SkewSecure: A QR-Based Tracking and Piracy Detection System for OTT Content

Addure Srihari
Reg. No. 24Q71F0001
srih41135@gmail.com
Department of Master of Computer Applications
Avanthi Institute of Engineering and Technology (Autonomous)
Vizianagaram, Andhra Pradesh, India
Under the guidance of Mr. S. Kesava Rao, M.Tech., (Ph.D.), Associate Professor
kesav546@gmail.com

***Abstract*—*With the rapid growth of Over-The-Top (OTT) platforms, digital content piracy has become a major challenge, leading to significant revenue loss and unauthorized distribution. This project proposes an innovative solution that integrates QR-code technology with content-tracking mechanisms. The system embeds a unique, dynamically generated QR code into each streamed video session, linking the content to a specific user or device. In the case of illegal screen recording or redistribution, the embedded QR code acts as a digital fingerprint, enabling identification of the source of the leak. The system incorporates real-time monitoring and detection techniques to track suspicious activities and flag potential piracy incidents, and enhances content security by combining encryption, watermarking, and user authentication while maintaining minimal impact on the viewing experience. Image-processing and machine-learning techniques improve the robustness of QR detection so that codes can be recovered even when pirated content has undergone compression, resizing, cropping, or format conversion. A secure backend maps each QR code to a database record of user, session, and device details, so that a detected pirated video can be automatically traced to its source. The prototype is implemented in Python using Django, OpenCV, the qrcode library, and a relational database, and was validated through ten functional test cases that all passed, presenting a scalable and efficient approach to combating digital piracy and strengthening the security of OTT ecosystems.*

Keywords*—*OTT Piracy Detection; QR Code; Digital Fingerprinting; Content Tracking; Video Watermarking; Encryption; Source Identification; Image Processing.

I. INTRODUCTION

With the rapid advancement of digital technologies and widespread availability of high-speed internet, the consumption of multimedia content has undergone a revolutionary transformation. Over-The-Top (OTT) platforms have emerged as a dominant medium for delivering entertainment directly to users over the internet, bypassing traditional cable and satellite distribution. Platforms such as Netflix, Amazon Prime Video, and Disney+ have gained immense popularity due to their convenience, affordability, and on-demand personalised content accessible across smartphones, tablets, laptops, and smart televisions.

However, along with the rapid expansion of OTT platforms, digital content piracy has emerged as a major challenge that threatens the sustainability of this ecosystem. Piracy involves the unauthorized

copying, recording, and redistribution of copyrighted content; with advanced screen-recording tools, file-sharing platforms, and illegal streaming services, pirated content can be captured and distributed widely within a short period, causing substantial financial losses and undermining intellectual-property rights. Traditional methods such as Digital Rights Management (DRM), encryption, and watermarking provide some protection but are not entirely effective: DRM can be bypassed, encryption does not prevent screen recording, and watermarks may be removed through editing. These approaches primarily focus on prevention rather than detection and traceability.

There is therefore a growing need for intelligent systems that not only protect digital content but also enable effective tracking and identification of piracy sources. The proposed system, SkewSecure, embeds a unique, dynamically generated QR code into each video stream during playback or distribution; the code encodes information such as user identification, device details, session data, and timestamp, collectively serving as a digital fingerprint. Compared with traditional watermarking, QR codes offer higher data capacity and improved robustness, making them suitable for tracking and enabling direct identification of the source of unauthorized content.

II. LITERATURE SURVEY

Traditional piracy-detection systems in digital media primarily rely on Digital Rights Management, encryption, and watermarking. DRM restricts unauthorized access by enforcing usage policies and access control but is often bypassed using screen-recording tools or software exploits. Encryption secures content during transmission but does not prevent piracy once the content is decrypted and played on a user device. Watermarking embeds visible or invisible marks to indicate ownership, but these marks can be removed or altered using video-editing tools, and conventional watermarking often suffers from low robustness against compression, cropping, and re-encoding, while static watermarks do not provide user-specific traceability.

