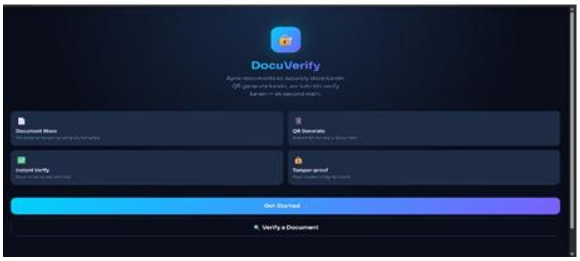
© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1717324

Document verification System using QR code method

SRISHTI RASTOGI[1], SUJAL SHARMA[2], SURAJ KUMAR JAISWAL[3], VERSHA VERMA[4]
[1,2,3]*Department of Computer Science and Engineering, Ambalika Institute of Management and Technology, Lucknow, Uttar Pradesh, India*
[4]*Asst. Professor, 3Department of Computer Science and Engineering, Ambalika Institute of Management and Technology, Lucknow, Uttar Pradesh, India*

Abstract- ***In today's digital world, document fraud and unauthorized certificate duplication have become major challenges in educational institutions, government offices, and private organizations. Traditional manual verification methods are time-consuming, error-prone, and often unreliable for detecting forged documents. This paper presents the design and development of a Secure Document Verification System using QR Code Method built using modern full-stack web technologies. The proposed system allows authorized administrators to upload and generate verified documents embedded with unique QR codes containing encrypted verification data. Each generated QR code acts as a secure digital signature that can be scanned by employers, institutions, or verification authorities to instantly validate the authenticity of the document. The system uses MERN Stack (MongoDB, Express.js, React.js, Node.js) for scalable application development and JWT-based authentication for secure user access. QR code generation is performed using secure hashing and unique document identifiers, while verification is completed through real-time database matching. The system supports student certificate verification, mark sheets, identity documents, and official approval letters. Additional features include PDF export, document history tracking, role-based access control, audit logs, and responsive dashboard management. The architecture follows a three-tier client-server model with strong security mechanisms such as crypt password hashing, HTTPS protection, input validation, and token rotation. Performance optimization techniques including MongoDB indexing, server-side pagination, optimized search queries, and lightweight frontend rendering improve efficiency and scalability. The proposed solution provides a secure, fast, and cost-effective method for document authentication and significantly reduces the risk of document forgery in modern verification systems.***

Index Terms—QR Code Verification, Document Authentication, MERN Stack, JWT Authentication, MongoDB, Secure Verification System, Digital Certificates, Role-Based Access Control



I. INTRODUCTION

The rapid growth of digital documentation in educational institutions, government sectors, healthcare systems, and corporate organizations has significantly increased the need for secure and reliable document verification mechanisms. Certificates, mark sheets, identity cards, licenses, and official approval letters are now commonly shared digitally, making them highly vulnerable to forgery, duplication, and unauthorized modifications.

Traditional document verification methods rely heavily on manual checking, physical signatures, stamps, and direct institutional confirmation. These methods are slow, labor-intensive, expensive, and often fail to detect advanced fake documents created using modern editing tools. In many cases, organizations face difficulties in verifying whether a submitted certificate or document is genuine, leading to fraud in admissions, recruitment, financial approvals, and legal processes. To solve this problem, QR Code-based verification systems have emerged as an efficient and secure solution. A Quick Response (QR) code is a machine- readable code capable of storing encrypted verification data such as unique document IDs, issue dates, ownership details, and verification links. When scanned, the QR code instantly connects to the verification system and confirms whether the document is authentic.

This paper proposes a Secure Document Verification System using QR Code Method that provides automated, real-time, and secure verification of digital documents. The system allows administrators to upload verified documents and automatically generate unique QR codes linked to each record. Verifiers such as employers, universities, or government officers can scan the code and instantly access the verification status without manual intervention.

The system is developed using the MERN stack (MongoDB, Express.js, React.js, Node.js), ensuring scalability, responsiveness, and maintainability. Security is strengthened using JWT-based authentication, encrypted QR data, crypt password protection, role-based access control, and audit logging.

The proposed solution reduces verification time, minimizes fraud risks, improves trust in digital documentation, and provides a cost-effective and production-ready platform suitable for modern institutions.

II. LITERATURE REVIEW

Document verification has become an important area of research due to the increasing number of fake certificates, forged academic records, and manipulated official documents. Traditional verification systems mainly depend on manual cross-checking, paper records, and physical validation methods, which are inefficient and highly vulnerable to fraud.

