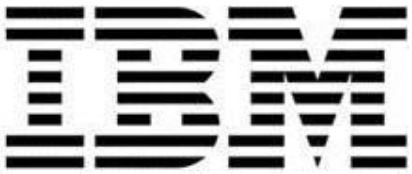


COLLEGE CODE:9123

COLLEGE NAME: SACS MAVMM ENGINEERING COLLEGE.

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING.

STUDENT NM ID:

CFB34EA92123C2D8F4742A1358F89EDC

D16827A562A6A729A3A45D7AAF6A3F93

193DCE73FF18EE79D1247F7F9EA2B8D8

39561F1F42801DB9D6742D4850F7685E

ROLL NO:

912323104023

912323104012

912323104034

912323104035

DATE:29/09/2025

Complete the project :

Phase User Registration With Validation Technology project

Name	Mobile number
KAVYA B	8248730761
DHARSHANA S	9342645618
PREETHISHARAN C	9715581398
PRIYADHARSHINI R	8438789396

Program:

index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initialscale=1.0">
<title>User Registration Form</title>
   <link rel="stylesheet" href="style.css">
</head>
<body>
   <div class="container">
      <div class="form-wrapper">
         <h1>Create Your Account</h1>
         <p class="subtitle">Join us today and get started</p>

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

```
            <!-- Full Name Field -->
            <div class="form-group">
                <label for="fullName">Full Name *</label>
                <input type="text" id="fullName" name="fullName" required>
                <span class="error-message" id="fullNameError"></span>
            </div>

            <!-- Email Field -->
            <div class="form-group">
                <label for="email">Email Address *</label>
                <input type="email" id="email" name="email" required>
                <span class="error-message" id="emailError"></span>
            </div>

            <!-- Password Field -->
            <div class="form-group">
                <label for="password">Password *</label>
                <div class="password-wrapper">
                    <input type="password" id="password" name="password" required>
                    <button type="button" class="toggle-password" id="togglePassword">
                        <span class="eye-icon">👁</span>
                    </button>
                </div>
                <div class="password-strength" id="passwordStrength">
                    <div class="strength-bar">
                        <div class="strength-fill" id="strengthFill"></div>
                    </div>
                    <span class="strength-text" id="strengthText">Password
strength</span>
                </div>
                <span class="error-message" id="passwordError"></span>
            </div>

            <!-- Confirm Password Field -->
            <div class="form-group">
```

```
            <label for="confirmPassword">Confirm Password *</label>
            <div class="password-wrapper">
                <input type="password" id="confirmPassword"
name="confirmPassword" required>
                <button type="button" class="toggle-password"
id="toggleConfirmPassword">
                    <span class="eye-icon">👁</span>
                </button>
            </div>
            <span class="error-message" id="confirmPasswordError"></span>
        </div>

        <!-- Phone Number Field -->
        <div class="form-group">
            <label for="phone">Phone Number</label>
            <input type="tel" id="phone" name="phone">
            <span class="error-message" id="phoneError"></span>
        </div>

        <!-- Date of Birth Field -->
        <div class="form-group">
            <label for="dateOfBirth">Date of Birth *</label>
            <input type="date" id="dateOfBirth" name="dateOfBirth" required>
            <span class="error-message" id="dateOfBirthError"></span>
        </div>              <!-- Submit Button -->
        <button type="submit" id="submitBtn" class="submit-btn"> Create
Account
        </button>          </form>

    <!-- Success Message -->
    <div class="success-message" id="successMessage">
        <div class="success-icon">☑</div>
        <h3>Registration Successful!</h3>
        <p>Your account has been created successfully. Welcome aboard!</p>
    </div>
```

```
      </div>
        <h2>Already have an account? <a href="login.html">Login here</a></h2>
  </div>

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

style.css

```
* {     margin: 0;     padding: 0;      box-
sizing: border-box;
}

body {      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);     min-
height:
100vh;     display: flex;     align-items: center;     justify-content: center;
padding: 20px;
}

.container {     width: 100%;      max-
width: 500px;
}
.form-wrapper {     background: white;     border-radius: 16px;     box-shadow: 0
20px 40px rgba(0, 0, 0, 0.1);     padding: 40px;     position: relative;      overflow:
hidden;
}
.form-wrapper::before {      content: '';
position: absolute;      top: 0;     left: 0;
right: 0;     height: 4px;
    background: linear-gradient(90deg, #667eea, #764ba2);  }

h1 {     color: #333;     font-size: 28px;     font-weight: 700;     margin-bottom: 8px;
    text-align: center;
```

```
}

.subtitle {     color: #666;     text-align: center;     margin-bottom: 32px; font-size:
16px;
}
.form-group {
   margin-bottom: 24px;
}

label {     display: block;     margin-bottom: 8px;     color: #333;     font-weight: 600;
font-size: 14px;
}

input {     width: 100%;     padding: 14px 16px;     border: 2px solid
#e1e5e9;     border-radius: 8px;     font-size: 16px;     transition: all 0.3s ease;
background-color: #fafbfc;
}

input:focus {     outline: none;     border-color: #667eea;     background-color: white;
box-shadow: 0 0 0 3px rgba(102, 126, 234, 0.1);
}

input:hover {     border-color: #c1c9d2;
}

/* Validation States */  input.valid {
border-color: #28a745;
background-color: #f8fff9;
}

input.invalid {     border-color: #dc3545;      background-
color: #fff8f8;
}

input.valid:focus {     box-shadow: 0 0 0 3px rgba(40, 167, 69, 0.1);  }
input.invalid:focus {     box-shadow: 0 0 0 3px rgba(220, 53, 69, 0.1);
```