Recent advances in machine learning and computer vision have enabled intelligent piracy-detection systems; convolutional neural networks and deep-learning models are widely used for video analysis, pattern recognition, and anomaly detection, although many approaches focus on detecting piracy after it occurs rather than tracing it. QR-code-based tracking has emerged as a promising solution because QR codes store large amounts of data compactly and can be detected with image-processing techniques; embedding dynamic QR codes into video frames assigns a unique identity to each distributed copy, enabling direct mapping between content and the user or device. Recent studies propose hybrid approaches combining watermarking, encryption, and QR-based encoding, and use machine learning to decode QR codes from distorted or low-quality frames, demonstrating improved robustness in real-world scenarios.

TABLE I. CONTENT-PROTECTION APPROACHES

S.No	Approach	Mechanism	Limitation / Note
1	Digital Rights Management	Access/usage policy enforcement	Bypassed by screen recording
2	Encryption	Secure transmission	No protection after decryption
3	Watermarking	Embedded ownership marks	Removable; low robustness

S.No	Approach	Mechanism	Limitation / Note
4	ML/CV detection	CNN-based video analysis	Detects after the fact
5	QR-based tracking	Dynamic QR digital fingerprint	User/device-level traceability
6	Hybrid (QR + crypto + ML)	Combined encoding & decoding	Robust under transformations

III. EXISTING SYSTEM AND PROPOSED SYSTEM

A. Existing System

Traditional OTT content-protection systems primarily rely on DRM, encryption, and watermarking to prevent unauthorized access and distribution. DRM controls how users access content by enforcing playback limits, device binding, and subscription-based access, but these mechanisms focus on prevention rather than detection and can be bypassed. Watermarks may be removed or degraded, and none of these methods reliably identifies the exact source of a leak once content has been captured and redistributed.

Limitations of the existing system:

- DRM can be bypassed using screen-recording tools or exploits.
- Encryption does not prevent capture after decryption.
- Watermarks can be removed or altered by editing.
- Static marks provide no user-specific traceability.
- Focus on prevention, not detection or source identification.

B. Proposed System

The proposed system introduces an advanced QR-based tracking and piracy-detection mechanism for OTT content. It embeds dynamically generated QR codes into video streams, creating a unique digital fingerprint for each user or session; the codes contain encoded information such as user ID, device details, and session data, enabling traceability of content distribution. Unlike traditional watermarking, QR codes provide higher data capacity and improved reliability, and the system uses image-processing and machine-learning techniques to detect and decode QR codes from video frames even after compression, cropping, or re-encoding. A secure backend maps each QR code to a database record so that a detected pirated video can be automatically traced to its source.

Advantages of the proposed system:

- User/session-level traceability through dynamic QR fingerprints.
- Higher data capacity and robustness than traditional watermarks.
- Reliable QR recovery under compression, cropping, and re-encoding.
- Automated source identification, reducing manual investigation.
- Encryption and authentication strengthen security.
- Minimal impact on video quality and user experience; scalable design.

