

HTML & JavaScript Form Validation Assignment

Introduction:

This assignment is about creating different types of forms using HTML and JavaScript.

The forms include Entrance Exam Registration, Marksheets Generator, SIM Registration, Course Enrollment, and Hospital Appointment Booking.

These forms collect user information and apply JavaScript validation to ensure correct data entry.

Validation prevents wrong or empty data from being submitted.

Objectives:

- ❖ To design interactive forms using HTML
- ❖ To apply JavaScript validation rules
- ❖ To ensure users enter correct data
- ❖ To learn client-side validation
- ❖ To understand real-life form systems

Tools Used:

- ❖ HTML
- ❖ JavaScript
- ❖ Web Browser

Forms Created:

1) Entrance Exam Registration Form:

This form collects student details like name, age, email, phone number, gender, and exam stream.

Validation ensures correct age, valid email, and required field completion.

A success message is shown and the page refreshes after 5 seconds.

Code Snapshot:

```
// Validation logic

function validateExamForm() {

    const name = nameInput.value.trim();

    const age = parseInt(document.getElementById('BarshaTimsina_06_age').value, 10);

    const phone = document.getElementById('BarshaTimsina_06_phone').value.trim();

    const stream = document.getElementById('BarshaTimsina_06_stream').value;

    const agree = document.getElementById('BarshaTimsina_06_agree').checked;

    const namePattern = /^[A-Za-z\s]+$/;

    if (!name) { alert('Please enter your full name.'); return false; }

    if (!namePattern.test(name)) { alert('Name should contain only letters and spaces.'); return false; }

    if (isNaN(age)) { alert('Please enter a valid age.'); return false; }

    if (age < 16 || age > 30) { alert('Age must be between 16 and 30.'); return false; }

    // phone digits only

    const digits = phone.replace(/\D/g, "");

    const start = digits.substring(0,2);
```

```
if (digits.length !== 10 || (start !== '98' && start !== '97')) {
    alert('Phone must be 10 digits and start with 98 or 97.');
    return false;
}

if (stream === 'none') { alert('Please select an exam stream.'); return false; }

if (!agree) { alert('Please confirm that all details are correct.'); return false; }

// if special visible, ensure filled
if (specialContainer.style.display === 'block' && !specialInput.value.trim()) {
    alert('Please fill the special ID field.');
    return false;
}

alert('Form submitted successfully. Preparing your application preview.');

// Demo: reset after short delay
setTimeout(() => { document.getElementById('examForm').reset(); updateSpecialVisibility(); }, 900);
return false; // prevent actual submission in demo
}

// Initialize visibility on load
updateSpecialVisibility();
```

Output Snapshot:

This page says

Form submitted successfully. Preparing your application preview.

OK

Entrance Exam Registration

Complete the form to register for our upcoming entrance exam. Provide accurate details — eligibility checks run automatically.

Age: 20 **Phone Number**: 9804908114

Email: timsinabarsha529@gmail.com

Gender: Male Female Other

Exam Stream: Management **Roll No**: 06

Special ID: BarshaTimsina_06

I confirm all details are correct

Submit Application

2) Student Mark Sheet Generator:

This form takes marks in English, Math, and Science. It calculates total marks, percentage, grade, and result. Validation ensures marks are between 0–100.