```
}
/* Password Wrapper */ .password-wrapper {     position: relative;
}
.toggle-password {     position: absolute;     right: 12px;     top: 50%;     transform:
translateY(-50%);     background: none;     border: none;     cursor:
pointer;     padding: 4px;     border-radius: 4px;
transition: background-color 0.2s ease;
}
.toggle-password:hover {  background-color:
#f0f0f0;
}
.eye-icon {     font-size: 18px;
opacity: 0.6;
}
/* Password Strength Meter */ .password-strength {      margin-
top: 8px;
}
.strength-bar {     width: 100%;     height: 4px;     background-color:
#e1e5e9;     border-radius: 2px;     overflow: hidden;      margin-
bottom: 4px;
}

.strength-fill {     height: 100%;     width: 0%;     transition: all 0.3s ease;      border-
radius: 2px;
}

.strength-fill.weak {     width: 25%;      background-
color: #dc3545;
}

.strength-fill.fair {     width: 50%;  background-color:
#ffc107;
}

.strength-fill.good {     width: 75%;  background-color:
   #17a2b8;
```

```
}

.strength-fill.strong {     width: 100%;      background-
color: #28a745;
}

.strength-text {     font-size: 12px;
color: #666;
}
/* Error Messages */ .error-message {     display: block;     color: #dc3545;     fontsize:
12px;     margin-top: 4px;     min-height: 16px;     opacity: 0;     transform:
translateY(-4px);
   transition: all 0.3s ease;
}
.error-message.show {     opacity: 1;
transform: translateY(0);
}
/* Submit Button */
.submit-btn { width: 100%; padding: 16px;      background: linear-
gradient(135deg, #667eea 0%, #764ba2 100%);     color: white;     border:
none;     border-radius: 8px;     font-size: 16px;     font-weight: 600;     cursor:
pointer;     transition: all 0.3s ease;     position: relative;      overflow: hidden;
}
.submit-btn:hover:not(:disabled) {     transform: translateY(-2px);      box-
shadow: 0 8px 25px rgba(102, 126, 234, 0.3);
}
.submit-btn:active:not(:disabled) {
transform: translateY(0);
}
.submit-btn:disabled {     background: #ccc;     cursor: not-allowed;     transform:
none;      box-shadow: none;
}
/* Success Message */ .success-message {     display: none;     text-align: center;
padding: 32px;     background: linear-gradient(135deg, #28a745,
#20c997);     color: white; border-radius: 12px;
margin-top: 24px;
```

```
}
.success-message.show          {
display: block;        animation:
slideIn 0.5s ease;
}
.success-icon {     font-size: 48px;      margin-
bottom: 16px;
}
.success-message h3 {     margin-bottom: 8px;      font-
size: 24px;
}
.success-message p {
opacity: 0.9;      font-size:
16px;
}
/* Animations */
@keyframes slideIn {
   from {        opacity: 0;
transform: translateY(20px);      }
to {        opacity: 1;
transform: translateY(0);
   } }
/* Responsive Design */
@media (max-width: 600px) { .form-wrapper {        padding: 24px;
margin: 10px;
   }          h1 {        font-size:
24px;
   }
       input {        padding: 12px 14px;        font-size: 16px; /* Prevents zoom on iOS
*/
   }

   .submit-btn {
padding: 14px;
   } }
@media (max-width: 400px) {
```

```
  .container {
padding: 10px;
  }

  .form-wrapper {
padding: 20px;
  } }
```

script.js

```
class RegistrationForm {    constructor() {        this.form =
document.getElementById('registrationForm');
this.submitBtn = document.getElementById('submitBtn');
      this.successMessage = document.getElementById('successMessage');

      this.fields = {            fullName:
document.getElementById('fullName'),            email:
document.getElementById('email'),            password:
document.getElementById('password'),            confirmPassword:
document.getElementById('confirmPassword'),            phone:
document.getElementById('phone'),
         dateOfBirth: document.getElementById('dateOfBirth')
      };

      this.errors = {            fullName:
document.getElementById('fullNameError'),            email:
document.getElementById('emailError'),            password:
document.getElementById('passwordError'),            confirmPassword:
document.getElementById('confirmPasswordError'),            phone:
document.getElementById('phoneError'),
         dateOfBirth: document.getElementById('dateOfBirthError')
      };

      this.validationState = {
fullName: false,            email: false,
```

```
password: false,
confirmPassword: false,
phone: true, // Optional field
dateOfBirth: false
        };
this.init();
    }
     init() {
this.setupEventListeners();
this.setupPasswordToggle();
this.updateSubmitButton();
    }

    setupEventListeners() {
// Real-time validation
        Object.keys(this.fields).forEach(fieldName => {             const field =
this.fields[fieldName];             field.addEventListener('input', () =>
this.validateField(fieldName));             field.addEventListener('blur', () =>
this.validateField(fieldName));
        });

        // Form submission         this.form.addEventListener('submit', (e)
=> this.handleSubmit(e));
    }

    setupPasswordToggle() {         const togglePassword =
document.getElementById('togglePassword');         const
toggleConfirmPassword =
document.getElementById('toggleConfirmPassword');

        togglePassword.addEventListener('click', () => {
this.togglePasswordVisibility('password');
        });

        toggleConfirmPassword.addEventListener('click', () => {
this.togglePasswordVisibility('confirmPassword');
```

```
        });
    }

    togglePasswordVisibility(fieldName)             {
const field = this.fields[fieldName];             const
isPassword = field.type === 'password';

        field.type = isPassword ? 'text' : 'password';

        const toggleBtn = fieldName === 'password' ?
document.getElementById('togglePassword') :
            document.getElementById('toggleConfirmPassword');

        const eyeIcon = toggleBtn.querySelector('.eye-icon');
eyeIcon.textContent = isPassword ? '🙈' : '👁';
    }

    validateField(fieldName) {         const
field = this.fields[fieldName];         const
value = field.value.trim();         let
isValid = false;         let errorMessage =
'';

        switch (fieldName) {             case 'fullName':
isValid = this.validateFullName(value);                 if
(!isValid) {                     if (value === '') {
errorMessage = 'Full name is required';
                } else if (value.length < 3) {                     errorMessage
= 'Full name must be at least 3 characters';
                } else if (!/^[a-zA-Z\s]+$/.test(value)) {
errorMessage = 'Full name can only contain letters and spaces';
                }
            }
            break;
        case
'email':
```

