Experiment No. 06

Title: Event handling using JavaScript to explore web browser environment.

Batch:SY-IT(B3) **Roll No.:16010423076** **Experiment No.:6**

Aim: Event handling using JavaScript to explore web browser environments.

Resources needed: Notepad++, Web Browser
Theory:

An HTML event can be something the browser does, or something a user does.

Here are some examples of HTML events:

An HTML web page has finished loading
An HTML input field was changed
An HTML button was clicked
Often, when events happen, you may want to do something.

JavaScript lets you execute code when events are detected.

HTML allows event handler attributes, with JavaScript code, to be added to HTML elements.

What can JavaScript Do?
Event handlers can be used to handle, and verify, user input, user actions, and browser actions:

- Things that should be done every time a page loads
- Things that should be done when the page is closed
- Action that should be performed when a user clicks a button
- Content that should be verified when a user inputs data

Many different methods can be used to let JavaScript work with events:

- HTML event attributes can execute JavaScript code directly
- HTML event attributes can call JavaScript functions
- You can assign your own event handler functions to HTML elements
- You can prevent events from being sent or being handled

Syntax:

```
<element event='some JavaScript'>
<element event="some JavaScript">
```

Example

In the following example, an onclick attribute (with code), is added to a <button> element:

```
<!DOCTYPE html>
<html>
<body>
<button onclick="document.getElementById('demo').innerHTML=Date()">The
time is?</button>

<p id="demo"></p>

</body>
</html>
```

JavaScript HTML DOM Events

HTML DOM allows JavaScript to react to HTML events:

A JavaScript can be executed when an event occurs, like when a user clicks on an HTML element. To execute code when a user clicks on an element, add JavaScript code to an HTML event attribute:

onclick=JavaScript

EXAMPLE

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

<h1 onclick="this.innerHTML='Ooops!'">Click on this text!</h1>

</body>
</html>
```

Activity:

Apply following JS events on your web pages

Input Events

- Onblur
- onreset

Mouse Events

- Onmouseover
- Onmousedown

Click Events

- Onclick
- Ondbclick

Apply following DOM events to your webpages

- Onload
- Onchange
- onmouseover

Results: (Program printout with output)

Code :
Only one script.js has all the functionality for all JavaScript events across all files:
(Full Project attached as zip with the experiment sheet)

```
//  Initialize features when DOM is fully loaded
document.addEventListener("DOMContentLoaded", function () {
    console.log(" DOM Loaded - Initializing Features...");

    initBackToTopButton();
    insertDynamicDate();
    initSearchBar();
    applySeverityLabels();
});

//  Back to Top Button
function initBackToTopButton() {
    const backToTop = document.createElement("button");
    backToTop.innerText = "⬆ Back to Top";
    backToTop.style.position = "fixed";
    backToTop.style.bottom = "20px";
    backToTop.style.right = "20px";
    backToTop.style.padding = "10px 15px";
    backToTop.style.background = "#8E24AA";
    backToTop.style.color = "white";
    backToTop.style.border = "none";
    backToTop.style.borderRadius = "8px";
    backToTop.style.cursor = "pointer";
    backToTop.style.display = "none";
    backToTop.style.zIndex = "1000";

    document.body.appendChild(backToTop);

    backToTop.addEventListener("click", scrollToTop);
    window.addEventListener("scroll", toggleBackToTopVisibility);
}

function scrollToTop() {
    window.scrollTo({ top: 0, behavior: "smooth" });
}

function toggleBackToTopVisibility() {
    const backToTop = document.querySelector("button");
    if (window.scrollY > 200) {
        backToTop.style.display = "block";
    } else {
        backToTop.style.display = "none";
    }
}

//  Dynamic Full Date in Footer
function insertDynamicDate() {
    const footer = document.querySelector("footer p");
    if (footer) {
        const now = new Date();
        const fullDate = now.toDateString(); // Example format: "Thu Mar 27 2025"
        footer.innerHTML = ` ${fullDate} <br/> © Cyber Awareness<br/>All rights reserved`;
    }
}

//  Show welcome message on the home page
if (window.location.pathname === "/" || window.location.pathname.includes("index.html")) {
    alert("Welcome to the Cyber Awareness Portal!");
}

//  Search Bar Functionality (Only for Advisory Page)
function initSearchBar() {
    if (!window.location.pathname.includes("advisory.html")) return;

    const searchInput = document.createElement("input");
    searchInput.type = "text";
    searchInput.placeholder = "Search advisories...";
    searchInput.style.margin = "20px auto";
```

```
    searchInput.style.padding = "10px";
    searchInput.style.width = "50%";
    searchInput.style.display = "block";
    searchInput.style.border = "1px solid #ccc";
    searchInput.style.borderRadius = "8px";
    searchInput.style.textAlign = "center";

    document.querySelector(".container").insertBefore(searchInput,
document.querySelector(".container").children[2]);
    searchInput.addEventListener("input", filterAdvisories);
}

function filterAdvisories() {
    let searchValue = this.value.toLowerCase();
    document.querySelectorAll(".box").forEach(box => {
        let title = box.querySelector("h3").innerText.toLowerCase();
        box.style.display = title.includes(searchValue) ? "inline-block" : "none";
    });
}

//  Threat Severity Indicator (Only for Advisory Page)
function applySeverityLabels() {
    if (!window.location.pathname.includes("advisory.html")) return;

    document.querySelectorAll(".box").forEach(box => {
        const severityLevels = ["Low", "Medium", "High"];
        const severityColors = { "Low": "green", "Medium": "orange", "High": "red" };

        const severity = severityLevels[Math.floor(Math.random() * severityLevels.length)];  //
Random severity
        const severityLabel = document.createElement("p");
        severityLabel.innerText = `Severity: ${severity}`;
        severityLabel.style.color = "white";
        severityLabel.style.backgroundColor = severityColors[severity];
        severityLabel.style.padding = "5px";
        severityLabel.style.borderRadius = "5px";
        severityLabel.style.fontWeight = "bold";
        severityLabel.style.display = "inline-block";

        box.appendChild(severityLabel);
    });
}

//  Other Event Listeners

//  Onblur event: Alerts when input loses focus (ONLY for text fields)
document.querySelectorAll("input[type='text']").forEach(input => {
    input.addEventListener("blur", function () {
        alert(`You left the field: ${this.name}`);
    });
});

