# **NAME: Sharvayu Zade**

# **PRN: 23070521135**

# **SEC: B(B2)**

# 

# **Practical 08**

**Final Task:**

To design and implement a **Gym Admission Form** with **comprehensive form handling and live validation** using **JavaScript events**, **form properties**, and **HTML5 validation features**.

### **Task Description:**

You are required to **build a complete Gym Admission Form** that dynamically validates all user inputs both **in real time** (while typing) and **on submission**.

Your form **must** include:

* All core HTML5 input types (text, email, number, date, password, checkbox, radio, select, file).
* All key **form events** (onsubmit, onblur, onchange, oninput, etc.).
* **Custom validation messages** using setCustomValidity().
* Use of **JavaScript form object properties**, including .elements, .value, .checked, .length, .action, .method, .name, etc.
* **Live (real-time)** validation messages that update as the user types.

### **Form Fields (Minimum Required):**

| **Field** | **Type** | **Validation Rules** |
| --- | --- | --- |
| Full Name | text | Required, min 3 chars, only alphabets |
| Email | email | Valid format, required |
| Age | number | 18–60 only |
| Gender | radio | Required |
| Membership Type | select (e.g. Monthly, Quarterly, Yearly) | Cannot be default option |
| Password | password | Must contain uppercase, lowercase, number, and special character |
| Confirm Password | password | Must match Password |
| Phone | text | Only digits, exactly 10 digits |
| Date of Birth | date | Not in the future |
| Profile Photo | file | Only .jpg/.png, ≤ 1MB |
| Terms & Conditions | checkbox | Must be checked |

### **JavaScript Requirements:**

Your JavaScript must demonstrate **ALL** of the following:

1. **Accessing Forms and Elements**Use both document.forms["formName"] and getElementById() in different parts.  
   Display total number of form elements (form.length).
2. **Form Events:**onsubmit → prevent submission until all fields valid.  
   onblur → show warning when a field loses focus and is invalid.  
   onchange → validate dropdowns and checkboxes.  
   oninput / onkeyup → live validation for fields like name, email, password.  
   oninvalid → show custom error messages.  
   onreset → clear all live validation messages.
3. **Validation Techniques:**Required field check (trim() based).  
   Data type validation (isNaN, regex for text-only fields).  
   Range and length validation (for age, password).  
   Regex-based pattern matching (for email, password, phone).  
   Comparing fields (passwords).  
   Checkbox / radio selection check.  
   Dropdown validation.  
   File upload validation (type + size).  
   Date validation (no future DOB).  
   Custom messages with setCustomValidity() and reportValidity().
4. **Live Error Display:**Each input must have a small <span> or <div> below it that displays the current validation status in real time (e.g. ✅ Valid / ❌ Error).  
   Errors must disappear immediately when corrected.
5. **Final Form Submission:**On successful validation, form should show a confirmation message like:  
    *“Form submitted successfully! Welcome to FitZone Gym.”*Use preventDefault() to handle custom submission logic.

### **Bonus Requirements (for extra marks):**

Display user’s **selected membership type**, **name**, and **email** in a confirmation preview box before final submission.  
Disable submit button until all fields are valid.  
Use **CSS transitions** or **color-coded validation indicators** (green/red borders).

### **Restrictions:**

**No external JavaScript libraries** (e.g., jQuery, React, or Bootstrap validation).  
**No AI-generated or copied code** every line must be self-written and explained.  
You must **comment your code** clearly explaining the purpose of each validation and event.

## **Sample Output Description (for Reference Only)**

### **Expected Layout Overview**

The **Gym Admission Form** should appear centered on the webpage, visually clean, and neatly structured.

Each input field should have:

* A **label** (e.g., “Full Name:”)
* An **input box**
* A small **validation message area** right below it (e.g., “✅ Looks good” or “❌ Name must be at least 3 letters”).

Example (visual description only):

----------------------------------------------

🏋️‍♂️ FitZone Gym Admission Form

----------------------------------------------

Full Name: [\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_]

❌ Name must be at least 3 letters

Email: [\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_]

✅ Valid email address

Age: [\_\_]

❌ Age must be between 18 and 60

Gender: ( ) Male ( ) Female ( ) Other

❌ Please select your gender

Membership Type: [ Monthly ▼ ]

✅ Membership selected

Password: [\*\*\*\*\*\*\*\*]

❌ Weak password – include uppercase, lowercase, number, special char.

Confirm Password: [\*\*\*\*\*\*\*\*]

✅ Passwords match

Phone: [\_\_\_\_\_\_\_\_\_\_]

❌ Must be exactly 10 digits

Date of Birth: [ yyyy-mm-dd ]

✅ Valid date

Profile Photo: [Choose File] No file chosen

❌ Only JPG/PNG under 1MB allowed

[ ] I agree to the terms & conditions

❌ Please accept terms to continue

----------------------------------------------

[ Submit ] [ Reset ]

----------------------------------------------

### **Live Validation Behavior**

1. **As the user types**:  
   The border of each field changes **red → green** dynamically.  
   Validation messages update immediately under each field.  
   Example: typing only two letters in “Full Name” instantly shows  
    ❌ Name must be at least 3 characters.
2. **When user moves to another field (onblur)**:  
   If the field is empty or invalid, a red error appears instantly.  
   Example: leaving “Email” blank triggers  
    ❌ Email is required.
3. **When the user changes dropdowns or radio buttons (onchange)**:  
   The related validation message appears or disappears automatically.
4. **Password fields**:  
   The form displays feedback like:  
   ❌ Too short  
   ❌ Must include number  
   ✅ Strong password  
   “Confirm Password” field must check equality **live** as the user types.
5. **File upload**:  
   Uploading a wrong format (e.g., PDF) shows  
    ❌ Only JPG or PNG allowed.  
   Uploading a large file shows  
    ❌ File size exceeds 1MB.
6. **Submit button behavior**:  
   Disabled until all validations pass.

**When all fields are correct, button activates and shows success message:** ✅ Form submitted successfully! Welcome to FitZone Gym.

If any field is invalid, form **does not reload** (uses preventDefault()).

1. **Reset button**:  
   Clears all fields and removes all validation messages in real time.

### **Visual Feedback Summary Table**

| **Color** | **Meaning** |
| --- | --- |
| 🔴 Red Border | Field invalid |
| 🟢 Green Border | Field valid |
| ⚠️ Yellow Border | Incomplete input (e.g., password partly strong) |
| ✅ Message | Validation success |
| ❌ Message | Validation failed |

### **Note to Students**

This is **only a visual and behavioral reference** — not a code template. You must design your own HTML structure, write your own CSS for styling, and implement **every validation** using **your own JavaScript logic**.

I may test your code by:

* Typing invalid data to see if your events work.
* Checking the browser console for errors.
* Asking you to explain how each event and property works in your script.

