IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429

Volume 5, Issue 2, November 2025

Impact Factor: 7.67

TrustQR: Smart QR + Blockchain for Fake Product Detection

Prof. Kapil D. Dere[1], Dr. Anand A. Khatri[2], Aditi Bibave[3], Suhani Pachpute[4], Vaishnavi Mandale[5]

Assistant Professor, Department of Computer Engineering[1]
Head of Department, Department of Computer Engineering[2]
Students, Department of Computer Engineering [3,4,5]
Jaihind College of Engineering, Kuran, Pune, India

Abstract: *TrustQR is a blockchain-powered Android application developed using Java/XML, Firebase Real-time Database, Firebase Authentication, and Cloud Storage to ensure product authenticity and consumer trust. The system integrates SHA-256 encryption with decentralized data storage to secure product information against tampering or duplication. It features two modules: the Admin app, where authorized sellers generate encrypted QR codes for each product, and the Customer app, where users scan QR codes to instantly verify authenticity. Each verification is logged in real time on Firebase, maintaining transparency and traceability across the product's lifecycle.*

An AI-driven review analysis engine, powered by Natural Language Processing (NLP), detects fake or biased reviews and calculates user reputation scores to promote credible feedback. This hybrid system of block chain and artificial intelligence ensures trust, accountability, and secure digital verification. TrustQR provides a scalable, mobile-friendly, and tamper-proof solution for combating counterfeit products across industries such as healthcare, cosmetics, electronics, and retail..

Keywords*:* blockchain authentication, SHA-256 encryption, Smart QR verification, Firebase Real-time Database, AI-based fake review detection, NLP sentiment analysis, product authenticity

I. INTRODUCTION

In today's global marketplace, counterfeit products have become one of the most serious challenges affecting consumer safety, brand reputation, and economic growth. From pharmaceuticals and cosmetics to electronics and luxury goods, the infiltration of fake items has created an urgent need for transparent and tamper-proof product authentication systems. Traditional verification techniques such as holograms, barcodes, or serial numbers are easily cloned or manipulated, leading to a growing demand for digital verification systems backed by advanced technology.

TrustQR is an innovative blockchain-powered Android application designed to address this challenge by ensuring end-to-end product authenticity using a combination of Smart QR verification, SHA-256 encryption, and AI-driven analysis. The system leverages Firebase Realtime Database, Firebase Authentication, and Cloud Storage to provide a secure, real- time, and scalable backend infrastructure. It bridges the gap between manufacturers, distributors, and customers by creating a trust chain that records and verifies each product's journey from production to purchase.

At the core of TrustQR lies the concept of Blockchain Authentication, where every product record is stored in an immutable, decentralized ledger using SHA-256 cryptographic hashing. This ensures that once a product's details are added by the admin, they cannot be altered or tampered with by any unauthorized user. Each product is assigned a unique encrypted QR code, which acts as its digital signature. When a customer scans the QR code through the mobile app, the app instantly verifies its authenticity by cross-checking with the blockchain data stored in Firebase.

Beyond product verification, TrustQR also incorporates AI-based fake review detection and NLP sentiment analysis to maintain the credibility of user feedback. Customers can write reviews after successful verification, but the system automatically flags suspicious or repetitive patterns to prevent misleading information. This dual-layer system combining

DOI: 10.48175/IJARSCT-29725

blockchain-based authenticity and AI-based credibility enhances transparency, strengthens consumer trust, and creates a more reliable digital marketplace.

From a technical perspective, the project uses Java/XML for front-end Android development, integrated with Firebase services for cloud-based data handling. The Admin app allows verified sellers or manufacturers to register products and generate encrypted QR codes, while the Customer app provides real-time verification, feedback submission, and reputation tracking.

In summary, TrustQR is not just a verification tool but a comprehensive trust ecosystem that merges blockchain security, smart QR verification, and AI intelligence. It empowers both consumers and manufacturers with transparency, accountability, and confidence in product authenticity creating a safer and more ethical marketplace for the digital era.

II. LITERATURE SURVEY

[1] H. Gupta and R. Sharma, "Transformer-Based Real-Time Mobile Inference for Text Classification," IEEE Access, 2024.

Findings: This paper explored lightweight transformer architectures for mobile-based NLP classification. It showed that optimized transformer models can perform accurate fake review detection directly on Android devices, reducing dependency on cloud inference and improving privacy.

