Ain Shams University
Faculty of Computer & Information Sciences
Information System Department

Identify Fake Products Using Blockchain Technology

By

Michael Medhat Mounir [Information System]

Mario Mamdouh Reda [Information System]

Philip Wagih George [Information System]

Michael Farid Fouad [Information System]

Kermina Ashraf Mikhail [Information System]

Under Supervision of
[Dr. Al Shaimaa Abo-Alian]
[Assistant Professor],
Information system Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

[TA. Esraa Karam]
[Teaching Assistant],
Information system Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

June 2023

Acknowledgements

First and foremost, we would like to thank everyone who has assisted us in working and deploying our graduation project, especially our supervisor Dr. Al Shaimaa Abo-Alian, who has motivated us and assisted us in overcoming every challenge we have encountered along the way. She did not spare any effort or time in aiding, encouraging, and leading us throughout the project, but instead gave us invaluable advice and information. We would have been in a bind without her ongoing help, direction, and wealth of knowledge in this field.

We would want to express our gratitude to our friends and family for their support, love, and inspiration throughout this trying road. Our successes have been motivated by their everlasting faith in our competence.

We also want to sincerely thank everyone who in any way contributed to the success of our effort. Finally, we would like to express our sincere gratitude to everyone who helped make this project a success. We are thankful for the chance to have taken on this project because your advice, support, and participation have been absolutely priceless.

Abstract

Counterfeiting is a persistent global issue that poses significant challenges to industries, consumers, and economies worldwide. This paper addresses the problem of identifying fake products through the implementation of blockchain technology. The primary objective is to establish a robust system that enables the authentication of products at each stage of the supply chain, thereby reducing reliance on traders and ensuring end-users receive genuine goods. To tackle this problem, a combined approach of decentralized blockchain technology and supply chain management is proposed. This approach allows for the tracking and verification of product authenticity from the manufacturer to the end-user. By leveraging blockchain's inherent transparency and immutability, each product's origin, manufacturing details, and subsequent transactions can be securely recorded and verified. The main results of this study reveal the effectiveness of the proposed system in combating counterfeiting. By authenticating products at every stage of the supply chain, it becomes significantly more difficult for counterfeiters to introduce fake products into the market. This enhances consumer trust, safeguards brand reputation, and helps ensure the delivery of quality goods to end-users. The conclusions drawn from this research underscore the potential of blockchain technology as a powerful tool in the fight against counterfeiting. The combination of decentralized blockchain infrastructure with supply chain management facilitates real-time product authentication, providing a transparent and immutable record of a product's journey. This not only improves consumer safety and confidence but also enables businesses to streamline their quality assurance processes and reduce costs associated with counterfeit goods.

خُلاصة

التزوير مشكلة عالمية مستمرة تواجه صناعات ومستهلكين واقتصادات العالم بأسره. يتناول هذا البحث مشكلة تحديد المنتجات المزيفة من خلال تطبيق تكنولوجيا البلوكشين. الهدف الرئيسي هو إنشاء نظام قوي يمكّن التحقق من صحة المنتجات في كل مرحلة من مراحل سلسلة التوريد، وبالتالي الاعتماد على التجار بشكل أقل وضمان تلقي المستخدمين النهائيين للسلع الحقيقية. لمعالجة هذه المشكلة، يُقترح نهج مجتمع يستخدم تقنية البلوكشين المركزية وإدارة سلسلة التوريد. يسمح هذا النهج بتتبع والتحقق من مصداقية المنتجات من الشركة المصنعة إلى المستخدم النهائي. من خلال استغلال شفافية التكنولوجيا وعدم قابلية التلاعب بها، يتم تسجيل والتحقق بأمان من أصل المنتج وتفاصيل التصنيع والمعاملات اللاحقة. توضح النتائج الرئيسية لهذه الدراسة فعالية النظام المقترح في مكافحة التزوير. من خلال التحقق من صحة المنتجات في كل مرحلة من سلسلة التوريد، يصبح من الصعب بشكل كبير للمزورين إدخال المنتجات المزيفة إلى السوق. وهذا يعزز ثقة المستهلك ويحمي سمعة العلامة التجارية ويساعد في ضمان تقديم السلع ذات الجودة للمستخدمين النهائيين. تستنتج هذه الدراسة إمكانات تكنولوجيا البلوكشين كأداة قوية في مكافحة التزوير. إن دمج البنية التحتية للبلوكشين المركزية مع إدارة سلسلة التوريد يسهم في التحقق من المنتجات في الوقت الحقيقي، وتوفير سجل شفاف وغير قابل للتلاعب برحلة المنتج. وهذا لا يحسن فقط سلامة وثقة المستهلك، بل يمكن الشركات من تبسيط عمليات ضمان الجودة وتقليل التكاليف المتعلقة بالسلع المزيفة.

Table of Contents

List of Abbreviations

UI: User Interface

DApp: Decentralized App

RFID: Radio Frequency Identification

AI: Artificial Intelligence

ML: Machine Learning

IOT: Internet of Things

EVM: Ethereum Virtual Machine

IDE: Integrated Development Environment

NPM: Node Package Manager

List of Figures

Chapter One: Introduction

1.1 Problem Definition:

We all buy goods almost daily. Every time we buy something we want to make sure that we're buying the original one. Recently, replicated, and fraudulent goods are more prevalent in this contemporary period.

As a result, many counterfeit products are produced, which causes consumer uncertainty and mistrust. Traditional methods of product verification and authentication have proven inadequate, necessitating innovative solutions to combat this problem effectively.

So, it is essential to have a system that allows customers to verify all the information about the product they are purchasing to determine whether it is authentic to improve user trust [1].

1.2 Motivation:

Building an application that identifies fake products using blockchain technology is a compelling endeavor with immense motivation. Counterfeit products not only undermine consumer trust but also pose significant risks to health, safety, and brand reputation. By leveraging the transparency and immutability of blockchain, our application aims to combat this global issue and foster a marketplace of authenticity [2].

