```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Employee Onboarding Form</title>
    <!-- <link rel="stylesheet" href="./On-boarding.css"> -->
     <style>

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: Arial, sans-serif;
}

body {
    background-color: #f5f5f5;
    padding: 20px;
}

.container {
    /* max-width: 1200px; */
    width: 100%;
    margin: 0 auto;
    background-color: white;
    padding: 30px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
}

.A {
    display: flex;
    align-items: center;
    margin-bottom: 30px;
}

.A img {
    height: 60px;
    width: 120px;
```

```
    object-fit: contain;
}

.A h1 {
    text-align: center;
    color: #333;
    margin: 0;
    flex-grow: 1;
    text-shadow: 2px 2px 6px #b3b3b3;
}

.form-section {
    margin-bottom: 30px;
    padding: 20px;
    border: 1px solid #ddd;
    border-radius: 8px;
}

.form-section h2 {
    color: #2c3e50;
    margin-bottom: 20px;
    padding-bottom: 10px;
    border-bottom: 2px solid #eee;
}

.form-section h3 {
    color: #34495e;
    margin: 15px 0;
}

.form-grid {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
    gap: 20px;
}

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

```css
label {
    display: block;
    margin-bottom: 5px;
    color: #555;
    font-weight: bold;
}

input, select, textarea {
    width: 100%;
    padding: 8px 12px;
    border: 1px solid #ddd;
    border-radius: 4px;
    font-size: 14px;
}

input[type="checkbox"] {
    width: auto;
}

.file-upload {
    margin: 10px 0;
}

.file-upload label {
    display: block;
    margin-bottom: 5px;
}

.file-upload input[type="file"] {
    width: 100%;
    padding: 8px;
    border: 1px solid #ddd;
    border-radius: 4px;
    background-color: #fff;
}

.upload-preview {
    margin-top: 5px;
    font-size: 12px;
    color: #666;
```

```
    display: flex;
    align-items: center;
    gap: 10px;
}

.upload-preview img {
    max-width: 100px;
    max-height: 100px;
    object-fit: contain;
}

.file-requirements {
    font-size: 12px;
    color: #666;
    margin-top: 3px;
    font-style: italic;
}

.checkbox-container {
    margin: 15px 0;
    display: flex;
    align-items: center;
    gap: 8px;
}

.education-entry, .employment-entry {
    padding: 15px;
    border: 1px solid #eee;
    border-radius: 4px;
    margin-bottom: 15px;
    background-color: #f8f9fa;
}

.btn-add {
    background-color: #28a745;
    color: white;
    padding: 8px 16px;
    border: none;
    border-radius: 4px;
    cursor: pointer;
```

```
    margin-bottom: 20px;
}

.btn-remove {
    background-color: #dc3545;
    color: white;
    padding: 4px 8px;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    font-size: 12px;
    margin-top: 10px;
}

.submit-btn {
    background-color: #007bff;
    color: white;
    padding: 12px 24px;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    font-size: 16px;
    width: 100%;
    margin-top: 20px;
}

.btn-add:hover, .submit-btn:hover {
    opacity: 0.9;
}

.loading {
    display: none;
    text-align: center;
    margin: 20px 0;
}

.loading::after {
    content: "...";
    animation: dots 1s infinite;
}
```

```css
@keyframes dots {
    0%, 20% { content: "."; }
    40% { content: ".."; }
    60%, 100% { content: "..."; }
}

.success-message {
    background-color: #d4edda;
    color: #155724;
    padding: 15px;
    border-radius: 4px;
    margin-bottom: 20px;
    display: none;
}

.error {
    color: #dc3545;
    font-size: 12px;
    margin-top: 5px;
}

.required::after {
    content: " *";
    color: #dc3545;
}

#permanentAddress {
    display: none;
}

#permanentAddress.show {
    display: block;
}

.invalid-feedback {
    color: #dc3545;
    font-size: 12px;
    margin-top: 5px;
    display: none;
```

```
}

/* Media Queries for Responsiveness */
@media (max-width: 1200px) {
    .container {
        padding: 20px;
    }

    .A {
        flex-direction: column;
        align-items: flex-start;
    }

    .A h1 {
        text-align: left;
        margin-top: 10px;
    }

    .form-grid {
        grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
    }
}

@media (max-width: 768px) {
    .container {
        padding: 15px;
    }

    .A {
        flex-direction: column;
        align-items: flex-start;
        margin-bottom: 20px;
    }

    .A img {
        margin-bottom: 10px;
    }

    .form-grid {
        grid-template-columns: 1fr;
```

```
        gap: 15px;
    }

    .form-group {
        margin-bottom: 10px;
    }

    input, select, textarea {
        font-size: 16px;
    }

    .submit-btn {
        font-size: 14px;
        padding: 10px 20px;
    }

    .btn-add, .btn-remove {
        font-size: 14px;
        padding: 6px 12px;
    }

    .checkbox-container {
        flex-direction: column;
        gap: 10px;
    }

    .loading::after {
        font-size: 16px;
    }
}

@media (max-width: 480px) {
    .container {
        padding: 10px;
    }

    .A img {
        width: 100px;
        height: 50px;
    }
```

```
    .A h1 {
        font-size: 18px;
    }

    .form-grid {
        grid-template-columns: 1fr;
    }

    input, select, textarea {
        font-size: 18px;
        padding: 10px 14px;
    }

    .submit-btn {
        font-size: 14px;
        padding: 10px 20px;
    }

    .btn-add, .btn-remove {
        font-size: 14px;
        padding: 6px 12px;
    }

    .checkbox-container {
        flex-direction: column;
        gap: 10px;
    }
}
     </style>
</head>
<body>
    <div class="container">
        <div class="A">
            <img src="ATS Wallpaper.jpg" alt="Logo">
            <h1>EMPLOYEE ONBOARDING FORM</h1>
        </div>
        <div id="successMessage" class="success-message">
            Form submitted successfully! Your employee ID will be shared
shortly.
```