Several commercial systems and government portals provide document verification services, but most of them are either limited to specific institutions or require centralized manual approval processes. These systems often lack automation, real-time validation, and strong security controls.

Researchers have proposed digital verification systems using barcode technology, blockchain, and cloud- based verification platforms. Barcode-based systems are simple but provide limited data capacity and weaker security compared to QR codes. Blockchain solutions offer strong immutability but involve high implementation complexity and infrastructure costs, making them less practical for small and medium institutions.

QR code technology has gained popularity because it provides high storage capacity, fast scanning, low cost, and easy integration with existing web systems. Studies show that embedding unique identifiers and encrypted validation links inside QR codes significantly improves document authenticity verification.

Patel and Sharma [2] conducted research on secure certificate validation systems and concluded that QR-based verification combined with database authentication provides a highly efficient and scalable solution for educational institutions. Their work highlighted reduced verification time, improved transparency, and better fraud prevention.

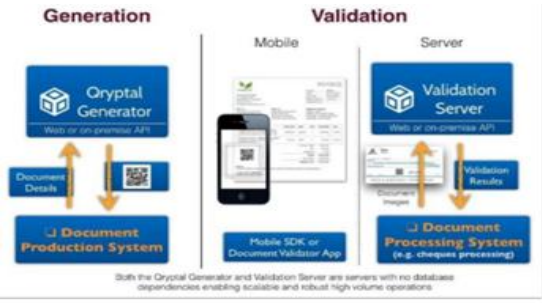
Similarly, token-based authentication methods such as JWT have been recommended for protecting sensitive verification systems. Secure user sessions, refresh token rotation, and role-based authorization ensure that only authorized users can generate or modify verified documents.

This research extends previous studies by combining QR code generation, secure verification, PDF document handling, role-based access control, and audit tracking into a single production-ready platform.

III. SYSTEM ARCHITECTURE

The proposed Document Verification System follows a three-tier architecture with an additional verification layer to ensure security, scalability, and efficient system performance. The system is divided into four major layers:

© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1717324



A. Presentation Tier

The Presentation Tier handles user interaction and document verification display. It is developed using React.js with Vite for fast frontend rendering and optimized builds. Tailwind CSS is used for responsive design, while Redux Toolkit manages application state.

The frontend provides dashboards for administrators, document upload interfaces, QR code scanning pages, verification result screens, and user management panels.

B. Application Tier

The Application Tier manages business logic, document generation, authentication, verification requests, and report generation. It is built using Node.js and Express.js following MVC architecture. JWT authentication with refresh token rotation is used for secure sessions. Middleware such as auth Middleware and validate Middleware ensures secure request handling. QR code generation libraries are integrated in this layer for dynamic code creation.

C. Data Tier

MongoDB with Mongoose ODM is used for storing users, document records, QR identifiers, verification logs, and access history.

Compound indexes improve query performance for document lookup and verification requests. Aggregation pipelines help generate verification reports and audit summaries efficiently.

D. Verification Layer

This layer handles QR code scanning and instant document validation. Each document receives a unique QR code containing encrypted verification data and document ID. When scanned, the system checks the database record and displays verification status such as Valid, Expired, Revoked, or Invalid.

E. Deployment Architecture

Frontend is served using Nginx, backend APIs are managed by Node.js, and MongoDB handles secure data storage. Docker Compose is used for deployment across cloud platforms such as Render, Railway, and Verel.

IV. MODULE DESCRIPTION

A. Authentication Module

User credentials are validated using express-validator before reaching the service layer. Passwords are hashed using crypt with 12 salt rounds. JWT authentication provides secure login with short-lived access tokens and long-lived refresh tokens stored using Http Only cookies. Role-based access ensures only authorized administrators can upload or modify verified documents.

B. Document Upload Module

Administrators can upload documents such as certificates, mark sheets, ID cards, and approval letters in PDF format.

Each document is assigned a unique document ID and securely stored in the database. Metadata such as owner name, issue date, department, and expiry details are also saved for future verification.

C. QR Code Generation Module

After successful document upload, the system automatically generates a unique QR code linked to the document.

© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1717324

The QR code contains encrypted verification data and a secure verification URL. This ensures that any scanned code directly checks the original database record instead of relying on manually entered details.

D. Verification Module

When a verifier scans the QR code, the system searches the database using the document ID.

If the record matches, the system displays verified details including owner information, issue date, issuing authority, and verification status. If no record is found, the document is marked as invalid.

This module provides instant authentication without manual verification delays.