[2] K. Wasnik, P. Deokar, and A. Gohane, "Detection of Counterfeit Products using Blockchain," IEEE International Conference on Advances in Computing and Communication (ICACC), 2022.

Findings: This paper proposed a blockchain-based framework for supply chain transparency where each product record is stored as a hash on a distributed ledger. It demonstrated how immutability and traceability of blockchain reduce counterfeit goods. The study validated that SHA-256 hashing effectively prevents tampering in product lifecycle data.

[3] R. Pitale, A. Gaikwad, and S. Patil, "Fake Product Identification using Blockchain," IEEE International Conference on Inventive Research in Computing Applications (ICIRCA), 2023.

Findings: This paper implemented a mobile interface that uses blockchain transaction IDs to confirm product originality. The researchers found blockchain integration improved verification speed and transparency while minimizing dependence on centralized authorities.

[4] R. Pitale, A. Gaikwad, and S. Patil, "Fake Product Identification using Blockchain," IEEE International Conference on Inventive Research in Computing Applications (ICIRCA), 2023.

Findings: This paper implemented a mobile interface that uses blockchain transaction IDs to confirm product originality. The researchers found blockchain integration improved verification speed and transparency while minimizing dependence on centralized authorities.

[5] T. Wang, H. Zheng, and C. You, "A Texture-Hidden Anti-Counterfeiting QR Code and Authentication Method," IEEE Conference on Image Processing and Security, 2023.

Findings: This research introduced texture-based hidden QR codes embedded with visual micro-patterns that prevent duplication. Their method significantly reduced code forgery rates by combining cryptographic signatures with physical pattern verification, laying the groundwork for Smart QR technology.

[6] X. Li and L. Chen, "Fake Review Detection Using Deep Neural Networks with Multimodal Feature Fusion," IEEE International Conference on Parallel and Distributed Systems (ICPADS), 2023.

Findings: The authors developed a neural network model that analyzes text semantics and reviewer behavior patterns to identify fake online reviews. The system achieved over 90% precision, proving that AI and NLP can effectively detect opinion spam in e-commerce platforms.

[7] S. Prathipa, R. Priyadharshini, and S. Priyanka, “Counterfeit Product Detection in Supply Chain Management with Blockchain,” IEEE Conference on Computational Science and Technology (ICCST), 2022.
Findings: The authors presented a system integrating IoT sensors with blockchain to verify authenticity at every logistics stage. Their results showed over 95% accuracy in detecting unauthorized product replacements, emphasizing decentralized ledgers as a secure verification layer for end-to-end authenticity.

[8] Y. Yan, H. Liu, and S. Li, “An IoT-Based Anti-Counterfeiting System Using Visual Features on QR Code,” IEEE Internet of Things Journal, 2019.
Findings: The paper proposed an IoT architecture for QR-based authentication that links visual QR code data with secure cloud verification. It demonstrated how combining IoT and QR verification can offer real-time authenticity validation and remote monitoring of product legitimacy.

III. METHODOLOGY

3.1 Overview

TrustQR fuses three layers: (1) Blockchain-style authentication via an append-only hash chain of product events, (2) Smart QR verification with signed payloads, and (3) AI/NLP to filter fake reviews and maintain user reputation. Android apps (Java/XML) talk to Firebase (Auth, Realtime DB, Cloud Storage, and Cloud Functions). All product states are hashed with SHA-256 and linked to a ledger stored as an append-only chain.

