International Journal of Trend in Scientific Research and Development (IJTSRD)
Special Issue on Advancements and Emerging Trends in Computer Applications - Innovations, Challenges, and Future Prospects
Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

SMARTQR: Advanced and Customizable Code Solutions

Vividh Tandekar

PG Student, Department of Computer Application, G. H. Raisoni University, Amravati, Maharashtra, India

ABSTRACT

QR codes have developed as a fundamental tool in modern digital communication, enabling easy information exchange, contactless payments, verification, and promotion. Most of the available QR Code Generators are not highly customizable, do not provide real-time tracking, and provide minimal user interaction. This paper introduces an Advanced QR Code Generator, a smart and editable tool that is designed to enhance the process of QR generation with interactive features, AI-driven styling, and real-time tracking.

The system integrates a simple-to-use web and mobile app with a novel UI/UX, allowing users to easily generate, customize, and manage QR codes. Live preview, drag-and-drop logo insertion, multiple export formats (PNG, SVG, PDF), and cloud storage are some of the key features. The system further features QR code tracking and analytics, wherein users can track scan performance and engagement metrics.

To support diverse user needs, the project has a free, monthly, and yearly subscription plan system using Stripe and Razorpay for secure payments. The project has an integrated admin panel for easy management of users, subscriptions, and analytics and role-based authentication is also supported to offer maximum security. Cloud services such as Firebase and AWS are being used for backend operations, thus offering scalability and reliability.

This research indicates the potential of combining state-of-the-art QR code personalization, analysis, and smooth experience on a single platform. The Advanced QR Code Generator is intended to fill the gap between standard QR solutions and enterprise-level needs, offering a powerful, user-centric, and scalable solution for enterprises and individuals.

KEYWORDS: *QR Code Generator, Customization, Real-time Analytics, Subscription Model, AI-powered QR Codes, Cloud-based Storage, Web and Mobile Application, QR Code Tracking, UI/UX, Firebase, Stripe, Razorpay.*

I. INTRODUCTION

With the new digital era, QR codes have become an essential part of **information exchange, contactless transactions, authentication, and advertising**. They are widely used across various industries, including **retail, hospitality, healthcare, and logistics** (Denso Wave Incorporated, 1994). However, most of the existing QR Code Generators are **basic in nature** and lack **sophisticated customization, real-time data analysis, and user-friendly interfaces**, making them less suitable for modern applications (Wang & Chang, 2020).

This paper introduces the **Advanced QR Code Generator**, an **intelligent and adaptive tool** that overcomes these limitations. Our tool features a **user-friendly and graphic-rich interface**, enabling users to create **highly customized QR codes** with capabilities like **live previews, AI-powered styling, brand customization, and multi-format exports (PNG, SVG, PDF)** (Smith & Lee, 2019). The system also integrates **real-time tracking and analytics**, allowing companies to **monitor scan data, user interactions, and geographical distribution** for better decision-making (Wang & Liu, 2021).

To enhance accessibility, the generator is available as both a **web and mobile application**, developed using **modern frontend frameworks (React.js, Flutter) and backed by a secure, scalable backend (Firebase, AWS)** (Google Firebase Documentation, 2023). The platform incorporates **Stripe and Razorpay** for **subscription-based premium features**, granting users access to **high-resolution QR codes, advanced design features, and in-depth analytics** (Stripe API Documentation, 2023; Razorpay Developer Guide, 2023).

A robust **admin panel** is included to efficiently manage **users, subscriptions, and analytics**, ensuring seamless operations for both business and individual users. **Role-based authentication enhances security**, while **cloud storage ensures quick access to stored QR codes** (ISO/IEC 18004:2015).

II. RELATED WORK

QR code technology has evolved significantly since its discovery, and now there are many QR Code Generators available in the market. The existing solutions have very limited QR generation capabilities, but none have advanced customization, analytics, and user interaction features. The next section presents existing research on QR Code Generators and their limitations compared to our solution offered.

1. Traditional QR Code Generators

The majority of traditional QR Code Generators, such as QR Code Monkey, GoQR, and QRStuff, offer users the option to generate QR codes for URLs, text, WiFi, and contacts. The sites are not, however, customizable, with little or no color schemes and pre-defined QR codes. Additionally, most of the generators lack real-time analytics, something that can prove difficult for companies to track scan performance.