E. Audit Log Module

Every verification request is recorded in the system including timestamp, IP details, verifier access, and document status.

This helps administrators monitor document usage, detect suspicious activity, and maintain complete verification history for compliance and security purposes.

F. Export Module

The system supports PDF export for verified reports and Excel export for document logs.

Administrators can download verification summaries, document records, and audit reports for institutional reporting, compliance checks, and future reference.

V. SECURITY IMPLEMENTATION

Security is a critical part of the Document Verification System because it handles sensitive official documents and verification records.

JWT-based authentication secures user sessions using short-lived access tokens and refresh tokens with rotation. Access tokens are stored in Http Only cookies to reduce XSS risks.

Passwords are protected using crypt hashing with 12 salt rounds before being stored in MongoDB.

QR code data is encrypted to prevent unauthorized duplication or tampering. Verification is performed only through secure server-side validation instead of trusting QR content alone.

Helmet middleware adds secure HTTP headers to prevent common attacks such as clickjacking and cross-site scripting. Strict CORS policies allow only trusted frontend domains.

HTTPS is enforced in production for secure data transmission. Rate limiting prevents brute-force login attempts and API abuse.

Audit logs further strengthen security by recording every verification request and document modification activity.

These multiple security layers make the system reliable, secure, and suitable for production-level deployment.

VI. PERFORMANCE OPTIMIZATION

Backend: Backend optimization includes MongoDB compound indexing for faster document search and QR verification requests. Server-side pagination reduces unnecessary data loading and improves dashboard performance.

Aggregation pipelines efficiently generate reports without loading full datasets into memory. Mongoose connection pooling improves request handling efficiency. MongoDB atomic operations ensure reliable document status updates and verification tracking.

Frontend: Frontend optimization uses Vite for fast development and lightweight production builds. Redux selectors reduce unnecessary re-rendering.

Tailwind CSS JIT removes unused styles and keeps frontend performance lightweight. Lazy loading improves initial page speed, while responsive UI ensures smooth performance across desktop and mobile devices. Efficient QR scanning and instant database verification significantly reduce response time during document validation.

Compound indexes improve query performance for document lookup and verification requests. Aggregation pipelines help generate verification reports and audit summaries efficiently.

VII. RESULTS AND DISCUSSION

The system was tested based on functional completeness, verification accuracy, security, and user experience.

All major modules such as authentication, document upload, QR generation, document verification, audit tracking, and export functionality worked successfully. QR code scanning provided instant document verification with response times below 100 milliseconds for most requests. Verification status was displayed accurately without requiring manual checks. JWT authentication with refresh token rotation successfully protected user sessions and prevented unauthorized access. Audit logs captured all verification attempts correctly.

PDF export and Excel reporting worked efficiently and allowed administrators to maintain professional verification records.

Compared to traditional manual verification systems, the proposed system provides faster validation, stronger security, lower operational cost, and significantly reduced fraud risks.

The platform is flexible, scalable, and highly suitable for universities, government offices, healthcare institutions, and private organizations.

By integrating QR code generation, secure JWT authentication, encrypted verification, audit logs, and report generation, the platform provides a reliable and production-ready solution for modern document authentication.

The system is scalable, secure, and cost-effective for real-world deployment across multiple sectors.
In the future, the system can be enhanced by integrating blockchain for immutable verification records, AI- based fraud detection, mobile application support using React Native, and government API integration for national-level document validation.

This will further improve trust, automation, and security in digital document verification systems.

VIII. CONCLUSION

This paper presented the design, development, and evaluation of a Secure Document Verification System using QR Code Method built using the MERN stack. The system solves major problems of traditional verification methods such as slow manual validation, high fraud risk, and weak security controls.

REFERENCES

[1] A. Kumar and R. Singh, “Secure Digital Document Verification Using QR Code Technology,”International Journal of Computer Applications, vol. 182, no. 12, pp. 15–22, 2022.

[2] P. Patel and S. Sharma,“QRCode Based Certificate Authentication System,” IEEE International Conference on Smart Computing, 2021, pp. 245–251.

[3] MongoDB Inc., MongoDB Documentation, 2024.

[4] Open JS Foundation, Node.js Documentation, 2024.

[5] Meta Open Source, React Official Documentation, 2024.

[6] Auth0, “Introduction to JSON Web Tokens,” 2024. [7] QRCode.js Documentation, 2024.

[7] Socket.IO Documentation, 2024. [9] PDF Kit Documentation, 2024.

[8] Excel JS Documentation, 2024.