# Web QR Attendance System — Full Code & Setup

This document contains a complete, copy-paste ready web project for a **teacher-operated QR/barcode attendance system**.

**Overview**

* Teacher opens a web page (Teacher Dashboard) and clicks **Start Attendance** which opens the camera and scans student QR codes.
* Each scanned code contains the student’s studentId. The front-end sends that to the backend which records attendance (date/time/subject).
* There’s also a Student QR page where a student can view/download/print their QR code.
* The backend uses **Node.js + Express** and **SQLite** (file DB) for simplicity.

## File list (create these files and folders exactly)

attendance-qr-project/  
├─ package.json  
├─ server.js  
├─ public/  
│ ├─ index.html # Teacher dashboard + scanner  
│ ├─ student\_qr.html # Student QR viewer / generator  
│ ├─ qrcode.min.js # QR generator lib  
│ ├─ html5-qrcode.min.js # QR scanner lib (from cdn or local)  
│ └─ styles.css  
└─ data/  
 └─ attendance.db # will be created automatically

## 1) package.json

{  
 "name": "attendance-qr-system",  
 "version": "1.0.0",  
 "description": "Simple web QR attendance system",  
 "main": "server.js",  
 "scripts": {  
 "start": "node server.js"  
 },  
 "dependencies": {  
 "cors": "^2.8.5",  
 "express": "^4.18.2",  
 "sqlite3": "^5.1.6",  
 "body-parser": "^1.20.2"  
 }  
}

## 2) server.js

const express = require('express');  
const bodyParser = require('body-parser');  
const sqlite3 = require('sqlite3').verbose();  
const path = require('path');  
const fs = require('fs');  
const cors = require('cors');  
  
const app = express();  
app.use(cors());  
app.use(bodyParser.json());  
app.use(express.static(path.join(\_\_dirname, 'public')));  
  
// ensure data dir  
if (!fs.existsSync(path.join(\_\_dirname, 'data'))) {  
 fs.mkdirSync(path.join(\_\_dirname, 'data'));  
}  
  
const DB\_PATH = path.join(\_\_dirname, 'data', 'attendance.db');  
const db = new sqlite3.Database(DB\_PATH);  
  
// create tables if not exist  
db.serialize(() => {  
 db.run(`CREATE TABLE IF NOT EXISTS students (  
 studentId TEXT PRIMARY KEY,  
 name TEXT,  
 class TEXT  
 )`);  
  
 db.run(`CREATE TABLE IF NOT EXISTS attendance (  
 id INTEGER PRIMARY KEY AUTOINCREMENT,  
 studentId TEXT,  
 date TEXT,  
 timeIn TEXT,  
 subject TEXT,  
 UNIQUE(studentId, date, subject)  
 )`);  
});  
  
// endpoints  
app.get('/api/students', (req, res) => {  
 db.all('SELECT \* FROM students', (err, rows) => {  
 if (err) return res.status(500).json({ error: err.message });  
 res.json(rows);  
 });  
});  
  
app.post('/api/students', (req, res) => {  
 const { studentId, name, class: cls } = req.body;  
 if (!studentId) return res.status(400).json({ error: 'studentId required' });  
 db.run('INSERT OR REPLACE INTO students(studentId, name, class) VALUES(?,?,?)', [studentId, name || '', cls || ''], function(err) {  
 if (err) return res.status(500).json({ error: err.message });  
 res.json({ success: true });  
 });  
});  
  
// mark attendance  
app.post('/api/mark', (req, res) => {  
 const { studentId, subject } = req.body;  
 if (!studentId) return res.status(400).json({ error: 'studentId required' });  
  
 // get date/time  
 const now = new Date();  
 const date = now.toISOString().split('T')[0]; // YYYY-MM-DD  
 const timeIn = now.toTimeString().split(' ')[0]; // HH:MM:SS  
  