Three major problems were the motivation behind the development of this project:

I. Counterfeit Goods:

The growing problem of counterfeit goods in the marketplace. Counterfeit products not only harm the reputation of the original brand but also pose significant risks to consumers, such as health hazards, financial loss, and even safety concerns.

II. Lack Of Control:

Current supply chain systems often lack proper control mechanisms to track and authenticate products effectively. This lack of control opens opportunities for counterfeiters to introduce fake products into the market, compromising brand reputation and consumer trust:

- If their products are duplicated by another unknown company
- How these products are existed in the market

III. Consumer Safety:

Counterfeit products can have serious consequences on consumer safety, particularly in industries such as pharmaceuticals, automotive parts, and electronics. Ensuring the authenticity and quality of products is crucial to safeguarding consumer well-being.

The market needs to promote transparency and accountability in the supply chain, making it easier to trace the origin of the product and identify any potentially fraudulent activities. As the problem of duplicating original products, it risky for consumers as:

- It may lead to financial loss.
- It may lead to safety issues.
- It may lead to death in health hazards.

In this context, it was necessary to develop an identifying fake products system using block chain that gives the possibility of ensuring that the product is original, Consumers can also benefit from this type of project by ensuring that the products they purchase are genuine and safe to use, allows companies can protect their brand reputation and prevent revenue loss due to counterfeit sales. Help to solve the previous three major problems [3].

1.3 Objectives:

To address the challenges posed by counterfeit products, this research aims to utilize blockchain technology to create a robust system for identifying fake products. The specific objectives are:

I. User-optimization:

To provide an optimized user experience by ensuring consumers have access to genuine products and accurate information about their origins. Which also needs to be highly cost-effective and time efficient.

II. Transparent:

Everyone connected to the network of the block chain can trace the transactions recorded on the block chain. Verifiable authenticity exists for every transaction. To enhance transparency in the supply chain, allowing stakeholders to trace the journey of products from manufacturer to end-user, thereby eliminating information asymmetry.

III. Decentralized:

Instead of storing the data on a single centralized server, decentralization enables the distribution of information on every node in the network, reducing the chances of a single point of failure. To leverage the decentralized nature of blockchain technology, ensuring no single entity has control over the system, thereby reducing the risk of manipulation and fraud.

IV. A Universal Ecosystem:

To establish a universal ecosystem where various stakeholders, including manufacturers, distributors, retailers, and consumers, can participate and collaborate to combat counterfeiting. Blockchain identity management doesn't set any geographical boundaries. So, users can use the system anywhere to verify their products from any company in the system.

V. Security:

To provide a highly secure environment where product data and transaction records are stored immutably on the blockchain, making them temper-proof and resistant to fraudulent activities, preventing the purchase of counterfeit products. Having more security of product details via a QR code. Also, it helps companies in identifying the products that are being duplicated or faked.

1.4 Methodology:

The methodology used to create this system includes the use of a high-end development tool called Ganache to run our own local Ethereum blockchain, a cryptocurrency wallet called MetaMask to interact with the Ethereum blockchain, the Remix IDE tool used to debug the Smart Contract Program [4].

1.5 Time Plan:

Figure 1.1: Time Plan

Developing an application that identifies fake products using blockchain technology requires careful planning and execution.
Most of the following processes were made in parallel with each other to move in sync with each other. For example, each new idea in the implementation or system functionality was added to the documentation as soon as possible.

Here is a time plan paragraph description that outlines the key steps and estimated timelines for building the application:

I. Understanding Blockchain Concepts (11 Weeks):

Dedicate time to thoroughly understand the fundamental concepts of blockchain technology, including its structure, consensus algorithms, and immutability. Familiarize yourself with various blockchain platforms and frameworks to determine the most suitable option for the application.

II. Reading Papers & Related Works (7 Weeks):

Conduct an extensive literature review and study existing research papers and related works on counterfeit detection using blockchain. Gain insights into successful case studies, approaches, potential challenges faced in similar projects.

III. Experimental Design &Development (5 Weeks):

Define the application's architecture, data models, and system requirements based on your understanding and research. Develop a detailed experimental plan to implement the counterfeit detection mechanism using blockchain technology.

IV. Implementation (17 Weeks):

Utilize the insights gained from the experimental design stage to implement the application. Develop the necessary smart contracts, backend infrastructure, and frontend components based on the chosen blockchain platform. Integrate the blockchain functionality, data management, and user interface to create a functional prototype.

V. Evaluation & System Analysis (10 Weeks):

Conduct rigorous testing and evaluation of the application's performance, accuracy, and efficiency. Collect and analyze data to assess the effectiveness of the counterfeit detection mechanism and identify areas for improvement.

VI. Documentation (Across 24 Weeks):

Document the entire development process, including the application's design, implementation details, experimental findings, and system analysis. Create user manuals, technical specifications, and API documentation to facilitate future maintenance, collaboration, and knowledge sharing.

VII. UI Modifications (4 Weeks):

Enhance the user interface based on user feedback and usability testing results. Optimize the application's visual appeal, intuitiveness, and accessibility to provide a seamless experience for users.

VIII. Testing & Closure (7 Weeks):

Conduct comprehensive testing, including functional, integration, and security testing, to ensure the application's stability and reliability. Address any identified bugs or issues and perform final refinements. Prepare the application for deployment, finalize legal and regulatory compliance, and obtain necessary certifications before closing the project.

1.6 Thesis Outline:

➢ Chapter 1: Introduction

This chapter provides an overview of the research project or system development, setting the context for the entire document. It outlines the problem statement, objectives, and research questions, highlighting the significance and relevance of the study. The chapter introduces the scope and limitations of the project and presents an outline of the subsequent chapters.