```
            isValid = this.validateEmail(value);
if (!isValid) {                 if (value === '') {
errorMessage = 'Email is required';
                } else {                    errorMessage = 'Please enter a
valid email address';
                }
}
break;

            case 'password':
                isValid = this.validatePassword(value);
this.updatePasswordStrength(value);             if
(!isValid) {                if (value === '') {
errorMessage = 'Password is required';
                } else {
                    errorMessage = 'Password must be at least 8 characters with uppercase,
lowercase, number, and special character';
                }
            }
            // Re-validate confirm password when password changes
if (this.fields.confirmPassword.value) {
this.validateField('confirmPassword');
            }
break;

            case 'confirmPassword':
                isValid = this.validateConfirmPassword(value);
if (!isValid) {                if (value === '') {
errorMessage = 'Please confirm your password';
                } else {                    errorMessage =
'Passwords do not match';                    }            }
break;

            case 'phone':
```

```
            isValid = this.validatePhone(value);                if
(!isValid && value !== '') {                    errorMessage = 'Phone
number must be 10-15 digits';
            }
break;

          case 'dateOfBirth':                isValid =
this.validateDateOfBirth(value);                if (!isValid) {
if (value === '') {                    errorMessage = 'Date
of birth is required';
                } else {                    errorMessage = 'You must be
at least 18 years old';
                }
}
break;
      }

      this.updateFieldUI(fieldName, isValid, errorMessage);
this.validationState[fieldName] = isValid;        this.updateSubmitButton();

      return isValid;
   }

   validateFullName(value) {        return value.length >= 3
&& /^[a-zA-Z\s]+$/.test(value);
   }

   validateEmail(value)  {              const  emailRegex  =
/^[^\s@]+@[^\s@]+\.[^\s@]+$/;                    return
emailRegex.test(value);
```

```
    }



    validatePassword(value) { const minLength
        = value.length >= 8; const hasUppercase
        = /[A-Z]/.test(value);
        const hasLowercase = /[a-z]/.test(value);        const
hasNumber = /\d/.test(value);        const hasSpecialChar =
/[!@#$%^&*(),.?":{}|<>]/.test(value);

        return minLength && hasUppercase && hasLowercase && hasNumber
&& hasSpecialChar;
    }

    validateConfirmPassword(value) {        return value ===
this.fields.password.value && value !== '';
    }

    validatePhone(value) {        if (value === '')
return true; // Optional field        const
phoneRegex = /^\d{10,15}$/;
        return phoneRegex.test(value.replace(/\D/g, ''));
    }

    validateDateOfBirth(value) {
if (!value) return false;

        const birthDate = new Date(value);        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--;
      }

      return age >= 18;
   updatePasswordStrength(password) {
      const strengthFill = document.getElementById('strengthFill'); const
      strengthText = document.getElementById('strengthText');

      if (password === '') {
strengthFill.className = 'strength-fill';
strengthText.textContent = 'Password strength';
return;
      }

      const checks = [
password.length >= 8,            /[A-
Z]/.test(password),
        /[a-z]/.test(password),
        /\d/.test(password),
        /[!@#$%^&*(),.?":{}|<>]/.test(password)
      ];

      const score = checks.filter(check => check).length;

      strengthFill.className = 'strength-fill';

      if (score <= 2) {
strengthFill.classList.add('weak');
strengthText.textContent = 'Weak password';
```

```
    }



        } else if (score === 3) {
strengthFill.classList.add('fair');
            strengthText.textContent = 'Fair password';
        } else if (score === 4) {
strengthFill.classList.add('good');
strengthText.textContent = 'Good password';
        } else {
strengthFill.classList.add('strong');
strengthText.textContent = 'Strong password'; }
    updateFieldUI(fieldName, isValid, errorMessage) {
        const field = this.fields[fieldName]; const
        errorElement = this.errors[fieldName];

        // Update field styling
field.classList.remove('valid', 'invalid');         if
(field.value.trim() !== '') {
field.classList.add(isValid ? 'valid' : 'invalid');         }

        // Update error message         if
(errorMessage) {
errorElement.textContent = errorMessage;
errorElement.classList.add('show');
        } else {
errorElement.classList.remove('show');
setTimeout(() => {                  if
(!errorElement.classList.contains('show')) {
errorElement.textContent = '';
                }
            }, 300);
        }
```

```
    }



    }

    updateSubmitButton() {         const allValid =
Object.values(this.validationState).every(state => state);
this.submitBtn.disabled = !allValid;
    }


handleSubmit(e) {
    e.preventDefault();

    // Validate all fields
let allValid = true;
```

```
    Object.keys(this.fields).forEach(fieldName =>
{         if (!this.validateField(fieldName)) {
allValid = false;
        }
    });

    if (allValid) {
        // ☑ Save user data in localStorage
const userData = {              fullName:
this.fields.fullName.value.trim(),              email:
this.fields.email.value.trim(),              password:
this.fields.password.value.trim(),              phone:
this.fields.phone.value.trim(),              dob:
this.fields.dateOfBirth.value.trim()
        };
        localStorage.setItem("registeredUser", JSON.stringify(userData));

        this.showSuccess();

        // After 3 sec → go to login.html
setTimeout(() => {
window.location.href = "login.html";
        }, 3000);
    }
}


    showSuccess() {
this.form.style.display = 'none';
        this.successMessage.classList.add('show');

        // Reset form after 3 seconds
setTimeout(() => {
this.resetForm();
        }, 3000);
```

```
    }
    resetForm() {
        this.form.reset();
this.successMessage.classList.remove('show');
this.form.style.display = 'block';