Code Snapshot:

```
<script>

function generateMarksheet() {

    const name = document.getElementById('Barsha_06_name').value.trim();

    const roll = document.getElementById('Barsha_06_roll').value;

    const english = parseFloat(document.getElementById('Barsha_06_english').value);

    const math = parseFloat(document.getElementById('Barsha_06_math').value);

    const science = parseFloat(document.getElementById('Barsha_06_science').value);

    const totalMarks = english + math + science;

    const percentage = (totalMarks / 300) * 100;

    const grade = calculateGrade(percentage);

    const result = calculateResult(percentage);

    const output = `Name: ${name}\nRoll: ${roll}\nEnglish: ${english}\nMath: ${math}\nScience: ${science}\nTotal Marks: ${totalMarks}\nPercentage: ${percentage}%
Grade: ${grade}\nResult: ${result}`;

    alert(output);
}

function calculateGrade(percentage) {
    if (percentage >= 90) return 'A';
    else if (percentage >= 80) return 'B';
    else if (percentage >= 70) return 'C';
    else if (percentage >= 60) return 'D';
    else return 'E';
}

function calculateResult(percentage) {
    if (percentage >= 90) return 'Pass';
    else return 'Fail';
}
```

```

const math = parseFloat(document.getElementById('Barsha_06_math').value);
const science = parseFloat(document.getElementById('Barsha_06_science').value);

// Validation
if (!name) { alert('Please enter student name.'); return false; }