➢ Chapter 2: Literature Review

In this chapter, a comprehensive review of existing literature and related works is presented. It involves an in-depth analysis of scholarly articles, research papers, and relevant sources to identify gaps, trends, and insights related to the topic. The literature review provides a theoretical framework and supports the research methodology adopted in the study.

➢ Chapter 3: System Architecture

This chapter delves into the technical aspects of the system architecture. It outlines the overall design, components, and infrastructure of the developed application. The chapter discusses the selection of technologies, frameworks, and tools used in the development process.

➢ Chapter 4: System Implementation & Results

This chapter focuses on the practical implementation of the system. It details the development process, including coding practices, algorithms, and data structures utilized. The chapter presents the results of the system implementation, such as performance metrics, accuracy rates, and validation tests.

- ## Chapter 5: Run the Application

  This chapter provides instructions on how to run and utilize the developed application. It guides users through the installation process, system setup, and configuration steps. The chapter explains the application's user interface, features, and functionalities.

- ## Chapter 6: Conclusion & Future Work

  The final chapter summarizes the key findings, contributions, and implications of the research or system development. It reflects on the objectives stated in the introduction and addresses the research questions. The chapter discusses the significance of the results and their potential impact on the field. Additionally, it suggests areas for future research, enhancements, or improvements to extend the system's capabilities and address any limitations identified during the project.

  These chapter descriptions provide a general understanding of the content covered in each chapter.

Chapter Two:
Literature Review

2.1 Introduction:

In this chapter, a comprehensive literature review is conducted to explore the existing knowledge and research related to identifying fake products using blockchain technology. The literature review aims to provide a solid foundation for understanding the concepts, technologies, and methodologies relevant to the project. By examining previous studies, this chapter sets the stage for the subsequent chapters, highlighting the gaps in the current knowledge and identifying the research objectives and contributions of the project. The problem of counterfeit products has been a persistent and growing issue for businesses and consumers worldwide, leading to significant economic, social, and health consequences. Various technologies and strategies have been developed to tackle this problem, but they often fall short due to their limitations and vulnerabilities. Blockchain technology has emerged as a promising solution to address the challenges of counterfeit products, leveraging its unique features of immutability, transparency, and decentralization. This literature review aims to provide an overview of the existing research on using blockchain in supply chain management and anti-counterfeiting efforts and to identify the strengths and weaknesses of current approaches [5].

2.2 Technology Background:

The theoretical background of the application includes many concepts and frameworks related to user experience, e-commerce, and blockchain technology. Here are some foundations of the development:

I. Blockchain Technology:

Blockchain is a distributed ledger technology that allows for secure and transparent transactions without the need for intermediaries. It is based on a decentralized network of nodes that collectively validate and record transactions, ensuring their integrity and immutability. It has several features that make it suitable for supply chain management and anti-counterfeiting efforts [6], including:

- **Immutability**:
  Once a transaction is recorded on the blockchain, it cannot be altered or deleted, ensuring the integrity and authenticity of the data.
- **Transparency:**
  All participants in the blockchain network can access and verify the transaction data, promoting transparency and accountability.
- **Security:**
  Blockchain uses advanced cryptographic techniques to ensure the confidentiality and authenticity of the data.
- **Decentralization:**
  Blockchain eliminates the need for intermediaries and central authorities, reducing the risk of fraud and boosting efficiency.

II. Scientific Background of the Blockchain:

The first decentralized blockchain was conceptualized by a person (or group of people) known as Satoshi Nakamoto in 2008. Nakamoto improved the design in an important way using a Hash cash-like method to timestamp blocks without requiring them to be signed by a trusted party and introducing a difficulty parameter to stabilize rate at which blocks are added to the chain. The design was implemented the following year by Nakamoto as a core component of the cryptocurrency Bitcoin which serves as the public ledger for all transactions on the network. Each block in the blockchain consists of:

1. Data
2. Hash of the current block
3. Hash of the previous block

When the first block is created the current hash of this block is the previous hash of next block [7]

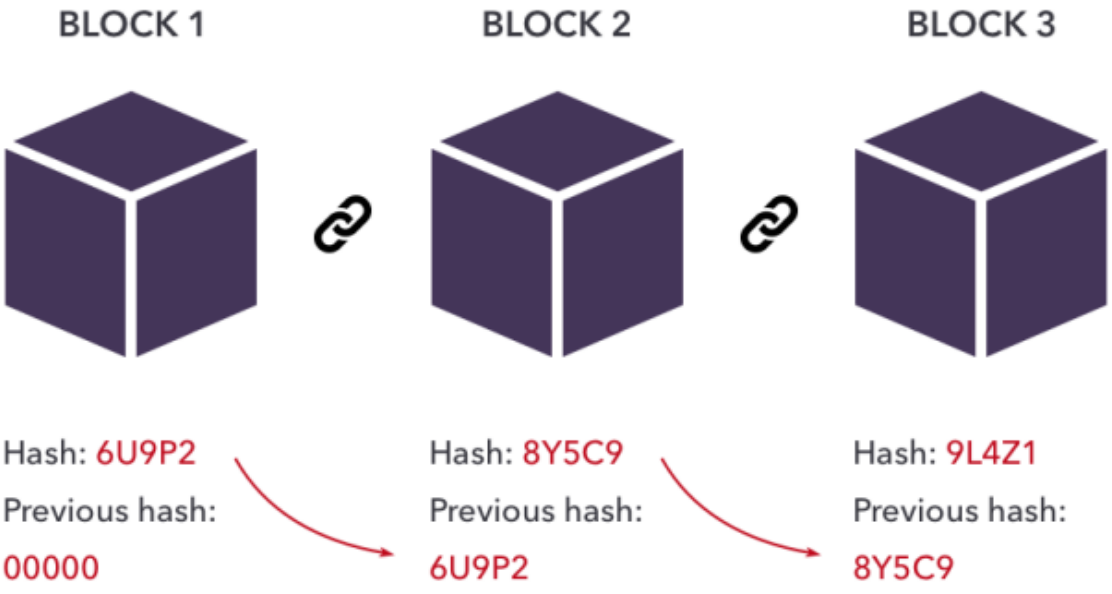

Figure 2.1: Simple Blockchain View

III. Anti-Counterfeiting Efforts:

Counterfeit products pose a significant threat to consumer safety and brand reputation, leading to economic losses and legal liabilities. Various approaches have been developed to tackle this problem, including:

- **Physical authentication:**
  This involves using physical features, such as holograms, barcodes, and serial numbers, to verify the authenticity of products. However, these features can be easily replicated or tampered with, reducing their effectiveness.
- **Digital authentication:**
  This approach involves using digital technologies, such as QR codes and RFID tags, to track and verify the authenticity of products. However, these technologies can be vulnerable to hacking and data breaches.
- **Supply chain management:**
  This approach involves using supply chain management systems to track and verify the movement of products from the source to the consumer. However, these systems can be complex and expensive to implement, and they may not be able to detect all instances of counterfeiting [8].

IV. Use of Blockchain in Anti-Counterfeiting Efforts:

Blockchain technology has been proposed as a potential solution to address the limitations of existing anti-counterfeiting approaches. Several studies have explored the use of blockchain in supply chain management and anti-counterfeiting efforts. It examines how blockchain's characteristics, such as immutability, transparency, and decentralization, can enhance the authentication and verification of products. The use of smart contracts and distributed ledger technology in creating secure and trustworthy systems is explored [9].

V. Limitations and Challenges:

Despite the potential benefits of using blockchain in anti-counterfeiting efforts, several limitations and challenges need to be addressed, including:

- **Scalability:**
  Blockchain technology can be slow and resource-intensive, making it difficult to scale up to large-scale supply chain networks.
- **Interoperability:**
  Blockchain networks may not be compatible with existing systems, requiring significant investment in infrastructure and integration.
- **Education and adoption:**
  Blockchain technology is still relatively new and complex, requiring education and adoption among stakeholders to achieve its full potential [10].

2.3 Related Works:

These systems are used within the boundaries of a single organization, while our system is developed for public use. Another remarkable difference is that our system is based on blockchain, which is not the case with the other systems. Various related works have explored different concepts and approaches to address the challenges of counterfeit products and improve customer satisfaction. These include database systems for product verification, real-time feedback mechanisms, integration of blockchain technology, and leveraging AI and ML algorithms.

Project Paper	Project Name	Author	Year	Technologies	Limitations & Weaknesses	Database Description
(ICCCI) Coimbatore, India	Blockchain Based Inventory Management by QR Code	G. Vidhya Lakshmi, S. Gogulamudi	27 –29 Jan 2021	OpenCV, Python, Peer to Peer Network	Using python to generate QR code which can be improvised by using algorithm in blockchain technology	Blockchain (EVM) Ethereum Virtual Machine
(IJRTE) ISSN: 2277-3878	Detecting Fake Drugs using Blockchain	A. Sanghi, Aayush, A. Katakwa	1, May 2021	Hyperledger Fabric	It has shown a lack of use cases. It has got a complex architecture. It has got minimum APIs and SDKs	LevelDB and CouchDB (Private State Databse)
(IJSRCSEIT) ISSN : 2456-3307	Product Identification System Using Block chain	Sushil Kumar, Vikas Gupta, Yash Pachori	15 - 22 May 2021	AES 128, Wampserver	heavy amount of internal component software. Not Cross Platform.	MySQL
(ICSCC) 2019.8843597	Fake News Detection in social media using blockchian	Shovon Paul, Jubair Islam Joy, Shaila Sarker	14 June 2019	Breadth First Search News Rating Statistics	It's difficult to detect the news based on politics and religion. For its veridical verification system, journals and news portals had to face job risk	Blockchain (EVM) Ethereum Virtual Machine

Figure 2.2: Related Paper Articles

It provides insights into the different approaches, methodologies, and outcomes of previous work. By examining related works, the project aims to build upon existing knowledge and contribute novel insights to the field.

In the realm of combating counterfeit products and ensuring customer satisfaction, various projects and initiatives have been undertaken, each with its unique concepts and approaches. One notable example is the implementation of robust database systems for product verification. These projects leverage advanced database technologies to create centralized repositories of product information, enabling users to authenticate and validate the authenticity of products. By storing relevant data, such as unique identifiers, manufacturing details, and supply chain records, in these databases, users can easily access and verify product information, enhancing transparency and trust [11].

	Our Product	chekkit	CHKFAKE
Intended Customer	Open Source for Everyone	Available for Download	Intended for Organizational Use
Database Management System	Block Chain Based Database	SQL Based Database	SQL Based Database
Access Certificate	✓	✓	✗
Application Platform	Web Application	Mobile Application	Mobile Application
Price	Free	Free	Paid
Customer Satisfaction	Not yet Applicant	★★★★	★★★

Figure 2.3: Related Applications

In Fariha Jahan, Mayel Mostafa, Shahrin Chowdhury "SHA-256 in Parallel Blockchain Technology: Storing Land Related Documents" International Journal of Computer Applications (0975 – 8887) Volume 175 – No. 35, December 2020 displayed a fake product detection system using blockchain where SHA-256 Algorithm was used to identify a product. A fully functional anti-product forgery system was designed by a group of researchers that uses digital signatures for verification. Product Ownership Management System was proposed. It displayed the use of blockchain-based systems over traditional RFID-based system. Blockchain was used to remove the limitation in post supply chain. To improve the current supply chain method a paper used blockchain combined with IoT to track product origin [12].