        // Reset validation states
        Object.keys(this.validationState).forEach(key => {
this.validationState[key] = key === 'phone'; // phone is optional
        });

        // Reset field styling
        Object.values(this.fields).forEach(field => {
field.classList.remove('valid', 'invalid');
        });

        // Reset error messages
        Object.values(this.errors).forEach(error => {
error.classList.remove('show');
error.textContent = '';
        });

        // Reset password strength
this.updatePasswordStrength('');

        // Reset password visibility        this.fields.password.type = 'password';
this.fields.confirmPassword.type = 'password';
document.querySelector('#togglePassword .eye-icon').textContent = '👁';
document.querySelector('#toggleConfirmPassword .eye-icon').textContent =
'👁';

        this.updateSubmitButton();
    }
}
```

```
// Initialize the form when DOM is loaded
document.addEventListener('DOMContentLoaded', () => {     new
RegistrationForm();
});
```

login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<title>Login Page</title>
<style>
  /* Reset some styles */
  * {     box-sizing:
border-box;
  }

  body {     font-family: 'Segoe UI', Tahoma, Geneva, Verdana,
sans-serif;     background: #f0f2f5;     margin: 0;     display: flex;
justify-content: center;     align-items: center;     height: 100vh;
  }

  .login-container {     background: white;
padding: 2rem 3rem;     border-radius: 8px;
box-shadow: 0 4px 12px rgba(0,0,0,0.1);
width: 100%;
   max-width: 400px;
 }
  h2 {   margin-bottom:
  1.5rem;
   text-align: center;
color: #333;
```

```
}

 .form-group {     margin-
bottom: 1.25rem;     position:
relative;
 }

 label {     display: block;
margin-bottom: 0.4rem;
font-weight: 600;     color:
#555;
 }

 input[type="email"],
input[type="password"] {
width: 100%;     padding:
0.65rem 1rem;     font-size:
1rem;     border: 1.5px solid
#ccc;     border-radius: 5px;
transition: border-color 0.3s;
 }

 input[type="email"]:focus,
input[type="password"]:focus   {
border-color: #007bff;     outline:
none;
 }

 .error-message {
color: #d93025;    font-
size: 0.85rem;
margin-top: 0.3rem;
display: none;
}
```

```
  .show-error {
display: block;
  }

  .toggle-password {
position: absolute;
right: 1rem;     top:
2.6rem;     cursor:
pointer;     user-
select: none;     font-
size: 1.1rem;
color: #666;
  }

  button {     width: 100%;
background-color: #007bff;
border: none;     color: white;
font-weight: 600;     font-size:
1.1rem;     padding: 0.75rem;
border-radius: 5px;     cursor:
pointer;     transition: background-
color 0.3s;
  }

  button:disabled {     background-
color: #a0c6ff;     cursor: not-
allowed;
 }
  button:hover:not(:disabled) {
  background-color: #0056b3; }

 .success-message {    background-
color: #d4edda;
    border: 1px solid #c3e6cb;
color: #155724;     padding:
1rem;     border-radius: 5px;
```

```
margin-top: 1rem;
display: none;     text-align:
center;     font-weight: 600;
  }
</style>
</head>
<body>

<div class="login-container">
  <h2>Login</h2>
  <form id="loginForm" novalidate>
    <div class="form-group">
      <label for="email">Email Address</label>
      <input type="email" id="email" placeholder="Enter your email" required />
      <div id="emailError" class="error-message">Please enter a valid email.</div>
</div>

    <div class="form-group">
      <label for="password">Password</label>
      <input type="password" id="password" placeholder="Enter your password"
required />
      <span id="togglePassword" class="toggle-password" title="Show/Hide
Password">👁</span>
      <div id="passwordError" class="error-message">Password must be at least 6
characters.</div>
    </div>
    <button type="submit" id="submitBtn" disabled>Login</button>
  </form>

  <div id="successMessage" class="success-message">Login successful!
Redirecting...</div>
</div>

<script>   class LoginForm {     constructor() {       this.form =
document.getElementById('loginForm');       this.emailField =
```

```
document.getElementById('email');       this.passwordField =
document.getElementById('password');       this.emailError =
document.getElementById('emailError');       this.passwordError =
document.getElementById('passwordError');       this.submitBtn =
document.getElementById('submitBtn');       this.successMessage =
document.getElementById('successMessage');       this.togglePasswordBtn
= document.getElementById('togglePassword');       this.validationState = {
email: false,         password: false,
     };
this.init();
   }
    init() {       this.emailField.addEventListener('input', () =>
this.validateEmail());       this.passwordField.addEventListener('input', () =>
this.validatePassword());       this.togglePasswordBtn.addEventListener('click',
() => this.togglePasswordVisibility());       this.form.addEventListener('submit',
(e) => this.handleSubmit(e));       this.updateSubmitButton();
   }

  validateEmail() {
     const value = this.emailField.value.trim();   const
   emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
   const isValid = emailRegex.test(value);

     if (!isValid) {
      this.emailError.classList.add('show-error');
       this.validationState.email = false;
this.emailField.classList.add('invalid');
     } else {         this.emailError.classList.remove('show-
error');         this.validationState.email = true;
this.emailField.classList.remove('invalid');
     }
     this.updateSubmitButton();
   }
```

```
   validatePassword() {       const value =
this.passwordField.value;       const isValid
= value.length >= 6;

     if (!isValid) {         this.passwordError.classList.add('show-
error');         this.validationState.password = false;
this.passwordField.classList.add('invalid');
     } else {         this.passwordError.classList.remove('show-
error');         this.validationState.password = true;
this.passwordField.classList.remove('invalid');
     }
     this.updateSubmitButton();
   }

   updateSubmitButton() {       const allValid =
Object.values(this.validationState).every(Boolean);
this.submitBtn.disabled = !allValid;
  }
   togglePasswordVisibility() {   const type = this.passwordField.type ===
   'password' ? 'text' : 'password';   this.passwordField.type = type;
    this.togglePasswordBtn.textContent = type === 'password' ? '👁' : '🙈';
  }

   handleSubmit(e) {
e.preventDefault();

 // Validate first
this.validateEmail();
this.validatePassword();

 if (Object.values(this.validationState).every(Boolean)) {
   // Get registered user from localStorage     const savedUser =
JSON.parse(localStorage.getItem("registeredUser"));
```

