

**Ex No: 2**

**Date:**

## **FORM AND VALIDATION**

### **AIM :**

The aim of this project is to create a registration form with proper validation for various input fields such as **First Name**, **Password**, **Email**, **Mobile Number**, **Last Name**, and **Address** using **JavaScript** for client-side validation and **HTML/CSS** for the structure and styling.

### **ALGORITHM :**

1. **Start**
  - Begin by setting up the registration form in HTML with fields for first name, last name, email, password, mobile number, and address.
2. **Capture User Input:**
  - When the user submits the form, capture the values entered in the form fields (first name, last name, password, email, mobile number, and address).
3. **Validate First Name:**
  - Check if the first name contains only alphabets using a regular expression (`^[A-Za-z]+$`).
  - Ensure the length of the first name is **at least 6 characters**.
  - If the validation fails, display an alert message and stop the form submission.
4. **Validate Password:**
  - Ensure that the password has a minimum length of **6 characters**.
  - If the password is shorter, show an alert and stop the form submission.
5. **Validate Email:**
  - Check if the email follows the standard pattern (e.g., `name@domain.com`) using a regular expression (`^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}$`).
  - If the email is invalid, display an error message and prevent the form from being submitted.
6. **Validate Mobile Number:**
  - Ensure the mobile number contains exactly **10 digits** using the regular expression (`^[0-9]{10}$`).
  - If it doesn't match, display an alert and prevent form submission.
7. **Validate Last Name:**

- Check if the last name is not empty.
- If it is empty, show an alert and halt the form submission.
- 8. **Validate Address:**
  - Ensure the address field is not empty.
  - If the address is left blank, show an alert and prevent form submission.
- 9. **Submit Form:**
  - If all the fields pass validation, display a **success message** (e.g., "Registration Successful!") and allow the form to be submitted.
- 10. **End**
  - End the validation process and, if everything is valid, the form is successfully submitted.

## CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Registration Form</title>
<style>
/* General Styling */ body {
font-family: 'Arial', sans-serif; background-color: #f4f7fc; margin: 0;
padding: 0;
color: #333;
}

h2 {
text-align: center; color: #4CAF50; padding-top: 20px;
}

/* Form Container */
.form-container { width: 50%; margin: 30px auto; padding: 20px;
background-color: #fff; border-radius: 8px;
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

/* Input Fields */ input[type="text"], input[type="password"],
input[type="email"], input[type="submit"], textarea {
width: 100%; padding: 12px; margin: 10px 0; border-radius: 4px;
border: 1px solid #ddd; font-size: 16px;
}

/* Textarea Specific Styling */ textarea {
resize: vertical; min-height: 100px;
}
```

```

input[type="submit"] { background-color: #4CAF50; color: white;
border: none; cursor: pointer; font-size: 18px;
transition: background-color 0.3s ease;
}

input[type="submit"]:hover { background-color: #45a049;
}

/* Labels */ label {
font-weight: bold; display: block; margin: 10px 0 5px;
}

/* Error and Success Messages */
.error-message { color: red;
font-size: 14px;
}

.success-message { color: green; font-size: 14px;
}

/* Responsive Design */
@media screen and (max-width: 768px) {
.form-container { width: 90%;
}
}
</style>
</head>
<body>

<h2>Registration Form</h2>

<div class="form-container">
<form id="registrationForm" onsubmit="return validateForm()">
<!-- First Name -->
<label for="firstName">First Name:</label>
<input type="text" id="firstName" name="firstName" required><br>

<!-- Last Name -->
231501058

```

```

<label for="lastName">Last Name:</label>
<input type="text" id="lastName" name="lastName" required><br>

<!-- Password -->
<label for="password">Password:</label>
<input type="password" id="password" name="password" required><br>

<!-- Email -->
<label for="email">Email:</label>

<input type="email" id="email" name="email" required><br>

<!-- Mobile Number -->
<label for="mobile">Mobile Number:</label>
<input type="text" id="mobile" name="mobile" required><br>

<!-- Address -->
<label for="address">Address:</label>
<textarea id="address" name="address" required></textarea><br>

<!-- Submit Button -->
<input type="submit" value="Register">
</form>
</div>

<script src="validate.js"></script>
</body>
</html>

```

JAVASCRIPT:

```

function validateForm() {
// Get values from form fields
const firstName = document.getElementById("firstName").value; const lastName =
document.getElementById("lastName").value; const password =
document.getElementById("password").value; const email =
document.getElementById("email").value;
const mobile = document.getElementById("mobile").value; const address =
document.getElementById("address").value;

// First Name Validation: Must contain only alphabets and length >= 6 const
namePattern = /^[A-Za-z]+$/;
231501058

```

```

if (!namePattern.test(firstName) || firstName.length < 6) {
  alert("First Name must contain only alphabets and be at least 6 characters long."); return false;
}

// Password Validation: Must be at least 6 characters long if (password.length < 6) {
alert("Password must be at least 6 characters long."); return false;
}

// Email Validation: Must follow standard email format
const emailPattern = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}$/; if
(!emailPattern.test(email)) {
  alert("Please enter a valid email address."); return false;
}

// Mobile Number Validation: Must contain exactly 10 digits const
mobilePattern = /^[0-9]{10}$/;
if (!mobilePattern.test(mobile)) {
  alert("Mobile number must be exactly 10 digits long."); return false;
}

// Last Name Validation: Must not be empty if (lastName.trim() === "") {
alert("Last Name cannot be empty."); return false;
}

// Address Validation: Must not be empty if (address.trim() === "") {
alert("Address cannot be empty."); return false;
}

// If all validations pass alert("Registration Successful!"); return true;
}

```

**OUTPUT :**

**Registration Form**

First Name:

Last Name:

Password:

Email:

Mobile Number:

Address:

**Register**

## **RESULT :**

Thus the give task is executed successfully.