if (!roll) { alert('Please enter roll number.'); return false; }
if (isNaN(english) || english < 0 || english > 100) { alert('English marks must be between 0 and 100.'); return false; }
if (isNaN(math) || math < 0 || math > 100) { alert('Math marks must be between 0 and 100.'); return false; }
if (isNaN(science) || science < 0 || science > 100) { alert('Science marks must be between 0 and 100.'); return false; }

// Calculations
const total = english + math + science;
const percentage = (total / 300 * 100).toFixed(2);

// Result
const pass = english >= 40 && math >= 40 && science >= 40;
const resultText = pass ? ' Pass' : ' Fail';

// Grade
let grade, gradeClass;
if (!pass) {
    grade = 'Fail';
    gradeClass = 'grade-fail';
} else if (percentage >= 80) {
    grade = 'A';
    gradeClass = 'grade-a';
} else if (percentage >= 60) {
    grade = 'B';
    gradeClass = 'grade-b';
} else {
    grade = 'C';
    gradeClass = 'grade-c';
}

// Display results

```

```

document.getElementById('totalMarks').textContent = total + '/300';

document.getElementById('percentage').textContent = percentage + '%';

document.getElementById('passFail').textContent = resultText;

const gradeEl = document.getElementById('grade');

gradeEl.textContent = grade;

gradeEl.className = gradeClass;

document.getElementById('marksheetResult').classList.add('show');

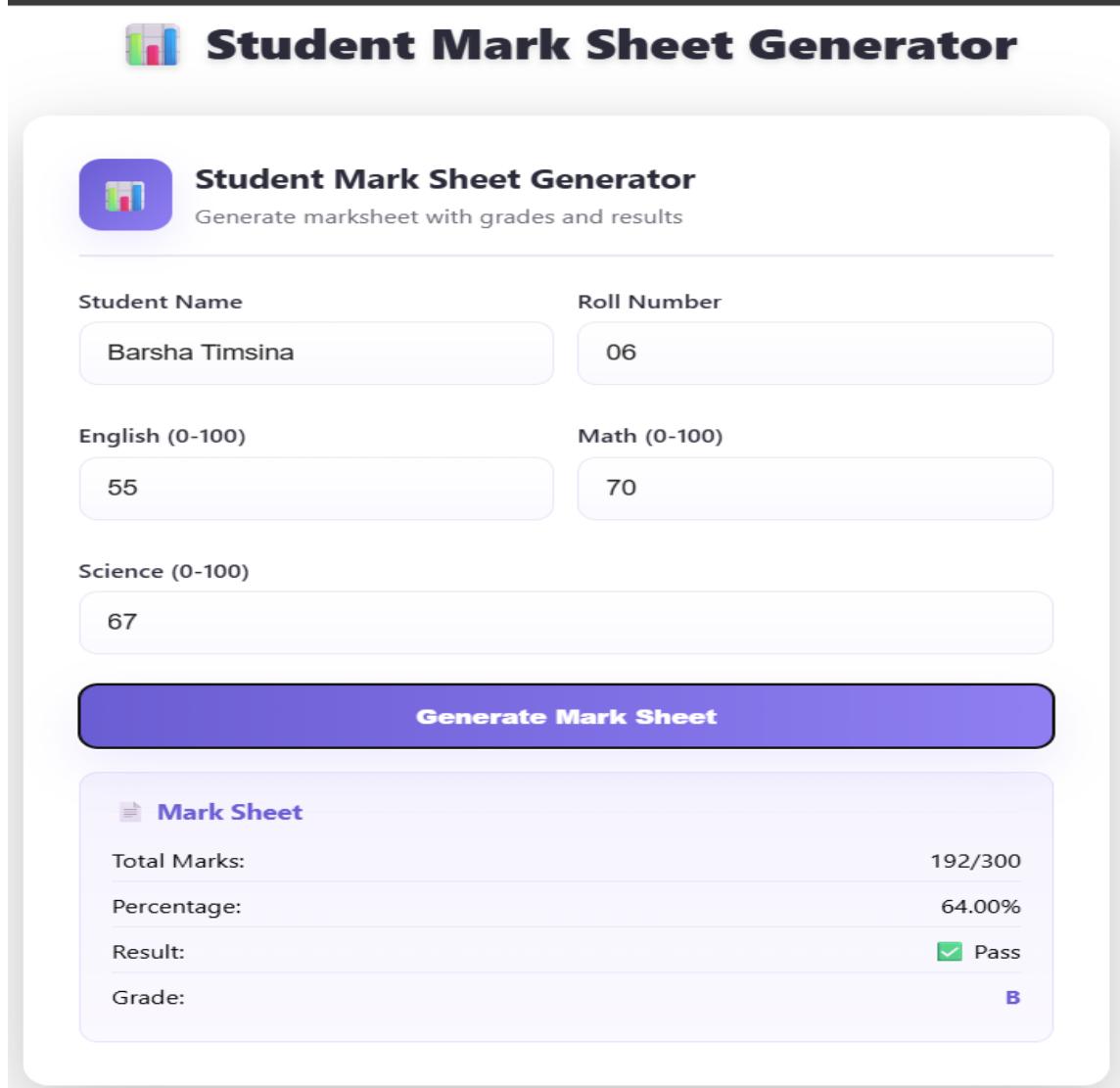
return false;

}

</script>