 // insert if not duplicate for same date+subject  
 const stmt = db.prepare('INSERT OR IGNORE INTO attendance(studentId, date, timeIn, subject) VALUES(?,?,?,?)');  
 stmt.run([studentId, date, timeIn, subject || 'General'], function(err) {  
 if (err) return res.status(500).json({ error: err.message });  
 if (this.changes === 0) {  
 // already exists  
 return res.json({ success: false, message: 'Already marked' });  
 }  
 res.json({ success: true, date, timeIn });  
 });  
 stmt.finalize();  
});  
  
// get attendance (filter by date optional)  
app.get('/api/attendance', (req, res) => {  
 const date = req.query.date; // optional  
 let sql = 'SELECT a.\*, s.name FROM attendance a LEFT JOIN students s ON a.studentId = s.studentId';  
 const params = [];  
 if (date) { sql += ' WHERE date = ?'; params.push(date); }  
 sql += ' ORDER BY timeIn';  
 db.all(sql, params, (err, rows) => {  
 if (err) return res.status(500).json({ error: err.message });  
 res.json(rows);  
 });  
});  
  
// serve index  
app.get('/', (req, res) => {  
 res.sendFile(path.join(\_\_dirname, 'public', 'index.html'));  
});  
  
const PORT = process.env.PORT || 3000;  
app.listen(PORT, () => console.log('Server started on port', PORT));

## 3) public/index.html (Teacher dashboard & scanner)

<!doctype html>  
<html>  
<head>  
 <meta charset="utf-8" />  
 <meta name="viewport" content="width=device-width,initial-scale=1" />  
 <title>Teacher — Attendance Scanner</title>  
 <link rel="stylesheet" href="./styles.css">  
 <!-- html5-qrcode script -->  
 <script src="https://unpkg.com/html5-qrcode@2.3.8/minified/html5-qrcode.min.js"></script>  
</head>  
<body>  
 <div class="container">  
 <h1>Attendance — Teacher Scanner</h1>  
  
 <div class="controls">  
 <label>Subject: <input id="subject" placeholder="Maths / Class 10A" /></label>  
 <button id="startBtn">Start Attendance</button>  
 <button id="stopBtn" disabled>Stop Attendance</button>  
 <button id="viewAttendance">View Today's Attendance</button>  
 </div>  
  
 <div id="reader" style="width:500px; max-width:100%; margin-top:12px;"></div>  
  
 <div id="log"></div>  
 </div>  
  
<script>  
const startBtn = document.getElementById('startBtn');  
const stopBtn = document.getElementById('stopBtn');  
const log = document.getElementById('log');  
const subjectInput = document.getElementById('subject');  
let html5QrcodeScanner;  
  
function appendLog(msg) {  
 const p = document.createElement('div');  
 p.textContent = '[' + new Date().toLocaleTimeString() + '] ' + msg;  
 log.prepend(p);  
}  
  
startBtn.addEventListener('click', () => {  
 startBtn.disabled = true;  
 stopBtn.disabled = false;  
 const qrRegionId = 'reader';  
 html5QrcodeScanner = new Html5Qrcode(qrRegionId);  
 const config = { fps: 10, qrbox: { width: 250, height: 250 } };  
 html5QrcodeScanner.start(  
 { facingMode: 'environment' },  
 config,  
 qrCodeSuccessCallback  
 ).catch(err => {  
 appendLog('Camera start error: ' + err);  
 startBtn.disabled = false;  
 stopBtn.disabled = true;  
 });  
});  
  
stopBtn.addEventListener('click', () => {  
 stopBtn.disabled = true;  
 startBtn.disabled = false;  
 if (html5QrcodeScanner) {  
 html5QrcodeScanner.stop().then(() => {  
 appendLog('Scanner stopped');  
 }).catch(err => appendLog('Stop error: ' + err));  
 }  
});  
  
function qrCodeSuccessCallback(decodedText, decodedResult) {  
 // decodedText expected to be studentId or JSON; we assume studentId  
 const studentId = decodedText.trim();  
 const subject = subjectInput.value || 'General';  
  