```
        </div>
        <form id="onboardingForm" novalidate>
            <!-- Personal Information -->
            <div class="form-section">
                <h2>Personal Information</h2>
                <div class="form-grid">
                    <div class="form-group">
                        <label class="required">First Name</label>
                        <input type="text" name="firstName" required
pattern="^[A-Za-z\s]{5,}$">
                        <div class="invalid-feedback">Please enter at
least 5 alphabetical characters</div>
                    </div>
                    <div class="form-group">
                        <label class="required">Last Name</label>
                        <input type="text" name="lastName" required
pattern="^[A-Za-z\s]{5,}$">
                        <div class="invalid-feedback">Please enter at
least 5 alphabetical characters</div>
                    </div>
                    <div class="form-group">
                        <label class="required">Date of Birth</label>
                        <input type="date" id="dob" name="dob" required>
                    </div>
                    <div class="form-group">
                        <label class="required">Gender</label>
                        <select name="gender" required>
                            <option value="">Select Gender</option>
                            <option value="Male">Male</option>
                            <option value="Female">Female</option>
                            <option value="Other">Other</option>
                            <option value="Prefer not to say">Prefer not
to say</option>
                        </select>
                    </div>
                    <div class="form-group">
                        <label>Blood Group</label>
                        <select name="bloodGroup">
                            <option value="">Select Blood Group</option>
                            <option value="A+">A+</option>
```

```
                            <option value="A-">A-</option>
                            <option value="B+">B+</option>
                            <option value="B-">B-</option>
                            <option value="O+">O+</option>
                            <option value="O-">O-</option>
                            <option value="AB+">AB+</option>
                            <option value="AB-">AB-</option>
                        </select>
                    </div>
                    <div class="form-group">
                        <label>Marital Status</label>
                        <select name="maritalStatus">
                            <option value="">Select Status</option>
                            <option value="Single">Single</option>
                            <option value="Married">Married</option>
                            <option value="Divorced">Divorced</option>
                            <option value="Widowed">Widowed</option>
                        </select>
                    </div>
                    <div class="form-group">
                        <label class="required">Guardian Name</label>
                        <input type="text" name="guardianName" required
pattern="^[A-Za-z\s]{5,}$">
                        <div class="invalid-feedback">Please enter at
least 5 alphabetical characters</div>
                    </div>
                    <div class="form-group">
                        <label class="required">Guardian Phone</label>
                        <input type="tel" name="guardianPhone" required
pattern="[0-9]{10}" title="Please enter a valid 10-digit phone number">
                    </div>
                    <div class="form-group">
                        <label class="required">Guardian Relation</label>
                        <input type="text" name="guardianRelation"
required pattern="^[A-Za-z\s]{5,}$">
                        <div class="invalid-feedback">Please enter at
least 5 alphabetical characters</div>
                    </div>
                </div>
            </div>
```

```
            <!-- ID Proof Section -->
            <div class="form-section">
                <h2>ID Proof Details</h2>
                <div class="form-grid">
                    <div class="form-group">
                        <label class="required">Aadhaar Number</label>
                        <input type="text" name="aadhaarNumber" required
pattern="[0-9]{12}">
                        <div class="file-upload">
                            <label class="required">Upload Aadhaar
Card</label>
                            <input type="file" name="aadhaarFile" required
accept=".pdf,.jpg,.jpeg,.png">
                            <div id="aadhaarPreview"
class="upload-preview"></div>
                            <div class="file-requirements">Accepted
formats: PDF, JPG, PNG (Max size: 2MB)</div>
                        </div>
                    </div>
                    <div class="form-group">
                        <label class="required">PAN Number</label>
                        <input type="text" name="panNumber" required
pattern="[A-Z]{5}[0-9]{4}[A-Z]{1}">
                        <div class="file-upload">
                            <label class="required">Upload PAN
Card</label>
                            <input type="file" name="panFile" required
accept=".pdf,.jpg,.jpeg,.png">
                            <div id="panPreview"
class="upload-preview"></div>
                            <div class="file-requirements">Accepted
formats: PDF, JPG, PNG (Max size: 2MB)</div>
                        </div>
                    </div>
                </div>
            </div>

            <!-- Contact Information -->
            <div class="form-section">
```

```
                <h2>Contact Information</h2>
                <div class="form-grid">
                    <div class="form-group">
                        <label class="required">Email</label>
                        <input type="email" name="email" required>
                    </div>
                    <div class="form-group">
                        <label class="required">Phone</label>
                        <input type="tel" name="phone" required
pattern="[0-9]{10}" title="Please enter a valid 10-digit phone number">
                    </div>
                    <div class="form-group">
                        <label class="required">Emergency Contact
Name</label>
                        <input type="text" name="emergencyContact"
required pattern="^[A-Za-z\s]{5,}$">
                        <div class="invalid-feedback">Please enter at
least 5 alphabetical characters</div>
                    </div>
                    <div class="form-group">
                        <label class="required">Emergency Contact
Relation</label>
                        <input type="text" name="emergencyRelation"
required pattern="^[A-Za-z\s]{5,}$">
                        <div class="invalid-feedback">Please enter at
least 5 alphabetical characters</div>
                    </div>
                    <div class="form-group">
                        <label class="required">Emergency Contact
Phone</label>
                        <input type="tel" name="emergencyPhone" required
pattern="[0-9]{10}" title="Please enter a valid 10-digit phone number">
                    </div>
                </div>
            </div>

            <!-- Address Information -->
            <div class="form-section">
                <h2>Address Information</h2>
```

