What is Validation
Validation is the process of checking whether the entered data is correct, complete, and in the proper format.

Validation म्हणजे user ने दिलेली माहिती बरोबर आहे का, पूर्ण आहे का आणि योग्य format मध्ये आहे का हे तपासणे.
Why is Validation Important?
1-Prevents wrong data
2-Improves user experience
3-Saves server time
4-Shows instant error messages

Example

- Email format check
- Password length check
- Check input field is empty

What is Verification
Verification is the process of confirming that the provided data is true or actually belongs to the user.
Verification म्हणजे दिलेली माहिती खरी आहे का हे confirm करणे.

◆ **Example**

- OTP verification
- Email verification link
- Captcha check

◆ **Real-life Example**

English:
User receives OTP → enters OTP → system verifies it
Marathi:
User ला OTP येतो → OTP टाकतो → system तपासते

What is Authentication
Authentication is the process of identifying a user by checking login credentials.
Authentication म्हणजे user कोण आहे हे तपासणे (login process).
Jo login krto to kon ahe->admin ka user etc

- **Example**
    - Username + Password
    - Login with OTP
    - Fingerprint / Face ID

What is Authorization

Authorization is the process of deciding what an authenticated user is allowed to do.

Authorization म्हणजे user ला कोणते काम करण्याची परवानगी आहे हे ठरवणे.

Example

- Admin can delete data
- User can only view data

Validation using js

Example

Html from

```
<form onsubmit="return validateForm()">
  Name: <input type="text" id="name">
  <input type="submit">
</form>
```

Js

1 -checking field is not empty or required Field Validation

```
function validateForm() {
  let name = document.getElementById("name").value;

  if (name ==="") {
    alert("Name must be filled");
    return false;
  }
  return true;
}
```

2-> Length Validation

```
if (password.length < 6) {
  alert("Password too short");
}
```

3-> Number Validation

```
let age = document.getElementById("age").value;
if (isNaN(age)) {
  alert("Enter number only");
}
```

4-> Email Validation (Important)

`@` आणि `.` असणे आवश्यक

```
let email = document.getElementById("email").value;
let pattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

if (!pattern.test(email)) {
  alert("Invalid Email");
}
```

Jr apn manually kel tr

```
let email = document.getElementById("email").value;
if (email == "") {
  alert("Email required");
  return false;
}
if (email.indexOf("@") == -1 || email.indexOf(".") == -1) {
  alert("Invalid Email");
  return false;
}
```

5->Mobile Number Validation

```
if (!/^[1-9]\d{9}$/.test(mobile)) {
  alert("Invalid Mobile Number");
}
```

Manually

```
let mobile = document.getElementById("mobile").value;
```

```
if (mobile.length != 10) {
  alert("Mobile number must be 10 digits");
  return false;
}
if (isNaN(mobile)) {
  alert("Mobile must contain only numbers");
  return false;
}
```

6->strong password
Password madhe uppercase ,lowercase ani special symbol pahije

```
let strong = /^(?=.*[A-Z])(?=.*\d)(?=.*[@$!]).{8,}$/;
if (!strong.test(password)) {
  alert("Weak Password");
}
```

Manually

```
let hasUpper = false;
let hasNumber = false;
let hasSpecial = false;
Let password=document.getElementById("password");
for (let i = 0; i < password.length; i++) {
  let ch = password.charAt(i);
  if (ch >= 'A' && ch <= 'Z') {
    hasUpper = true;
  } else if (ch >= '0' && ch <= '9') {
    hasNumber = true;
  } else {
    hasSpecial = true;
  }
}
if (!(hasUpper && hasNumber && hasSpecial)) {
  alert("Password must contain uppercase, number and special character");
  return false;
}
```

7->Confirm Password

```
if (pwd !== cpwd) {
  alert("Password mismatch");
}
```

8->checkbox validation

```
if (!document.getElementById("agree").checked) {
  alert("Accept terms");
}
```

9->Radio button

```
          let gender = document.querySelector('input[name="gender"]:checked');

if (gender == null) {
  alert("Select gender");
}
```

10->drop down list

```
if (city == "") {
  alert("Select city");
}
```

11->Stop Form Submission

```
  return false;
```

Jr apn return true kel tr ch form submit hoto

Example

html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
  <form onsubmit="return validateForm()">
  Name: <input type="text" id="name"><br><br>
  Age: <input type="text" id="age"><br><br>
  Email: <input type="text" id="email"><br><br>
  Mobile: <input type="text" id="mobile"><br><br>
  Password: <input type="password" id="password"><br><br>
  Confirm Password: <input type="password" id="cpwd"><br><br>


  Gender:
  <input type="radio" name="gender">Male
  <input type="radio" name="gender">Female<br><br>

  City:
  <select id="city">
    <option value="">Select</option>
    <option>Pune</option>
    <option>Mumbai</option>
  </select><br><br>