//  Onreset event: Confirms before resetting the form
document.querySelectorAll("form").forEach(form => {
    form.addEventListener("reset", function (event) {
        if (!confirm("Are you sure you want to reset the form?")) {
            event.preventDefault();
        }
    });
});

//  Onmouseover event: Changes color when hovering over advisory boxes
document.querySelectorAll(".box").forEach(box => {
    box.addEventListener("mouseover", function () {
        console.log("🔍 Mouseover detected on:", this);
        this.style.backgroundColor = "#dfe6f1";
    });

    //  Onmouseout event: Resets the background color
    box.addEventListener("mouseout", function () {
        this.style.backgroundColor = "#ffffff";
    });

    //  Onclick event: Changes border when clicking on advisory box
    box.addEventListener("click", function () {
        console.log("🖱️ Click event detected on:", this);
        this.style.border = "2px solid blue";  // Blue border on click
    });
```

```
    //  Ondblclick event: Expands advisory box smoothly
    box.addEventListener("dblclick", function () {
        console.log("🖱️ Double-click event detected on:", this);
        this.style.height = "auto";
        this.style.transition = "height 0.3s ease";
        this.style.border = "2px solid green";  // Green border on double-click
    });
});

//  Onclick event: Alerts when clicking on "Read" or "Watch" links in #resources
document.querySelectorAll("#resources a").forEach(link => {
    link.addEventListener("click", function (event) {
        event.preventDefault(); // Prevents unnecessary page reload
        alert("You clicked on a resource link.");
    });
});

//  Onchange event: Alerts only for dropdowns and checkboxes (to avoid too many alerts)
document.querySelectorAll("select, input[type='checkbox']").forEach(input => {
    input.addEventListener("change", function () {
        alert(`Value changed in: ${this.name}`);
    });
});
```

complaint.html

```
<script>
        document.addEventListener("DOMContentLoaded", function () {
            const complaintForm = document.getElementById("complaintForm");
            const viewComplaintBtn =
document.getElementById("viewComplaintBtn");
            const complaintSummary =
document.getElementById("complaintSummary");

            // Function to store complaint data in localStorage
            function storeComplaint(event) {
                event.preventDefault(); // Prevents form submission

                const complaintData = {
                    name: document.getElementById("name").value,
                    email: document.getElementById("email").value,
                    phone: document.getElementById("phone").value,
                    date: document.getElementById("incident_date").value,
                    type: document.getElementById("incident_type").value,
                    details: document.getElementById("details").value
                };

                localStorage.setItem("cyberComplaint",
JSON.stringify(complaintData));
                alert("Complaint submitted successfully!");
                complaintForm.reset();
            }

            // Function to display complaint summary
            function displayComplaint() {
                const storedComplaint =
```

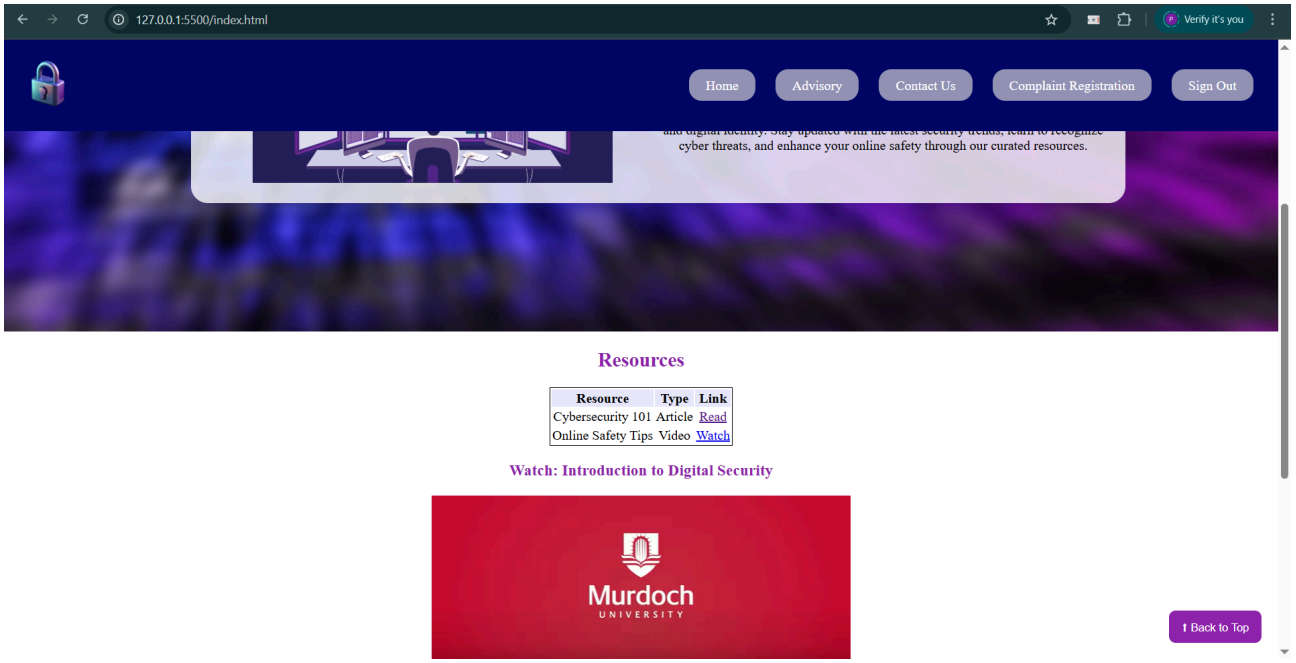
```
localStorage.getItem("cyberComplaint");

