```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Employee Attendance Portal</title>
    <style>
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
            font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI',
Roboto, sans-serif;
        }

        body {
            min-height: 100vh;
            background: linear-gradient(135deg, #f6f8fd 0%, #e9f0ff 100%);
            padding: 2rem;
            display: flex;
            align-items: center;
            justify-content: center;
        }

        .container {
            width: 100%;
            max-width: 500px;
            background: rgba(255, 255, 255, 0.95);
            backdrop-filter: blur(10px);
            border-radius: 20px;
            padding: 2rem;
            box-shadow: 0 8px 32px rgba(31, 38, 135, 0.15);
            transform: translateY(0);
            transition: transform 0.3s ease;
        }

        .container:hover {
            transform: translateY(-5px);
        }
```

```css
.card-header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: 2rem;
    position: relative;
}

.card-title {
    color: #1a365d;
    font-size: 1.5rem;
    font-weight: 600;
}

.clock-icon {
    width: 32px;
    height: 32px;
    color: #1a365d;
}

.input-group {
    margin-bottom: 1.5rem;
}

.input-label {
    display: block;
    margin-bottom: 0.5rem;
    color: #4a5568;
    font-size: 0.875rem;
}

.input-field {
    width: 100%;
    padding: 0.75rem;
    border: 2px solid #e2e8f0;
    border-radius: 8px;
    font-size: 1rem;
    background: rgba(255, 255, 255, 0.9);
}
```

```css
.input-field:focus {
    outline: none;
    border-color: #4299e1;
    box-shadow: 0 0 0 3px rgba(66, 153, 225, 0.2);
}

.status-container {
    background: rgba(255, 255, 255, 0.9);
    border-radius: 12px;
    padding: 1rem;
    margin: 1.5rem 0;
}

.status-row {
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin: 0.5rem 0;
}

.status-badge {
    display: inline-flex;
    align-items: center;
    padding: 0.5rem 1rem;
    border-radius: 999px;
    font-size: 0.875rem;
    font-weight: 500;
}

.status-dot {
    width: 8px;
    height: 8px;
    border-radius: 50%;
    margin-right: 8px;
    position: relative;
}

.status-in {
    background-color: rgba(72, 187, 120, 0.1);
    color: #48bb78;
```

```css
        }

        .status-out {
            background-color: rgba(229, 62, 62, 0.1);
            color: #e53e3e;
        }

        .button {
            width: 100%;
            padding: 1rem;
            border-radius: 8px;
            background: linear-gradient(135deg, #1a365d 0%, #2a4365 100%);
            color: white;
            border: none;
            font-size: 1rem;
            font-weight: 500;
            cursor: pointer;
            transition: all 0.3s ease;
        }

        .button:disabled {
            background: linear-gradient(135deg, #cbd5e0 0%, #a0aec0 100%);
            cursor: not-allowed;
        }

        .button:not(:disabled):hover {
            transform: translateY(-2px);
            box-shadow: 0 4px 12px rgba(26, 54, 93, 0.25);
        }

        @media (max-width: 640px) {
            body {
                padding: 1rem;
            }

            .container {
                padding: 1.5rem;
            }

            .card-title {
```

```
                font-size: 1.25rem;
            }

            .button {
                padding: 0.875rem;
            }
        }

    </style>
</head>
<body>
    <div class="container">
        <div class="card-header">
            <h2 class="card-title">Punch-IN/OUT Portal</h2>
            <svg class="clock-icon" viewBox="0 0 24 24" fill="none"
stroke="currentColor" stroke-width="2">
                <circle cx="12" cy="12" r="10"></circle>
                <polyline points="12 6 12 12 16 14"></polyline>
            </svg>
        </div>
        <div class="input-group">
            <label class="input-label" for="employeeId">Employee
ID</label>
            <input type="text" id="employeeId" class="input-field"
placeholder="Enter your 7-digit ID">
        </div>
        <div class="input-group">
            <label class="input-label" for="employeeName">Full
Name</label>
            <input type="text" id="employeeName" class="input-field"
placeholder="Enter your full name">
        </div>
        <div class="status-container">
            <div class="status-row">
                <span class="status-label">Current Status:</span>
                <div class="status-badge status-out">
                    <span class="status-dot"></span>
                    <span class="status-text">Punched Out</span>
                </div>
            </div>
```

```html
            <div class="status-row">
                <span class="status-label">Time:</span>
                <span class="time">--:--:--</span>
            </div>
        </div>
        <button class="button" id="punchButton" disabled>Punch In</button>
    </div>

    <script>
        const employeeIdInput = document.getElementById('employeeId');
        const employeeNameInput = document.getElementById('employeeName');
        const statusBadge = document.querySelector('.status-badge');
        const statusText = document.querySelector('.status-text');
        const punchButton = document.getElementById('punchButton');
        const timeDisplay = document.querySelector('.time');

        let status = 'out';

        function validateEmployeeId(input) {
            const employeeIdRegex = /^ATS0\d{3}$/;
            return employeeIdRegex.test(input);
        }

        function validateEmployeeName(input) {
            const alphabetSpaceRegex = /^[a-zA-Z\s]{5,}$/;
            return alphabetSpaceRegex.test(input);
        }

        function updateValidation() {
            const isIdValid = validateEmployeeId(employeeIdInput.value);
            const isNameValid =
validateEmployeeName(employeeNameInput.value);
            punchButton.disabled = !(isIdValid && isNameValid);
        }

        function updateTime() {
            const now = new Date();
            timeDisplay.textContent = now.toLocaleTimeString();
        }
```

```
        function toggleStatus() {
            status = status === 'in' ? 'out' : 'in';

            statusBadge.className = `status-badge status-${status}`;
            statusText.textContent = `Punched
${status.charAt(0).toUpperCase() + status.slice(1)}`;
            punchButton.textContent = `Punch ${status === 'in' ? 'Out' :
'In'}`;

            const record = {
                id: employeeIdInput.value,
                name: employeeNameInput.value,
                status: status,
                timestamp: new Date().toLocaleTimeString(),
                date: new Date().toLocaleDateString()
            };

            const records =
JSON.parse(localStorage.getItem('punchRecords') || '[]');
            records.push(record);
            localStorage.setItem('punchRecords', JSON.stringify(records));

            employeeIdInput.disabled = status === 'in';
            employeeNameInput.disabled = status === 'in';
        }

        employeeIdInput.addEventListener('input', updateValidation);
        employeeNameInput.addEventListener('input', updateValidation);
        punchButton.addEventListener('click', toggleStatus);

        setInterval(updateTime, 1000);
        updateTime();
    </script>
</body>
</html>
```