2024 1st International Conference on Advances in Computing, Communication and Networking (ICAC2N) | 979-8-3503-5681-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICAC2N63387.2024.10895390

Developing a Python-Based QR Code Generation System

Priyanka Kaushik
Dept. of CSE
Chandigarh University
Chandigarh, India
kaushik.priyanka17@gmail.com
https://orcid.org/0000-0003-4766-8772

Shamna Parveen S
Dept. of CSE
Dhanlaxmi Srinivasan College Of Engineering
Coimbatore, India
shamnaparvin@gmail.com

Shalu
Dept. of Computer S. & Technology
Manav Rachna University
Faridabad, India
singhshalu2609@gmail.com

Sucheta Panda
Dept. of Computing and IT
Usha Martin University
Jharkhand, India
suchetapanda10@gmail.com

Swati Agarwal
Dept. of Mathematics
Usha Martin University
Jharkhand, India
swati.agarwal@mangalayatan.edu.in

P.R.Sukanya Sridevi
Dept. of Computer Science
Meenakshi A.H.E.R
Chennai, India

***Abstract*—This project entails importing necessary modules, creating the main window, and obtaining user inputs such as text or URL, the location to store the QR code, the name of the QR code, and its size. The paper delves into the underlying principles of QR code technology, covering aspects such as its structure, error correction capabilities, and applications across various industries. It highlights how Python serves as a powerful, flexible, and user-friendly tool for generating QR codes. This project is designed to be utilized by software developers, researchers, and enthusiasts, enabling them to gain a comprehensive understanding of QR code generation and to create their own customized QR code generation applications. By providing a detailed technical guide for QR code generation using Python, this research paper makes a significant contribution to the field of QR code technology.**

***Index Terms*—Python, QR code, Generation**

I. Introduction

A. Problem Definition

The aim of this project is to develop a software tool that can quickly and easily generate QR codes. QR codes have become increasingly popular across various fields, such as advertising, marketing, and inventory management. The primary objective is to create a user-friendly tool that allows users to efficiently generate QR codes from various types of input data, including URLs, plain text, email addresses, and phone numbers. The tool should also offer customization options like changing the color, size, and shape of the QR code. Additionally, the output should be available in multiple formats, such as PNG, JPG, and SVG. Ultimately, the goal is to create an efficient and versatile QR code generator that caters to a wide range of user needs.

B. Problem Overview

A QR code generator is a software application designed to create QR codes quickly and easily. QR codes are two-dimensional barcodes that can contain various types of information, such as URLs, text, email addresses, and phone numbers. This tool automates the process of generating QR codes from input data, making the process more efficient and less error-prone.

Manually creating QR codes can be time-consuming and susceptible to errors. A QR code generator addresses these issues by automating the process, saving time, and minimizing mistakes. These tools are widely used in various applications, including advertising, marketing, and inventory management.

One of the main challenges in developing a QR code generator is ensuring that the generated QR codes are accurate, reliable, and secure. It is crucial to validate input data to ensure its suitability for encoding into a QR code. The generated QR codes should be tamper-proof and resistant to duplication. Additionally, the tool must be user-friendly, with an intuitive interface that guides users through the QR code generation process.

To address these challenges, the QR code generator should be built with a robust algorithm that meets industry standards. It should include error-checking and validation mechanisms to ensure the accuracy of input data. Security features such as encryption and authentication should be integrated to make the QR codes secure and tamper-proof. Overall, the QR code generator is an essential tool for various applications, and it must be designed and implemented carefully to meet user needs effectively.

C. Hardware Specifications

The hardware requirements for a QR code generator depend on its implementation and expected workload. Generally, the following minimum hardware specifications are recommended:

- Processor: A modern dual-core processor, such as an Intel Core i3 or AMD Ryzen 3.

- RAM: At least 4GB of RAM for smooth operation.

- Hard Drive: Sufficient storage space for the program and generated QR codes.