                if (storedComplaint) {
                    const complaint = JSON.parse(storedComplaint);
                    complaintSummary.innerHTML = `
                        <h3>Submitted Complaint</h3>
                        <p><strong>Name:</strong> ${complaint.name}</p>
                        <p><strong>Email:</strong> ${complaint.email}</p>
                        <p><strong>Phone:</strong> ${complaint.phone}</p>
                        <p><strong>Incident Date:</strong>
${complaint.date}</p>
                        <p><strong>Cyber Crime Type:</strong>
${complaint.type}</p>
                        <p><strong>Details:</strong>
${complaint.details}</p>
                    `;
                } else {
                    complaintSummary.innerHTML = "<p>No complaint found.
Please submit one first.</p>";
                }
            }

            // Event Listeners
            complaintForm.addEventListener("submit", storeComplaint);
            viewComplaintBtn.addEventListener("click", displayComplaint);
        });
    </script>
```

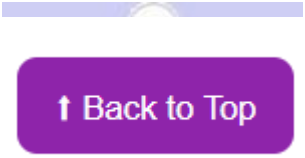
Output :

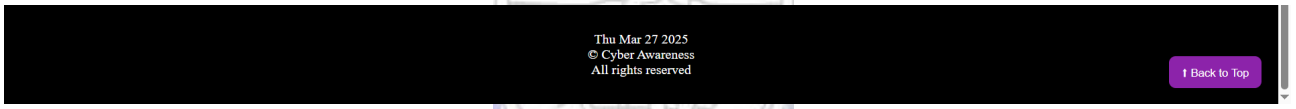
functions were created for:

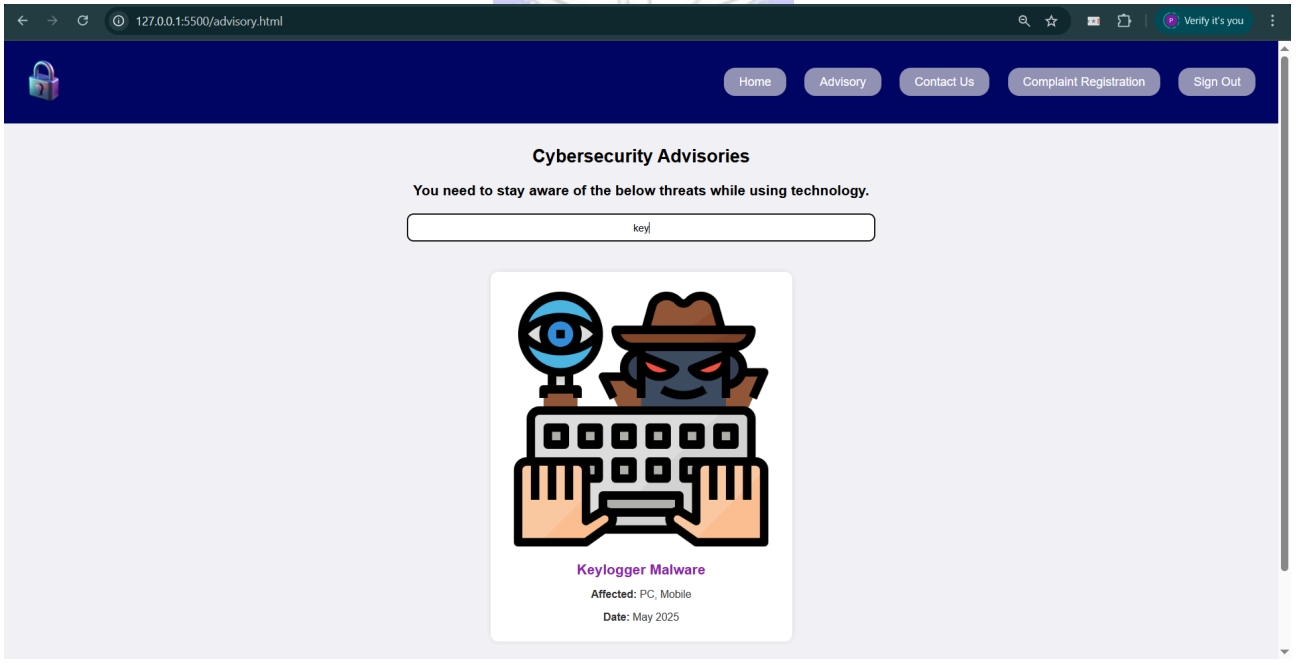
initBackToTopButton() → Manages the back-to-top button.
insertDynamicDate() → Adds the current date to the footer.
initSearchBar() → Adds a search bar for advisories.
