**Perform from the following to develop interactive web pages using JavaScript:**

**a. Variables, Operators, Conditions, Loops,**

**b. Functions, Events, Classes and Objects,**

**Html**

<!DOCTYPE html>

<html>

<head>

<title>Interactive Web Page</title>

</head>

<body>

<h1>JavaScript Basics</h1>

<!-- Variables and Operators -->

<script>

// Variables

var x = 5;

var y = 10;

// Operators

var sum = x + y;

var product = x \* y;

document.write("Sum: " + sum + "<br>");

document.write("Product: " + product + "<br>");

</script>

<!-- Conditions -->

<script>

var age = 20;

if (age >= 18) {

document.write("You are an adult.<br>");

} else {

document.write("You are not an adult.<br>");

}

</script>

<!-- Loops -->

<script>

for (var i = 1; i <= 5; i++) {

document.write("Iteration " + i + "<br>");

}

</script>

<h1>JavaScript Advanced</h1>

<!-- Functions -->

<script>

function greet(name) {

alert("Hello, " + name + "!");

}

greet("John");

</script>

<!-- Events -->

<button id="myButton">Click Me</button>

<script>

var button = document.getElementById("myButton");

button.addEventListener("click", function() {

alert("Button Clicked!");

});

</script>

<!-- Classes and Objects -->

<script>

class Person {

constructor(name, age) {

this.name = name;

this.age = age;

}

getInfo() {

return this.name + " is " + this.age + " years old.";

}

}

var person1 = new Person("Alice", 25);

var person2 = new Person("Bob", 30);

document.write(person1.getInfo() + "<br>");

document.write(person2.getInfo() + "<br>");

</script>

</body>

</html>

**Perform from the following to develop interactive web pages using JavaScript:**

**Error handling, Validations, Arrays, String, Date**  
<!DOCTYPE html>

<html>

<head>

<title>Interactive Web Page</title>

</head>

<body>

<h1>Interactive Web Page</h1>

<!-- Error Handling and Validations -->

<script>

function validateEmail(email) {

const emailRegex = /^[a-zA-Z0-9.\_-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;

return emailRegex.test(email);

}

try {

const userEmail = "user@example.com";

if (validateEmail(userEmail)) {

console.log("Valid email address.");

} else {

throw new Error("Invalid email address.");

}

} catch (error) {

console.error("Validation error:", error.message);

}

</script>

<!-- Arrays and Strings -->

<script>

const fruits = ["Apple", "Banana", "Cherry"];

fruits.push("Date");

fruits.pop();

console.log("Updated Fruits:", fruits);

const sentence = "JavaScript is an amazing language";

const words = sentence.split(" ");

console.log("Words:", words);

const reversedSentence = words.reverse().join(" ");

console.log("Reversed Sentence:", reversedSentence);

</script>

<!-- Date Handling -->

<script>

const currentDate = new Date();

console.log("Current Date:", currentDate);

const futureDate = new Date(currentDate);

futureDate.setDate(currentDate.getDate() + 7);

console.log("Date 7 days from now:", futureDate);

const formattedDate = currentDate.toLocaleDateString();

console.log("Formatted Date:", formattedDate);

</script>

</body>

</html>

**Create a responsive webpage to design login form and Validation of Username and Password using JavaScript**

Html:

<!DOCTYPE html>

<html>

<head>

<title>Login Form</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f5f5f5;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.container {

background-color: #fff;

border: 1px solid #ddd;

padding: 20px;

border-radius: 5px;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

width: 300px;

text-align: center;

}

input[type="text"],

input[type="password"] {

width: 100%;

padding: 10px;

margin: 5px 0;

border: 1px solid #ccc;

}

button {

background-color: #003366;

color: #fff;

padding: 10px;

border: none;

width: 100%;

cursor: pointer;

}

.error {

color: red;

font-size: 12px;

}

.password-toggle {

display: flex;

align-items: center;

margin-top: 10px;

}

</style>

</head>

<body>

<div class="container">

<h2>Login</h2>

<input type="text" id="username" placeholder="Username">

<span class="error" id="usernameError"></span>

<div class="password-toggle">

<input type="password" id="password" placeholder="Password">

<span id="togglePassword" onclick="togglePasswordVisibility()">Show</span>

</div>

<span class="error" id="passwordError"></span>

<label for="rememberMe">

<input type="checkbox" id="rememberMe"> Remember Me

</label>

<button onclick="validateLogin()">Submit</button>

<p><a href="#">Forgot Password?</a></p>

<p>Don't have an account? <a href="#">Sign Up</a></p>

</div>

<script>

let passwordVisible = false;

function togglePasswordVisibility() {

const passwordField = document.getElementById("password");

const toggleText = document.getElementById("togglePassword");

passwordVisible = !passwordVisible;

if (passwordVisible) {

passwordField.type = "text";

toggleText.textContent = "Hide";

