



COLLEGE CODE: 9602

COLLEGE NAME: ARUNACHALA COLLEGE OF ENGINEERING FOR WOMEN

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING

STUDENT NM-ID: 1FAA98F1BD4F7A283654FA43CE1CA14F

ROLL NO: 960223104085

DATE:8-10-2025

Completed the project named as

Phase 5 ENHANCEMENTS&DEPLOYMENT

NAME: User Registration with Validation

SUBMITTED BY

NAME: Raesa AP

MOBILE NO: 9445773597

Final demo walkthrough:

The final demonstration of the User Registration with Validation.

User side:

- To collect necessary user information.
- To create a unique user identity for login.
- To store user details securely in the database.
- To allow personalized access and features.

Admin side:

- Manage all registered users.
- Verify and approve new registration.
- Maintain data integrity and security.
- View , edit, or delete user records.
- Handle system updates or errors.

Live Features:

- To provide immediate feedback to users.
- To make the system interactive and responsive.

- To help administrators monitor data in real time.
- To reduce errors and improve system efficiency.

Deployment:

- To make the system accessible to users.
- To ensure all components (frontend, backend, database) work together.
- To maintain and update the system easily.
- To monitor and fix errors that occur during real-time use.

Project Report:

The project report includes:

Introduction: The User Registration with Validation system is a web-based application that allows users to create an account by entering their details such as name, email, and password.

Validation ensures that the data entered by the user is correct and secure before it is stored in the database.

This project focuses on building a simple, secure, and user-friendly registration process.

Features Implemented:

- Real-time form validation

- Username and email availability check
- Password strength indicator
- Responsive web design (mobile friendly)
- Admin panel for user management

Conclusion & Future scope:

Future Enhancement:

Add OTP verification for email or mobile.

Include forgot password and reset features.

Integrate profile photo upload.

Add two-step authentication for security.

Conclusion:

The User Registration with Validation system ensures accurate data collection, enhances security, and provides an efficient way to register and manage users.

It improves the reliability and usability of web applications by validating input at both user and admin levels.

Screenshots/API Documentation:

Screenshots:

The image shows a web browser window with a single tab titled "User Registration with Validation". The address bar displays "127.0.0.1:5500/user.html". The main content area has a blue gradient background with a white registration form in the center. The form is titled "User Registration" and contains four input fields: "Username" (placeholder: "Enter your username"), "Email" (placeholder: "Enter your email"), "Password" (placeholder: "Enter your password"), and "Confirm Password" (placeholder: "Re-enter your password"). Below these fields is a blue "Register" button. The Windows taskbar is visible at the bottom, showing various application icons, a search bar, and system status icons on the right, including the date and time (19:33, 08-10-2025).

Challenges & Solutions:

1. Challenge: Handling Invalid or Missing Input

Description:

Users often leave required fields empty or enter incomplete information (e.g., missing username or email).

Example: Leaving the password field blank or entering a space as username.

Solution:

Add mandatory field checks in both frontend and backend.

Display clear, user-friendly error messages for each missing input.

Use HTML attributes like required or backend validation before database insertion.

2. Challenge: Validating Email Format

Description:

Users may enter incorrect or fake email formats like raeesa@ or raeesa@com.

Solution:

Use regular expressions (Regex) to ensure the email follows the correct pattern (e.g., name@example.com).

Send a verification email to confirm the address is valid.

3. Challenge: Weak Passwords

Description:

Many users choose weak or easy-to-guess passwords (e.g., 123456, password).

Solution:

Enforce password strength rules (minimum length, uppercase/lowercase, numbers, symbols).

Provide real-time feedback on password strength using visual indicators (e.g., weak → strong).

4. Challenge: Password Confirmation Errors

Description:

Users often mistype their password in the confirmation field.

Solution:

Include a “Confirm Password” field.

Validate both password fields match before submission.

Add a “Show Password” toggle to help users check their input.

