTML tag

Example 1: Button Click Event

Display a message when a button is clicked.

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

```
        <button onclick="alert('Button Clicked')">Click Me</button>
    </body>
</html>
```

Example 2: Change Text Color

Change text color on button click.

```
<!DOCTYPE html>
<html>
    <body>
        <p id="demo">Hello JavaScript</p>
        <button onclick="document.getElementById('demo').style.color='purple'">
        Change Color
        </button>
    </body>
</html>
```

---

2. Internal Event Handling

JavaScript code is written inside the <script> tag in the same HTML file.

Example 3: onclick using Function

Display a message using internal JavaScript**.**

```
<!DOCTYPE html>
<html>
    <body>
        <button onclick="showMessage()">Click</button>
```

```
        <script>
            function showMessage()
            {
             document.write("Welcome to Event Handling");
            }
        </script>
    </body>
</html>
```

Example 4: Mouse Over Event

Show message when mouse is moved over text.

```
<!DOCTYPE html>
<html>
    <body>
        <p onmouseover="mouseOver()">Move Mouse Here</p>
            <script>
                function mouseOver()
                {
                    alert("Mouse Over Event Occurred");
                }
            </script>
    </body>
</html>
```

---

3. External Event Handling

JavaScript code is written in a separate .js file and linked to HTML.

HTML File (event.html)

```
<!DOCTYPE html>
<html>
    <body>
        <button onclick="display()">Click Me</button>
        <script src="event.js"></script>
    </body>
</html>
```

JavaScript File (event.js)

```
function display()
{
  alert("External JavaScript Event");
}
```

Form Events

Example 5: onchange Event

Display selected value when dropdown changes.

```
<!DOCTYPE html>
<html>
<body>
    <select onchange="showValue(this.value)">
```

```
                <option>Select</option>
                <option>Java</option>
                <option>Python</option>
                <option>JavaScript</option>
        </select>
<script>
        function showValue(val)
        {
                alert("Selected Course: " + val);
        }
</script>
</body>
</html>
```

Example 6: onsubmit Event

Validate form submission.

```
<!DOCTYPE html>
<html>
        <body>
                <form onsubmit="return validate()">
                        Enter Name: <input type="text" id="name">
                        <input type="submit">
                </form>
        <script>
                function validate()
```

```
        {
         let name = document.getElementById("name").value;
         if(name == "")
        {
                alert("Name cannot be empty");
                return false;
         }
      alert("Form Submitted Successfully");
      return true;
    }
    </script>
</body>
</html>
```

Advantages of Event Handling

- Improves user interaction
- Makes web pages dynamic
- Reduces page reloads
- Enhances user experience

Applications of Event Handling

- Button click actions
- Form validation
- Interactive menus

- Games and animations
- Real-time input handling

Results:

1. Students should use Different events such as onclick, onmouseover, onchange, and onsubmit were executed correctly and produced the expected output based on user interaction.

Observations –

1. Student should able to triggered specific event action only when a specific user action occurs**.**

Prepared By:

Prof. K.A. Bhosale, Prof. S. N. Patil