applySeverityLabels() → Assigns random severity labels.
storeComplaint() → Stores form data into browser local storage
displayComplaint() → Displays form data from localstorage

Helper functions:
scrollToTop() → Scrolls smoothly to the top.
toggleBackToTopVisibility() → Shows/hides the back-to-top button.
filterAdvisories() → Filters advisory boxes based on search.


127.0.0.1:5500/index.html
Verify it's you
Home
Advisory
Contact Us
Complaint Registration
Sign Out
cyber threats, and enhance your online safety through our curated resources.
Resources
Resource Type Link
Cybersecurity 101 Article Read
Online Safety Tips Video Watch
Watch: Introduction to Digital Security
Murdoch
UNIVERSITY
↑ Back to Top


↑ Back to Top


Thu Mar 27 2025
© Cyber Awareness
All rights reserved
↑ Back to Top


127.0.0.1:5500/advisory.html
Verify it's you
Home
Advisory
Contact Us
Complaint Registration
Sign Out
Cybersecurity Advisories
You need to stay aware of the below threats while using technology.
key
Keylogger Malware
Affected: PC, Mobile
Date: May 2025

127.0.0.1:5500/advisory.html

Verify it's you

Home | Advisory | Contact Us | Complaint Registration | Sign Out

Search advisories...

Phishing Attacks

Affected: Emails, Social Media

Date: February 2025

Ransomware Threat

Affected: Windows, Linux

Date: March 2025

Severity: Medium

127.0.0.1:5500/complaint.html

Verify it's you