```

Output Snapshot:



The screenshot shows a web-based application titled "Student Mark Sheet Generator". The interface is clean with a light gray background and white cards for input fields.

Header:

Logo: A purple square icon containing a bar chart with three bars of increasing height.

Title: **Student Mark Sheet Generator**

Subtext: Generate marksheets with grades and results

Input Fields:

- Student Name:** Barsha Timsina
- Roll Number:** 06
- Subject & Marks:**
 - English (0-100): 55
 - Math (0-100): 70
 - Science (0-100): 67

Buttons:

- A large blue button labeled "Generate Mark Sheet" in white text.

Output Card:

Section: **Mark Sheet**

Total Marks:	192/300
Percentage:	64.00%
Result:	<input checked="" type="checkbox"/> Pass
Grade:	B

3)SIM Card Registration Form:

This form collects personal details for SIM registration.

It validates citizenship number, age (18+), and SIM type selection.

Terms and conditions must be accepted.

Code Snapshot:

```
<script>

function validateSIM() {

    const name = document.getElementById('Barsha_06_fullname').value.trim();

    const citizenship = document.getElementById('Barsha_06_citizenship').value.trim();

    const dob = document.getElementById('Barsha_06_dob').value;

    const nationality = document.getElementById('Barsha_06_nationality').value;

    const simType = document.querySelector('input[name="Barsha_06_simType"]:checked');

    const terms = document.getElementById('Barsha_06_terms').checked;

    const namePattern = /^[A-Za-z\s]+$/;

    if (!name) { alert('Please enter your full name.'); return false; }

    if (!namePattern.test(name)) { alert('Name must contain only letters.'); return false; }

    if (citizenship.length < 10) { alert('Citizenship number must be at least 10 characters.'); return false; }

    // Age check

    if (dob) {

        const birthDate = new Date(dob);

        const today = new Date();

        let age = today.getFullYear() - birthDate.getFullYear();

        const monthDiff = today.getMonth() - birthDate.getMonth();

        if (monthDiff < 0 || (monthDiff === 0 && today.getDate() < birthDate.getDate())) {

            age--;

        }

        if (age < 18) { alert('You must be 18 or above to register.'); return false; }

    } else {

}
```

```

        alert('Please enter date of birth.');
        return false;
    }

    if (!nationality) { alert('Please select nationality.');?>
    if (!simType) { alert('Please select SIM type.');?>
    if (!terms) { alert('You must accept Terms & Conditions.');?>
    return false;

}

document.getElementById('simSuccess').classList.add('show');

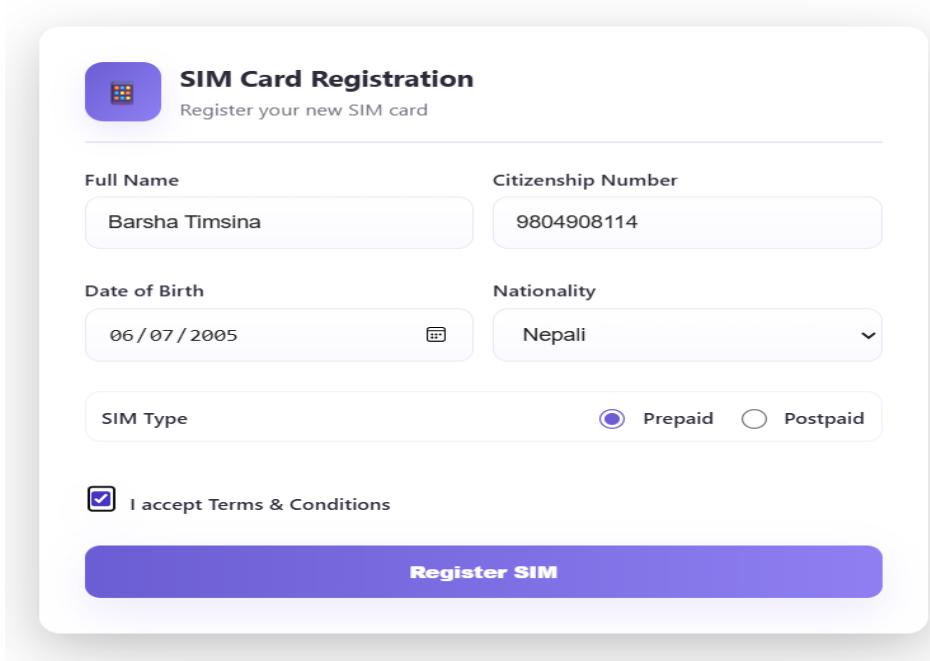
setTimeout(() => {
    document.getElementById('simForm').reset();
    document.getElementById('simSuccess').classList.remove('show');
}, 3000);

return false;
}

```

</script>

Output Snapshot:



The screenshot shows a mobile-style SIM Card Registration form. At the top, there's a purple header bar with a small icon and the text "SIM Card Registration". Below it, the main title "SIM Card Registration" is displayed next to a small purple square icon. A sub-instruction "Register your new SIM card" is shown in a smaller font.

The form fields are arranged in two columns:

- Full Name:** Barsha Timsina
- Citizenship Number:** 9804908114
- Date of Birth:** 06 / 07 / 2005
- Nationality:** Nepali
- SIM Type:** Prepaid (radio button selected)
- I accept Terms & Conditions:** A checked checkbox followed by the text.

At the bottom is a large blue "Register SIM" button.

4)Online Course Enrollment Form:

This form allows students to enroll in courses.
It checks valid email and required selections.
If the user has experience, they must provide details.