GitHub README & Setup Guide:

README contents:

- Project description, features, and tech stack.
- Screenshots and live demo link.

Steps to run locally:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>User Registration with Validation</title>
```

```
<style>
```

```
body {
```

```
font-family: Arial, sans-serif;
```

```
background: linear-gradient(135deg, #74ABE2, #5563DE);  
  
display: flex;  
  
justify-content: center;  
  
align-items: center;  
  
height: 100vh;  
  
}
```

```
.container {  
  
background: #fff;  
  
padding: 30px 40px;  
  
border-radius: 10px;  
  
box-shadow: 0 4px 20px rgba(0, 0, 0, 0.2);  
  
width: 350px;  
  
}
```

```
h2 {  
  
text-align: center;  
  
color: #333;  
  
margin-bottom: 20px;
```

```
}
```

```
.form-group {  
  margin-bottom: 15px;  
}
```

```
label {  
  display: block;  
  font-weight: bold;  
  margin-bottom: 5px;  
}
```

```
input {  
  width: 100%;  
  padding: 10px;  
  border: 1px solid #ccc;  
  border-radius: 6px;  
  font-size: 14px;  
}
```

```
input:focus {  
  
  outline: none;  
  
  border-color: #5563DE;  
  
}
```

```
.btn {  
  
  width: 100%;  
  
  padding: 10px;  
  
  background-color: #5563DE;  
  
  border: none;  
  
  color: white;  
  
  border-radius: 6px;  
  
  font-size: 16px;  
  
  cursor: pointer;  
  
}
```

```
.btn:hover {  
  
  background-color: #3b49b4;
```

```
}
```

```
.message {
```

```
  text-align: center;
```

```
  margin-top: 10px;
```

```
  font-size: 14px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
  <div class="container">
```

```
    <h2>User Registration</h2>
```

```
    <form id="registrationForm">
```

```
      <div class="form-group">
```

```
        <label for="username">Username</label>
```

```
        <input type="text" id="username" placeholder="Enter your username">
```

```
      </div>
```

```
      <div class="form-group">
```

```
<label for="email">Email</label>
```

```
<input type="email" id="email" placeholder="Enter your email">
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="password">Password</label>
```

```
<input type="password" id="password" placeholder="Enter your password">
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="confirmPassword">Confirm Password</label>
```

```
<input type="password" id="confirmPassword" placeholder="Re-enter your  
password">
```

```
</div>
```

```
<button type="submit" class="btn">Register</button>
```

```
<div class="message" id="message"></div>
```

```
</form>
```

```
</div>
```

```
<script>
```

```
    document.getElementById("registrationForm").addEventListener("submit",  
function(event) {
```

```
        event.preventDefault();
```

```
        let username = document.getElementById("username").value.trim();
```

```
        let email = document.getElementById("email").value.trim();
```

```
        let password = document.getElementById("password").value.trim();
```

```
        let confirmPassword =  
document.getElementById("confirmPassword").value.trim();
```

```
        let message = document.getElementById("message");
```

```
        // Username validation
```

```
        if (username === "") {
```

```
            message.textContent = "Username is required.";
```

```
            message.style.color = "red";
```

```
            return;
```

```
        }
```

```
        // Email validation
```

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

if (!emailPattern.test(email)) {

    message.textContent = "Please enter a valid email address.";

    message.style.color = "red";

    return;

}


// Password validation

if (password.length < 6) {

    message.textContent = "Password must be at least 6 characters long.";

    message.style.color = "red";

    return;

}


if (password !== confirmPassword) {

    message.textContent = "Passwords do not match.";

    message.style.color = "red";

    return;

}
```

```
// Success

message.textContent = "Registration successful!";

message.style.color = "green";


// (Optional) Reset form

document.getElementById("registrationForm").reset();

});

</script>

</body>

</html>
```

Final Submission (Remo + Depolyed Link):

Git Hub deployed link:

<https://github.com/raeesaa.1794-byte/naan-mudhalvan>