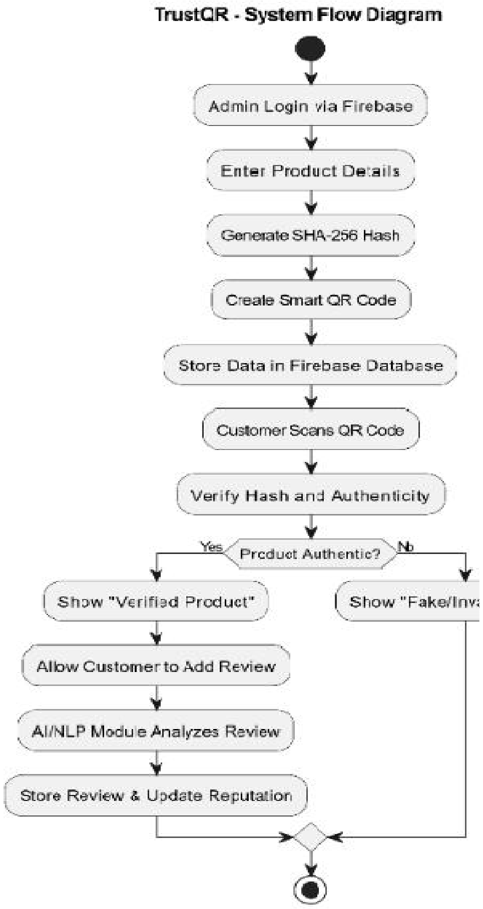
TrustQR - System Flow Diagram
Admin Login via Firebase
Enter Product Details
Generate SHA-256 Hash
Creale Smart QR Code
Store Data in Firebase Database
Customer Scans QR Code
Verify Hash and Authenticity
Yes
Product Authentic?
No
Show "Verified Product"
Show "Fake/Inv
Allow Customer to Add Review
AI/NLP Module Analyzes Review
Store Review & Update Reputation

Fig 1: System Flow Diagram

3.2 Core Steps (End-to-End)

A) Admin (Product Onboarding → QR Issuance)

1. Authenticate: Admin signs in using Firebase Authentication (email/OTP).
2. Create Product: Admin enters product metadata (PID, batch, MFD/EXP, SKU).
3. Hash & Seal: Cloud Function computes hash = SHA256(pid||meta||prev_hash||ts) and links it to the previous block (prev_hash) forming a hash chain.
4. Sign Payload: Trust service signs {pid, ts, nonce} with the platform key (e.g., ECDSA).
5. Generate Smart QR: QR encodes {pid, ts, nonce, sig} (no plaintext secrets).
6. Persist:

o Realtime DB: canonical product record + current state (MINTED).
o Cloud Storage: QR PNG/SVG + any media.
o Ledger Node: {index, pid, ts, prev_hash, hash} appended as immutable event.

7. Distribute: Admin prints/embeds the QR on the physical product.

B) Customer (Scan → Verify → Review)

1. Scan: Customer app reads the Smart QR and extracts {pid, ts, nonce, sig}.
2. Verify Signature: App/Cloud Function verifies sig against the platform public key.
3. Anti-Replay: Trust service checks nonce status (unused → accept, used → reject).
4. Ledger Check: Recompute hash; validate prev_hash link in the ledger (append-only).
5. State Fetch: Pull product state/version from Real-time DB (e.g., MINTED, IN_STORES, SOLD).
6. Decision:

o All checks pass → AUTHENTIC (green).
o Any check fails → SUSPECT/FAKE (red) with reason codes.

7. Record Verification: Append a VERIFIED event (hash-linked) and update last-seen metadata.
8. Review Flow: After a valid scan, user can post a review.
9. AI/NLP Filter:

o Text features: embedding (BERT-like), sentiment, burstiness.
o Behaviour features: device/IP entropy, purchase-verified flag, account age.
o Output: spam score + explanation tag.

10. Reputation Update: Reviewer's score adjusted; risky reviews down-weighted/flagged.

C) Security & Reliability Controls

1. Key Management: Sign with server-side keys only (Cloud Functions); rotate quarterly.
2. Dynamic QR: Include nonce + timestamp; single-use nonce; short expiry.
3. Rate Limits: Per device/IP verification throttles; CAPTCHAs on anomalies.
4. Integrity Locks: Store a hash digest of each Firebase product document in the ledger; any off-chain edit breaks the digest.
5. Audit Trails: Every state change (TRANSFER, SOLD, VERIFIED, REVIEWED) is a chain event.
6. PII Minimalism: Store only what's needed; tokenized identifiers in reviews.

D) Data Model (Sketch)

- /products/{pid} → core details, current_state, media_refs, last_hash.
- /ledger/{index} → {pid, ts, prev_hash, hash, event, signer}.
- /reviews/{pid}/{rid} → {uid, text, stars, ai_score, verified_purchase}.
- /nonces/{nonce} → {pid, ts, used: bool}.