- Display: A monitor with a minimum resolution of 1024x768 for optimal display. -Input Devices: A keyboard and mouse or touchscreen for user interaction. - Camera: A high-quality camera for scanning existing QR codes, if needed. - Internet Connection: Required if the QR code generator will be used online.

Most modern computers should be able to handle the workload of a QR code generator without any issues

D. Software Specifications

The software requirements for developing a QR code generator vary depending on the programming language and development tools used. The general software requirements include:

- Operating System: Compatible with Windows, macOS, or Linux.

- Integrated Development Environment (IDE):** Tools like Visual Studio Code, Eclipse, or IntelliJ IDEA for writing, debugging, and testing code.

-Programming Language: Python is recommended for this project.

- QR Code Generator Library: Libraries such as qrcode, PyQRCode, ZXing, or qrcode.js.

- Version Control: A system like Git for managing source code and collaboration.

- Testing Framework: Tools like Pytest or JUnit for testing the code.

- Documentation Tool: Tools such as Sphinx or Doxygen for creating user and developer documentation.

These software requirements are straightforward and most developers should already have the necessary tools and libraries installed in their development environment.

II. LITERATURE SURVEY

Papathanasiou et al. (2024) introduced the ”BEC Defender,” a novel QR code-based methodology aimed at preventing Business Email Compromise (BEC) attacks. Their study, published in Sensors, outlines the mechanism and effectiveness of using QR codes to authenticate business communications, thereby enhancing security protocols. The methodology has been validated through various simulations showing a reduction in phishing success rates, providing a robust defense against increasingly sophisticated email scams [1]. Prakash (2024) explores the influence of big data analytics on organizational performance in his article in the International Journal for Global Academic & Scientific Research. Through a comprehensive analysis, Prakash discusses how data-driven strategies can transform decision-making processes and improve operational efficiencies. The study asserts that organizations integrating big data tools into their operations witness significant enhancements in productivity and decision-making accuracy, thus impacting overall performance positively [2]. Vavilova et al. (2022) focused on the development of applications for generating and reading QR codes specifically tailored for the Micro1 model, as discussed at the 2022 conference on Systems of Signals Generating and Processing in the Field of on Board Communications. This research emphasizes the technical specifications and enhancements needed to optimize QR code utilization in communication systems onboard vehicles, which could improve both functionality and safety in automotive systems [3]. In their study published in the same journal as Prakash, Rathore and Rathore (2024) delve into the application of machine learning techniques to predict employee turnover and performance. By leveraging historical data, their models enable HR departments to identify potential leavers and high performers, thereby informing recruitment and retention strategies. This predictive capability could revolutionize human resource management by providing more targeted and effective interventions for employee management [4]. Lastly, Chen et al. (2023) address a common issue in QR code usage—variable lighting conditions that affect scan reliability. Their research, published in Electronics, presents an adaptive binarization method that adjusts to dynamic illumination, ensuring QR codes are legible in a wider range of lighting environments. This advancement supports the broader application of QR technology in diverse settings, enhancing usability and access [5]. Al Malki (2023) explores the sustainability challenges faced by Small and Medium Enterprises (SMEs), emphasizing the importance of incorporating sustainable practices to ensure long-term growth. The review identifies key barriers such as financial limitations, lack of expertise, and regulatory challenges, advocating for enhanced support systems to facilitate sustainable development in SMEs [6]. The work of Wang et al. (2023) introduces an innovative anti-counterfeiting QR code that integrates texture-hidden features for enhanced security. Their research presents a novel authentication method that leverages convolutional neural networks (CNNs) to discern authentic QR codes from forgeries, offering significant implications for digital security in various sectors [7]. Kuraishi (2023) discusses the crucial role of analytics in the supply chain management sector in Lithuania, highlighting how big data and AI can optimize operations, reduce costs, and enhance decision-making processes. This study provides valuable insights into the integration of technology in traditional industries, promoting efficiency and strategic planning [8]. Yang et al. (2023) present a CNN-based system for reading QR codes on packages using unmanned aerial vehicles (UAVs). This technology significantly improves the efficiency and accuracy of package handling and tracking, demonstrating the potential of integrating AI with UAV technology for logistic applications [9]. The evolution and application of QR code technology have seen significant