```
                <!-- Current Address -->
                <h3>Current Address</h3>
                <div class="form-grid">
                    <div class="form-group">
                        <label class="required">Street Address</label>
                        <textarea name="currentAddress" required
pattern="^[A-Za-z0-9\s,.-]{5,}$" rows="2"></textarea>
                        <div class="invalid-feedback">Please enter at
least 5 characters</div>
                    </div>
                    <div class="form-group">
                        <label class="required">City</label>
                        <input type="text" name="currentCity" required
pattern="^[A-Za-z\s]{5,}$">
                        <div class="invalid-feedback">Please enter at
least 5 alphabetical characters</div>
                    </div>
                    <div class="form-group">
                        <label class="required">State</label>
                        <input type="text" name="currentState" required
pattern="^[A-Za-z\s]{5,}$">
                        <div class="invalid-feedback">Please enter at
least 5 alphabetical characters</div>
                    </div>
                    <div class="form-group">
                        <label class="required">Pincode</label>
                        <input type="text" name="currentPincode" required
pattern="[0-9]{6}" title="Please enter a valid 6-digit pincode">
                    </div>
                </div>

                <div class="checkbox-container">
                    <input type="checkbox" id="sameAsCurrentAddress"
name="sameAsCurrentAddress">
                    <label for="sameAsCurrentAddress">Permanent Address
same as Current Address</label>
                </div>

                <!-- Permanent Address -->
                <div id="permanentAddress">
```

```
                    <h3>Permanent Address</h3>
                    <div class="form-group">
                        <label class="required">Street Address</label>
                        <textarea name="permanentAddress"
pattern="^[A-Za-z0-9\s,.-]{5,}$" rows="2"></textarea>
                        <div class="invalid-feedback">Please enter at
least 5 characters</div>
                    </div>
                    <div class="form-group">
                        <label class="required">City</label>
                        <input type="text" name="permanentCity"
pattern="^[A-Za-z\s]{5,}$">
                        <div class="invalid-feedback">Please enter at
least 5 alphabetical characters</div>
                    </div>
                    <div class="form-group">
                        <label class="required">State</label>
                        <input type="text" name="permanentState"
pattern="^[A-Za-z\s]{5,}$">
                        <div class="invalid-feedback">Please enter at
least 5 alphabetical characters</div>
                    </div>
                    <div class="form-group">
                        <label class="required">Pincode</label>
                        <input type="text" name="permanentPincode"
pattern="[0-9]{6}" title="Please enter a valid 6-digit pincode">
                    </div>
                </div>
            </div>
        </div>

        <!-- Educational Details -->
        <div class="form-section" style="background-color: white;">
            <h2>Educational Details</h2>
            <button type="button" class="btn-add"
onclick="addEducation()">Add Education</button>
            <div id="educationContainer"></div>
        </div>

        <!-- Employment History -->
```

```
        <div class="form-section" style="background-color: white;">
            <h2>Employment History</h2>
            <button type="button" class="btn-add"
onclick="addEmployment()">Add Employment</button>
            <div id="employmentContainer"></div>
        </div>

        <!-- Bank Details -->
        <div class="form-section" style="background-color: white;">
            <h2>Bank Details</h2>
            <div class="form-grid">
                <div class="form-group">
                    <label class="required">Account Holder Name</label>
                    <input type="text" name="accountHolderName" required
pattern="^[A-Za-z\s]{5,}$">
                    <div class="invalid-feedback">Please enter at least 5
alphabetical characters</div>
                </div>
                <div class="form-group">
                    <label class="required">Bank Name</label>
                    <input type="text" name="bankName" required
pattern="^[A-Za-z\s]{5,}$">
                    <div class="invalid-feedback">Please enter at least 5
alphabetical characters</div>
                </div>
                <div class="form-group">
                    <label class="required">Account Number</label>
                    <input type="text" name="accountNumber" required
pattern="[0-9]{9,18}" title="Please enter a valid account number">
                </div>
                <div class="form-group">
                    <label class="required">IFSC Code</label>
                    <input type="text" name="ifscCode" required
pattern="^[A-Za-z]{4}0[A-Z0-9]{6}$" title="Please enter a valid IFSC
code">
                </div>
            </div>
        </div>

        <div id="loading" class="loading">Submitting form</div>
```

```
        <button type="submit" class="submit-btn">Submit</button>
    </form>
</div>
<script>
    // Function to validate text fields
function validateTextField(value) {
    return /^[A-Za-z\s]{5,}$/.test(value);
}

// Function to validate address fields (allows numbers, spaces, commas,
periods, and hyphens)
function validateAddressField(value) {
    return /^[A-Za-z0-9\s,.-]{5,}$/.test(value);
}

// Function to validate if at least one education entry exists and is
valid
function validateEducation() {
    const educationEntries =
document.querySelectorAll('.education-entry');
    if (educationEntries.length === 0) {
        return false;
    }

    // Validate each education entry
    let isValid = true;
    educationEntries.forEach(entry => {
        const requiredFields = entry.querySelectorAll('[required]');
        requiredFields.forEach(field => {
            if (!validateField(field)) {
                isValid = false;
            }
        });
    });
    return isValid;
}

// Function to validate if at least one employment entry exists and is
valid
function validateEmployment() {
```

```
    const employmentEntries =
document.querySelectorAll('.employment-entry');
    if (employmentEntries.length === 0) {
        return false;
    }

    // Validate each employment entry
    let isValid = true;
    employmentEntries.forEach(entry => {
        const requiredFields = entry.querySelectorAll('[required]');
        requiredFields.forEach(field => {
            if (!validateField(field)) {
                isValid = false;
            }
        });
    });
    return isValid;
}

function addEducation() {
    const container = document.getElementById('educationContainer');
    const educationEntry = document.createElement('div');
    educationEntry.className = 'education-entry';
    educationEntry.innerHTML = `
        <div class="form-grid">
            <div class="form-group">
                <label class="required">Education Level</label>
                <select name="education[][level]" required>
                    <option value="">Select Level</option>
                    <option value="High School">High School</option>
                    <option value="Bachelor's">Bachelor's</option>
                    <option value="Master's">Master's</option>
                    <option value="Doctorate">Doctorate</option>
                </select>
            </div>
            <div class="form-group">
                <label class="required">Stream/Specialization</label>
                <input type="text" name="education[][stream]" required>
                <div class="error-message" id="percentageError"></div>
            </div>
```