DOI: 10.48175/IJARSCT-29725

E) AI/NLP Pipeline (Review Credibility)

1. Ingest: On submit, route review to AI service.
2. Pre-process: Clean text, language detect, tokenize.
3. Embed/Classify: BERT-like text encoder → spam classifier; fuse behaviour features.
4. Score & Explain: Generate spam_score, sentiment, and an explanation tag (e.g., "duplicate phrasing").
5. Store & Act: Save scores, update user reputation, flag or shadow-reduce visibility.

F) Performance & Scale

- Cold-start: Pre-warm Functions; keep model weights cached.
- Ledger Partitioning: Per-brand PID namespaces; periodic snapshotting.
- Offline Mode: Cache last known result; queue verification to retry on network.

IV. SYSTEM ARCHITECTURE

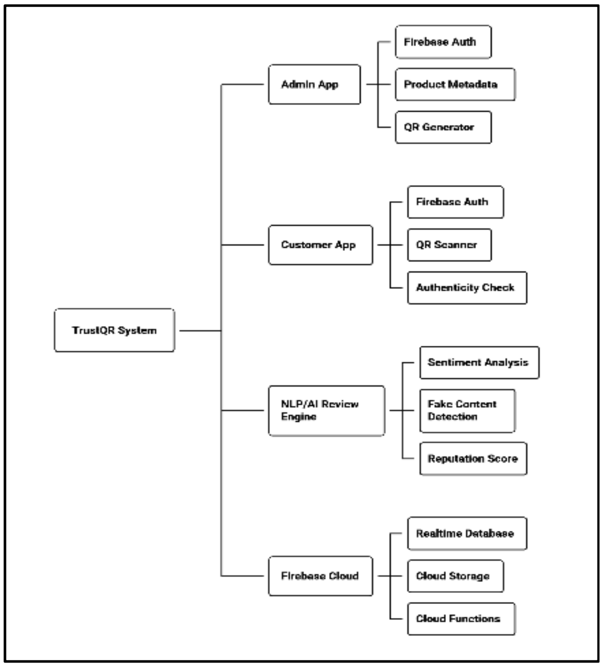

Fig. 2: System Architecture

1. (Mobile Client)

- The Admin signs in using Firebase Auth.
- When adding a product, the app sends product metadata (name, batch, date, etc.) to the QR Generator service.
- The QR Generator sends the product metadata to the SHA-256 Hasher, which generates a unique hash for that product.
- The product data (hash + metadata) is written into Firebase Realtime Database, while the generated QR image is uploaded to Firebase Cloud Storage.
- The admin can also update product info later or publish new product batches.

2. Customer App (Mobile Client)

- The Customer signs in via Firebase Auth.
- When the customer scans a product QR code, the app extracts the hash embedded in it.
- The app queries Firebase (via Cloud Functions) to fetch the product record matching that hash.

• The system checks authenticity if the hash exists and matches the stored one, it's a genuine product; otherwise, it's fake or tampered.
• The authenticity result is shown instantly in the app.

3. NLP / AI Review Engine

• When a customer submits a review, the text is sent to the AI Review Engine.
• The engine uses Natural Language Processing (NLP) to analyze sentiment and detect fake or biased content.
• Based on the analysis, it assigns a reputation score to both the review and the user.
• The processed data (review + reputation) is stored in the Firebase Realtime Database.

4. Firebase Cloud (Backend)

• Firebase Auth handles user logins for both Admin and Customer apps.
• Realtime Database stores:
o Product records (hash, metadata)
o User details
o QR scan logs
o Reviews and reputation scores
o Audit and event logs

V. WORKING

The TrustQR system works through a seamless interaction between the Admin App, Customer App, and Firebase Cloud Services, combining blockchain authentication, smart QR verification, and AI/NLP-based review analysis to ensure complete product authenticity and consumer trust.

1. Admin Authentication and Product Registration

The process begins when the admin logs in securely through Firebase Authentication. Once verified, the admin enters product details such as name, batch number, manufacturing date, and other identifiers through the Admin App interface. These details are securely stored in the Firebase Realtime Database.

2. Hash Generation and QR Code Creation

Each product record is encrypted using the SHA-256 algorithm, which generates a unique digital hash. This hash acts as a digital fingerprint that cannot be altered. Using this encrypted data, the system generates a Smart QR Code containing the hash value, timestamp, and digital signature. This QR code is then printed or attached to the product packaging.

3. Data Storage and Blockchain Verification

The product's encrypted information is stored in Firebase Cloud Storage and linked to a block chain-like hash chain structure within the Real-time Database. Each new entry references the previous hash, ensuring data immutability and tamper resistance. This mechanism prevents unauthorized changes or duplication of product records.