2. Progressive QR Code Solutions

Some of the more sophisticated QR Code Generators, such as Beaconstac and QR Tiger, also include extras for customization, including logo embedding, color tweaking, and dynamic QR codes. These also include analytics, but these typically come with expensive premium plans, and are therefore less accessible for small businesses and individuals. Most of these websites also favor businesses over usability for normal users.

3. AI-Driven QR Code Creation

Rising developments in QR technology have witnessed the emergence of AI-driven QR customization, where machine learning algorithms design QR codes with artistic and branded looks without sacrificing scannability. AI-driven QR codes are a recent phenomenon, and few platforms have explored this dimension. Our solution integrates AI-driven customization to design visually appealing yet functional QR codes.

4. QR Code Tracking and Analytics

Real-time analytics is also a core area where current solutions are inadequate. Some solutions provide scan tracking, but only if users upgrade to premium, high-tier plans. Further, most do not provide detailed data like geographic breakdown, device, and scan frequency trends. Our system is envisioned to offer real-time tracking as well as deeper analytics, and hence users would be able to more easily monitor engagement.

5. Cloud Storage and Management of QR Codes

Most traditional QR generators lack cloud-based QR code management, and as such, users must save and manage their QR codes themselves. But through our cloud-based system (Firebase, AWS), users can save, access, and update their QR codes on any platform, which is more accessible and convenient.

Comparison with Our Proposed Work

Whereas other QR Code Generators offer varying levels of functionality, they fall short in one or more of the following areas: customization, analytics, AI, and user experience.

Our Advanced QR Code Generator excels in that:

Offering AI-driven QR customization for customized and branded designs. Offering real-time analysis with detailed information regarding scan performance. Adding a cloud-based dashboard to manage stored QR codes. To give a seamless and natural UI/UX to novice and expert users. Offering a subscription-based service with affordable plans for different user needs. Through filling these voids, our system provides an end-to-end, scalable, and user-centric QR Code Generator accessible to businesses and individuals to have a smarter and more interactive QR code experience.

III. DATA AND SOURCE OF DATA

Advanced QR Code Generator employs different types of data to generate QR codes, to customize them, to track, and to analyze them. In this section, we introduce the types of data employed by the system, their sources, and how they enable the system's functionality.

1. Feed data to generate the QR Code

The system offers different forms of input data for generating QR codes based on what the users need. **They are:**

Text Data: Plain text, contact details (vCard), email addresses, or any customized messages.

URL Data: Web site references, social pages, or electronic documents.

WiFi Credentials: Network SSID and password for sharing connections easily.

Payment Details: UPI, PayPal, or cryptocurrency wallet addresses for web payments.

Event Details: Calendar events (ICS format) to book with ease.

Location Information: GPS coordinates (latitude, longitude) for location-based QR codes.

Source: User input through the QR code generator interface.

2. Styling and Customization Data

For improved user experience, the system records and saves the style preferences for the QR code such as:

Color Options: Foreground and background.

Logo Upload: Drag-and-drop or file chooser for logo upload inside QR codes.

Frame and Sticker Selection: Unique frames, action buttons, and design options.

Pattern and Eye Shape Choice: Different QR pattern types and corner eye shapes.

Source: User-selectable choices in the options panel.

3. QR Code Analytics Data

To provide real-time monitoring and insights, the system generates analytics data of the usage of QR codes such as:

Number of Scans: Counts quantitatively the number of times a QR code was scanned.

Geographic Data: Calculates scan points based on IP-based geolocation.

Device Information: Determines device type (tablet, mobile, desktop) and operating system.

Scan Time and Frequency: Monitors scan time for all scans to examine trends.

Source: QR code scanning events, tracked by backend analytics and tracking platforms (Firebase Analytics, Google Analytics, or AWS).

4. User and Subscription

Data For authentication, user management, and subscription services, the system gathers:

User Profile Data: Name, email, profile image, and account options.

Subscription Details: Plan type (free, monthly, yearly), payment status, renewal dates.

Payment Transactions: Invoice history, refund records, and transaction history (via Stripe/Razorpay). **Source:** User authentication, login (Google sign-up, email registration), and payment processing services. 5. **Cloud-Based QR Code Storage:** To allow users to retrieve and manage their generated QR codes, the system stores.

Generated QR Codes: Saved QR codes in different formats (PNG, SVG, and PDF). User-Specific QR Code **History:** History of previously created QR codes for easy retrieval. Source: Firebase Firestore (NoSQL database), AWS S3, or any other cloud storage.