advancements in recent years, marked by innovations that enhance both functionality and efficiency. Chen and Ti (2021) explore the design of a multi-purpose image-based QR code, introducing a versatile approach that could potentially support various applications in digital interactions and security (Chen & Ti, 2021). Their work represents a substantial leap in making QR codes more adaptable to different uses, highlighting the ongoing development in QR code technology [10]. A. Prakash (2024) explores the application of blockchain technology in enhancing supply chain management by increasing transparency and efficiency. The paper details how blockchain's decentralized nature can solve traditional challenges like fraud, inefficiency, and lack of transparency in supply chains. It highlights successful case studies across various industries and discusses the automation of processes through smart contracts. Despite challenges like the need for substantial technological investment and the lack of regulatory frameworks, Prakash optimistically predicts the growing integration of blockchain in supply chain operations as technological and regulatory landscapes evolve [11]. In parallel, the integration of QR codes in complex systems is being enhanced through sophisticated image recognition techniques. Liu et al. (2024) propose a novel data matrix code recognition method that employs an L-shaped dashed edge localization using a central prior, which significantly improves the accuracy and reliability of QR code scanning under varied operational conditions (Liu et al., 2024). This method addresses the challenges posed by conventional QR code scanners, especially in environments where quick and precise reading is required [12]. Further extending the scope of QR code utility, Anderson (2024) introduces QRGB+, a Python-based QR code generator that utilizes RGB color channels. This development is aimed at enhancing data capacity and extending the functionality of QR codes beyond black and white encoding (Anderson, 2024). This unpublished work suggests potential new pathways for increasing the information density and aesthetic variability of QR codes [13].

Beyond traditional applications, QR codes are being explored for their utility in secure and efficient systems. Sutheebanjard and Premchaiswadi (2010) and Tiwari (2016) both discuss the development of QR code generators, with Tiwari specifically focusing on their role in information technology environments (Sutheebanjard & Premchaiswadi, 2010; Tiwari, 2016). These studies emphasize the adaptability of QR codes in various IT applications, from security enhancements to streamlined data management processes [14] [15]. The development and application of QR codes in various domains highlight significant advancements in data encoding and security. Anderson (2024) introduces a novel application, "QRGB," which enhances the information density of QR codes through a 3-in-1 method that combines QR code generation with an additive color generation method (RGB) using Python programming [16]. This approach not only increases the capacity of QR codes but also extends their usability in complex applications. Additionally, Abdul-Jabbar and Farhan (2023) address security concerns specific to the healthcare sector by developing a secure QR-code generation technique that promises to safeguard sensitive medical data [17]. The segmentation of bank customers using advanced machine learning techniques has been explored to enhance customer service and targeted marketing. Rathore et al. (2024) conducted a segmentation study using recurrent neural networks (RNN), demonstrating effective categorization based on customer behavior and transaction patterns [18]. Concurrently, Sharma et al. (2024) utilized the K-means clustering algorithm in combination with the RFM (Recency, Frequency, Monetary) model to partition customers, which aids in identifying potential avenues for revenue growth within next-generation banking services [19]. The integration of machine learning techniques in SMART environments to tackle complex societal issues represents a crucial area of research. Kaushik et al. (2024) explored the application of ensemble learning and long short-term memory (LSTM) networks to combat hate speech in SMART environments. Their approach underscores the potential of hybrid models in enhancing the accuracy and responsiveness of predictive systems in real-time applications [20]. In the realm of smart cities, improving traffic management through technological integration is vital for enhancing urban mobility and safety. Sikarwar et al. (2024) investigated the use of OpenCV and Internet of Things (IoT) technology to empower lane management. This study demonstrates how integrating computer vision with IoT can significantly improve traffic flow and reduce congestion in urban areas [21].