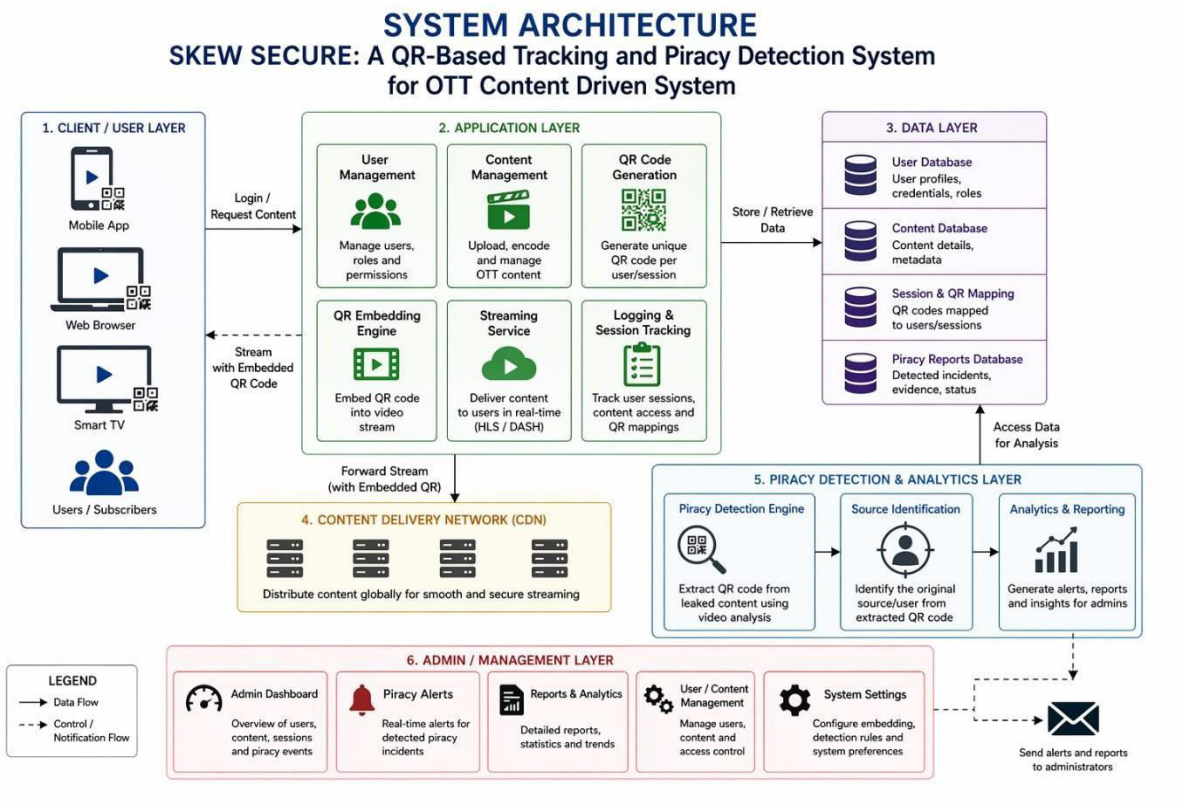
IV. SYSTEM DESIGN AND METHODOLOGY

A. System Analysis

Functionally, the system must support user authentication, content upload, unique QR-code generation from user/session data, embedding of QR codes into video frames, streaming of QR-embedded content without noticeable quality loss, session tracking, and extraction of QR codes from leaked content followed by source identification. Non-functional requirements include security (encryption and tamper-resistance), scalability to large OTT workloads, reliability of QR recovery under transformations, and a user-friendly administrative interface.

B. System Architecture

The architecture comprises a QR-generation module, a video-processing/embedding module, a streaming module, a detection-and-decoding module, and a secure backend with a web-based administrative interface. QR codes are generated per user/session with encoded and encrypted information; the video-processing module extracts frames using OpenCV and embeds the QR code at intervals using overlay techniques that preserve video quality; the streaming module delivers QR-embedded content; and the detection module locates and decodes QR codes from suspect video, using machine-learning reconstruction when codes are degraded. The backend database stores user credentials, session information, and QR mappings used for source identification.



C. Workflow

A user authenticates and requests content; the system generates a unique encrypted QR code from the user/session data, embeds it into the video frames, and streams the QR-embedded video while logging session details. If a pirated copy is found, the system extracts and decodes the QR code, decrypts the embedded data, and matches it against the database to identify the responsible user or device, supporting fast enforcement action.

V. SYSTEM IMPLEMENTATION

A. Technology Stack

TABLE II. TECHNOLOGY STACK

Component	Technology / Tool
Programming Language	Python
Web Framework	Django
Video / Image Processing	OpenCV, MoviePy
QR Generation / Detection	qrcode library; OpenCV QRCodeDetector
Numerical Computing	NumPy
Encryption	cryptography (Fernet); AES/RSA for payload
Database	MySQL / PostgreSQL
ML Frameworks (robust decode)	TensorFlow / PyTorch

B. QR Generation and Embedding

The QR-generation module creates a unique QR code for each user or streaming session, encoding user ID, device information, session timestamp, and keys. To enhance security, the encoded data is encrypted (for example using AES/RSA or the cryptography library's Fernet) before being converted into the QR image using the Python qrcode library. The video-processing module extracts frames using OpenCV and embeds the QR code into selected frames at regular or dynamic intervals using overlay techniques such as alpha blending or pixel-level manipulation, keeping the code visually imperceptible or minimally intrusive, and recombines the frames while maintaining audio–video synchronisation.

C. Detection, Decoding, and Source Identification

For piracy detection, the system extracts frames from suspect video and uses QR-detection algorithms (such as OpenCV's QRCodeDetector) to locate and decode embedded codes. When QR codes are partially damaged or distorted by compression, resizing, or cropping, machine-learning models trained on distorted QR samples reconstruct and decode the information. The decrypted payload is matched against the backend database to identify the user, device, or session that is the source of the leak, automating what would otherwise be a manual investigation.