 // send to server  
 fetch('/api/mark', {  
 method: 'POST',  
 headers: { 'Content-Type': 'application/json' },  
 body: JSON.stringify({ studentId, subject })  
 }).then(r => r.json()).then(resp => {  
 if (resp.success) appendLog('Marked: ' + studentId + ' — ' + subject + ' @ ' + resp.timeIn);  
 else appendLog('Not marked (maybe duplicate): ' + studentId + ' — ' + (resp.message || ''));  
 }).catch(err => appendLog('Network error: ' + err));  
}  
  
// view today's attendance  
document.getElementById('viewAttendance').addEventListener('click', async () => {  
 const today = new Date().toISOString().split('T')[0];  
 const res = await fetch('/api/attendance?date=' + today);  
 const rows = await res.json();  
 if (!rows || rows.length === 0) {  
 appendLog('No attendance for today yet.');  
 return;  
 }  
 appendLog('--- Today\'s Attendance ---');  
 rows.forEach(r => appendLog(r.studentId + (r.name ? (' ('+r.name+')') : '') + ' — ' + r.timeIn + ' — ' + r.subject));  
});  
</script>  
</body>  
</html>

## 4) public/student\_qr.html (Student QR generator / viewer)

<!doctype html>  
<html>  
<head>  
 <meta charset="utf-8" />  
 <meta name="viewport" content="width=device-width,initial-scale=1" />  
 <title>Student — Your QR</title>  
 <link rel="stylesheet" href="./styles.css">  
</head>  
<body>  
 <div class="container">  
 <h1>Your Student QR</h1>  
 <div>  
 <label>Student ID: <input id="studentId" placeholder="e.g. S12345" /></label>  
 <label>Name: <input id="name" placeholder="Your name (optional)" /></label>  
 <label>Class: <input id="class" placeholder="Class/Section (optional)" /></label>  
 <button id="generate">Generate QR</button>  
 <button id="saveStudent">Save to Server</button>  
 </div>  
  
 <div style="margin-top:12px;">  
 <canvas id="qrcanvas"></canvas>  
 </div>  
  
 <a id="downloadLink" style="display:block;margin-top:8px;">Download QR</a>  
 </div>  
  
 <!-- QR lib (qrcodejs) -->  
 <script src="https://unpkg.com/qrcode@1.5.1/build/qrcode.min.js"></script>  
 <script>  
 const generateBtn = document.getElementById('generate');  
 const saveBtn = document.getElementById('saveStudent');  
 const studentIdInput = document.getElementById('studentId');  
 const nameInput = document.getElementById('name');  
 const classInput = document.getElementById('class');  
 const canvas = document.getElementById('qrcanvas');  
 const downloadLink = document.getElementById('downloadLink');  
  
 generateBtn.addEventListener('click', () => {  
 const id = studentIdInput.value.trim();  
 if (!id) return alert('Enter student ID');  
 // draw QR to canvas  
 QRCode.toCanvas(canvas, id, { width: 300 }, function (error) {  
 if (error) return alert('QR error: ' + error);  
 downloadLink.href = canvas.toDataURL();  
 downloadLink.download = id + '\_qr.png';  
 downloadLink.textContent = 'Download QR as PNG';  
 })  
 });  
  
 saveBtn.addEventListener('click', async () => {  
 const id = studentIdInput.value.trim();  
 if (!id) return alert('Enter student ID');  
 const payload = { studentId: id, name: nameInput.value, class: classInput.value };  
 const res = await fetch('/api/students', { method: 'POST', headers: {'Content-Type':'application/json'}, body: JSON.stringify(payload) });  
 const data = await res.json();  
 if (data.success) alert('Saved'); else alert('Error: ' + (data.error || ''));  
 });  
 </script>  
</body>  
</html>

## 5) public/styles.css

body { font-family: Arial, sans-serif; background:#f5f7fb; color:#222; }  
.container { max-width:800px; margin:30px auto; background:white; padding:18px; border-radius:8px; box-shadow:0 6px 20px rgba(0,0,0,0.06);}  
.controls { display:flex; gap:8px; align-items:center; flex-wrap:wrap; }  
#log { margin-top:12px; max-height:260px; over