III. PROBLEM FORMULATION

Here, the problem formulation is required as: We need to formulate a program for developing a QR code generator as follows: Given an input data, the problem is to design and implement a

A software tool generating a QR code, then easily machine-scanned by camera or any other input device; the QR code must encode accurately the input data. Besides, the generated QR code has to be resistant and tamperproof to duplication. It should also be able to provide options for customizing a QR code that modulates its color, size, and even shape of the code. The tool should offer different format options, like PNG, JPG, or SVG, for yielding a generated QR code. Basically, design of a robust algorithm that generates QR codes up to standard—interested in embedding iterative notions of error checking and validation, which would give assurance that the input data is correct enough or suitable for encoding. It should also capture security features in which the code is assured secure and tamper-proof; this basically involves encryption and authentication. It must also possess a user-friendly interface that would assist or help the user in generating a QR code and further customize said code . Output shall be available to users in various formats, suiting varied needs. The generated QR codes have to be tested in order for it to prove the same would be easily readable by a camera or any other input device in proof of the efficiency of the solution. The code must also be validated such that it validates the input which is typed into it, and it exactly develops a code which, by design, is proof

against tampering and duplication. Furthermore, user interface testing has to be conducted to make it user-friendly and easy to work in. The overall testing process must also include analysis regarding the running of the tool for different data input and customization options to ensure that this tool works efficiently with respect to all prospects.

IV. OBJECTIVES

The following are the major objectives for a QR code generator:

Generate QR codes quickly and easily: The primary objective of a QR code generator is to provide an efficient and automated way to generate QR codes. The tool This should allow the creation of a QR code with speed and ease and without requiring any special technical knowledge . Input data must be encoded precisely: Whatever input data is used should be encoded precisely on the generated QR code. This way, the target audience will have an easy time accessing and using the encoded information.

Provide customization options: The tool should allow users to personalize QR code according to their requirements. This, therefore provides the user with an opportunity to change QR codes to accommodate his/her branding or design specifications. Security and tamper-proofing: This will ensure that any generated QR code is not accessible to any unauthorized party and is not duplicated through such a thing. Such a tool may include security features like encryption and authentication. Generating Output to Multiple Formats : The QR code resulted from the tool is output as such in multiple formats: PNG, JPG, or SVG, which are flexible in nature for opting by the user among available formats that might possibly be ideally in tune with their use . The tool should have a user-friendly interface that will assist the users in the various ways the QR code can be created and customized. The user interface must be designed in mind to make it user-friendly to everyone, including the layman, and not complicated.

V. METHODOLOGY

A QR code generator in Python can be implemented using the following steps:

1) Importing of necessary modules: This is the very first step in generating QR codes. For creating a QR code, either qrcode module and for the user interface, the tkinter module is used.
2) Building the chief window: Construction of the main window for the QR code generator application is now in order. This window should contain all fields that might be needed, such as text/URL, location to store the QR code, a name under which the QR code shall be saved, and the size of the QR code.
3) Taking user inputs: The program, after creating the main window, the The application should be able to take user inputs. This means that an input field can be created using the tkinter module. The input fields, in turn, should be designed with a user-friendly usability so the user can easily fill all the necessary inputs.
4) QR code generation: After taking the user inputs, the application should generate the QR code. This can be done with the help of the qrcode module. The text/URL input has to be encoded using a qrcode module followed by saving the generated QR code at a specified location with the specified name.
5) QR Code Display: The developed QR code should be displayed to the user by the application. Probably, it will make use of a module named Pillow, which might be utilized in generating an image of the QR code. Thereafter, this image will be presented inside the application's main window.
6) Error Handling: The application should also handle all the possible errors to prevent disorganization while generating the QR codes. For example, if the user provided an entry that is not a valid URL, there should be an error message that popped up on the display.
7) Testing: After the development of the QR code generator, it should be tested to ensure that it is working as per expectations. The application shall be tested against a variety of different inputs for assurance that it is working correctly.