VI. SYSTEM TESTING AND RESULTS

The system was validated through ten functional test cases covering user login, invalid login, content upload, QR-code generation, QR embedding, streaming of QR-embedded content, video-quality verification, session tracking, QR extraction from a leaked video, and source identification. All test cases passed and behaved as expected.

TABLE III. REPRESENTATIVE TEST CASES

ID	Scenario	Input	Expected Output	Status
TC01	User login	Valid credentials	Login successful	**Pass**
TC03	Content upload	Valid video file	Video uploaded and stored	**Pass**
TC04	QR code generation	User/session data	Unique QR code generated	**Pass**
TC05	QR embedding	Video + QR code	QR embedded into video stream	**Pass**
TC07	Video quality check	Streamed video	Video quality not affected	**Pass**
TC09	QR extraction (leaked)	Pirated video input	QR code extracted successfully	**Pass**
TC10	Source identification	Extracted QR code	User/source identified	**Pass**

A. Observed Results

The implementation demonstrates that embedding QR codes within video streams allows reliable detection of pirated content without significantly affecting user experience or video quality. The system successfully performs QR generation, embedding, extraction, and source identification, improving accountability and security in content delivery, and features such as real-time monitoring, alert generation, and analytical reporting help administrators respond quickly to piracy incidents. The source reports these outcomes qualitatively; no specific numeric metrics are claimed here, and robustness against highly sophisticated tampering remains a challenge.

Representative screenshots from the prototype implementation:

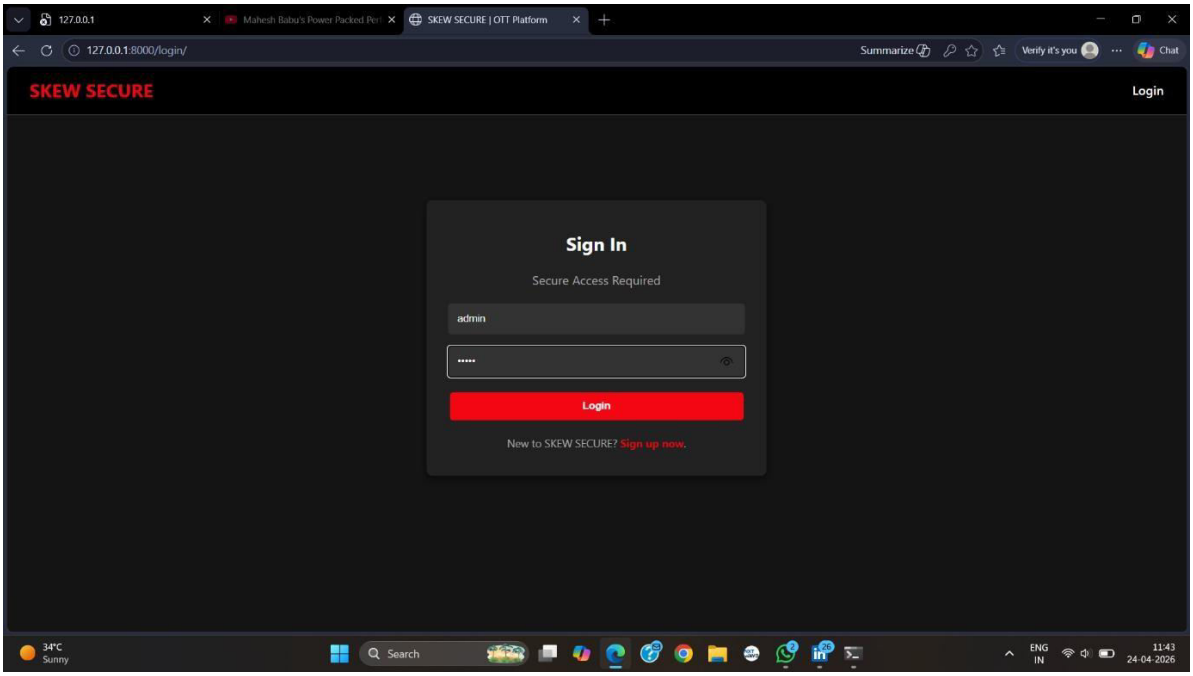


Fig. 1. Administrator login and dashboard.