4. Customer Verification Process

When a customer purchases a product, they scan the QR code using the Customer App. The app decrypts the QR code content and sends the details to Firebase for verification. The system compares the scanned hash with the stored hash value to check authenticity.
o If the data matches, the product is marked as "Verified", confirming its legitimacy.
o If any mismatch is found, the app alerts the user with a "Fake Product" warning.

DOI: 10.48175/IJARSCT-29725

5. AI/NLP Review Analysis

After verification, the user can submit a product review. The system analyzes each review using AI and Natural Language Processing (NLP) algorithms to detect fake, repetitive, or biased reviews. Each user receives a reputation score based on authenticity and review consistency, ensuring that only credible feedback is highlighted.

6. Result and Transparency

Verified results, customer reviews, and trust scores are displayed in real time on the app. This transparent process allows both manufacturers and customers to track product legitimacy and maintain trust in the brand ecosystem.

VI. ALGORITHMS

1) Admin "Enroll Product" (QR minting)

Goal: Create an immutable record and a QR payload that can't be forged without keys.
Inputs: product_id, batch_id, mfg_date, expiry, attrs, admin_private_key
State: last_block_hash (from Firebase), block_index++
Steps:
1. payload_core = canonical_json(product_id, batch_id, mfg, exp, attrs, block_index, last_block_hash)
2. data_hash = SHA256(payload_core)
3. sig = Ed25519.sign(admin_private_key, data_hash)
4. block = {payload_core, data_hash, sig, ts_now, issuer_pubkey, block_index}
5. Write block to Firebase at /ledger/{batch_id}/{block_index} with transaction semantics.
6. qr_payload = {product_id, batch_id, block_index, data_hash, issuer_pubkey, ts_now}
7. qr_string = base64url (CBOR (qr_payload)) → render as QR.
Update last_block_hash = SHA256(block) at /ledger_heads/{batch_id}.

2) Customer "Verify Scan" Inputs: qr_payload, phone_time Steps:
1. Parse & schema-validate qr_payload.
2. Pull /ledger/{batch_id}/{block_index} from Firebase.
3. Recompute SHA256(payload_core) and compare with qr_payload.data_hash.
4. Verify signature: Ed25519.verify(issuer_pubkey, data_hash, sig)
5. Check chain: compare payload_core.last_block_hash with /ledger_heads/{batch_id} trail by walking back or validating merkle/prev pointers.
6. Check business rules: expiry, revocation list, recall flags, scan-rate anomalies.
7. Verdict: Genuine / Revoked / Expired / Tampered / Unknown.
8. Log scan (coarse-grained, privacy-safe) at /telemetry/scans.

3) Fake-Review Detection (NLP + heuristics)

Inputs: review_text, lang, user_id, product_id** Pipeline:
1. Text cleaning: normalize, strip URLs, collapse whitespace.
2. Language detect & route.
3. Embedding: on-device or server (e.g., 384–768D sentence embeddings).
4. Classifier:
o Base: logistic regression / lightGBM on features below.
o Features: sentiment polarity; subjectivity; repetitiveness (n-gram overlap with user's past posts); burstiness vs user timeline; brand lexicon overuse; star/text mismatch; URL/affiliate density; perplexity vs product domain.
5. Graph checks: shared device/IP heuristics, co-review bursts, reciprocal rating rings (basic).
6. Output: fraud_prob ∈ [0,1], sentiment ∈ {-1,0,+1}, explanations.

DOI: 10.48175/IJARSCT-29725

V. CONCLUSION

The TrustQR system successfully integrates Blockchain Authentication, Smart QR Verification, and AI/NLP-based Review Analysis to create a secure and transparent product verification framework. By combining SHA-256 encryption with Firebase Realtime Database, the system ensures immutability and prevents tampering of product data. The admin module efficiently generates encrypted QR codes for every product, while the customer module verifies authenticity within seconds, making it practical for real-world deployment. The inclusion of AI-powered fake review detection further enhances user trust by filtering out fraudulent feedback and maintaining credible consumer insights. Through testing and validation, TrustQR proved to be a scalable, tamper-proof, and efficient solution for counterfeit prevention. It bridges the gap between manufacturers and customers, offering an ecosystem where trust, transparency, and technology converge to ensure genuine product delivery in modern marketplaces.