VI. EXPERIMENTAL SETUP

1) To make a QR code generator experiment, first we get a setup. The following are the steps that we can make an implementation for:
2) Define the requirements: Before starting the experiment, we need to define the requirements of the QR code generator, for example, type of input data, formatting, customization, security features, output formats, and user interface.
3) Choose a programming language and a development platform: We identify the programming A programming language and development platform with which the QR code generator can be built. Some common programming languages and platforms used to construct QR code generators include Python, Java, PHP, and JavaScript.
4) Setup the Development Environment: Involves the process of installing the development environment and the software, tools, and libraries to be employed to build the application in progressive phases. This could be done in-person on a local machine or in a cloud-based development environment.
5) Design the algorithm: We need to design the algorithm to generate QR codes meeting their Requirements: The algorithm must be designed to encode input data in a correct and secure manner and must entail a bundle of error checking and validation processing.
6) Carry out the algorithm: Implement the algorithm in the designated programming language and development platform. The implementation should be modular and well structured to ensure ease of support and scalability.
7) Test the QR code generator: We need to test the QR code generator using various input data and customization options to ensure that it generates QR codes should

be reliable and precise in testing. To obtain the best and most robust test results, the QR code should be tested against a variety of input data formats and output formats, and the user interface needs to be thoroughly tested to ensure that it is friendly and simple to carry-out.

8) Deploy the QR code generator: Once the QR code generator has been tested and validated, it can be deployed to a production environment. The The deployment should include setting up the infrastructure that the tool would need to use, such as servers and databases, and integrating the tool with whatever other systems are important in the context.
9)

VII. DESIGN IMPLEMENTATION

User Interface Design: Design a user interface using Tkinter, which is one of the Python libraries to be used in creating the Graphical User Interface for this project. This would house fields where data to be encoded would appear, custom options to personalize your QR code, and finally, a preview of how created QR code would look. QR code generation: the qrcode is a python library to generate QR codes. This library provides functions which encode data into QR codes, saving it as an image file.Customization Options: It must also allow for options to customize size and error correction level of QR code. Logotype, image, and custom carrier options for QR code may also be added. In view of this, those customization options can be implemented then through the Pillow library—a package in Python for image processing.

File Handling: The project should implant the generated QR code as an image file. In this procedure, a Python Imaging Library (PIL) can be used, which is forked from the Pillow library. PIL has functions that save images in multiple file formats, such as PNG, JPG, and BMP.

Systems Integration: QR Code Generator can also be integrated with other systems, like inventory management systems, e-commerce platforms, and marketing automation tools. This would be possible with the APIs that most of the systems provide for integration with other applications. One can autoenerate QR codes by using data in other systems. Wrapped Mobile Application: One can wrap a mobile application around the project for either Kivy or some other libraries like Pygame; both turn out excellent and hold the added advantage of having cross-platform mobile apps, all developed in Python. More security features: The QR code generator will add additional stipulations inside the QR code, which IT will make use of at the QR code generator stage to implement some security check features, provide password protection on the QR code itself, or even create encrypted QR codes that only authorized parties can decrypt. Concretely, these could be implemented by libraries such as pycrypto or cryptography.

VIII. RESULT

It is quite hard to provide an exact result regarding a python QR code generator project since the results may vary depending on what purpose the project will serve and other implementation details. Maybe these are some of the possible results you could arrive at with a Python QR code generator project.

Fig. 1. Phython QR Code Generator

Fig. 2. QR code

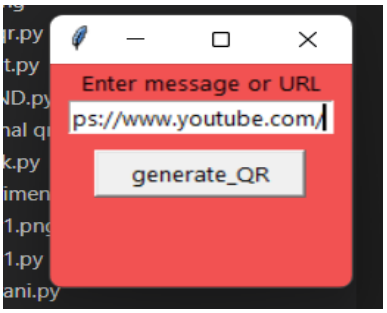

Fig. 3. QR Generator For Link(a)

Fig. 4. QR Generator For Link(b)