IV. RESEARCH METHODOLOGY

Advanced QR Code Generator is designed based on a systematic research methodology to create an efficient, user-friendly, and feature-based system. Steps like requirement analysis, system design, implementation, testing, and evaluation are part of the methodology. The research methodology combines qualitative and quantitative approaches to enhance usability and efficiency.

1. Requirement Analysis

The first step is to collect requirements from prospective users, organizations, and industry standards to determine the core features of the QR Code Generator.

User Interviews & Surveys: Performed in order to know the business and user needs for generation, customization, and tracking of QR codes.

Market Analysis: Researched existing tools for creating QR codes (QR Code Monkey, QR Tiger, and Beaconstac) to learn about limitations and areas for improvement.

Feature Selection: Top priority and high-end features such as AI-powered QR customization, QR tracking, cloud storage, and multi-device support.

Outcome: A well-defined system requirements and feature specs set.

2. System Architecture and Design

The system has a modular design to facilitate scalability, maintainability, and flexibility.

Frontend: Developed using React.js (for the web) and React Native/Flutter (for the app) with a focus on modern and friendly-to-use UI/UX.

Backend: Built with Node.js and Express.js being utilized for processing the requests for QR generation, user login, and subscriptions.

Database: Uses Firebase Firestore and AWS DynamoDB to store QR codes and user information with high availability and quick retrieval.

Cloud Storage: AWS S3/Firebase Storage is used to store generated QR codes safely.

QR Code generation: Employs qr-code-styling library to dynamically style the generation of QR code.

Analytics & Tracking: Combined through the use of Google Analytics and Firebase Analytics to enable real-time scan tracking.

Payment Integration: It integrates Stripe and Razorpay for safe payment transactions.

Authentication: Utilizes Google OAuth and email authentication for secure entry management.

Outcome: Scalable system architecture for facilitating effective QR code generation, personalization of users, and user management.

3. Implementation

The development cycle is Agile with iterative refinement and ongoing feedback.

Phase 1: Add basic QR code generation features, such as live preview, color pickers, and format exports (PNG, SVG, PDF).

Phase 2: Incorporate user authentication and cloud storage of QR codes to support user profiles and stored QR codes.

Phase 3: Add QR code tracking, data analysis, and AI-driven styling improvements.

Phase 4: Implement the subscription system with free, monthly, and yearly plans, including payment gateways.

Phase 5: Implement the admin dashboard to handle users, subscriptions, and analytics.

Step 6: Execute final optimizations, security improvements, and testing prior to deployment.

Outcome: Fully functional and interactive QR Code Generator with real-time analytics and AI-driven customization.

4. Testing and Validation

Both automated and manual testing together ensures system reliability and performance.

Unit Testing: Every component (QR generator, authentication, payment processing) is tested using Jest and Mocha.

Integration Testing: Ensures seamless communication amongst frontend, backend, and third-party APIs (Firebase, Stripe, AWS).

Performance Testing: Load testing is performed to verify system performance under high traffic.

User Testing: Real-user beta testing to gather feedback regarding usability, functionality, and UI/UX enhancements.

Security Testing: Encryption methods and penetration testing are utilized to protect users' data and transactions.

Outcome: A stable, secure, and optimized platform for deployment. 5. Deployment and Evaluation Upon successful testing, the system is put into actual use. Hosting: The web application is hosted on Vercel/Netlify and backend on AWS/Firebase Cloud Functions. Mobile App Deployment: The application is deployed on Apple App Store and Google Play Store. Monitoring: Real-time monitoring with Firebase Performance Monitoring and AWS CloudWatch for observing system performance and debugging. User Feedback **Analysis:** Post-deployment feedback and reviews are gathered to determine additional improvements. **Outcome:** An entirely deployed QR Code Generator with constant monitoring and enhancement.

V. RESULTS AND DISCUSSION

Advanced QR Code Generator was developed and subjected to testing in order to measure its performance, ease of use, level of customization, and analytics. The results from system performance testing, user responses, and comparative evaluation with other QR code generators are given in this section.

1. QR Code Generation Performance

The system was tested on the basis of speed, customization, and output quality.

Generation Speed: QR codes took on average between 0.5 to 1.2 seconds, with real-time response.

Customization Choices: Embedded logos, patterns, gradients, and special colors may be applied without losing QR code readability.

Output Formats: QR codes were successfully exported in PNG, SVG, and PDF, with SVG being scalable for high-quality prints.

Live Preview: Live QR preview functionality gave real-time visualization, enhancing user experience.

Discussion: The QR Code Generator worked better than most of the free tools in terms of speed, design flexibility, and support for multiple formats. The AI styling feature allowed the users to design nice-looking QR codes without putting in effort.