} else {

passwordField.type = "password";

toggleText.textContent = "Show";

}

}

function validateLogin() {

const username = document.getElementById("username").value;

const password = document.getElementById("password").value;

const usernameError = document.getElementById("usernameError");

const passwordError = document.getElementById("passwordError");

usernameError.textContent = "";

passwordError.textContent = "";

if (username === "") {

usernameError.textContent = "Username is required.";

return;

}

if (password === "") {

passwordError.textContent = "Password is required.";

return;

}

// Add your custom validation logic here

if (username === "exampleuser" && password === "password123") {

alert("Login successful!");

} else {

alert("Login failed. Please check your credentials.");

}

}

</script>

</body>

</html>

**Implement the different array methods in Javascript.**

HTML:

const fruits = ['apple', 'banana', 'cherry', 'date'];

// push(): Add elements to the end of the array

fruits.push('elderberry', 'fig');

console.log(fruits); // ['apple', 'banana', 'cherry', 'date', 'elderberry', 'fig']

// pop(): Remove the last element

const removedFruit = fruits.pop();

console.log(removedFruit); // 'fig'

console.log(fruits); // ['apple', 'banana', 'cherry', 'date', 'elderberry']

// shift(): Remove the first element

const firstFruit = fruits.shift();

console.log(firstFruit); // 'apple'

console.log(fruits); // ['banana', 'cherry', 'date', 'elderberry']

// unshift(): Add elements to the beginning of the array

fruits.unshift('apricot', 'grape');

console.log(fruits); // ['apricot', 'grape', 'banana', 'cherry', 'date', 'elderberry']

// concat(): Combine arrays

const moreFruits = ['kiwi', 'lemon'];

const allFruits = fruits.concat(moreFruits);

console.log(allFruits); // ['apricot', 'grape', 'banana', 'cherry', 'date', 'elderberry', 'kiwi', 'lemon']

// slice(): Create a shallow copy of a portion of the array

const selectedFruits = fruits.slice(2, 5);

console.log(selectedFruits); // ['banana', 'cherry', 'date']

// forEach(): Execute a function for each element

fruits.forEach(function(fruit) {

console.log(fruit.toUpperCase());

});

// map(): Create a new array with modified elements

const capitalizedFruits = fruits.map(function(fruit) {

return fruit.toUpperCase();

});

console.log(capitalizedFruits);

// filter(): Create a new array with filtered elements

const filteredFruits = fruits.filter(function(fruit) {

return fruit.length > 5;

});

console.log(filteredFruits);

// reduce(): Reduce the array to a single value

const totalLength = fruits.reduce(function(accumulator, fruit) {

return accumulator + fruit.length;

}, 0);

console.log(totalLength);

**Create a responsive webpage Pharmacy Management System using HTML, CSS and Javascript.**

<!DOCTYPE html>

<html>

<head>

<title>Pharmacy Management System</title>

<link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

<header>

<h1>Pharmacy Management System</h1>

</header>

<nav>

<ul>

<li><a href="#">Home</a></li>

<li><a href="#">Inventory</a></li>

<li><a href="#">Sales</a></li>

<li><a href="#">Orders</a></li>

<li><a href="#">Customers</a></li>

<li><a href="#">Suppliers</a></li>

</ul>

</nav>

<main>

<section id="inventory">

<h2>Inventory Management</h2>

<!-- Inventory-related content here -->

</section>

<section id="sales">

<h2>Sales Management</h2>

<!-- Sales-related content here -->

</section>

<section id="orders">

<h2>Order Management</h2>

<!-- Order-related content here -->

</section>

<section id="customers">

<h2>Customer Management</h2>

<!-- Customer-related content here -->

</section>

<section id="suppliers">

<h2>Supplier Management</h2>

<!-- Supplier-related content here -->

</section>

</main>

<footer>

<p>&copy; 2023 Pharmacy Management System</p>

</footer>

<script src="script.js"></script>

</body>

</html>

CSS:  
/\* Reset some default browser styles \*/

body, h1, h2, ul, li {

margin: 0;

padding: 0;

}

body {

font-family: Arial, sans-serif;

background-color: #f5f5f5;

}

header {

background-color: #003366;

color: #fff;

text-align: center;

padding: 20px;

}

nav {

background-color: #003366;

color: #fff;

padding: 10px;

}

nav ul {

list-style-type: none;

text-align: center;

}

nav li {

display: inline;

margin: 0 20px;

}

nav a {

text-decoration: none;

color: #fff;

}

main {

padding: 20px;

}

section {

background-color: #fff;

border: 1px solid #ddd;

border-radius: 5px;

padding: 20px;

margin: 20px 0;

}

footer {

background-color: #003366;

color: #fff;

text-align: center;

padding: 10px;

}