```
            <div class="form-group">
                <label class="required">Institution</label>
                <input type="text" name="education[][institution]"
required required>
                <div class="error-message" id="percentageError"></div>
            </div>
            <div class="form-group">
                <label class="required">Year of Completion</label>
                <input type="number" name="education[][year]" required
min="1900" max="2099">
            </div>
            <div class="form-group">
                <label class="required">Percentage/CGPA</label>
                <input type="number" step="0.01"
name="education[][percentage]" required min="0" max="100">
            </div>
            <div class="file-upload">
                <label class="required">Upload Document</label>
                <input type="file" name="panFile" required
accept=".pdf,.jpg,.jpeg,.png">
                <div id="Edu" class="uploadEdu"></div>
                <div class="file-requirements">Accepted formats: PDF, JPG,
PNG (Max size: 2MB)</div>
            </div>
        </div>
        <button type="button" class="btn-remove"
onclick="this.parentElement.remove()">Remove</button>
    `;
    container.appendChild(educationEntry);

    // Add validation listeners to new fields
    const newFields = educationEntry.querySelectorAll('input[pattern]');
    newFields.forEach(field => {
        field.addEventListener('input', function() {
            validateField(this);
        });
    });
}

// Set max date for date of birth (20 years ago)
```

```
const currentDate = new Date();
const twentyYearsAgo = new
Date(currentDate.setFullYear(currentDate.getFullYear() - 20));
const formattedDate = twentyYearsAgo.toISOString().split('T')[0];
document.getElementById('dob').setAttribute('max', formattedDate);

function addEmployment() {
    const container = document.getElementById('employmentContainer');
    const employmentEntry = document.createElement('div');
    employmentEntry.className = 'employment-entry';
    employmentEntry.innerHTML = `
        <div class="form-grid">
            <div class="form-group">
                <label class="required">Company Name</label>
                <input type="text" name="employment[][company]" required>
                <div class="error-message" id="percentageError"></div>
            </div>
            <div class="form-group">
                <label class="required">Designation</label>
                <input type="text" name="employment[][designation]"
required>
                <div class="error-message" id="percentageError"></div>
            </div>
            <div class="form-group">
                <label class="required">Employment Type</label>
                <select name="employment[][type]" required>
                    <option value="">Select Type</option>
                    <option value="Full-time">Full-time</option>
                    <option value="Part-time">Part-time</option>
                    <option value="Contract">Contract</option>
                    <option value="Internship">Internship</option>
                </select>
            </div>
            <div class="form-group">
                <label class="required">Start Date</label>
                <input type="date" name="employment[][startDate]"
required>
            </div>
            <div class="form-group">
                <label>End Date</label>
```

```
                <input type="date" name="employment[][endDate]">
            </div>
            <div class="file-upload">
                <label class="required">Upload Document</label>
                <input type="file" name="panFile" required
accept=".pdf,.jpg,.jpeg,.png">
                <div id="Emp" class="upload-Emp"></div>
                <div class="file-requirements">Accepted formats: PDF, JPG,
PNG (Max size: 2MB)</div>
            </div>
            <div class="form-group"><br><br>
                <label>
                    <input type="checkbox"
name="employment[][currentEmployer]">
                    Current Employer
                </label>
            </div>
        </div>
        <button type="button" class="btn-remove"
onclick="this.parentElement.remove()">Remove</button>
    `;
    container.appendChild(employmentEntry);

    // Add validation listeners to new fields
    const newFields = employmentEntry.querySelectorAll('input[pattern]');
    newFields.forEach(field => {
        field.addEventListener('input', function() {
            validateField(this);
        });
    });
}

// Function to validate fields
function validateField(field) {
    const isValid = field.checkValidity();
    const feedbackElement = field.nextElementSibling;
    if (feedbackElement &&
feedbackElement.classList.contains('invalid-feedback')) {
        feedbackElement.style.display = isValid ? 'none' : 'block';
    }
```

```
    return isValid;
}

// Handle same as current address checkbox
document.getElementById('sameAsCurrentAddress').addEventListener('change',
function() {
    const permanentAddressDiv =
document.getElementById('permanentAddress');
    const permanentFields = permanentAddressDiv.querySelectorAll('input,
textarea');

    if (this.checked) {
        permanentAddressDiv.style.display = 'none';
        permanentFields.forEach(field => {
            field.removeAttribute('required');
        });
    } else {
        permanentAddressDiv.style.display = 'block';
        permanentFields.forEach(field => {
            if (field.name !== 'sameAsCurrentAddress') {
                field.setAttribute('required', 'required');
            }
        });
    }
});

// Add validation listeners when DOM is loaded
document.addEventListener('DOMContentLoaded', function() {
    // Add validation to all fields with pattern attribute
    const patternFields = document.querySelectorAll('input[pattern],
textarea[pattern]');
    patternFields.forEach(field => {
        field.addEventListener('input', function() {
            validateField(this);
        });
    });

    // Form submission handler
    document.getElementById('onboardingForm').addEventListener('submit',
function(e) {
```

```
        e.preventDefault();

        // Validate all required fields
        const requiredFields = this.querySelectorAll('[required]');
        let isValid = true;

        requiredFields.forEach(field => {
            if (!validateField(field)) {
                isValid = false;
            }
        });

        // Validate education section
        if (!validateEducation()) {
            alert('Please add and fill at least one education entry');
            return;
        }

        // Validate employment section
        if (!validateEmployment()) {
            alert('Please add and fill at least one employment entry');
            return;
        }

        if (!isValid) {
            alert('Please fill all required fields correctly');
            return;
        }

        document.getElementById('loading').style.display = 'block';
        document.querySelector('.submit-btn').disabled = true;

        // Process form submission
        const formData = new FormData(this);
        try {
            const employeeData = {
                employeeId: 'EMP' + Date.now(),
                submissionDate: new Date().toISOString(),
                status: 'Pending'
            };
```