Home | Advisory | Contact Us | Complaint Registration | Go Back

Full Name:

Email:

Phone Number:

Incident Date:

mm/dd/yyyy

Type of Cyber Crime:

Select Type

Incident Details:

Submit Complaint

View Complaint

Submitted Complaint

Name: Ritesh

Email: rj@hotmail.com

Phone: 5345543

Incident Date: 2025-03-07

Cyber Crime Type: hacking

Details: dfcghdgh

File C:/Users/EXAM/Documents/16010423076-WPL(S)/20Feb/exp6javascript/exp6javascript/index.html

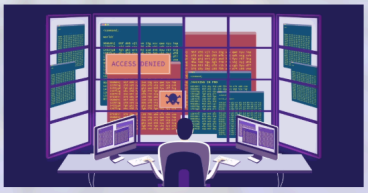
This page says

Welcome to the Cyber Awareness Portal!

OK

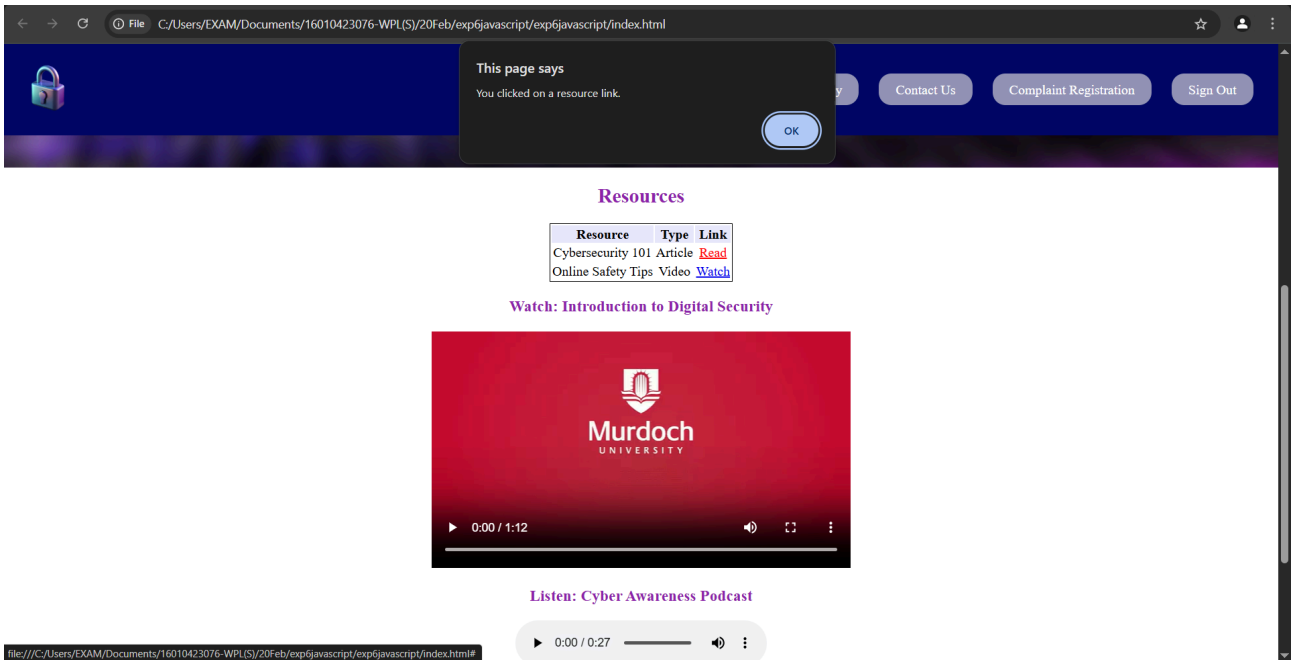
Contact Us | Complaint Registration | Sign Out

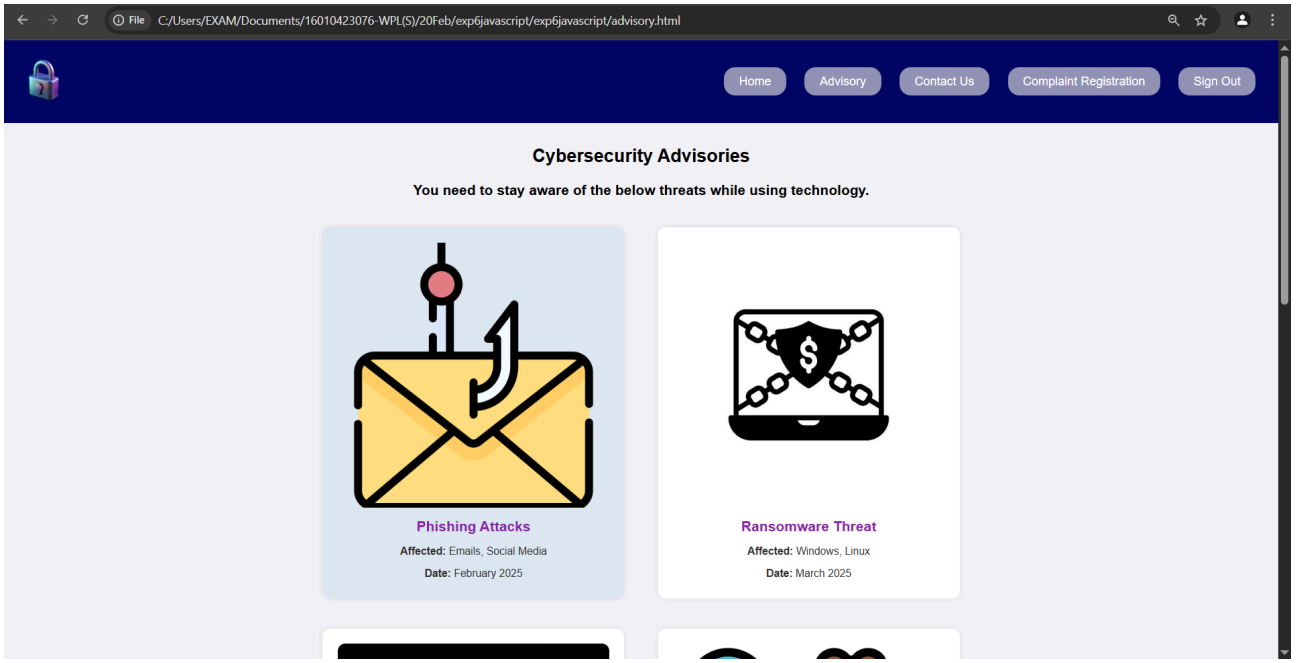
Welcome to Cyber Awareness Portal

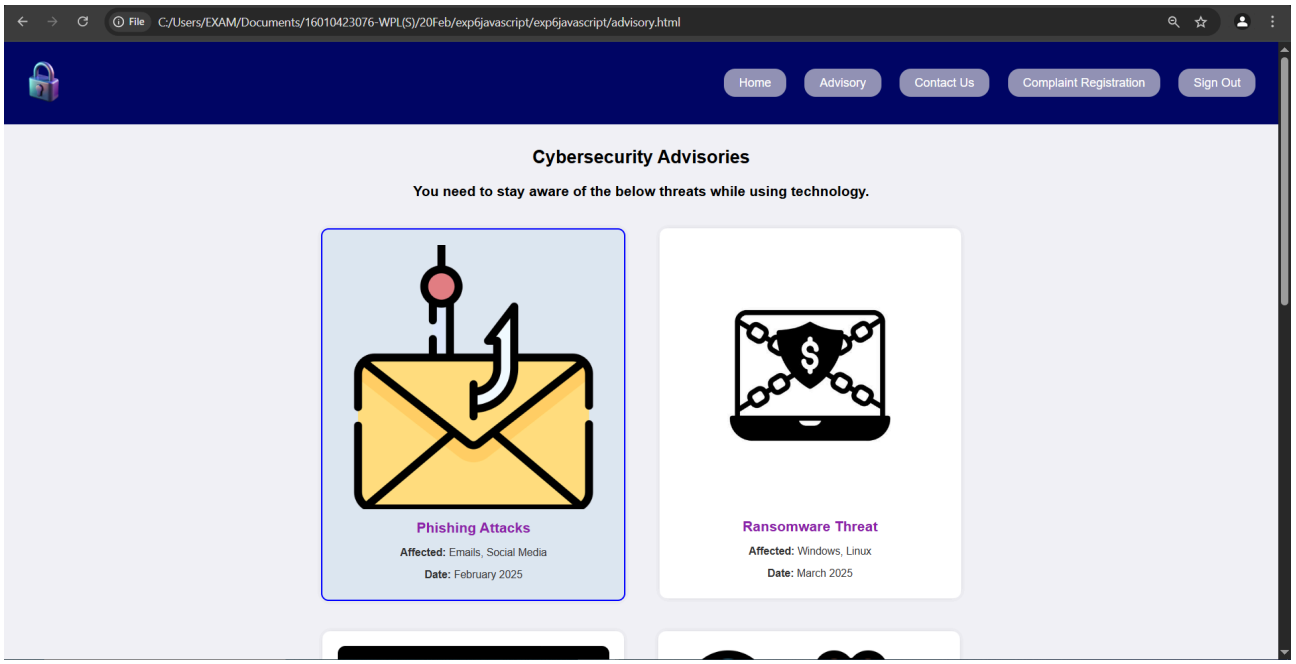


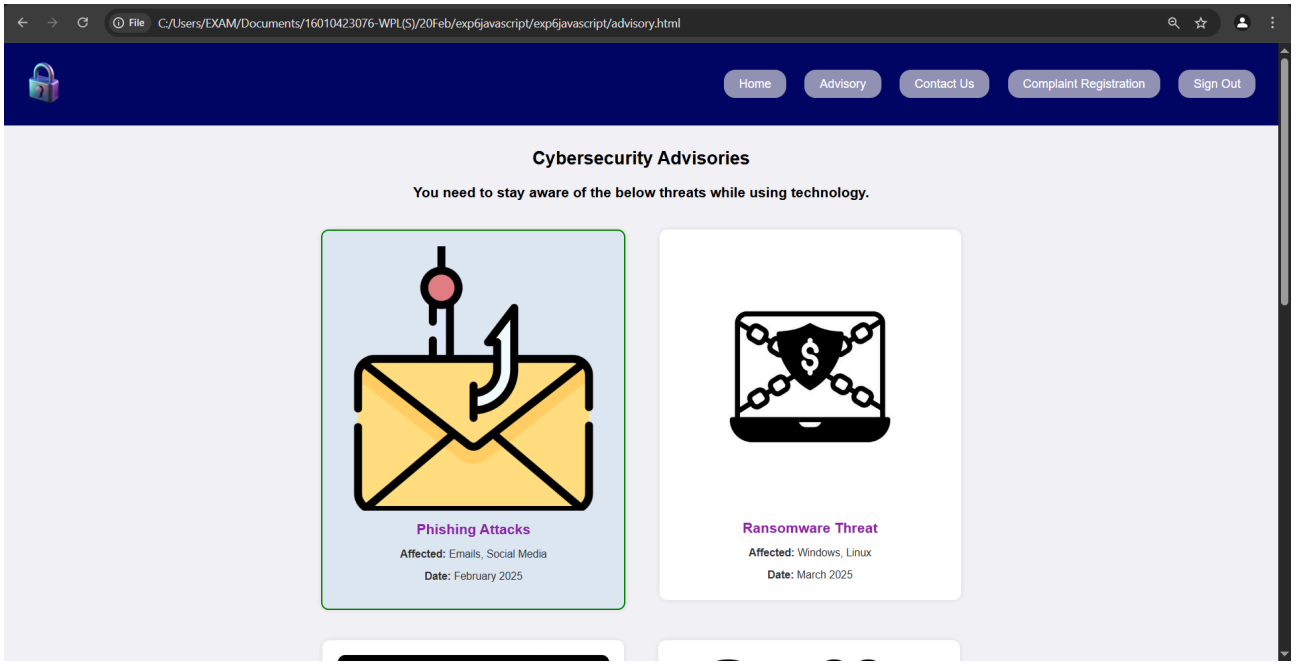
About The Portal

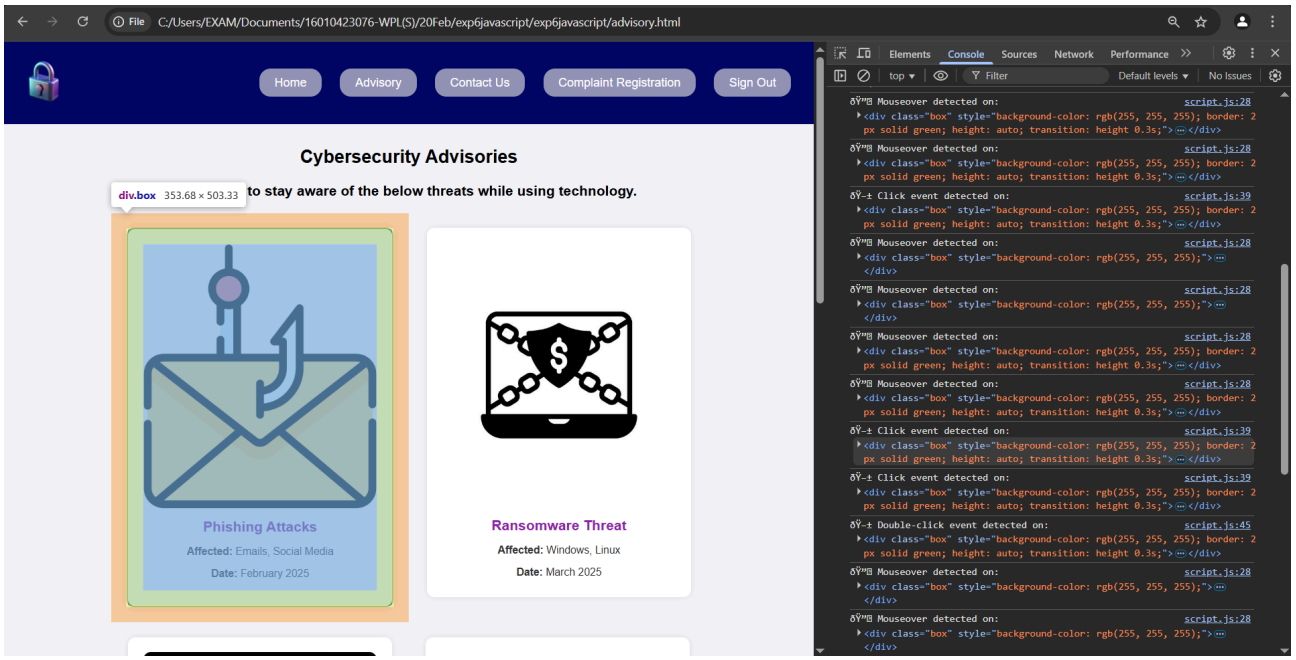
Cyber Awareness is essential in today's digital era. Cyber threats are constantly evolving, and protecting yourself online is crucial. This portal provides essential cybersecurity knowledge, tips, and best practices to safeguard your personal data and digital identity. Stay updated with the latest security trends, learn to recognize cyber threats, and enhance your online safety through our curated resources.

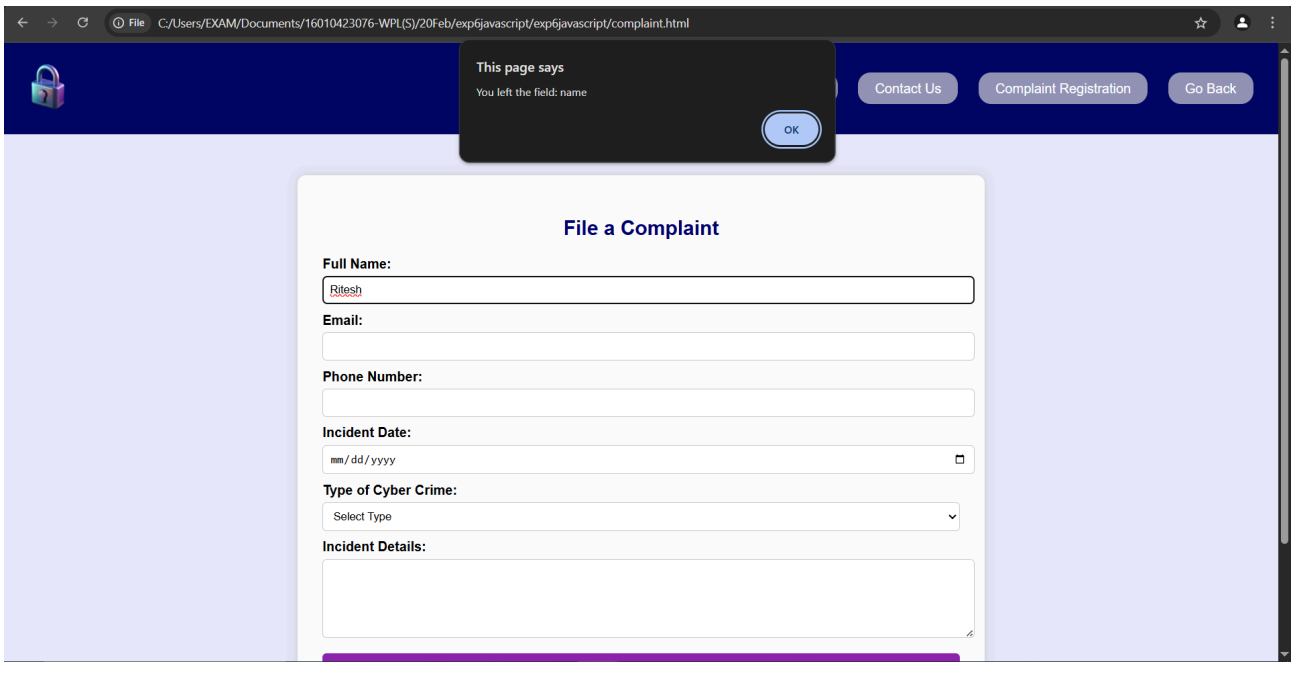

File C:/Users/EXAM/Documents/16010423076-WPL(S)/20Feb/exp6javascript/exp6javascript/index.html
This page says
