```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Student Attendance QR System</title>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/qrcodejs/1.0.0/qrcode.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/jsqr@1.4.0/dist/jsQR.min.js"></script>
    <style>
        * {
            box-sizing: border-box;
            margin: 0;
            padding: 0;
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
        }

        body {
            background-color: #f5f7fa;
            color: #333;
            line-height: 1.6;
            padding: 20px;
        }

        .container {
            max-width: 1200px;
            margin: 0 auto;
        }

        header {
            text-align: center;
            margin-bottom: 30px;
            padding: 20px;
            background: linear-gradient(135deg, #4a6fa5, #2c4a7a);
            color: white;
            border-radius: 10px;
            box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
        }

        h1 {
            font-size: 2.2rem;
            margin-bottom: 10px;
        }

        .description {
```

```
    font-size: 1.1rem;
    opacity: 0.9;
}

.main-content {
    display: grid;
    grid-template-columns: 1fr 1fr;
    gap: 30px;
}

@media (max-width: 768px) {
    .main-content {
        grid-template-columns: 1fr;
    }
}

.section {
    background-color: white;
    border-radius: 10px;
    padding: 25px;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}

.section-title {
    font-size: 1.5rem;
    margin-bottom: 20px;
    color: #4a6fa5;
    border-bottom: 2px solid #eaeaea;
    padding-bottom: 10px;
}

.form-group {
    margin-bottom: 20px;
}

label {
    display: block;
    margin-bottom: 8px;
    font-weight: 600;
}

input, select {
    width: 100%;
    padding: 12px;
```

```
    border: 1px solid #ddd;
    border-radius: 5px;
    font-size: 1rem;
}

button {
    background-color: #4a6fa5;
    color: white;
    border: none;
    padding: 12px 20px;
    border-radius: 5px;
    cursor: pointer;
    font-size: 1rem;
    font-weight: 600;
    transition: background-color 0.3s;
    margin-top: 10px;
}

button:hover {
    background-color: #3a5a80;
}

.qr-container {
    text-align: center;
    margin: 20px 0;
    min-height: 250px;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    border: 1px solid #ddd;
    border-radius: 10px;
    padding: 20px;
}

#qrCode {
    display: inline-block;
    padding: 15px;
    background-color: white;
    border-radius: 10px;
}

.scanner-container {
    position: relative;
```

```
    width: 100%;
    max-width: 500px;
    margin: 0 auto;
}

#video {
    width: 100%;
    height: auto;
    border-radius: 10px;
    background-color: #000;
}

#canvas {
    display: none;
}

.scanner-overlay {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    border: 2px solid #4a6fa5;
    border-radius: 10px;
    box-sizing: border-box;
}

.scanner-line {
    position: absolute;
    width: 100%;
    height: 2px;
    background-color: #4a6fa5;
    top: 50%;
    animation: scan 2s infinite linear;
}

@keyframes scan {
    0% { top: 10%; }
    50% { top: 90%; }
    100% { top: 10%; }
}

.attendance-table {
    width: 100%;
```

```
    border-collapse: collapse;
    margin-top: 20px;
    overflow-x: auto;
}

.attendance-table th, .attendance-table td {
    border: 1px solid #ddd;
    padding: 12px;
    text-align: left;
}

.attendance-table th {
    background-color: #4a6fa5;
    color: white;
}

.attendance-table tr:nth-child(even) {
    background-color: #f2f2f2;
}

.download-section {
    margin-top: 20px;
    display: flex;
    gap: 10px;
}

.download-section input {
    flex: 1;
}

.message {
    padding: 10px;
    margin: 10px 0;
    border-radius: 5px;
    text-align: center;
}

.success {
    background-color: #d4edda;
    color: #155724;
    border: 1px solid #c3e6cb;
}

.error {
```

```
    background-color: #f8d7da;
    color: #721c24;
    border: 1px solid #f5c6cb;
}

.hidden {
    display: none;
}

.camera-controls {
    display: flex;
    gap: 10px;
    margin-top: 15px;
}

.camera-controls button {
    flex: 1;
}

.stop-btn {
    background-color: #dc3545;
}

.stop-btn:hover {
    background-color: #c82333;
}

.sample-data {
    margin-top: 20px;
    padding: 15px;
    background-color: #f8f9fa;
    border-radius: 5px;
    border-left: 4px solid #4a6fa5;
}

.sample-data h3 {
    margin-bottom: 10px;
    color: #4a6fa5;
}

.sample-data ul {
    list-style-type: none;
}
```

```
        .sample-data li {
            padding: 5px 0;
        }

        .empty-state {
            text-align: center;
            padding: 20px;
            color: #6c757d;
        }

        .qr-placeholder {
            color: #6c757d;
            font-style: italic;
        }

        .button-group {
            display: flex;
            gap: 10px;
        }

        .button-group button {
            flex: 1;
        }
    </style>
</head>
<body>
    <div class="container">
        <header>
            <h1>Student Attendance QR System</h1>
            <p class="description">Generate QR codes with student name and section, then scan
them for attendance tracking</p>
        </header>

        <div class="main-content">
            <!-- QR Code Generator Section -->
            <div class="section">
                <h2 class="section-title">Generate Student QR Code</h2>
                <div class="form-group">
                    <label for="studentName">Student Name</label>
                    <input type="text" id="studentName" placeholder="Enter student name">
                </div>
                <div class="form-group">
                    <label for="studentSection">Section</label>
                    <input type="text" id="studentSection" placeholder="Enter section">
```

```
            </div>

            <div class="button-group">
                <button id="generateQR">Generate QR Code</button>
                <button id="testQR" style="background: #28a745;">Test with Sample
Data</button>
            </div>

            <div class="qr-container">
                <div id="qrCode" class="qr-placeholder">
                    QR Code will appear here after generation
                </div>
            </div>

            <button id="downloadQR" class="hidden">Download QR Code</button>

            <div class="sample-data">
                <h3>Sample Data for Testing</h3>
                <ul>
                    <li><strong>Name:</strong> Maria Santos, <strong>Section:</strong> 10-A</li>
                    <li><strong>Name:</strong> Juan Dela Cruz, <strong>Section:</strong>
10-B</li>
                    <li><strong>Name:</strong> Ana Reyes, <strong>Section:</strong> 11-A</li>
                </ul>
            </div>
        </div>

        <!-- QR Code Scanner Section -->
        <div class="section">
            <h2 class="section-title">Scan QR Code</h2>
            <div class="scanner-container">
                <video id="video" playsinline></video>
                <canvas id="canvas"></canvas>
                <div class="scanner-overlay">
                    <div class="scanner-line"></div>
                </div>
            </div>

            <div class="camera-controls">
                <button id="startScanner">Start Scanner</button>
                <button id="stopScanner" class="stop-btn hidden">Stop Scanner</button>
            </div>

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