Successful generation of QR codes is one of the major goals in a QR code generator project. If it efficiently generates the QR codes, it would be called successful. Customization options: Logging, color, and other design elements can be enacted into a QR code. A successful QR code generator project would be one in which it gives options to customize it in order to come up with a unique QR code. Error correction: QR codes inherently contain error correction, which helps them be readable even when partly damaged or blurred. A

successful QR code generator project could implement error correction to ensure that the generated QR codes are strong and can be read under most adverse conditions

Other tools can be integrated with a QR code generator project, for example, using libraries that process images or their Web frameworks to enhance its functionality. If successful, it will mean returning a more versatile and useful project. Usability: A QR code generator project would need to be easy to work with and user-friendly. This could mean the development of user interface tools, like a simple intuitive user interface, or documents explaining how the project is used.

IX. Conclusion

This paper illustrates the effective design and development of a QR code generator using Python, which simplifies and automates the creation of QR codes. The study thoroughly explored the technical and practical aspects, from the selection of programming tools to user interface design. Through its user-friendly interface and robust backend, the application ensures reliable QR code generation, offering customization options and secure, accurate outputs. This QR code generator stands out for its adaptability to various user needs and its integration capabilities with other systems, making it a versatile tool in digital information sharing. The research confirms that employing advanced coding standards and encryption can enhance data security, making QR codes a viable tool for versatile applications across different industries.

References

[1] A. Papathanasiou, G. Liontos, G. Paparis, V. Liagkou, and E. Glavas, "BEC Defender: QR Code-Based Methodology for Prevention of Business Email Compromise (BEC) Attacks," Sensors, vol. 24, no. 5. MDPI AG, p. 1676, Mar. 05, 2024. doi: 10.3390/s24051676.

[2] D. Prakash, "Data-Driven Management: The Impact of Big Data Analytics on Organizational Performance," International Journal for Global Academic & Scientific Research, vol. 3, no. 2. International Consortium of Academic Professionals for Scientific Research, pp. 12–23, Jul. 02, 2024. doi: 10.55938/ijgasr.v3i2.74.

[3] Z. A. Vavilova, E. A. Chirikov and N. V. Tkachenko, "Development of Applications for Generating and Reading QR Codes for the Micro1 Model," 2022 Systems of Signals Generating and Processing in the Field of on Board Communications, Moscow, Russian Federation, 2022, pp. 1-7, doi: 10.1109/IEEECONF53456.2022.9744343.

[4] R. Rathore and S. Pratap Singh Rathore, "Machine Learning Applications in Human Resource Management: Predicting Employee Turnover and Performance," International Journal for Global Academic & Scientific Research, vol. 3, no. 2. International Consortium of Academic Professionals for Scientific Research, pp. 48–59, Jul. 02, 2024. doi: 10.55938/ijgasr.v3i2.77.

[5] 5. R. Chen et al., "A Fast Adaptive Binarization Method for QR Code Images Based on Dynamic Illumination Equalization," Electronics, vol. 12, no. 19. MDPI AG, p. 4134, Oct. 04, 2023. doi: 10.3390/electronics12194134.

[6] M. Al Malki, "A Review of Sustainable Growth challenges faced by Small and Medium Enterprises," International Journal for Global Academic & Scientific Research, vol. 2, no. 1. International Consortium of Academic Professionals for Scientific Research, pp. 53–67, Mar. 30, 2023. doi: 10.55938/ijgasr.v2i1.39.

[7] T. Wang, H. Zheng, C. You, and J. Ju, "A Texture-Hidden Anti-Counterfeiting QR Code and Authentication Method," Sensors, vol. 23, no. 2. MDPI AG, p. 795, Jan. 10, 2023. doi: 10.3390/s23020795.

[8] Z. Mohammad Kuraishi, "Role of Analytics in Supply Chain Management Industry in Lithuania: Big Data Analytics & AI," International Journal for Global Academic & Scientific Research, vol. 2, no. 4. International Consortium of Academic Professionals for Scientific Research, pp. 44–53, Dec. 31, 2023. doi: 10.55938/ijgasr.v2i4.65.