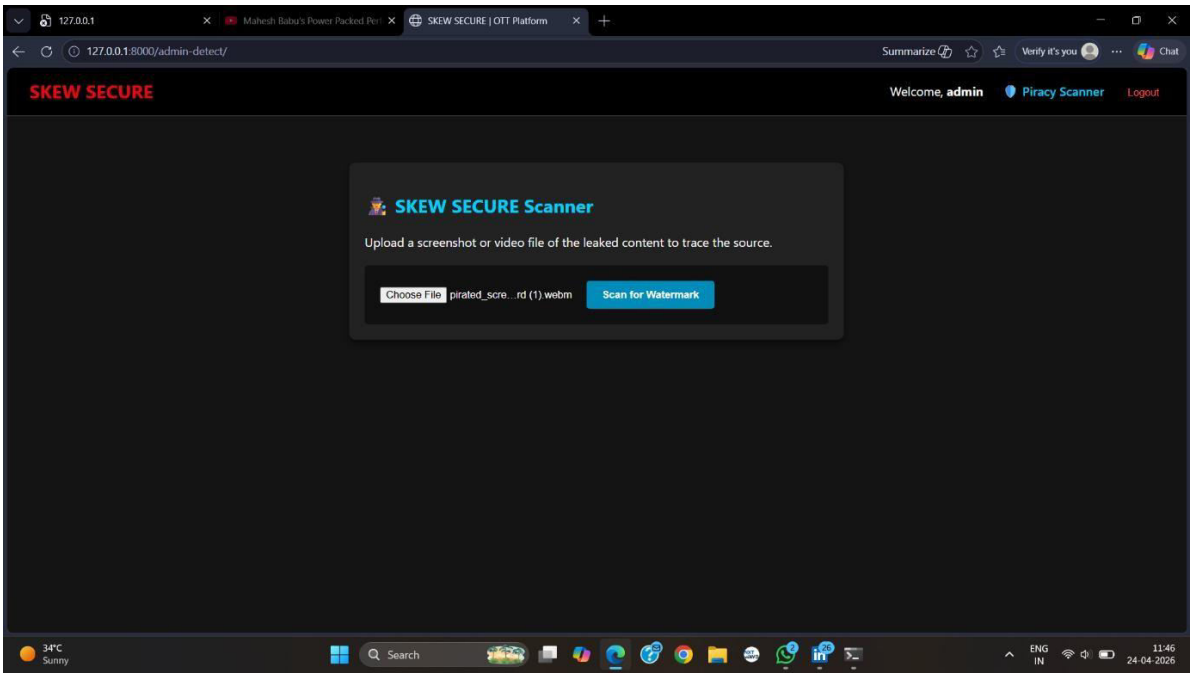


Fig. 2. Content upload and QR-code generation.