VI. FUTURE SCOPE

The future of the TrustQR system aims to improve product authenticity, data integrity, and scalability with new technologies. A major update is on-chain anchoring. This involves periodically linking Firebase block hashes to public blockchains like Polygon or Bitcoin. This approach offers tamper-proof verification without having to move the entire database on-chain. To prevent misuse, hardware-bound QR codes that use Android StrongBox or Play Integrity will ensure that only verified devices can create real codes. Supplier sub-ledgers will allow each supplier to keep their own hash chains while TrustQR manages a Merkle root registry for safe cross-verification and separation of trust domains.

Other advancements include zero-knowledge proofs (ZK) for confirming product details like quantity or expiration date without sharing sensitive data. There will also be fraud detection graphs to spot fake reviews using graph-based algorithms. Offline verification with short-lived signed tokens will support secure authentication without needing internet access. Additionally, recall and alert systems will quickly inform users about unsafe or recalled products. Other features include regulatory evidence lockers for audit storage, federated review models for private analysis, and counterfeit hotspot maps for identifying clusters of fake products. These improvements will make TrustQR a secure, transparent, and smart solution for product verification and consumer trust.

VII. ACKNOWLEDGMENT

The development of this project would not have been possible without the guidance and encouragement of our Project Guide and faculty members, whose valuable insights into blockchain, Android development, and artificial intelligence helped shape this system.

We would also like to thank the Department of Computer Engineering for providing the necessary infrastructure, resources, and testing facilities to complete this project successfully. Special thanks to our peers and mentors who provided constructive feedback during each development phase. Their support was instrumental in achieving this outcome.

REFERENCES

[1] H. Gupta and R. Sharma, “Transformer-Based Real-Time Mobile Inference for Text Classification,” IEEE Access, 2024.

[2] R. Verma, M. Agarwal, and K. Das, “AI-Assisted Review Authenticity Detection for E-Commerce,” IEEE Transactions on Computational Social Systems, 2024.

[3] R. Pitale, A. Gaikwad, and S. Patil, “Fake Product Identification using Blockchain,” IEEE ICIRCA, 2023.

[4] T. Wang, H. Zheng, and C. You, “A Texture-Hidden Anti-Counterfeiting QR Code and Authentication Method,” IEEE Conference on Image Processing and Security, 2023.

[5] X. Li and L. Chen, “Fake Review Detection Using Deep Neural Networks with Multimodal Feature Fusion,” IEEE ICPADS, 2023.

[6] S. Lee and J. Kim, “Blockchain-Based Supply Chain Authentication Using Secure QR Codes,” IEEE Transactions on Industrial Informatics, 2023.

[7] D. Kumar and S. Raj, "Design of Decentralized Trust Framework for IoT Authentication," IEEE Internet of Things Journal, 2023.
[8] C. Park and D. Lee, "Integrating Machine Learning for Fake Review Detection on Blockchain Platforms," IEEE BigData Conference, 2023.
[9] L. Chen and S. Zhou, "Deep Learning for Sentiment and Trust Analysis in Online Product Reviews," IEEE Transactions on Affective Computing, 2023.
[10] J. Luo, T. Xu, and M. Lin, "A Lightweight Blockchain-Based Verification Scheme for Smart Retail," IEEE Sensors Journal, 2022.
[11] K. Patel, R. Joshi, and N. Mehta, "Hybrid Cloud–Blockchain Model for Secure QR Code Verification," IEEE International Conference on Smart Computing, 2022.
[12] P. Singh and A. Kaur, "A Secure Product Verification Framework Using Blockchain and Mobile Computing," IEEE Access, 2022.
[13] S. Prathipa, R. Priyadharshini, and S. Priyanka, "Counterfeit Product Detection in Supply Chain Management with Blockchain," IEEE ICCST, 2022.
[14] K. Wasnik, P. Deokar, and A. Gohane, "Detection of Counterfeit Products using Blockchain," IEEE International Conference on Advances in Computing and Communication (ICACC), 2022.
[15] Y. Yan, H. Liu, and S. Li, "An IoT-Based Anti-Counterfeiting System Using Visual Features on QR Code," IEEE Internet of Things Journal, 2019

DOI: 10.48175/IJARSCT-29725