Abhinav Sanghi, Aayush, Ashutosh Katakwar, Anshul Arora, Aditya Kaushik, "Detecting Fake Drugs using Blockchain", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-10 Issue-1, May 2021. proposed a blockchain-based system to help in tracking the movement of drugs from the industry to the patient. Mainly, the Hyperledger fabric is used for implementing the entire model. In this model, the manufacturer must upload the details of a drug to a website which is sent further to the government for approval. Once the government approval is done, the pharmacies can request the approved drugs with the help of blockchain technology. Further, if any patient needs to get some medicine or drugs, then a request is made to the blockchain network. After that, a medical officer or doctor will approve or reject the request. Because the entire model is implemented in a blockchain network, it can help in preventing the counterfeiting of drugs and the movement of drugs can be tracked from the manufacturer up to the patient [13].

G. Vidhya Lakshmi, Subbarao Gogulamudi, Bodapati Nagaeswari, Shaik Reehana, "Blockchain-Based Inventory Management by QR Code Using Open CV", International Conference on Computer Communication and Informatics (ICCCI - 2021) Coimbatore, INDIA, Jan. 27 – 29, 2021 utilized blockchain technology and Python to generate QR codes. In other words, they used the features of blockchain and QR codes to create a reliable and transparent inventory management system. Using Python, they can create QR codes that are customized for different products. The details of the sold products are then broadcasted through the P2P network. A manufacturer can quickly compute inventory by retrieving product details from the blockchain database [14].

2.4 Conclusion:

Blockchain technology has the potential to transform supply chain management and anti-counterfeiting efforts, leveraging its unique features of immutability, transparency, and decentralization. However, several challenges need to be addressed to ensure its effective implementation, including scalability, interoperability, data privacy, and education and adoption. Future research should focus on developing practical and scalable solutions that leverage blockchain's strengths while overcoming its limitations.

In conclusion, this chapter provides a comprehensive literature review that establishes the background and context for the project. It explores the technological foundations of blockchain, discusses existing anti-counterfeiting efforts, examines the use of blockchain in combating counterfeiting, and highlights the limitations and challenges associated with its implementation. The review of related works sets the stage for the subsequent chapters, where the project's methodology, implementation, and findings will be presented.

Chapter Three:
System Architecture

3.1 System Architecture:

At the core of the architecture is the blockchain network, which serves as a distributed ledger for recording and storing product-related transactions and information. The blockchain network is decentralized, ensuring that no single entity has control over the system and reducing the risk of manipulation or tampering. It provides immutability, meaning that once data is recorded, it cannot be altered, ensuring the integrity of the information stored on the blockchain.

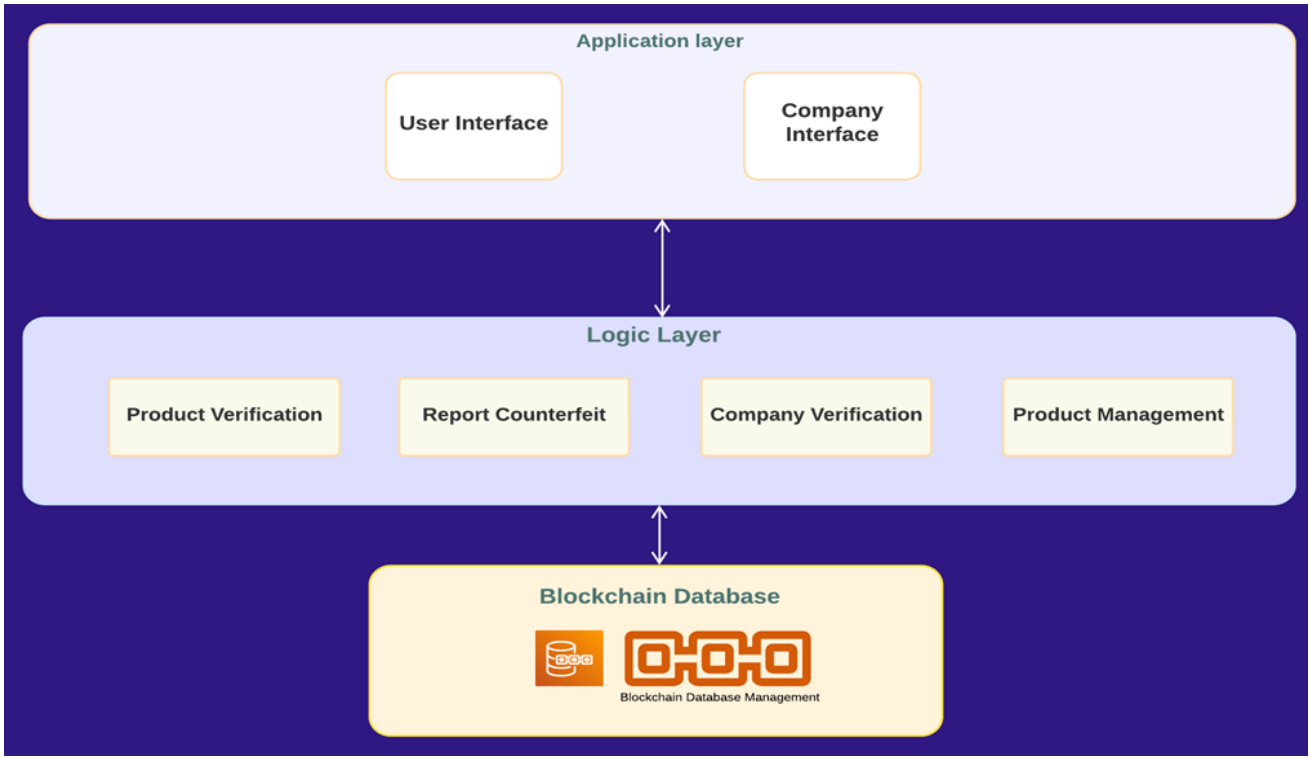

Figure 3.1: System Architecture

Explanation of the System Architecture:

The Application's system architecture is divided into three layers, Blockchain Database, logic layer and the application layer with the user interface. Basically, the Blockchain Database layer indicates the used database in the project. Including the companies' database with their products also it carries the reports of the customer's complaints. The Application layer is where all the functions are placed in, such as Product Verification, Report Counterfeit, Company Verification and Product Management. Finally, we have the Application Layer, is where the user can interact with the system thought a great User Experience, Such as scan the QR code either scan with camera or upload an image, Also user can make a report if the product's fake. We created this layer using ReactJS.