You clicked on a resource link.
OK
Contact Us
Complaint Registration
Sign Out
Resources
Resource Type Link
Cybersecurity 101 Article Read
Online Safety Tips Video Watch
Watch: Introduction to Digital Security
Murdoch
UNIVERSITY
0:00 / 1:12
Listen: Cyber Awareness Podcast
0:00 / 0:27
file:///C:/Users/EXAM/Documents/16010423076-WPL(S)/20Feb/exp6javascript/exp6javascript/index.html#

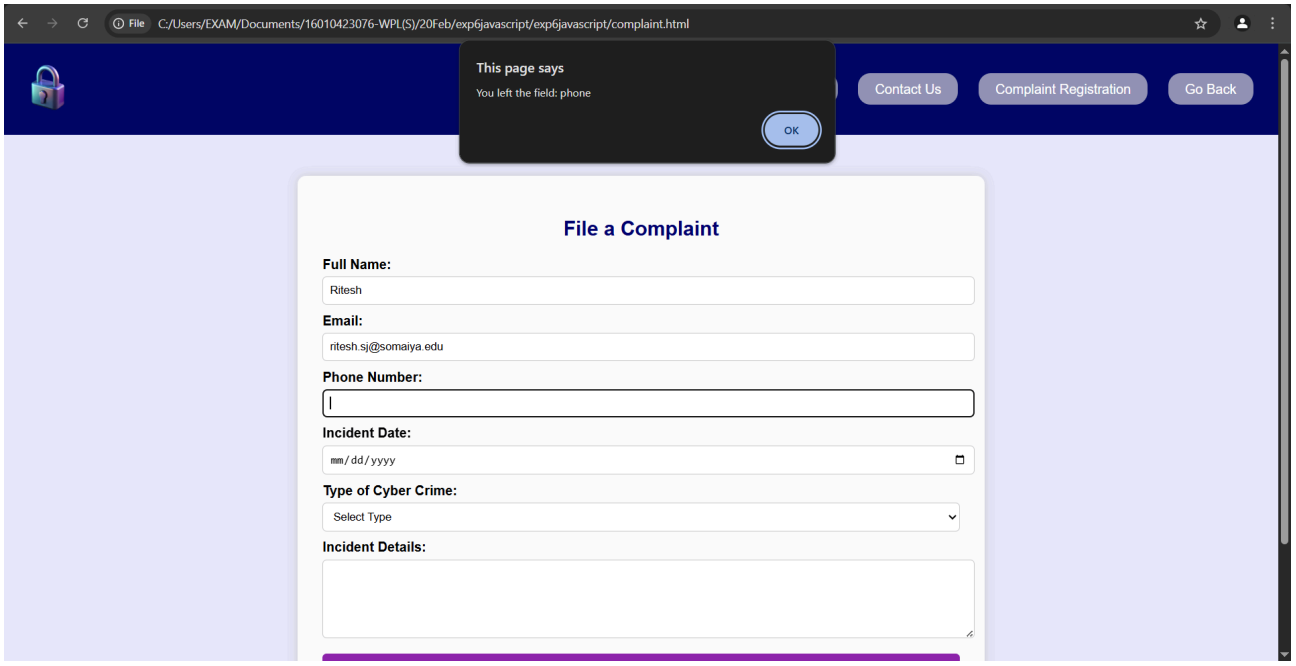

File C:/Users/EXAM/Documents/16010423076-WPL(S)/20Feb/exp6javascript/exp6javascript/advisory.html
Home
Advisory
Contact Us
Complaint Registration
Sign Out
Cybersecurity Advisories
You need to stay aware of the below threats while using technology.
Phishing Attacks
Affected: Emails, Social Media
Date: February 2025
Ransomware Threat
Affected: Windows, Linux
Date: March 2025

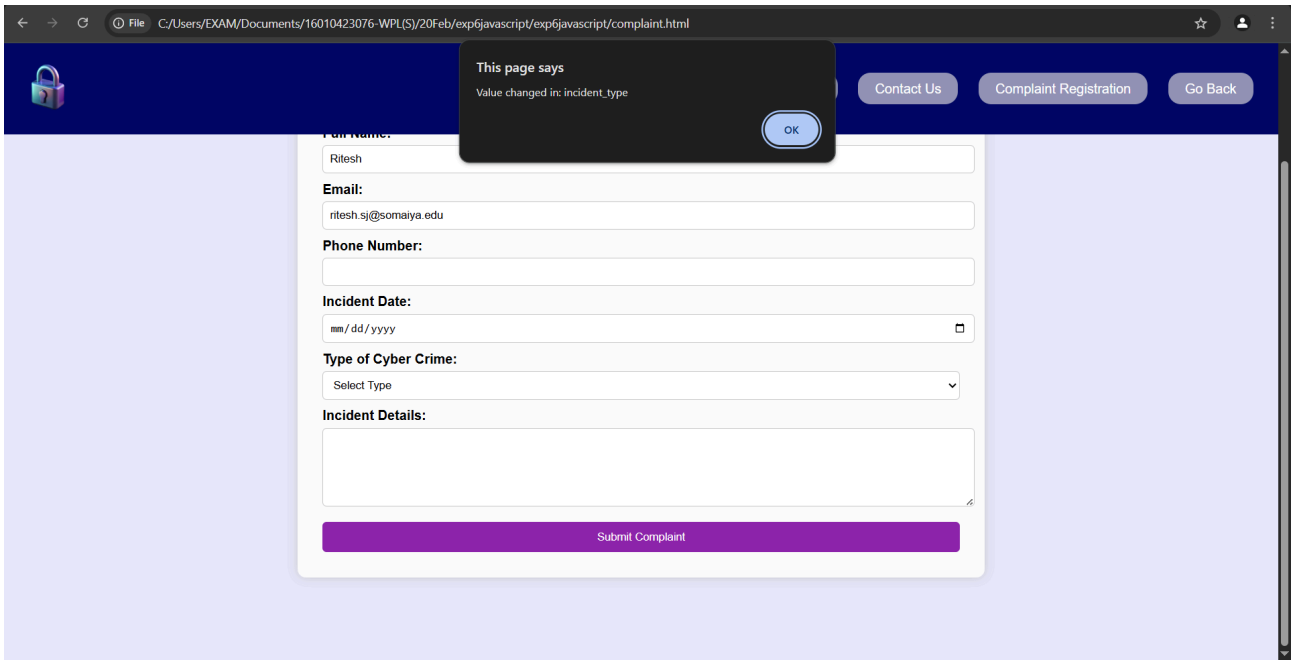

File C:/Users/EXAM/Documents/16010423076-WPL(S)/20Feb/exp6javascript/exp6javascript/advisory.html
Home
Advisory
Contact Us
Complaint Registration
Sign Out
Cybersecurity Advisories
You need to stay aware of the below threats while using technology.
Phishing Attacks
Affected: Emails, Social Media
Date: February 2025
Ransomware Threat
Affected: Windows, Linux
Date: March 2025


File C:/Users/EXAM/Documents/16010423076-WPL(S)/20Feb/exp6javascript/exp6javascript/advisory.html
Home
Advisory
Contact Us
Complaint Registration
Sign Out
Cybersecurity Advisories