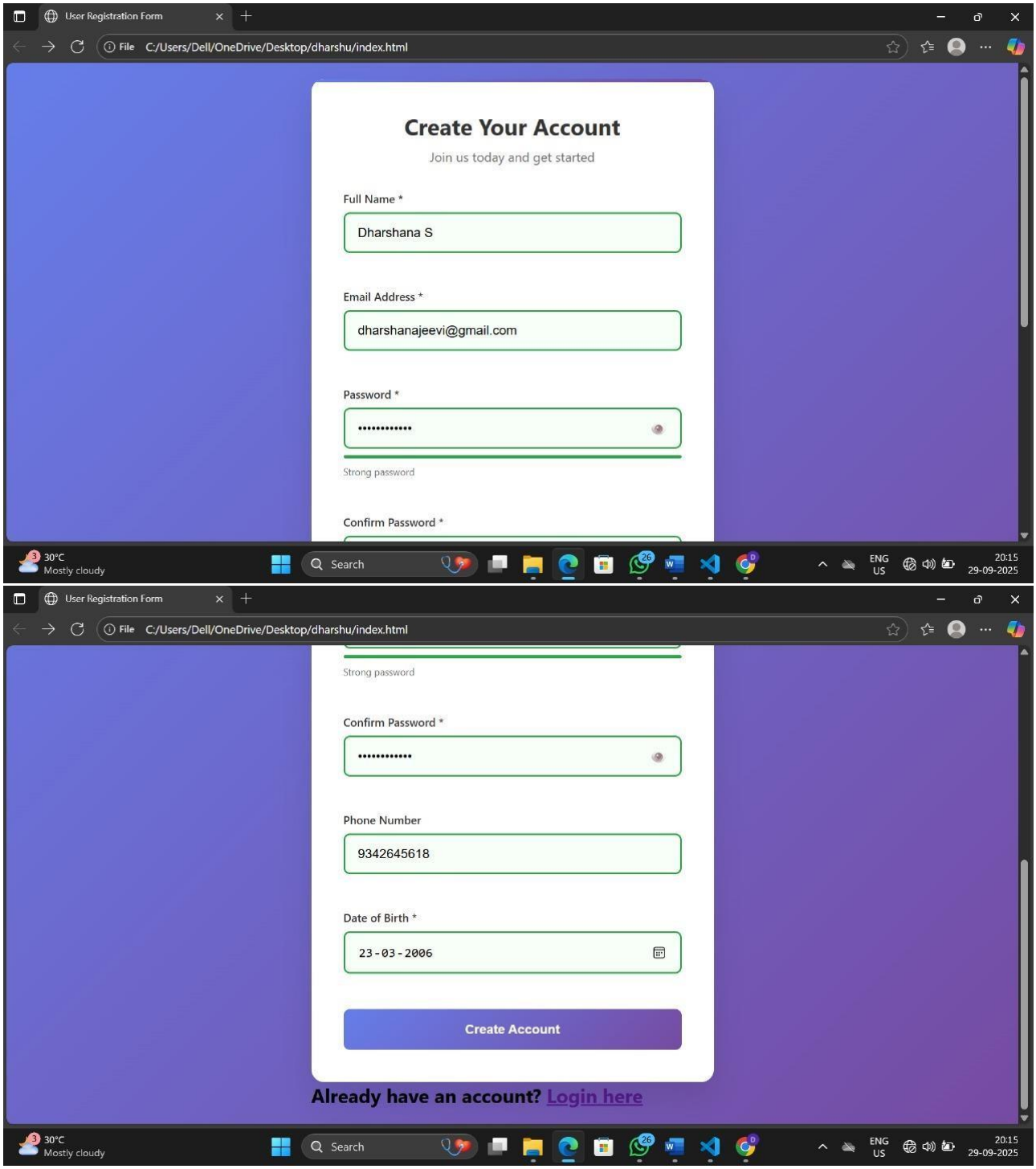
```
   if (savedUser &&          savedUser.email ===
this.emailField.value.trim() &&          savedUser.password ===
this.passwordField.value.trim()) {

     // ☑ Successful login
this.form.style.display = "none";
     this.successMessage.style.display = "block";

     setTimeout(() => {         window.location.href = "welcome.html"; //
redirect to welcome page       }, 2000);

   } else {       alert("❌ Invalid email or password. Please
register first.");
    }
  }
}

   resetForm() {
     this.form.reset();
    this.successMessage.style.display = 'none';
    this.form.style.display = 'block';
     this.validationState.email = false;
this.validationState.password = false;
this.updateSubmitButton();
     this.emailError.classList.remove('show-error');
this.passwordError.classList.remove('show-error');
this.emailField.classList.remove('invalid');
this.passwordField.classList.remove('invalid');
this.passwordField.type = 'password';
this.togglePasswordBtn.textContent = '👁';
    }
  }

  document.addEventListener('DOMContentLoaded', () => {
new LoginForm();
```