```
            </div>
        </div>

        <!-- Attendance Records Section -->
        <div class="section" style="margin-top: 30px;">
            <h2 class="section-title">Attendance Records</h2>
            <div class="download-section">
                <input type="text" id="fileName" placeholder="Enter file name (default:
attendance_date)">
                <button id="downloadTable">Download Attendance</button>
            </div>

            <table class="attendance-table">
                <thead>
                    <tr>
                        <th>Student Name</th>
                        <th>Section</th>
                        <th>Date & Time</th>
                    </tr>
                </thead>
                <tbody id="attendanceBody">
                    <tr class="empty-state">
                        <td colspan="3">No attendance records yet. Scan QR codes to add
records.</td>
                    </tr>
                </tbody>
            </table>
        </div>
    </div>

    <script>
        // DOM Elements
        const generateQRBtn = document.getElementById('generateQR');
        const downloadQRBtn = document.getElementById('downloadQR');
        const startScannerBtn = document.getElementById('startScanner');
        const stopScannerBtn = document.getElementById('stopScanner');
        const downloadTableBtn = document.getElementById('downloadTable');
        const testQRBtn = document.getElementById('testQR');
        const qrCodeDiv = document.getElementById('qrCode');
        const video = document.getElementById('video');
        const canvas = document.getElementById('canvas');
        const scanResult = document.getElementById('scanResult');
        const attendanceBody = document.getElementById('attendanceBody');
        const fileNameInput = document.getElementById('fileName');
```

```
// Variables
let currentQR = null;
let stream = null;
let scanning = false;
let attendanceRecords = [];

// Generate QR Code - USING WORKING CODE
generateQRBtn.addEventListener('click', generateQR);
testQRBtn.addEventListener('click', testQR);

function generateQR() {
    const studentName = document.getElementById('studentName').value.trim();
    const studentSection = document.getElementById('studentSection').value.trim();

    if (!studentName || !studentSection) {
        alert('Please fill in both student name and section!');
        return;
    }

    // Create student data
    const studentData = {
        name: studentName,
        section: studentSection,
        timestamp: new Date().toISOString()
    };

    const jsonString = JSON.stringify(studentData);

    // Clear previous QR
    qrCodeDiv.innerHTML = '';

    // Generate new QR
    currentQR = new QRCode(qrCodeDiv, {
        text: jsonString,
        width: 200,
        height: 200,
        colorDark: "#000000",
        colorLight: "#ffffff",
        correctLevel: QRCode.CorrectLevel.H
    });

    // Show download button
    downloadQRBtn.classList.remove('hidden');
```

```
    console.log('QR Generated for:', studentName);
}

function testQR() {
    // Fill with sample data
    document.getElementById('studentName').value = 'Maria Santos';
    document.getElementById('studentSection').value = '10-A';