You need to stay aware of the below threats while using technology.
Phishing Attacks
Affected: Emails, Social Media
Date: February 2025
Ransomware Threat
Affected: Windows, Linux
Date: March 2025


File C:/Users/EXAM/Documents/16010423076-WPL(S)/20Feb/exp6javascript/exp6javascript/advisory.html
Home
Advisory
Contact Us
Complaint Registration
Sign Out
Cybersecurity Advisories
div.box 353.68 × 503.33
to stay aware of the below threats while using technology.
Phishing Attacks
Affected: Emails, Social Media
Date: February 2025
Ransomware Threat
Affected: Windows, Linux
Date: March 2025
Elements
Console
Sources
Network
Performance
Mouseover detected on:
Click event detected on:
Double-click event detected on:


File C:/Users/EXAM/Documents/16010423076-WPL(S)/20Feb/exp6javascript/exp6javascript/complaint.html
This page says
You left the field: name
OK
Contact Us
Complaint Registration
Go Back
File a Complaint
Full Name:
Ritesh
Email:
Phone Number:
Incident Date:
mm/dd/yyyy
Type of Cyber Crime:
Select Type
Incident Details:





Questions:

Q1) What are the different types of load events

Load events are actions triggered when a webpage or a specific element finishes loading. The different types include onload, which fires when the whole page (including images and scripts) is fully loaded, DOMContentLoaded, which triggers when only the HTML is completely loaded and parsed, and onbeforeunload or onunload, which occur when a user is about to leave or has left the page.

Q2) Explain Onkeypress , onkeyup events

The onkeypress event is triggered when a key is pressed down and held, mainly detecting character keys (letters, numbers, and symbols) but not special keys like Shift or Arrow keys. The onkeyup event, on the other hand, fires when a key is released after being pressed, making it useful for capturing the final input after typing or detecting when a user stops pressing a key.

Outcomes: CO3: Apply JavaScript and JSON for web application development

Conclusion: (Conclusion to be based on the outcomes achieved)

From this experiment, I learned how JavaScript enables event-driven programming to create dynamic and interactive web pages. By implementing various event handlers such as onclick, onmouseover, onblur, and onchange, I explored how user interactions trigger specific actions, enhancing user experience. Additionally, I understood the significance of the DOMContentLoaded event, ensuring that scripts execute only after the document is fully loaded. Through practical implementation, I gained insights into event handling, DOM manipulation, and user input validation, reinforcing the importance of JavaScript in modern web development.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

- "Web technologies: Black Book", Dreamtech Publications
- http://www.w3schools.com