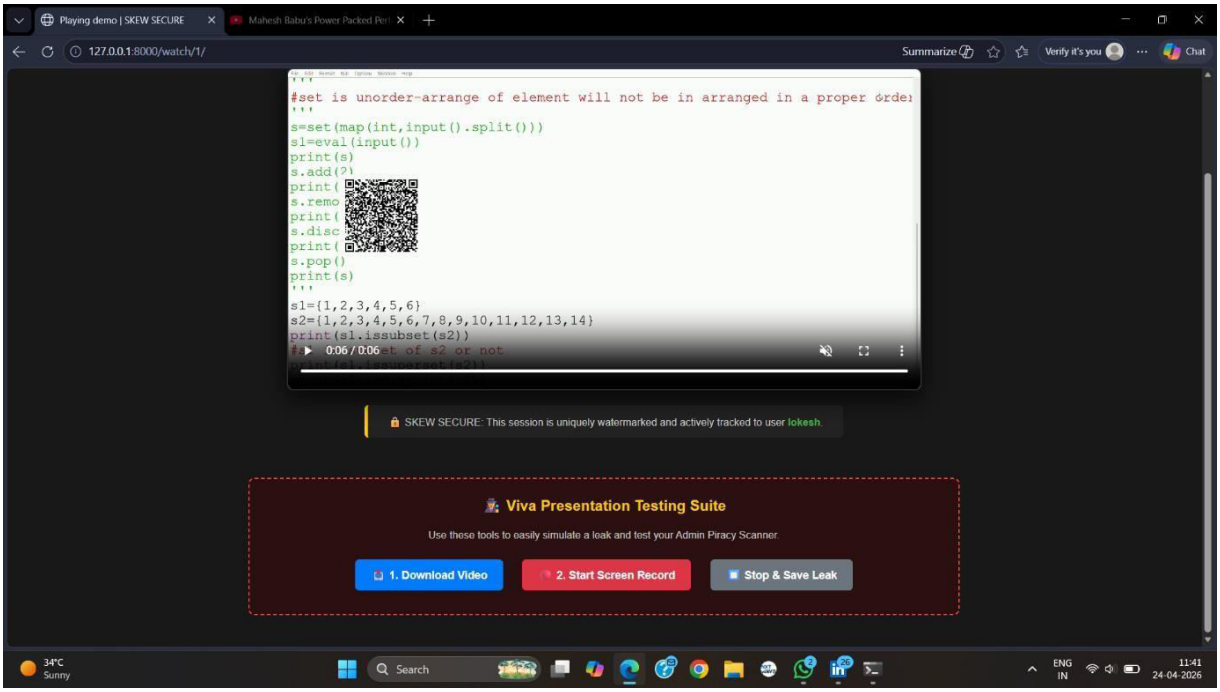


Fig. 3. QR embedding into the video stream.

VII. CONCLUSION AND FUTURE SCOPE

The project provides an effective and innovative solution to the growing problem of digital content piracy on OTT platforms. By integrating QR-based tracking with content streaming, the system ensures that each video session is uniquely identifiable, enabling accurate tracing of unauthorized distribution. The implementation demonstrates that embedding QR codes within video streams allows reliable detection of pirated content without significantly affecting user experience or video quality, and the system successfully performs QR generation, embedding, extraction, and source identification, thereby improving accountability and security in content delivery. Features such as real-time monitoring, alert generation, and analytical reporting enhance administrators' ability to respond quickly to piracy incidents; while challenges such as handling sophisticated piracy methods and ensuring robustness against tampering remain, the system proves to be a scalable and practical approach for modern OTT platforms.

Future work can strengthen robustness and scale. Deep-learning models can be further trained on diverse real-world distortions to improve QR recovery from heavily degraded pirated content; frequency-domain or multi-layer embedding can increase resistance to removal; and cloud-native, distributed processing can support millions of concurrent streams. Integration with automated web-crawling and content-matching could detect leaked content across the internet, and stronger cryptographic protection of the embedded payload would further harden the system against tampering and forgery.

REFERENCES

[1] I. J. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, Digital Watermarking and Steganography, 2nd ed. Burlington, MA, USA: Morgan Kaufmann, 2007.

[2] F. Hartung and M. Kutter, "Multimedia Watermarking Techniques," Proceedings of the IEEE, vol. 87, no. 7, pp. 1079–1107, 1999.

[3] J. Fridrich, Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge, U.K.: Cambridge University Press, 2009.

[4] M. Arnold, M. Schmucker, and S. Wolthusen, Techniques and Applications of Digital Watermarking and Content Protection. Norwood, MA, USA: Artech House, 2003.

[5] R. Chandramouli, M. Kharrazi, and N. Memon, "Image Steganography and Steganalysis: Concepts and Practice," in Proc. Int. Workshop on Digital Watermarking, 2003.

[6] C. T. Hsu and J. L. Wu, "Multiresolution Watermarking for Digital Video," IEEE Trans. Circuits and Systems for Video Technology, vol. 9, no. 1, pp. 109–120, 1999.

[7] S. Katzenbeisser and F. A. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking. Norwood, MA, USA: Artech House, 2000.

[8] K. Solanki, N. Jacobsen, and B. S. Manjunath, "Robust Image Watermarking Using Feature-Based Techniques," IEEE Trans. Image Processing, 2005.

[9] ISO, "Information Technology — Coding of Audio-Visual Objects (MPEG Standards)." [Online]. Available: https://www.iso.org/

[10] Denso Wave, "QR Code Standard (ISO/IEC 18004)." [Online]. Available: https://www.qrcode.com/