  <input type="checkbox" id="agree"> I Agree<br><br>

  <input type="submit">
</form>

    <script src="./js/FetchMethod.js"></script>
  </body>
</html>
```

JS

```
function validateForm() {

  let name = document.getElementById("name").value;
  if (name === "") {
    alert("Name must be filled");
    return false;
  }
  let password = document.getElementById("password").value;
  if (password.length < 6) {
    alert("Password too short");
    return false;
  }
  let age = document.getElementById("age").value;
  if (isNaN(age)) {
    alert("Enter number only");
    return false;
  }

  let email = document.getElementById("email").value;
  let pattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
  if (!pattern.test(email)) {
    alert("Invalid Email");
    return false;
  }

  let mobile = document.getElementById("mobile").value;
  if (!/^[6-9]\d{9}$/.test(mobile)) {
    alert("Invalid Mobile Number");
    return false;
  }

  let strong = /^(?=.*[A-Z])(?=.*\d)(?=.*[@$!]).{8,}$/;
  if (!strong.test(password)) {
    alert("Weak Password");
    return false;
  }

  let cpwd = document.getElementById("cpwd").value;
  if (password !== cpwd) {
    alert("Password mismatch");
    return false;
  }
```

```
  if (!document.getElementById("agree").checked) {
    alert("Accept terms");
    return false;
  }

  let gender = document.querySelector('input[name="gender"]:checked');
  if (gender == null) {
    alert("Select gender");
    return false;
  }

  let city = document.getElementById("city").value;
  if (city == "") {
    alert("Select city");
    return false;
  }
          //sagle fild validate zale tr ch return krnar ani form submit honar
  return true;
}
```

Validation using html
Khalchya attribute apn as property mhnun use krayche

1 required

Makes the input field mandatory. Form cannot be submitted if empty.
हा attribute वापरल्यास field रिकामी ठेवता येत नाही.
<input type="text" required>

2 autofocus

Automatically places the cursor in the input field when the page loads.
Page load होताच cursor त्या field मध्ये जातो.
<input type="text" autofocus>

3 maxlength

Sets the maximum number of characters allowed.
किती जास्तीत जास्त अक्षरं टाकता येतील ते ठरवतो.
<input type="text" maxlength="10">

4 minlength

Sets the minimum number of characters required.
किमान किती अक्षरं हव्यात ते ठरवतो.
<input type="password" minlength="6">

5 pattern

Validates input using Regular Expression.
Regex वापरून input validate करतो.
<input type="text" pattern="[0-9]{10}">

6 type

Ensures specific input type like email, number, date.
विशिष्ट प्रकारचा input घेण्यासाठी वापरतो.
<input type="email">

7 placeholder

Shows hint text inside the input field.
Field मध्ये hint text दाखवतो.
<input type="text" placeholder="Enter name">

8 readonly

User can see but not edit the value.
Value दिसते पण बदलता येत नाही.
<input type="text" readonly>

9 disabled

Disables the input field completely.
Field पूर्णपणे बंद करतो.
<input type="text" disabled>

10 min and max

Defines minimum and maximum value.
किमान आणि कमाल value ठरवतो.
<input type="number" min="18" max="60">

11 checked

Pre-selects the checkbox or radio button.
आधीच checkbox / radio select करतो.
<input type="checkbox" checked>

Regular Expression and pattern
RegEx is a pattern used to match, search, or validate text.

RegEx म्हणजे text तपासण्यासाठी वापरले जाणारे नियम (pattern)
उदा. email, mobile, password योग्य आहे का ते तपासणे.

Rule of Regex ->regEx always / pattern / madhe ch lihaycha

Symbol	Meaning (English)	Marathi अर्थ
`^`	Start of string	सुरुवात
`$`	End of string	शेवट
`.`	Any one character	कोणतंही एक अक्षर
`*`	0 or more times	0 किंवा जास्त
`+`	1 or more times	किमान 1
`?`	Optional	ऐच्छिक
`\d`	Digit (0–9)	अंक
`\s`	Space	space
`[abc]`	a or b or c	a/b/c
`[0-9]`	Any digit	कोणताही अंक

Simple Example

Only number

[0-9]

Only digit

[a-zA-Z]

Email RegEx

^[^\s@]+@[^\s@]+\.[^\s@]+$

Email मध्ये @ आणि . असणं गरजेचं

Part	Meaning
^	Start
[^\s@]+	text before @
@	must have @
[^\s@]+	domain name
\.	dot
$	end

Mobile number

^[6-9]\d{9}$

- ^[6-9] → starts with 6–9
- \d{9} → next 9 digits
- $ → end

Marathi: 10 अंकी नंबर जो 6–9 ने सुरू होतो

Html madhe ks use kryach

```
<input type="text" pattern="[0-9]{10}" required>
```

Js madhe pattern or regEx use ks kraych

```
let pattern = /^[6-9]\d{9}$/;
let mobile = "9876543210";   //yethe apn user krun getEleById ni input gheto

if (pattern.test(mobile)) {
  alert("Valid");
} else {
  alert("Invalid");
}
```