```
            // Convert FormData to object
            for (let [key, value] of formData.entries()) {
                if (!key.includes('[]')) {
                    employeeData[key] = value;
                }
            }

            // Save to localStorage
            const existingEmployees =
JSON.parse(localStorage.getItem('hrEmployees')) || [];
            existingEmployees.push(employeeData);
            localStorage.setItem('hrEmployees',
JSON.stringify(existingEmployees));

            // Show success message
            document.getElementById('successMessage').style.display =
'block';

            // Reset form
            this.reset();
            document.getElementById('educationContainer').innerHTML = '';
            document.getElementById('employmentContainer').innerHTML = '';

            // Scroll to top
            window.scrollTo({ top: 0, behavior: 'smooth' });

            // Hide success message after 5 seconds
            setTimeout(() => {
                document.getElementById('successMessage').style.display =
'none';
            }, 5000);
        } catch (error) {
            console.error('Error saving employee data:', error);
            alert('Error saving employee data. Please try again.');
        } finally {
            document.getElementById('loading').style.display = 'none';
            document.querySelector('.submit-btn').disabled = false;
        }
    });
```

```
});
</script>
</body>
</html>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>HR Dashboard - Employee Onboarding</title>
    <!-- <link rel="stylesheet" href="./On-boarding.css"> -->
     <style>

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Inter', -apple-system, BlinkMacSystemFont, sans-serif;
}

:root {
    --primary-color: #2563eb;
    --secondary-color: #64748b;
    --success-color: #22c55e;
    --danger-color: #ef4444;
    --background-color: #f8fafc;
    --card-background: #ffffff;
    --border-color: #e2e8f0;
    --text-primary: #1e293b;
    --text-secondary: #64748b;
    --shadow-sm: 0 1px 2px 0 rgba(0, 0, 0, 0.05);
    --shadow-md: 0 4px 6px -1px rgba(0, 0, 0, 0.1);
    --shadow-lg: 0 10px 15px -3px rgba(0, 0, 0, 0.1);
}

body {
    background-color: var(--background-color);
    color: var(--text-primary);
    line-height: 1.5;
}
```

```css
/* Container and Header */
.container {
    max-width: 1280px;
    margin: 0 auto;
    padding: 1.5rem;
}

.header {
    display: flex;
    flex-direction: column;
    gap: 1.5rem;
    margin-bottom: 2rem;
}

@media (min-width: 768px) {
    .header {
        flex-direction: row;
        justify-content: space-between;
        align-items: center;
    }
}

.header h1 {
    font-size: 1.875rem;
    font-weight: 700;
    color: var(--text-primary);
    animation: slideDown 0.5s ease-out;
}

/* Search and Filter Section */
.search-bar {
    display: flex;
    gap: 1rem;
    flex-wrap: wrap;
}

.search-bar input,
.filter-section select {
    padding: 0.75rem 1rem;
```

```
    border: 1px solid var(--border-color);
    border-radius: 0.5rem;
    background-color: var(--card-background);
    transition: all 0.2s ease;
    font-size: 0.875rem;
}

.search-bar input {
    flex: 1;
    min-width: 250px;
}

.search-bar input:focus,
.filter-section select:focus {
    outline: none;
    border-color: var(--primary-color);
    box-shadow: 0 0 0 3px rgba(37, 99, 235, 0.1);
}

.filter-section {
    display: flex;
    gap: 1rem;
    margin-bottom: 2rem;
    flex-wrap: wrap;
}

.filter-section select {
    min-width: 150px;
}

/* Employee Grid */
.employees-grid {
    display: grid;
    gap: 1.5rem;
    grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
    margin-bottom: 2rem;
}

.employee-card {
    background-color: var(--card-background);
```

```
    border-radius: 0.75rem;
    padding: 1.5rem;
    box-shadow: var(--shadow-sm);
    transition: all 0.3s ease;
    cursor: pointer;
    animation: fadeIn 0.5s ease-out;
}

.employee-card:hover {
    transform: translateY(-2px);
    box-shadow: var(--shadow-md);
}

.employee-card h3 {
    font-size: 1.125rem;
    margin-bottom: 1rem;
    color: var(--text-primary);
}

.employee-info {
    display: flex;
    flex-direction: column;
    gap: 0.5rem;
}

.employee-info p {
    color: var(--text-secondary);
    font-size: 0.875rem;
}

/* Status Badges */
.status {
    display: inline-block;
    padding: 0.25rem 0.75rem;
    border-radius: 9999px;
    font-size: 0.75rem;
    font-weight: 500;
    text-transform: uppercase;
}
```

```css
.status-pending {
    background-color: #fef3c7;
    color: #92400e;
}

.status-approved {
    background-color: #dcfce7;
    color: #166534;
}

.status-rejected {
    background-color: #fee2e2;
    color: #991b1b;
}

/* Modal Styles */
.modal {
    display: none;
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background-color: rgba(0, 0, 0, 0.5);
    z-index: 1000;
    animation: fadeIn 0.3s ease-out;
}

.modal-content {
    position: relative;
    background-color: var(--card-background);
    margin: 2rem auto;
    padding: 2rem;
    border-radius: 1rem;
    max-width: 800px;
    max-height: 90vh;
    overflow-y: auto;
    animation: slideUp 0.4s ease-out;
}
```

```css
.close-btn {
    position: absolute;
    top: 1rem;
    right: 1.5rem;
    font-size: 1.5rem;
    cursor: pointer;
    color: var(--text-secondary);
    transition: color 0.2s ease;
}

.close-btn:hover {
    color: var(--text-primary);
}

/* Employee Details Styling */
.header-section {
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: 2rem;
    padding-bottom: 1rem;
    border-bottom: 1px solid var(--border-color);
}

.employee-header {
    display: flex;
    align-items: center;
    gap: 1rem;
}

.employee-avatar {
    width: 3.5rem;
    height: 3.5rem;
    background-color: var(--primary-color);
    color: white;
    border-radius: 50%;
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 1.25rem;
```

