

QR Code Attendance System: A Web-Based Smart Attendance Solution

Mrs. Suguna M. K, *Associate Prof, SMVIT,*

Debashish Choudhary , *Student, SMVIT,*

Harikesh deep, *Student, SMVIT,*

Jishnu Kumar, *Student, SMVIT,*

Karan Kumar, *Student, SMVIT*

Abstract

This research presents the design and implementation of a QR Code-Based Smart Attendance System aimed at modernizing the traditional attendance tracking process. Existing manual or biometric-based methods suffer from drawbacks such as inefficiency, inaccuracies, proxy attendance, and lack of hygiene. To address these challenges, the proposed system utilizes unique QR codes assigned to each user and a real-time scanning interface built with OpenCV and Python Flask. Attendance is logged into a relational database (MySQL), providing accurate, fast, and secure records. The system includes role-based dashboards for students and administrators, emphasizing modularity, scalability, and user-friendliness.

I. Introduction

In the era of digital transformation, institutions are increasingly adopting technology to improve operational efficiency. One such domain is attendance tracking, a critical but traditionally tedious process. Manual systems are time-consuming and susceptible to human error. RFID systems, though popular, require physical infrastructure and carry costs. Biometric systems, especially post-pandemic, raise hygiene concerns.

This paper explores the development of a QR code-based system as a cost-effective and hygienic solution. The platform uses unique QR codes to identify students, scanned via webcam or mobile camera. The system is designed with a modular architecture, consisting of a Flask-based backend, OpenCV-powered QR scanner, and a responsive HTML/CSS/JS frontend. It aims to automate attendance, minimize human intervention, and offer real-time logging and reporting.

II. Literature Survey

Recent advancements in computer vision and web development have led to numerous attendance automation systems. Biometric and facial recognition

methods have gained popularity, yet issues such as environmental constraints, lighting conditions, and privacy remain. RFID/NFC solutions improve over manual systems but depend on physical card readers.

QR codes provide a flexible and cost-effective medium for encoding unique identity data. They can be generated dynamically, printed or stored digitally, and scanned with standard camera hardware. Research studies and pilot implementations in schools and universities show that QR-based attendance can reduce costs, simplify deployments, and prevent proxy attendance.

III. System Objectives

The key objectives of the system are:

- To eliminate the possibility of proxy or false attendance entries.
- To develop a touch-free, user-friendly attendance mechanism.
- To enable real-time logging and centralized reporting.
- To build a scalable system deployable in schools, colleges, and offices.
- To ensure minimal setup with cost-effective tools and technologies.
- To provide analytical insights on attendance trends and user behavior.

IV. System Architecture

The system architecture comprises three main components:

1. **Frontend (Presentation Layer)**:

Developed using HTML5, CSS3, Bootstrap, and JavaScript, it provides login pages, QR scanning interface, and dashboards. The design is responsive, supporting desktops, tablets, and mobiles.

2. **Backend (Application Logic Layer)**:

Powered by Flask, it handles user authentication, QR code generation, scanning logic, and interaction with the database. The backend uses RESTful endpoints to

ensure secure communication and data exchange.

3. ****Database (Data Storage Layer)**:**

A MySQL database is used to store user details, attendance logs, and generated QR metadata. The database schema supports normalization, referential integrity, and indexing to speed up queries and ensure data consistency.

V. Module Description

1. ****User Registration**:** Admins register users by providing name, email, and role. The system generates a unique QR code for each user and stores it on the server.

2. ****QR Code Generation**:** QR codes are generated using the `qrcode` Python library, encoding user IDs. The code is saved as a PNG file and displayed to the user.

3. ****QR Code Scanning**:** OpenCV captures live feed from a webcam, detecting and decoding QR data. If valid, attendance is logged.

4. ****Attendance Logging**:** The Flask backend checks the decoded data, verifies the user in the database, and records the date/time stamp.

5. ****Admin Panel**:** The dashboard allows viewing reports, adding users, and exporting logs.

6. ****Security & Validation**:** Input fields are sanitized, and role-based access control is implemented to prevent unauthorized use.

VI. Database Design

The database includes the following normalized tables:

- `users` (user_id, name, email, password_hash, role)
- `qr_codes` (qr_id, user_id, qr_path)
- `attendance` (att_id, user_id, scan_time)

Proper use of foreign keys ensures integrity. Indexes on `user_id` and `scan_time` support faster queries. Auto-incrementing IDs maintain record uniqueness. Optional tables like `logs` and `audit_trail` may be added for analytics.

VII. Technologies Used

- PHP
- Flask
- OpenCV
- MySQL
- HTML5, CSS3, JavaScript

- Bootstrap
- QR CODE library
- Jinja2 Templates

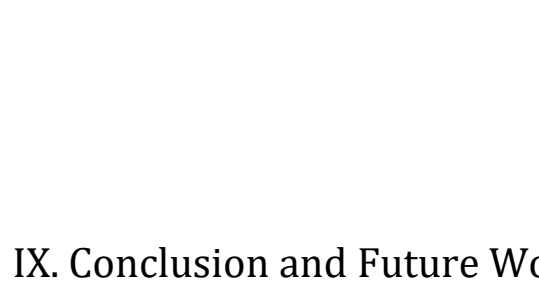
VIII. Results and Screenshots

Functional testing was conducted to evaluate usability, performance, and robustness. Sample test cases included:

- Scanning valid/invalid QR codes
- Verifying login access levels
- Testing UI responsiveness on different devices

Screenshots were captured for login pages, scan interface, and admin dashboard. Reports were successfully exported in CSV format.





IX. Conclusion and Future Work

The QR Code Attendance System proves to be an efficient, cost-effective, and scalable solution for modernizing attendance tracking. It simplifies the process, improves accuracy, and eliminates manual errors. The system's modular architecture allows easy maintenance and future upgrades. In the future, integration with facial recognition can enhance security. Deployment on cloud platforms will ensure scalability and real-time access. Additionally, a mobile application can offer convenience for scanning and record management. Implementing SMS or email notifications can further boost user engagement and transparency.

References

1. Flask Documentation - <https://flask.palletsprojects.com>
2. OpenCV Guide - <https://docs.opencv.org>
3. Python QR Code Generator - <https://pypi.org/project/qrcode>
4. MySQL Reference - <https://dev.mysql.com/doc>
5. Bootstrap - <https://getbootstrap.com>
6. QR Code Use Cases - IEEE Papers (2020-2024)
7. Web Development Stack Tutorials - W3Schools