    // Generate immediately
    setTimeout(generateQR, 100);
}

// Download QR Code
downloadQRBtn.addEventListener('click', function() {
    if (!currentQR) {
        alert('Please generate a QR code first!');
        return;
    }

    const studentName = document.getElementById('studentName').value.trim();
    const studentSection = document.getElementById('studentSection').value.trim();
    const fileName = `${studentName}_${studentSection}_qr`.replace(/\s+/g, '_');

    const canvas = qrCodeDiv.querySelector('canvas');
    if (canvas) {
        const link = document.createElement('a');
        link.download = `${fileName}.png`;
        link.href = canvas.toDataURL('image/png');
        link.click();
    }
});

// Start QR Scanner
startScannerBtn.addEventListener('click', async function() {
    try {
        // Request back camera
        const constraints = {
            video: {
                facingMode: { ideal: 'environment' } // Prefer back camera
            }
        };

        stream = await navigator.mediaDevices.getUserMedia(constraints);
```

```
            video.srcObject = stream;
            await video.play();

            // Show/hide buttons
            startScannerBtn.classList.add('hidden');
            stopScannerBtn.classList.remove('hidden');

            // Start scanning
            scanning = true;
            scanQRCode();
        } catch (err) {
            console.error('Error accessing camera:', err);

            // Fallback to any camera if back camera is not available
            try {
                stream = await navigator.mediaDevices.getUserMedia({ video: true });
                video.srcObject = stream;
                await video.play();

                startScannerBtn.classList.add('hidden');
                stopScannerBtn.classList.remove('hidden');
                scanning = true;
                scanQRCode();

                scanResult.textContent = 'Using available camera (back camera not found)';
                scanResult.className = 'message error';
                scanResult.classList.remove('hidden');
            } catch (fallbackError) {
                alert('Error accessing camera. Please make sure you have granted camera
permissions.');
            }
        }
    });

    // Stop QR Scanner
    stopScannerBtn.addEventListener('click', function() {
        scanning = false;

        if (stream) {
            stream.getTracks().forEach(track => track.stop());
        }

        // Show/hide buttons
```

```
            startScannerBtn.classList.remove('hidden');
            stopScannerBtn.classList.add('hidden');

            // Hide scan result
            scanResult.classList.add('hidden');
        });

        // QR Code Scanning Function
        function scanQRCode() {
            if (!scanning) return;

            if (video.readyState === video.HAVE_ENOUGH_DATA) {
                canvas.width = video.videoWidth;
                canvas.height = video.videoHeight;

                const ctx = canvas.getContext('2d');
                ctx.drawImage(video, 0, 0, canvas.width, canvas.height);

                const imageData = ctx.getImageData(0, 0, canvas.width, canvas.height);
                const code = jsQR(imageData.data, imageData.width, imageData.height);

                if (code) {
                    try {
                        const data = JSON.parse(code.data);
                        console.log("QR Code scanned:", data);

                        // Add timestamp if not present
                        if (!data.timestamp) {
                            data.timestamp = new Date().toISOString();
                        }

                        // Check if this record already exists (same name and within 5 minutes)
                        const recordExists = attendanceRecords.some(record =>
                            record.name === data.name &&
                            record.section === data.section &&
                            Math.abs(new Date(data.timestamp) - new Date(record.timestamp)) < 300000
// 5 minutes
                        );

                        if (!recordExists) {
                            // Add to attendance records
                            attendanceRecords.push(data);

                            // Update table
```

```
                    updateAttendanceTable();

                    // Show success message
                    scanResult.textContent = `Attendance recorded for ${data.name}
(${data.section})`;
                    scanResult.className = 'message success';
                    scanResult.classList.remove('hidden');
                } else {
                    // Show info message
                    scanResult.textContent = `Attendance already recorded for ${data.name}`;
                    scanResult.className = 'message error';
                    scanResult.classList.remove('hidden');
                }

                // Stop scanning for a moment
                scanning = false;
                setTimeout(() => {
                    scanning = true;
                    scanQRCode();
                }, 2000);

                return;
            } catch (e) {
                console.error('Error parsing QR code data:', e);
                scanResult.textContent = 'Invalid QR code format';
                scanResult.className = 'message error';
                scanResult.classList.remove('hidden');
            }
        }
    }

    // Continue scanning
    requestAnimationFrame(scanQRCode);
}

// Update Attendance Table
function updateAttendanceTable() {
    // Clear table
    attendanceBody.innerHTML = '';

    if (attendanceRecords.length === 0) {
        const emptyRow = document.createElement('tr');
        emptyRow.className = 'empty-state';
```

```
            emptyRow.innerHTML = '<td colspan="3">No attendance records yet. Scan QR codes
to add records.</td>';
            attendanceBody.appendChild(emptyRow);
```