Every user of the DApp must be authenticated before logging in. This authentication system has been implemented using the solidity code. After successful authentication, the company can add their product to the DApp and enroll products of the company. The data of the company is provided to the website and stored in the blockchain network. After a product has been included in the blockchain, it is assigned a QR code for verification. The users can verify products from the manufacturer after registration. The ownership transfer of the product can be tracked through the QR code. This process is show in the system architecture.

3.2 Description of the Methods & Procedures Used:

3.2.1 Functional Requirements:

Functional requirements shown in this section explain a software system's potential outcomes of what the system must achieve.

I. User Functions

1. User Registration:

- **Description:** The user registration function develops smart contracts on the Ethereum blockchain. This function allows new users to register by providing their personal information and creating a new user account in the system. It is commonly used in decentralized applications (DApp) where user registration and identity management are required.
- **Pre/Post Conditions:** The prerequisite for using the register user function is that it gets accurate user data, including usernames and emails that serve as distinctive identifiers as well as other pertinent information like names, addresses, and phone numbers. The function of successfully registering the user and creating a new user account in the system is the post-condition allowing the user to scan products and verify its authenticity.
- **Inputs:** Username, Password, Email & MetaMask Address.
- **Outputs:** The user is registered successfully to the system.
- **Processing:**
    - Check and Receive the user's personal information.
    - Check for the uniqueness of the provided identifier.
    - If the identifier is unique, create a new user account with the provided information and store it in a user mapping.
    - Emit an event or return a successful user registration.
    - If the identifier is not unique, handle the error condition appropriately, such as throwing an exception.

2. User Login:

- **Description:** This function verifies the user's credentials and grants access to their account, allowing them to interact with the application's features and perform authorized actions.
- **Pre/Post Conditions:** The user must supply valid login credentials, which normally consist of a distinctive identifier, like a username or email, and a corresponding password, in order to utilize the login functionality. The function successfully authenticates the user's credentials & grants access to accounts.
- **Inputs:** Username, Password & MetaMask Address.
- **Outputs:** Boolean expression indicating whether the user is logged in successfully to the system or not.
- **Processing:**
    - Check and Receive the user's login credentials.
    - Retrieve the user's account information based on the provided identifier which is the username & password.
    - Verify that the provided password matches the stored password for the corresponding user account.
    - If the password is correct, grant access to the user's account and return a successful login.
    - If the password is incorrect, handle the error condition appropriately, to inform the user that the login failed due to invalid credentials.

3. Scan QR Code:

- **Description:** Allows users to scan QR codes in real time using their device's camera or upload the corresponding QR code image. This function facilitates quick and efficient data capture and image uploading for various applications such as inventory management, ticket verification, or product authentication.
- **Pre/Post Conditions:** The precondition includes the user having access to a device with a camera and the application having the necessary permissions to access the camera. The function must correctly scan the QR code and upload the corresponding image for the operation to succeed.
- **Inputs:** QR Code.
- **Outputs:** Keccak 265 Hash for the scanned product.
- **Processing:**
    - Activate the device's camera and stream the live feed to the application to identify the QR Code.
    - Analyze the camera for QR codes, detecting and decoding them in real-time.
    - Once a QR code is successfully scanned, capture the corresponding image from the camera feed.
    - Upload the image to a designated server or storage location, associating it with the scanned QR code data.
    - Calculate the hash of the given product and store it in the blockchain system with the product data and company in correspondence.
    - Provide feedback to the user indicating the successful scanning and image upload.

4. Verify Product:

- **Description:** Allows users to verify the authenticity of a product using the product produced hash. This function ensures that users can authenticate a product and confirm its legitimacy before making a purchase or using it.
- **Pre/Post Conditions:** The action of scanning the QR code, which creates the hash that serves as the product's unique identity, is the prerequisite. The function successfully confirms the product's legitimacy using the supplied data.
- **Inputs:** Product Keccak 265 Hash.
- **Outputs:** Product Id, Name, Model, Description, Company Name, and Image Link.
- **Processing:**
    - Obtain the product information through the QR code
    - Access a product database or external source to retrieve the product hash unique identifier.
    - Compare the product hash with the expected attributes of an authentic product.
    - Perform validation checks through cryptographic verification by verifying the hash values associated with the product.
    - Determine the authenticity of the product based on the comparison and validation results.
    - Response indicating whether the product is verified as genuine or potentially counterfeit.
    - Provide feedback for the user showing the verification outcome.
- **Modification:** The old version of this function only returned a Boolean True or False indicating whether the product is fake or authentic but now it returns the product information is authentic and Null values if fake.

5. Send Report:

- **Description:** Enables people to provide feedback or reports about different problems or situations. Users can utilize this feature to voice their concerns, offer feedback, or report any instances of spotting counterfeit goods while utilizing the program or interacting with its features.
- **Pre/Post Conditions:** The user has access to the application's reporting capabilities and has relevant data or details to include in the report. The function successfully delivers the report, ensuring that the company is informed of the user's concerns or feedback for follow-up action.
- **Inputs:** Location, Description, Company Name.
- **Outputs:** The report is stored successfully in the mapping of the system and sent to the corresponding company.
- **Processing:**
    - Obtain Receive the user's report as input, including the description of the issue or incident,
    - Validate the report to ensure that it contains necessary information and meets any predefined criteria.
    - Assign a unique identifier or reference number to the report for tracking purposes.
    - Store the report in a designated mapping in the system
    - Send acknowledgments to the user, indicating that their report has been successfully submitted.
    - Send the report to the company that a fake product related to it has been identified to notify it about the incident of counterfeiting.