```
    font-weight: 600;
}

.details-section {
    margin-bottom: 2rem;
    animation: fadeIn 0.5s ease-out;
}

.details-section h3 {
    color: var(--text-primary);
    margin-bottom: 1rem;
    font-size: 1.125rem;
}

.details-grid {
    display: grid;
    gap: 1.5rem;
    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
}

.detail-item label {
    display: block;
    color: var(--text-secondary);
    font-size: 0.875rem;
    margin-bottom: 0.25rem;
}

.detail-item div {
    color: var(--text-primary);
    font-weight: 500;
}

/* Document Preview */
.document-preview {
    display: flex;
    align-items: center;
    gap: 0.5rem;
    margin-top: 0.5rem;
    padding: 0.5rem;
    background-color: var(--background-color);
```

```
    border-radius: 0.5rem;
    font-size: 0.875rem;
}

/* Action Buttons */
.action-buttons {
    display: flex;
    gap: 1rem;
    margin-top: 2rem;
}

.btn {
    padding: 0.75rem 1.5rem;
    border: none;
    border-radius: 0.5rem;
    font-weight: 500;
    cursor: pointer;
    transition: all 0.2s ease;
}

.btn-approve {
    background-color: var(--success-color);
    color: white;
}

.btn-reject {
    background-color: var(--danger-color);
    color: white;
}

.btn:hover {
    opacity: 0.9;
    transform: translateY(-1px);
}

/* Pagination */
.pagination {
    display: flex;
    justify-content: center;
    gap: 0.5rem;
```

```
    margin-top: 2rem;
}

.pagination button {
    padding: 0.5rem 1rem;
    border: 1px solid var(--border-color);
    background-color: var(--card-background);
    border-radius: 0.5rem;
    cursor: pointer;
    transition: all 0.2s ease;
}

.pagination button.active {
    background-color: var(--primary-color);
    color: white;
    border-color: var(--primary-color);
}

.pagination button:hover:not(.active) {
    background-color: var(--background-color);
}

/* Animations */
@keyframes fadeIn {
    from {
        opacity: 0;
    }
    to {
        opacity: 1;
    }
}

@keyframes slideDown {
    from {
        transform: translateY(-20px);
        opacity: 0;
    }
    to {
        transform: translateY(0);
        opacity: 1;
```

```
    }
}

@keyframes slideUp {
    from {
        transform: translateY(20px);
        opacity: 0;
    }
    to {
        transform: translateY(0);
        opacity: 1;
    }
}

/* Loading State */
.loading {
    text-align: center;
    padding: 2rem;
    color: var(--text-secondary);
    animation: pulse 1.5s infinite;
}

@keyframes pulse {
    0% {
        opacity: 1;
    }
    50% {
        opacity: 0.5;
    }
    100% {
        opacity: 1;
    }
}

/* Responsive Adjustments */
@media (max-width: 640px) {
    .container {
        padding: 1rem;
    }
```

```
    .modal-content {
        margin: 1rem;
        padding: 1rem;
    }

    .header-section {
        flex-direction: column;
        gap: 1rem;
        align-items: flex-start;
    }

    .action-buttons {
        flex-direction: column;
    }

    .btn {
        width: 100%;
    }
}
     </style>
</head>
<body>
    <div class="container">
        <div class="header">
            <h1>HR Dashboard - Employee Onboarding</h1>
            <div class="search-bar">
                <input type="text" id="searchInput" placeholder="Search
employees...">
            </div>
        </div>

        <div class="filter-section">
            <select id="statusFilter">
                <option value="">All Status</option>
                <option value="Pending">Pending</option>
                <option value="Approved">Approved</option>
                <option value="Rejected">Rejected</option>
            </select>
            <select id="sortBy">
                <option value="newest">Newest First</option>
```

```
                <option value="oldest">Oldest First</option>
                <option value="name">Name A-Z</option>
            </select>
        </div>

        <div id="employeesGrid" class="employees-grid">
            <!-- Employee cards will be dynamically added here -->
        </div>

        <div class="pagination" id="pagination">
            <!-- Pagination buttons will be dynamically added here -->
        </div>
    </div>

    <!-- Modal for detailed view -->
    <div id="employeeModal" class="modal">
        <div class="modal-content">
            <span class="close-btn" onclick="closeModal()">&times;</span>
            <div id="employeeDetails">
                <!-- Detailed employee information will be added here -->
            </div>
        </div>
    </div>
    <script>

let employees = [];
const ITEMS_PER_PAGE = 6;
let currentPage = 1;

// Load employees from localStorage
function loadEmployees() {
    const storedEmployees = localStorage.getItem('hrEmployees');
    employees = storedEmployees ? JSON.parse(storedEmployees) : [];
    displayEmployees();
}

// Filter and sort employees
function getFilteredEmployees() {
    const searchTerm =
document.getElementById('searchInput').value.toLowerCase();
```

```
    const statusFilter = document.getElementById('statusFilter').value;
    const sortBy = document.getElementById('sortBy').value;

    let filtered = employees.filter(emp => {
        const matchesSearch = (emp.firstName + ' ' +
emp.lastName).toLowerCase().includes(searchTerm);
        const matchesStatus = !statusFilter || emp.status ===
statusFilter;
        return matchesSearch && matchesStatus;
    });

    // Sort employees
    filtered.sort((a, b) => {
        switch(sortBy) {
            case 'newest':
                return new Date(b.submissionDate) - new
Date(a.submissionDate);
            case 'oldest':
                return new Date(a.submissionDate) - new
Date(b.submissionDate);
            case 'name':
                return (a.firstName + ' ' +
a.lastName).localeCompare(b.firstName + ' ' + b.lastName);
            default:
                return 0;
        }
    });

    return filtered;
}

// Display employees
function displayEmployees() {
    const filteredEmployees = getFilteredEmployees();
    const grid = document.getElementById('employeesGrid');
    grid.innerHTML = '';

    const startIndex = (currentPage - 1) * ITEMS_PER_PAGE;
    const endIndex = startIndex + ITEMS_PER_PAGE;
```