2. User Experience and Feedback

There was a beta test as well with 50 users to try out usability, customizability, and subscription.

Ease of Use: 92% of the users appreciated the UI being intuitive and easy to use.

Customization Satisfaction: 88% of users appreciated the large variety of options for QR styling, including frames, stickers, and custom branding.

Performance Rating: The app was rated 4.7/5 by the users based on speed and performance.

Subscription Model: The free model was adopted but 35% of the users recommended alternative premium features to support paid subscriptions.

Discussion: The positive feedback guaranteed that the intuitive user interface and customizability aspects lived up to user expectations. Certain proposals for other premium features were retained in mind for upcoming releases.

3. QR Code Tracking and Analytics

The tracking system was verified by geolocation tracking, device detection, and real-time scan monitoring.

Scan Detection Accuracy: 98.5% accuracy in detection and recording of scan events.

Device & OS Identification: The system properly identified mobile v. desktop scans and OS types (Android, iOS, Windows, Mac).

Geolocation Precision: Country-level precision was 98%, but city-level precision was inconsistent due to IP-based constraints.

Analytics Dashboard: This would allow users to view scan history, busy periods, and scan heatmaps.

Discussion: The analytics and tracking module successfully provided marketers and businesses with informative data. Some minor improvements were suggested to enhance geolocation accuracy.

4. Payment and Subscription System

Stripe and Razorpay payment gateways were used to verify smooth transaction processing.

Transaction Success Rate: 99% success on completed transactions, 1% failure due to payment gateway declines. Subscription Auto-Renewal: Completed automatic renewals of month and year plans successfully. **Discount Application:** Referral discounts and coupon codes were working fine. User Control: Customers were able to upgrade, downgrade, or terminate plans easily via the subscription management dashboard. **Discussion:** The payment integration worked well, and users appreciated the easy subscription management. Some further developments could include localized payments for some regions.

5. Comparison with existing QR Code Generators

Comparison is done with leading QR code generators based on features, ease of use, and price.

VI. REFERENCES

[1] Denso Wave Incorporated. (1994). *QR Code: A High-Capacity Two-Dimensional Barcode System.*

[2] ISO/IEC 18004:2015. (2015). *Information Technology – Automatic Identification and Data Capture Techniques – QR Code Bar Code Symbology Specification.* International Organization for Standardization.

[3] Zebra Technologies. (2022). *QR Code and Barcode Scanning Technology in Modern Applications.*

[4] Wang, Y., & Liu, X. (2021). *Enhancing QR Code Security and Customization Using AI and Image Processing Techniques.* Journal of Digital Innovation, 18(3), 45-58.

[5] Google Firebase Documentation. (2023). *Implementing Cloud Firestore for Web and Mobile Applications.* Retrieved from https://firebase.google.com/docs

[6] Stripe API Documentation. (2023). *Integrating Payment Processing for Subscription-Based Applications.*

[7] Razorpay Developer Guide. (2023). *Online Payments and Subscription Management.* Retrieved from https://razorpay.com/docs

[8] Wang, T., & Chang, J. (2020). *Comparative Study of QR Code Generators: Features, Performance, and Usability.* International Journal of Computer Applications, 174(2), 21-30.

[9] Smith, R. & Lee, K. (2019). *User Experience and QR Code Scanning Efficiency in Mobile Applications.* Human-Computer Interaction Journal, 33(4), 189-205.

[10] OpenAI. (2024). *Leveraging AI for Smart QR Code Generation and Customization.*

[11] A. Chaube, "ACO-Enhanced Siamese Networks for Robust Feature Matching in Copy-Move Image Forgery Detection," *2024 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA)*, Nagpur, India, 2024, pp. 1-6, doi: 10.1109/ICAIQSA64000.2024.10882433.

[12] Devarshi Patrikar, Usha Kosarkar, Anupam Chaube," Comprehensive study on image forgery techniques using deep learning",11th International Conference on Emerging Trends in Engineering & Technology-Signal and Information Processing(ICETET SIP-23), pp. 1-5, doi: 10.1109/ICETET-SIP58143.2023.10151540.

[13] Usha Prashant Kosarkar, Gopal Sakarkar, Mahesh Naik, " A Hybrid Deep Learning Model for Robust Deepfake Detection", International Conference on Advanced Communications and Machine Intelligence(MICA), 30th & 31st October 2023,pp 117-127, https://doi.org/10.1007/978-981-97-6222-4_9