II. Company Functions

1. Company Registration:

- **Description:** Enables businesses or organizations to create their presence within the system and register their information. Companies can use this feature to set up their profiles, supply pertinent information, and finish the registration process required to be recognized as businesses within the program.
- **Pre/Post Conditions:** The business has the necessary data and paperwork to finish the registration process. The function successfully registers the business and creates its profile within the program.
- **Inputs:** Company Name, Password, Email & MetaMask Address, Company Certificate Number.
- **Outputs:** The user is registered successfully to the system.
- **Processing:**
    - Receive the company's information as input, including the legal name, email address, password, and any additional required information or documentation.
    - Validate the provided information to ensure completeness and correctness.
    - Check for any specific requirements or eligibility criteria for company registration within the application.
    - Create a new company profile within the system, associating it with the provided information.
    - Store the company's profile and relevant details in a company mapping in the blockchain system.
    - Return the successful registration of the company.
    - If any errors or issues are encountered during the registration process, display error messages indicating that the registration data is not acceptable.

2. Company Login:

- **Description:** This function verifies the companies' credentials and grants access to their account, allowing them to interact with the application's features and perform authorized actions.
- **Pre/Post Conditions:** The company must supply valid login credentials, which normally consist of a distinctive identifier, like a username or email, and a corresponding password, in order to utilize the login functionality. The function successfully authenticates the company's credentials & grants access to its accounts.
- **Inputs:** Username, Password & MetaMask Address.
- **Outputs:** Boolean expression indicating whether the company is logged in successfully to the system or not.
- **Processing:**
    - Check and Receive the company's login credentials.
    - Retrieve the company's account information based on the provided identifier which is the username & its corresponding password.
    - Verify that the provided password matches the stored password for the corresponding company account.
    - If the password is correct, grant access to the company's account and return a successful login.
    - If the password is incorrect, handle the error condition appropriately, to inform the company that the login failed due to invalid credentials.

3. Upload Products:

- **Description:** Enables signed-up companies or institutions to post their items to the network. Companies can use this capability to showcase and promote their products within the application by adding product specifications, descriptions, photos, and other relevant information.
- **Pre/Post Conditions:** The company has the required authorization to upload items and is logged in. The function must correctly upload the goods to the program so that they may be browsed, searched for, and potentially purchased by users.
- **Inputs:** Array of Products send through a CSV containing the product's information.
- **Outputs:** The products are added successfully to the mapping of products and return an array containing the calculated hashes for each product sent to the system
- **Processing:**
    - Receive the product information from the company as input, including the product name, description, model, images, and any other relevant details.
    - Validate the provided information to ensure completeness and correctness.
    - Create a new product entry within the application's database, associating it with the provided information.
    - Store the product details, images, and other related information in a designated product mapping.
    - Assign a unique identifier which is the calculated hash for identification and tracking purposes.
    - Return a success status to indicate the successful upload of the products, informing the company that their products are now available for users to verify.

4. Receive Products Reports:

- **Description:** Enables registered companies to obtain and handle user-submitted reports. Companies may effectively handle and address consumer complaints, suggestions, or issues pertaining to their goods, services, or general customer experience.
- **Pre/Post Conditions:** The business has access rights and is logged in to access and manage user reports. The function correctly receives and registers the reports, enabling the business to take the necessary measures for the response.
- **Inputs:** Company Name.
- **Outputs:** Array of reports submitted to the company.
- **Processing:**
    - Store the reports in a designated mapping associating them with the corresponding company names.
    - Assign unique identifiers to each report for tracking and reference purposes for each company.
    - Notify the companies about the newly received reports, through the reports tab designated for each company.
    - Provide an interface company representative to access and manage the received reports, to review, prioritize, assign, or take necessary actions.

5. Delete Report:

- **Description:** Gives registered companies or organizations the ability to remove particular reports that have been submitted by users. Companies now have the option to delete reports from their system if they are no longer essential, relevant, or necessary to keep them.
- **Pre/Post Conditions:** The organization is logged in and has the required rights to remove reports. The function must successfully remove the requested report from the company's system and the designated mapping.
- **Inputs:** Company Name, Report Index.
- **Outputs:** The specific report is successfully deleted from the company mapping.
- **Processing:**
    - Receive the identifier of the report to be deleted.
    - Validate the input to ensure its correctness and that the report has a valid index in the mapping.
    - Locate the specified report within the company's system based on the provided identifier.
    - Confirm the deletion action with appropriate safeguards to prevent accidental deletion.
    - Remove the report from the mapping, ensuring that it is no longer accessible or retained within the system.
    - Return a success status to indicate the successful deletion of the report.

6. Delete All Reports:

- **Description:** Allows registered companies to remove all user-submitted reports. Companies now have the option to delete every report from their system thanks to this feature, which may be required in specific circumstances like system maintenance, data cleanup, or when reports are no longer required.
- **Pre/Post Conditions:** The company is logged in and has the required rights to remove reports. The function must successfully erase all reports from the company's system for there to be no reports left accessible or saved.
- **Inputs:** Company Name.
- **Outputs:** All reports have been deleted successfully and the mapping of the company reports is now empty.
- **Processing:**
    - Confirm the intent to delete all reports, through a confirmation step.
    - Validate the confirmation and ensure that the company has the necessary authorization to delete all reports.
    - Remove all reports from the reports mapping, ensuring that none of the reports are accessible or retained within the system.
    - Return a success status to indicate the successful deletion of all reports.