```
    const paginatedEmployees = filteredEmployees.slice(startIndex,
endIndex);

    paginatedEmployees.forEach(emp => {
        const card = document.createElement('div');
        card.className = 'employee-card';
        card.onclick = () => showEmployeeDetails(emp);

        const submissionDate = new
Date(emp.submissionDate).toLocaleDateString();

        card.innerHTML = `
            <h3>${emp.firstName} ${emp.lastName}</h3>
            <div class="employee-info">
                <p>Employee ID: ${emp.employeeId}</p>
                <p>Submission Date: ${submissionDate}</p>
                <p>Email: ${emp.email}</p>
                <span class="status
status-${emp.status.toLowerCase()}">${emp.status}</span>
            </div>
        `;

        grid.appendChild(card);
    });

    updatePagination(filteredEmployees.length);
}

// Update pagination
function updatePagination(totalItems) {
    const totalPages = Math.ceil(totalItems / ITEMS_PER_PAGE);
    const pagination = document.getElementById('pagination');
    pagination.innerHTML = '';

    for (let i = 1; i <= totalPages; i++) {
        const button = document.createElement('button');
        button.innerText = i;
        button.className = i === currentPage ? 'active' : '';
        button.onclick = () => {
            currentPage = i;
```

```
            displayEmployees();
        };
        pagination.appendChild(button);
    }
}

// Show employee details in modal
function showEmployeeDetails(employee) {
const modal = document.getElementById('employeeModal');
const details = document.getElementById('employeeDetails');

const submissionDate = new
Date(employee.submissionDate).toLocaleDateString();
const dob = new Date(employee.dob).toLocaleDateString();

details.innerHTML = `
<div class="header-section">
    <div class="employee-header">
        <div
class="employee-avatar">${employee.firstName[0]}${employee.lastName[0]}</d
iv>
        <div>
            <h2>${employee.firstName} ${employee.lastName}</h2>
            <p style="color: #6c757d;">Employee ID:
${employee.employeeId}</p>
        </div>
    </div>
    <span class="status-badge
status-${employee.status.toLowerCase()}">${employee.status}</span>
</div>

<div class="details-section">
    <h3>Personal Information</h3>
    <div class="details-grid">
        <div class="detail-item">
            <label>Date of Birth</label>
            <div>${dob}</div>
        </div>
        <div class="detail-item">
            <label>Gender</label>
```

```
            <div>${employee.gender}</div>
        </div>
        <div class="detail-item">
            <label>Blood Group</label>
            <div>${employee.bloodGroup || 'Not specified'}</div>
        </div>
        <div class="detail-item">
            <label>Marital Status</label>
            <div>${employee.maritalStatus || 'Not specified'}</div>
        </div>
    </div>
</div>

<div class="details-section">
    <h3>Guardian Information</h3>
    <div class="details-grid">
        <div class="detail-item">
            <label>Guardian Name</label>
            <div>${employee.guardianName}</div>
        </div>
        <div class="detail-item">
            <label>Guardian Phone</label>
            <div>${employee.guardianPhone}</div>
        </div>
        <div class="detail-item">
            <label>Guardian Relation</label>
            <div>${employee.guardianRelation}</div>
        </div>
    </div>
</div>

<div class="details-section">
    <h3>ID Proof Details</h3>
    <div class="details-grid">
        <div class="detail-item">
            <label>Aadhaar Number</label>
            <div>${employee.aadhaarNumber}</div>
            <div class="document-preview">
                <div class="document-icon">📄</div>
                <span>Aadhaar Card Document</span>
```

```
            </div>
        </div>
        <div class="detail-item">
            <label>PAN Number</label>
            <div>${employee.panNumber}</div>
            <div class="document-preview">
                <div class="document-icon">📄</div>
                <span>PAN Card Document</span>
            </div>
        </div>
    </div>
</div>

<div class="details-section">
    <h3>Contact Information</h3>
    <div class="details-grid">
        <div class="detail-item">
            <label>Email</label>
            <div>${employee.email}</div>
        </div>
        <div class="detail-item">
            <label>Phone</label>
            <div>${employee.phone}</div>
        </div>
        <div class="detail-item">
            <label>Emergency Contact</label>
            <div>${employee.emergencyContact}</div>
        </div>
        <div class="detail-item">
            <label>Emergency Contact Phone</label>
            <div>${employee.emergencyPhone}</div>
        </div>
        <div class="detail-item">
            <label>Emergency Contact Relation</label>
            <div>${employee.emergencyRelation}</div>
        </div>
    </div>
</div>

<div class="details-section">
```

```
    <h3>Current Address</h3>
    <div class="details-grid">
        <div class="detail-item">
            <label>Street Address</label>
            <div>${employee.currentAddress}</div>
        </div>
        <div class="detail-item">
            <label>City</label>
            <div>${employee.currentCity}</div>
        </div>
        <div class="detail-item">
            <label>State</label>
            <div>${employee.currentState}</div>
        </div>
        <div class="detail-item">
            <label>Pincode</label>
            <div>${employee.currentPincode}</div>
        </div>
    </div>
</div>

${employee.permanentAddress ? `
<div class="details-section">
    <h3>Permanent Address</h3>
    <div class="details-grid">
        <div class="detail-item">
            <label>Street Address</label>
            <div>${employee.permanentAddress}</div>
        </div>
        <div class="detail-item">
            <label>City</label>
            <div>${employee.permanentCity}</div>
        </div>
        <div class="detail-item">
            <label>State</label>
            <div>${employee.permanentState}</div>
        </div>
        <div class="detail-item">
            <label>Pincode</label>
            <div>${employee.permanentPincode}</div>
```

