|  |  |  |
| --- | --- | --- |
| **Name : Harsh Jain** | **Class/Roll No. :D11AD/25** | **Grade :** |

**Title of Experiment :Personal portfolio.**

**Objective of Experiment : The objective of this code is to create a web page with a contact form that, when submitted, captures user input, validates it, and logs the data with the submission date. Additionally, the page displays the current date when loaded.**

**Outcome of Experiment : The expected outcome is that users can submit the form, and their data is collected and displayed on the console. The page also shows the current date upon loading, serving as a basic example of form handling and date display in web development.**

**Problem Statement : Enhance your personal website that you made in Experiment 01-04 using JavaScript features like Functions, Validations, Arrays, String, Date.**

**Description / Theory :**

1. **Event Handling:**

Event handling in JavaScript involves using event listeners to respond to user interactions. For example, you can use the **addEventListener** method to execute functions when users click buttons, submit forms, or perform other actions on your website. This creates an interactive user experience.

1. **Classes and Objects:**

JavaScript supports object-oriented programming, allowing you to define classes and create objects. Classes serve as blueprints for creating instances (objects) with shared properties and methods. You can use classes to represent entities in your website, such as users, products, or comments, making code more organized and maintainable.

1. **Error Handling:**

JavaScript provides mechanisms like **try**, **catch**, and **throw** to handle and manage errors. Error handling is essential for providing a better user experience by gracefully dealing with unexpected issues. It allows you to log errors for debugging and show meaningful error messages to users, enhancing the robustness of your site.

1. **Dynamic Content Manipulation:**

JavaScript enables you to manipulate content dynamically without reloading the entire page. You can use DOM manipulation methods to add, remove, or modify elements in response to user actions. This leads to a more fluid and responsive website, enhancing user engagement.

1. **AJAX (Asynchronous JavaScript and XML):**

AJAX allows your website to make asynchronous requests to the server without reloading the page. It's commonly used for fetching and sending data. With AJAX, you can update parts of a page without disrupting the user experience, creating a more seamless interaction between your site and the server.

1. **Data Validation:**

Data validation is essential to ensure the accuracy and security of user input. JavaScript can perform client-side validation by checking input fields before form submission, providing instant feedback to users. However, server-side validation is equally crucial to prevent malicious data submission and maintain data integrity.

1. **Responsive Design:**

Responsive design is about creating a website that adapts to different screen sizes and devices. JavaScript can be used to adjust the layout and behavior of your site based on the user's device. This ensures a consistent and user-friendly experience, whether users access your site on a desktop, tablet, or mobile phone.

1. **User Authentication:**

User authentication features are essential for securing specific parts of your website. JavaScript, in combination with server-side technologies, allows you to implement user registration, login systems, and session management. This feature ensures that only authorized users can access certain content or perform privileged actions on your site, enhancing its security and functionality.

**Program :**

|  |
| --- |
| function submitForm(event) {  event.preventDefault();    const name = document.getElementById('name').value;  const email = document.getElementById('email').value;  const message = document.getElementById('message').value;    if (!name || !email || !message) {  alert('Please fill in all fields.');  return;  }    const submissionDate = new Date();    const formData = {  name,  email,  message,  submissionDate,  };    console.log('Form Data:', formData);    document.getElementById('name').value = '';  document.getElementById('email').value = '';  document.getElementById('message').value = '';    alert('Form submitted successfully!');  }    function displayCurrentDate() {  const currentDateElement = document.getElementById('currentDateDisplay');    const currentDate = new Date();    const options = { year: 'numeric', month: 'long', day: 'numeric' };  const formattedDate = currentDate.toLocaleDateString('en-US', options);    currentDateElement.textContent = `Current Date: ${formattedDate}`;  }    const form = document.getElementById('contactForm');  form.addEventListener('submit', submitForm);    window.addEventListener('load', displayCurrentDate); |