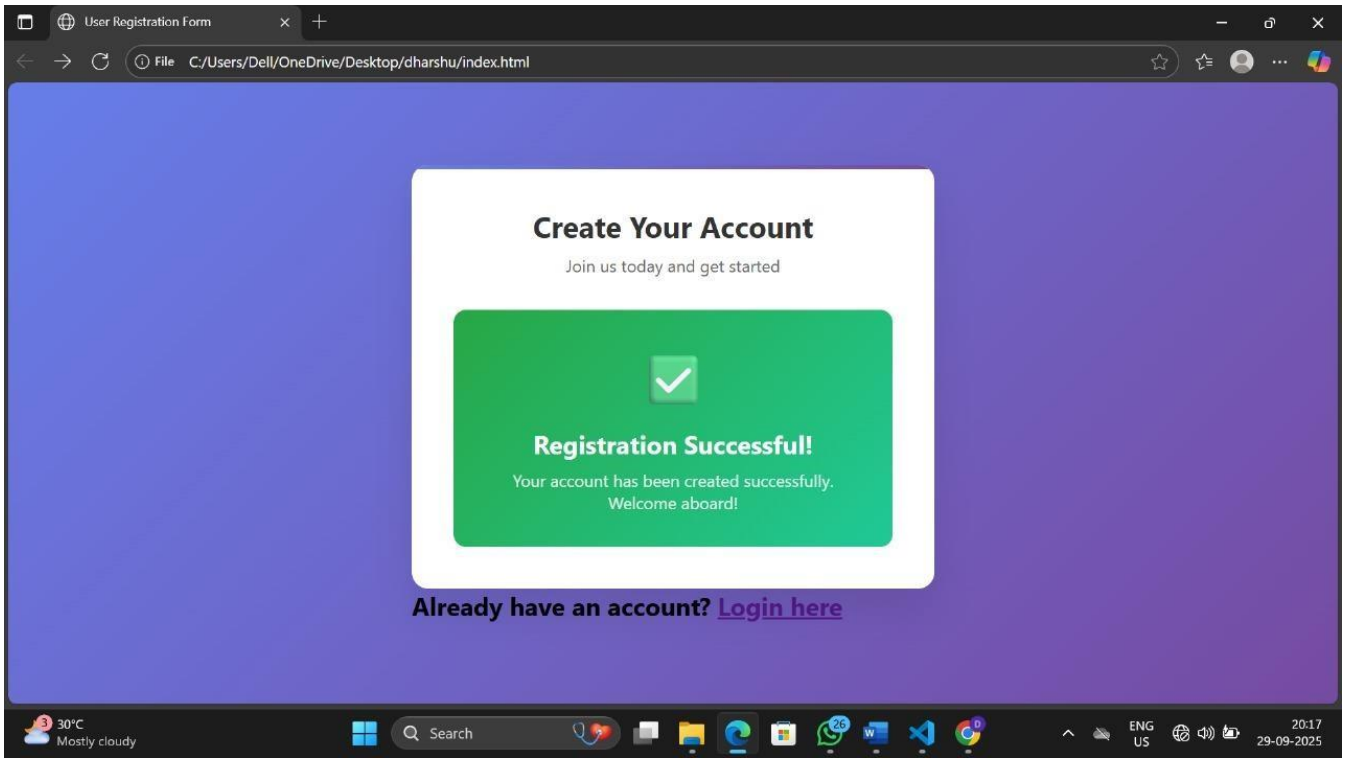
```
  });
</script>

</body>
</html>
```