```
        </div>
    </div>
</div>
` : ''}

<div class="details-section">
    <h3>Bank Details</h3>
    <div class="details-grid">
        <div class="detail-item">
            <label>Account Holder Name</label>
            <div>${employee.accountHolderName}</div>
        </div>
        <div class="detail-item">
            <label>Bank Name</label>
            <div>${employee.bankName}</div>
        </div>
        <div class="detail-item">
            <label>Account Number</label>
            <div>${employee.accountNumber}</div>
        </div>
        <div class="detail-item">
            <label>IFSC Code</label>
            <div>${employee.ifscCode}</div>
        </div>
    </div>
</div>

<div class="action-buttons">
    <button class="btn btn-approve"
onclick="updateStatus('${employee.employeeId}',
'Approved')">Approve</button>
    <button class="btn btn-reject"
onclick="updateStatus('${employee.employeeId}',
'Rejected')">Reject</button>
</div>
`;

modal.style.display = 'block';
}
```

```
// Close modal
function closeModal() {
    document.getElementById('employeeModal').style.display = 'none';
}

// Update employee status
function updateStatus(employeeId, status) {
    const employeeIndex = employees.findIndex(emp => emp.employeeId ===
employeeId);
    if (employeeIndex !== -1) {
        employees[employeeIndex].status = status;
        localStorage.setItem('hrEmployees', JSON.stringify(employees));
        displayEmployees();
        closeModal();
    }
}

// Event listeners
document.getElementById('searchInput').addEventListener('input', () => {
    currentPage = 1;
    displayEmployees();
});

document.getElementById('statusFilter').addEventListener('change', () => {
    currentPage = 1;
    displayEmployees();
});

document.getElementById('sortBy').addEventListener('change',
displayEmployees);

// Close modal when clicking outside
window.onclick = function(event) {
const modal = document.getElementById('employeeModal');
if (event.target === modal) {
    closeModal();
}
};

// Handle escape key press to close modal
```

```
document.addEventListener('keydown', function(event) {
if (event.key === 'Escape') {
    closeModal();
}
});

// Generate dummy data for testing if no data exists
function generateDummyData() {
if (!localStorage.getItem('hrEmployees')) {
    const dummyEmployees = [
        {
            employeeId: 'EMP001',
            firstName: 'John',
            lastName: 'Doe',
            email: 'john.doe@example.com',
            phone: '1234567890',
            dob: '1990-01-01',
            gender: 'Male',
            bloodGroup: 'O+',
            currentAddress: '123 Main St',
            currentCity: 'New York',
            currentState: 'NY',
            currentPincode: '100001',
            accountHolderName: 'John Doe',
            bankName: 'City Bank',
            accountNumber: '123456789',
            ifscCode: 'CITY0000123',
            submissionDate: new Date(2024, 0, 1).toISOString(),
            status: 'Pending'
        },
        {
            employeeId: 'EMP002',
            firstName: 'Jane',
            lastName: 'Smith',
            email: 'jane.smith@example.com',
            phone: '9876543210',
            dob: '1992-05-15',
            gender: 'Female',
            bloodGroup: 'A+',
            currentAddress: '456 Oak Ave',
```

```
            currentCity: 'Los Angeles',
            currentState: 'CA',
            currentPincode: '200001',
            accountHolderName: 'Jane Smith',
            bankName: 'Global Bank',
            accountNumber: '987654321',
            ifscCode: 'GLOB0000456',
            submissionDate: new Date(2024, 0, 15).toISOString(),
            status: 'Approved'
        }
    ];
    localStorage.setItem('hrEmployees', JSON.stringify(dummyEmployees));
}
}

// Export data to CSV
function exportToCSV() {
const filteredEmployees = getFilteredEmployees();
if (filteredEmployees.length === 0) {
    alert('No data to export');
    return;
}

// Get headers from first employee object
const headers = Object.keys(filteredEmployees[0]);

// Create CSV content
let csvContent = headers.join(',') + '\n';

// Add data rows
filteredEmployees.forEach(emp => {
    const row = headers.map(header => {
        let cell = emp[header] || '';
        // Wrap in quotes if contains comma
        if (cell.toString().includes(',')) {
            cell = `"${cell}"`;
        }
        return cell;
    });
    csvContent += row.join(',') + '\n';
```

```
});

// Create and trigger download
const blob = new Blob([csvContent], { type: 'text/csv;charset=utf-8;' });
const link = document.createElement('a');
if (navigator.msSaveBlob) {
    // IE 10+
    navigator.msSaveBlob(blob, 'employee_data.csv');
} else {
    // Other browsers
    link.href = URL.createObjectURL(blob);
    link.setAttribute('download', 'employee_data.csv');
    document.body.appendChild(link);
    link.click();
    document.body.removeChild(link);
}
}

// Add export button to header
function addExportButton() {
const searchBar = document.querySelector('.search-bar');
const exportBtn = document.createElement('button');
exportBtn.className = 'btn';
exportBtn.style.backgroundColor = '#2563eb';
exportBtn.style.color = 'white';
exportBtn.textContent = 'Export to CSV';
exportBtn.onclick = exportToCSV;
searchBar.appendChild(exportBtn);
}

// Initialize dashboard
function initDashboard() {
generateDummyData();
loadEmployees();
addExportButton();
}

// Add loading state handlers
function showLoading() {
const grid = document.getElementById('employeesGrid');
```

```
grid.innerHTML = '<div style="text-align: center; padding:
20px;">Loading...</div>';
}

function hideLoading() {
displayEmployees();
}

// Refresh data periodically
function setupAutoRefresh() {
setInterval(() => {
    loadEmployees();
}, 30000); // Refresh every 30 seconds
}

// Error handling
function handleError(error) {
console.error('Error:', error);
const grid = document.getElementById('employeesGrid');
grid.innerHTML = `
    <div style="text-align: center; padding: 20px; color: #ef4444;">
        An error occurred. Please try refreshing the page.
    </div>
`;
}

// Initialize the dashboard with error handling
try {
initDashboard();
setupAutoRefresh();
} catch (error) {
handleError(error);
}

    </script>
</body>
</html>
```