[9] S.-Y. Yang, H.-C. Jan, C.-Y. Chen, and M.-S. Wang, "CNN-Based QR Code Reading of Package for Unmanned Aerial Vehicle," Sensors, vol. 23, no. 10. MDPI AG, p. 4707, May 12, 2023. doi: 10.3390/s23104707.

[10] S.-K. Chen and Y.-W. Ti, "A Design of Multi-Purpose Image-Based QR Code," Symmetry, vol. 13, no. 12. MDPI AG, p. 2446, Dec. 18, 2021. doi: 10.3390/sym13122446.

[11] A. Prakash, "Blockchain Technology for Supply Chain Management: Enhancing Transparency and Efficiency," International Journal for Global Academic & Scientific Research, vol. 3, no. 2. International Consortium of Academic Professionals for Scientific Research, pp. 01–11, Jul. 02, 2024. doi: 10.55938/ijgasr.v3i2.73.

[12] Y. Liu, Y. Song, G. Gu, J. Luo, T. Wang, and Q. Jiang, "A Data Matrix Code Recognition Method Based on L-Shaped Dashed Edge Localization Using Central Prior," Sensors, vol. 24, no. 13. MDPI AG, p. 4042, Jun. 21, 2024. doi: 10.3390/s24134042.

[13] I. F. Anderson, "QRGB+: Python-based QR code generator with RGB color channels using Replit online IDE to enhance data capacity.," Unpublished, 2024, doi: 10.13140/RG.2.2.12977.19046.

[14] 15. P. Sutheebanjard and W. Premchaiswadi, "QR-code generator," 2010 Eighth International Conference on ICT and Knowledge Engineering. IEEE, Nov. 2010. doi: 10.1109/ictke.2010.5692920.

[15] S. Tiwari, "An Introduction to QR Code Technology," 2016 International Conference on Information Technology (ICIT). IEEE, Dec. 2016. doi: 10.1109/icit.2016.021.

[16] I. Anderson, "QRGB: App for QR Code Generation (3-in-1 Method), Additive Color Generation Method (RGB), Using Python Programming Code, to Increase Accumulated Information Density." MDPI AG, Jul. 19, 2024. doi: 10.20944/preprints202407.1384.v2.

[17] S. S. Abdul-Jabbar and A. k. Farhan, "Secure QR-Code Generation in Healthcare," Karbala International Journal of Modern Science, vol. 9, no. 2. University of Kerbala - KIJOMS, May 09, 2023. doi: 10.33640/2405-609x.3294.

[18] S. P. S. Rathore, N. Anute, H. Raje, D. S. Ubale, A. P. Narkhede and P. Kaushik, "Segmentation Study on Bank Customers Based on RNN," 2023 International Conference on Smart Devices (ICSD), Dehradun, India, 2024, pp. 1-6, doi: 10.1109/ICSD60021.2024.10751183.

[19] V. Sharma, P. Agarwal, H. Y. Shaikh, R. M. Lenka, S. K. Manjhi and R. Rathore, "Smart Next-Generation Revenue Growth: A Methodology for Partitioning Customers Utilizing the K-Means Algorithm and RFM Model," 2023 International Conference on Smart Devices (ICSD), Dehradun, India, 2024, pp. 1-6, doi: 10.1109/ICSD60021.2024.10751627.

[20] P. Kaushik, R. Rohilla, P. Walia, S. Shankar, M. M. Kaushik and T. S. Gupta, "Confronting Hate Speech in SMART Environments: An Approach that Uses Ensemble Learning and LSTM," 2023 International Conference on Smart Devices (ICSD), Dehradun, India, 2024, pp. 1-6, doi: 10.1109/ICSD60021.2024.10751441.

[21] S. S. Sikarwar, N. Chawla, P. Agarwal, M. F. Afzal, S. P. S. Rathore and N. Valecha, "Improving Lane Management: Leveraging OpenCV and IoT to Empower Smart Environments," 2023 International Conference on Smart Devices (ICSD), Dehradun, India, 2024, pp. 1-5, doi: 10.1109/ICSD60021.2024.10751495.