Code Snapshot:

```
<script>

function toggleExperience() {

    const expYes = document.getElementById('Barsha_06_expYes').checked;

    const container = document.getElementById('expDetailsContainer');

    const textarea = document.getElementById('Barsha_06_expDetails');

    if (expYes) {
        container.classList.remove('hidden');

        textarea.required = true;
    } else {
        container.classList.add('hidden');

        textarea.required = false;

        textarea.value = '';
    }
}

function validateCourse() {

    const name = document.getElementById('Barsha_06_name').value.trim();

    const email = document.getElementById('Barsha_06_email').value.trim();

    const course = document.getElementById('Barsha_06_course').value;

    const mode = document.querySelector('input[name="Barsha_06_mode"]:checked');

    const experience = document.querySelector('input[name="Barsha_06_experience"]:checked');

    const expDetails = document.getElementById('Barsha_06_expDetails').value.trim();

    const emailPattern = /^[^@\s]+@[^\s@]+\.[^\s@]+$/;
}
```

```

if (!name) { alert('Please enter your name.'); return false; }

if (!email) { alert('Please enter your email.'); return false; }

if (!emailPattern.test(email)) { alert('Please enter a valid email address.'); return false; }

if (!course) { alert('Please select a course.'); return false; }

if (!mode) { alert('Please select mode of study.'); return false; }

if (!experience) { alert('Please indicate if you have previous experience.'); return false; }

if (experience.value === 'Yes' && !expDetails) {

    alert('Please provide experience details.); return false;

}

alert('🎓 Enrollment Successful! Welcome to the course.);

document.getElementById('courseForm').reset();

toggleExperience();

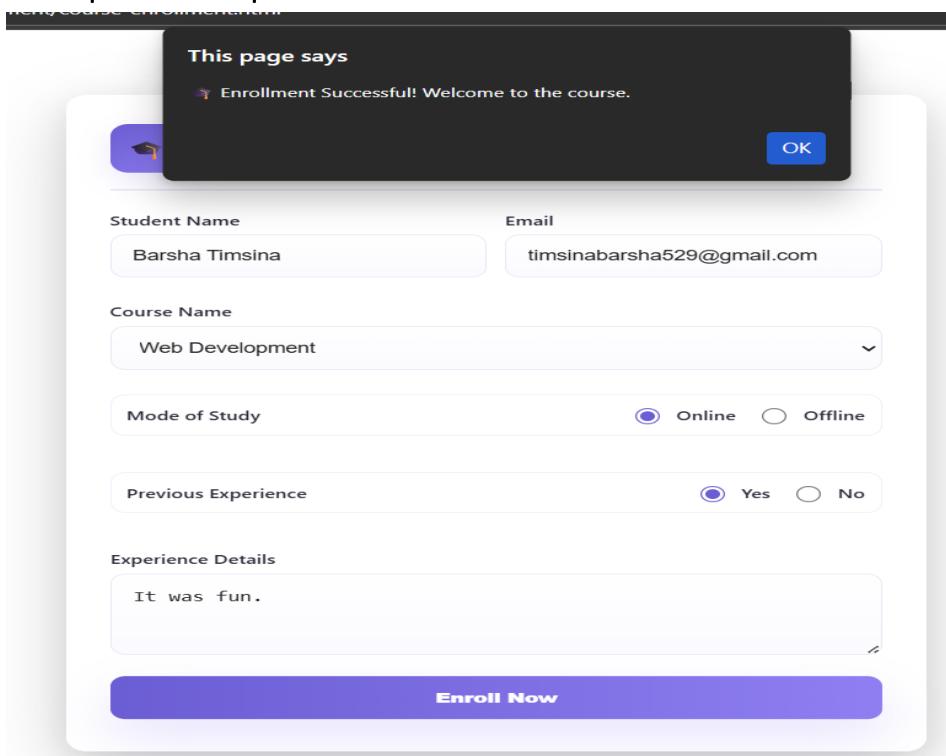
return false;

}

```

</script>

Output Snapshot:



5) Hospital Appointment Booking Form:

This form collects patient details and appointment date.

Validation prevents past date booking and ensures valid phone numbers.

A confirmation alert is shown with the patient name.