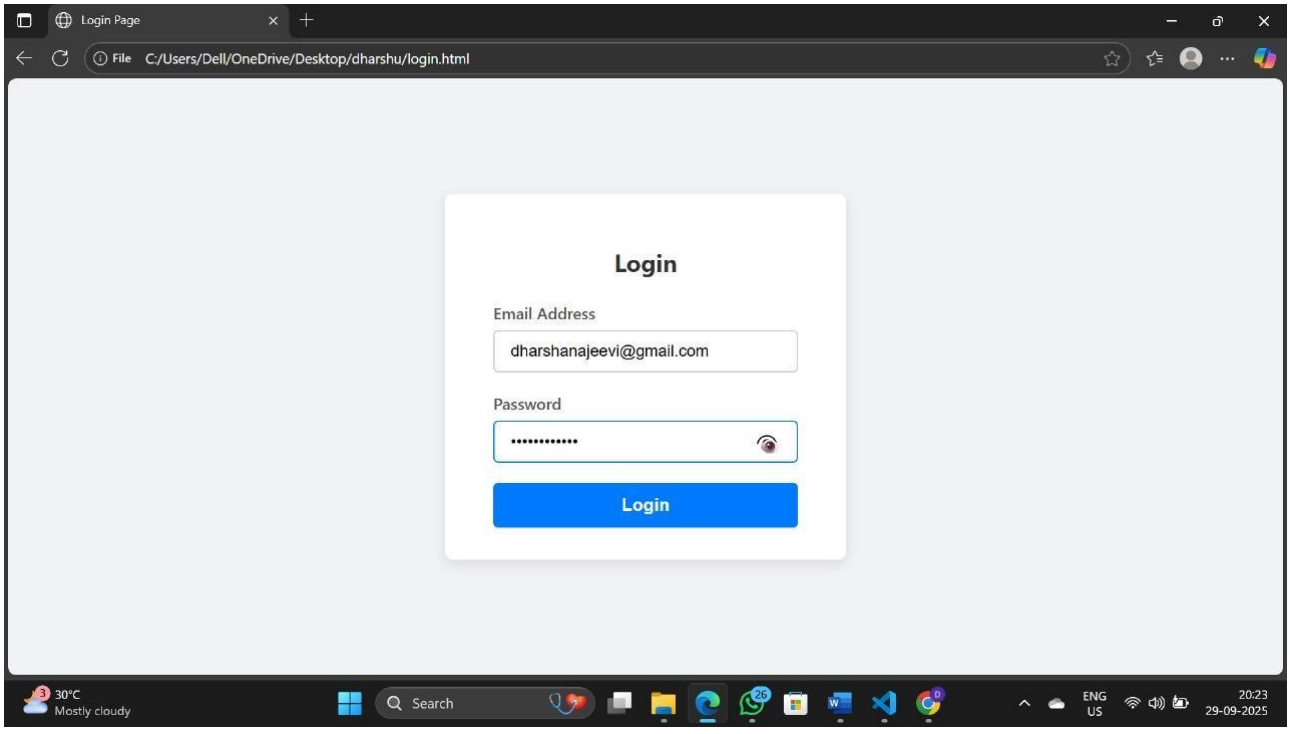
Welcome.html

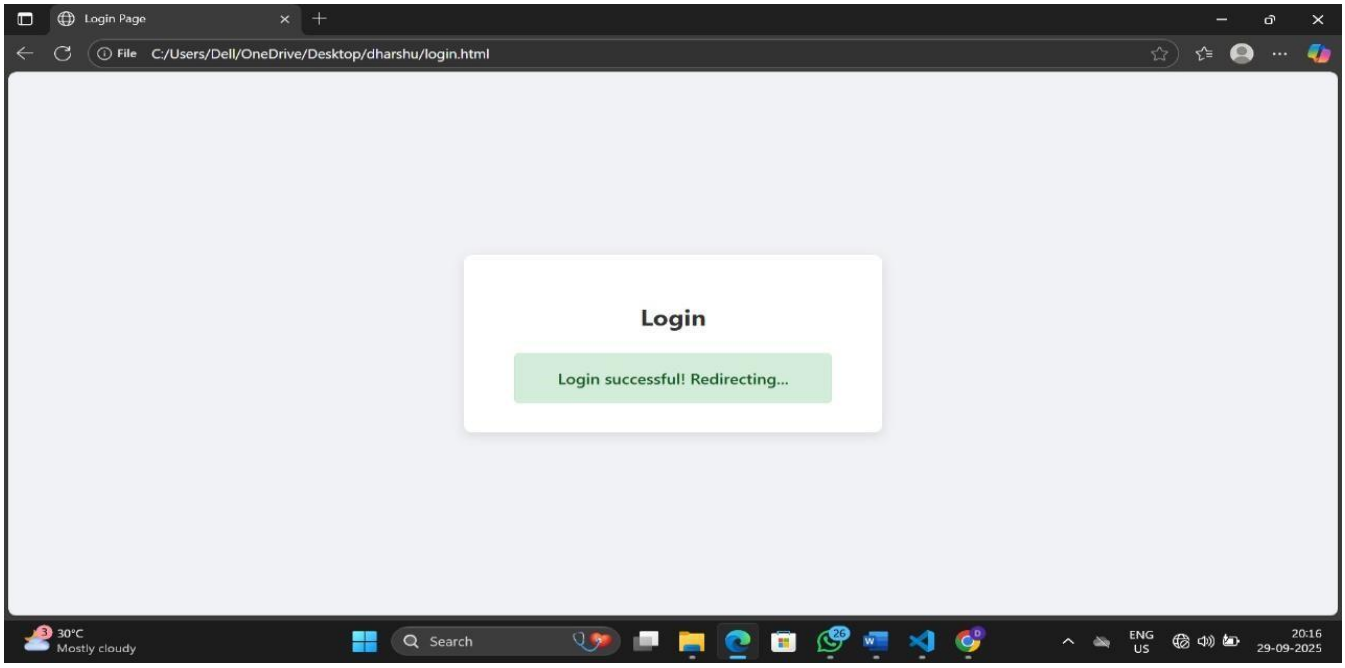
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Welcome</title>
</head>
<body>
  <h1>🎉 Welcome!</h1>
  <p id="userName"></p>
  <button onclick="logout()">Logout</button>
  <script>     const savedUser =
JSON.parse(localStorage.getItem("registeredUser"));     if (savedUser) {
document.getElementById("userName").textContent =
       "Hello, " + savedUser.fullName + " 👋";
   } else {       window.location.href = "login.html"; // if no user,
back to login
   }

   function logout() {
localStorage.removeItem("registeredUser"); // clear login
window.location.href = "login.html";
   }
  </script>
</body>
</html>
```

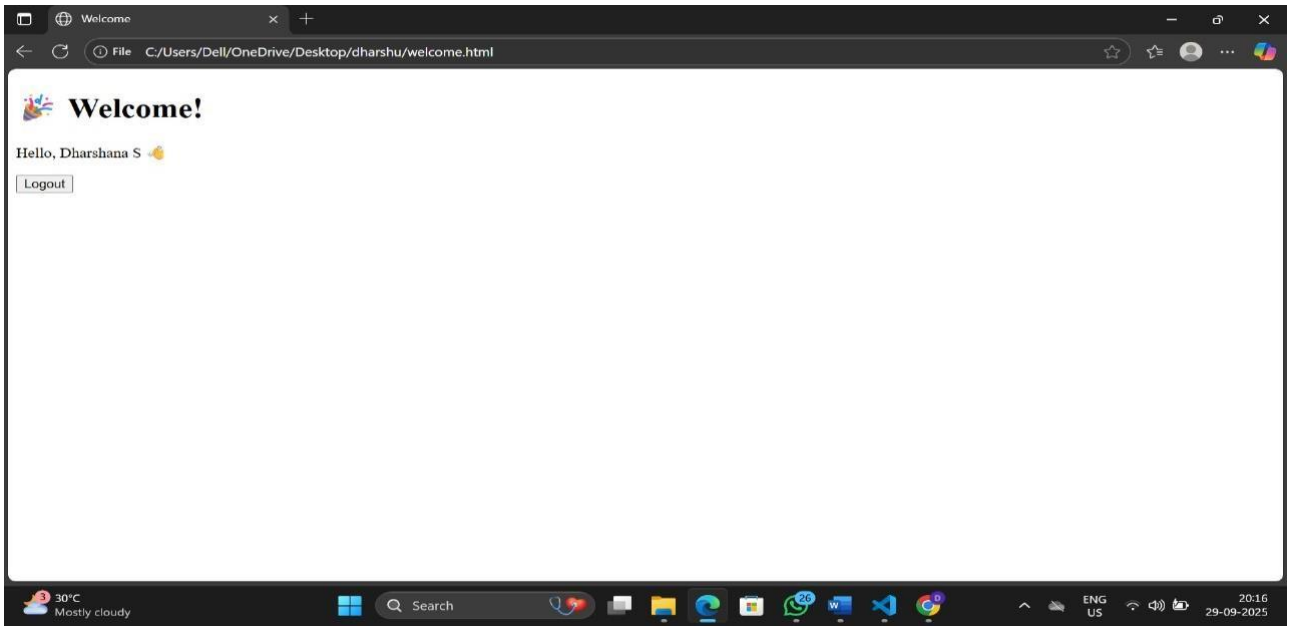
OUTPUT:

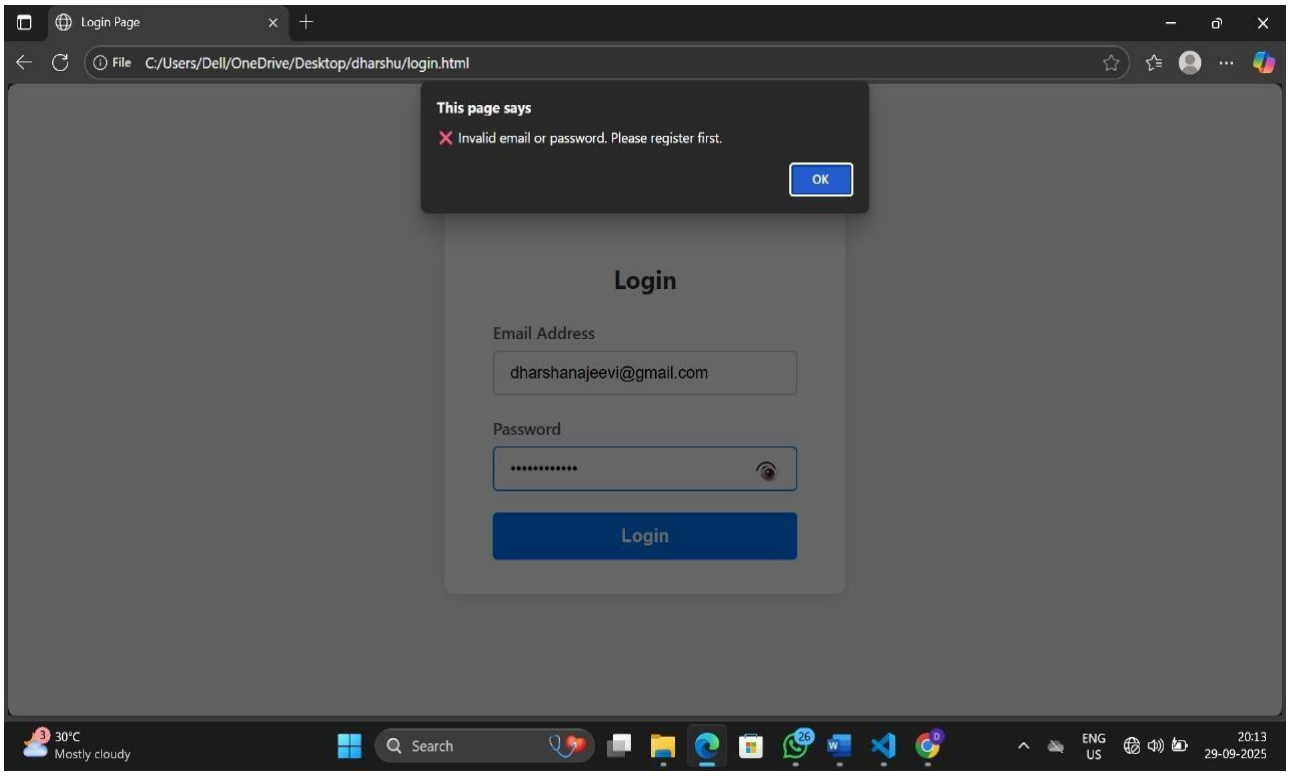

User Registration Form
File C:/Users/Dell/OneDrive/Desktop/dharshu/index.html
Create Your Account
Join us today and get started
Full Name *
Dharshana S
Email Address *
dharshanajeevi@gmail.com
Password *
Strong password
Confirm Password *
Strong password
Confirm Password *
Phone Number
9342645618
Date of Birth *
23-03-2006
Create Account
Already have an account? Login here
30°C
Mostly cloudy
Search
ENG
US
20:15
29-09-2025


User Registration Form
File C:/Users/Dell/OneDrive/Desktop/dharshu/index.html
Create Your Account
Join us today and get started
Registration Successful!
Your account has been created successfully.
Welcome aboard!
Already have an account? Login here
30°C
Mostly cloudy
Search
ENG
US
20:17
29-09-2025


Login Page
File C:/Users/Dell/OneDrive/Desktop/dharshu/login.html
Login
Email Address
dharshanajeevi@gmail.com
Password
Login
30°C
Mostly cloudy
Search
ENG
US
20:23
29-09-2025


Login Page
File C:/Users/Dell/OneDrive/Desktop/dharshu/login.html
Login
Login successful! Redirecting...
30°C
Mostly cloudy
Search
ENG
US
20:16
29-09-2025


Welcome
File C:/Users/Dell/OneDrive/Desktop/dharshu/welcome.html
Welcome!
Hello, Dharshana S
Logout
30°C
Mostly cloudy
Search
ENG
US
20:16
29-09-2025


Login Page
File C:/Users/Dell/OneDrive/Desktop/dharshu/login.html
This page says
Invalid email or password. Please register first.
OK
Login
Email Address
dharshanajeevi@gmail.com
Password
Login
30°C
Mostly cloudy
Search
ENG
US
20:13
29-09-2025