Code Snapshot:

```
<script>

function validateHospital() {

    const name = document.getElementById('Barsha_06_name').value.trim();

    const age = parseInt(document.getElementById('Barsha_06_age').value);

    const phone = document.getElementById('Barsha_06_phone').value.trim();

    const department = document.getElementById('Barsha_06_department').value;

    const date = document.getElementById('Barsha_06_date').value;

    const time = document.getElementById('Barsha_06_time').value;

    const namePattern = /^[A-Za-z\s]+$/;

    const phoneDigits = phone.replace(/\D/g, "");

    const phoneStart = phoneDigits.substring(0,2);

    if (!name) { alert('Please enter patient name.'); return false; }

    if (!namePattern.test(name)) { alert('Name should contain only letters.'); return false; }

    if (isNaN(age) || age <= 0) { alert('Age must be greater than 0.'); return false; }

    if (phoneDigits.length !== 10 || (phoneStart !== '98' && phoneStart !== '97')) {

        alert('Phone must be 10 digits and start with 98 or 97.); return false;

    }

    // Date validation - cannot be in past

    if (date) {

        const selectedDate = new Date(date);

        const today = new Date();

        today.setHours(0,0,0,0);

    }

}
```

```

if (selectedDate < today) {
    alert('Appointment date cannot be in the past.');?>
}
} else {
    alert('Please select appointment date.');?>
}

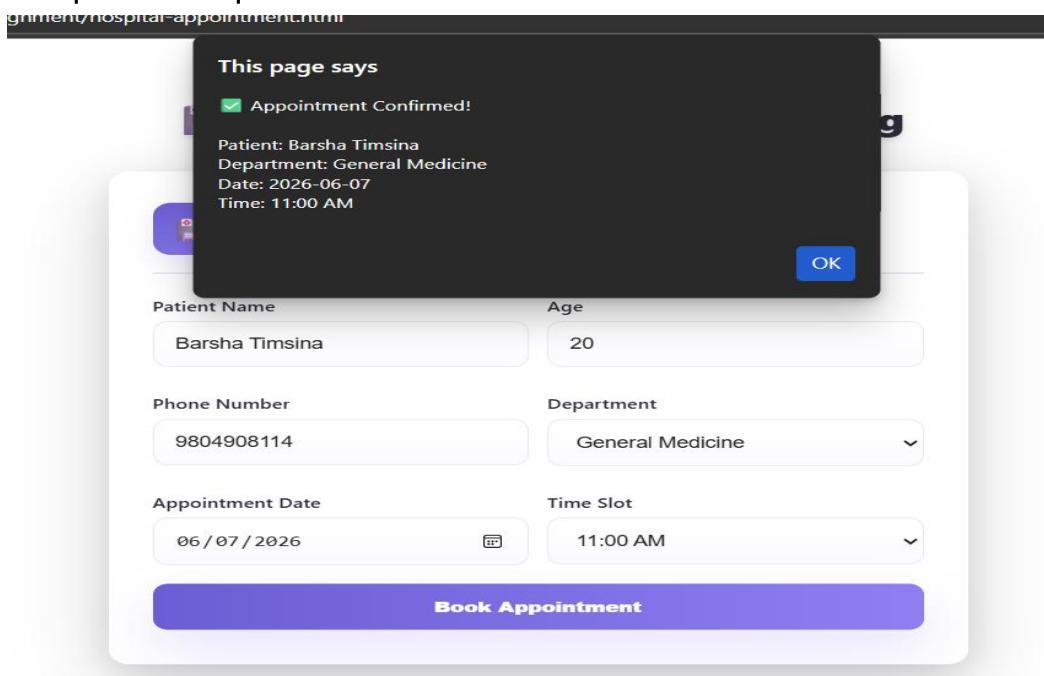
if (!department) { alert('Please select department.');?>
if (!time) { alert('Please select time slot.');?>
}

alert(`☑ Appointment Confirmed!\n\nPatient: ${name}\nDepartment: ${department}\nDate: ${date}\nTime: ${time}`);
document.getElementById('hospitalForm').reset();
return false;
}

// Set minimum date for appointment to today
document.addEventListener('DOMContentLoaded', function() {
    const today = new Date().toISOString().split('T')[0];
    document.getElementById('Barsha_06_date').setAttribute('min', today);
});

```

Output Snapshot:



Conclusion:

In this assignment, multiple forms were created using HTML and JavaScript.

Validation helps ensure users enter correct information before submission.

This assignment improved understanding of client-side validation and real-world form usage.

It also showed how forms are used in education, hospitals, telecom, and online systems.