7. Retrieve Company Products:

- **Description:** Enables registered companies to access and retrieve the products they have submitted. Companies can browse and manage their product listings and carry out other related tasks using this service.
- **Pre/Post Conditions:** The company has the required authorization to access the product data. The function's ability to effectively obtain and provide the company's product listings so they can browse and manage their products.
- **Inputs:** Company Name.
- **Outputs:** Array of products containing all the products produced by a specific company.
- **Processing:**
    - Validate the company's authentication and authorization to access its product data.
    - Retrieve the product listings associated with the company's account from the products mapping.
    - Apply any specified parameters or filters to refine the retrieved product listings by company name.
    - Format and present the product data to the company, through a designated page for the company products.
    - Provide options for the company to modify or perform other actions on their products.
    - Handle any potential errors or exceptions that may occur during the retrieval process, ensuring a graceful user experience and appropriate error messaging.

8. Delete Product by Model:

- **Description:** Allows registered companies to remove a specific product from their listings based on its model or identification. Companies can use this function to remove products from their inventory or offerings when they are no longer offered, have been discontinued, or for other uses.
- **Pre/Post Conditions:** The business is logged in and has the required rights to remove products. The function must correctly remove the requested product from the company's listings based on its model in order for it to no longer be visible or accessible.
- **Inputs:** Company Name, Product Model.
- **Outputs:** Remove all products in the company mapping that follows the filtration criteria by the model of the product.
- **Processing:**
    - Receive the model of the product to be deleted.
    - Validate the input to ensure its correctness and that the specified model is actually presented.
    - Locate the specified product within the company's listings based on the provided model.
    - Confirm the deletion action with appropriate safeguards to prevent accidental deletion.
    - Remove the product from the products mapping ensuring that it is no longer accessible or displayed within the company's listings.
    - Return a success status to indicate the successful deletion of the product.

3.2.2 Non-Functional Requirements

1. **Performance:**

   To guarantee a flawless and responsive user experience, the program must operate at an exceptionally high level of performance. Aiming for a response time of less than one second, it should offer low latency real-time product authenticity verification. To ensure that consumers can swiftly validate products without delays or timeouts, the system should be able to handle a large volume of verification requests concurrently.

2. **Security:**

   The application must give security measures top priority because product authentication is a critical process. To safeguard user data during transmission and storage, it should utilize robust encryption mechanisms. Mechanisms for user authentication and access control ought to be set up to stop unauthorized access to the system. To preserve user confidence and protect their privacy, compliance with data protection and privacy rules should be enforced.

3. **Scalability:**

   The program must scale smoothly as the user base expands and the number of products requiring verification rises. It should have a horizontally scalable architecture that enables scalability, ensuring that more resources may be added to meet rising demand. Thousands of concurrent verification transactions should be supported by the system without any performance or responsiveness deterioration.

4. Availability:

High availability must be guaranteed by the program to guarantee users' uninterrupted service. To reduce downtime, it should be built with redundant systems and fail over procedures. To ensure consistent access to product verification services and allow users to authenticate products with no interruptions of service outages.

5. Reliability:

To retain reliability in the product verification process, it is essential to ensure the application's dependability. To manage unforeseen mistakes or failures, the system must have strong fault tolerance techniques. This will reduce downtime and guarantee continuous availability. To protect against data loss or corruption, frequent data backups and disaster recovery protocols should be in place. The system should be available 24 hours a day, and the system shouldn’t crash if invalid input is entered and produces a suitable error.

6. Ease of Use:

The application should prioritize ease of use to ensure a seamless and intuitive experience for users. The user interface is designed with a user-centric approach, offering clear instructions and intuitive navigation. The product verification process should be simple, requiring minimal effort and technical expertise from the users. Visual cues, tooltips, and informative feedback should be provided to guide users through each step of the verification process. The application should also support various scanning methods, such as QR code scanning or image recognition, to accommodate different user preferences and accessibility needs. Individuals shall spend 10-15 minutes learning how to use the system.

Chapter Four:
System Implementation & Results

4.1 Description of Materials (Datasets):

4.1.1 Nike products Data set

A	B	C	D	E	F	G	H
ID	Name	Model	Description	LinkImage			
1	Nike Air Force 1	airForce	The first Air Jordan was produced for use by Michael Jor	https://static.nike.com/a/images/t_PDP_1280_v1/f_auto,q_auto:eco/b7d92			
2	Nike Air Jordan I	AirJordan	The first Air Jordan was produced for use by Michael Jor	https://static.nike.com/a/images/t_PDP_1280_v1/f_auto,q_auto:eco/2028cf			
3	Nike Air Max 90	AirMax	Nothing as fly, nothing as comfortable, nothing as proven	https://static.nike.com/a/images/t_PDP_1280_v1/f_auto,q_auto:eco/0afb8e			
4	Nike Men's Invincible R	Invincible Run	The Invincible 3 has high-support technology, with cushic	https://encrypted-tbn2.gstatic.com/images?q=tbn:ANd9GcRNGjtoIDlZUOpr			
5	Nike Men's Pegasus 40	Running	The Nike Air Zoom Pegasus 40 is a daily trainer with a st	https://images.stockx.com/images/Nike-Zoom-Mercurial-Vapor-15-Elite-FG			
6	Nike Free Metcon 5	Men's Training Shoes	the Nike Free Metcon 5 can meet you in the depths, help	https://static.nike.com/a/images/t_PDP_1728_v1/f_auto,q_auto:eco/fbbdd9			
7	Nike City Connect	Men's T-Shirt	The Nike City Connect (MLB Los Angeles Angels) T-Shir	https://static.nike.com/a/images/t_PDP_1728_v1/f_auto,q_auto:eco/673ae			
8	San Diego Wave Esser	Women's Nike Soccer Shor	For day games and warm summer nights, keep cool and	https://static.nike.com/a/images/t_PDP_1728_v1/f_auto,q_auto:eco/89555			
9	Golden State Warriors	Men's Jordan NBA Jacket	Throw on this lightweight Golden State Warriors Jacket w	https://static.nike.com/a/images/t_PDP_1728_v1/f_auto,q_auto:eco/747e1			
10	Nike Sideline Coach Lo	Men's Short-Sleeve Jacket	The Nike Sideline Coach Lockup (NFL Kansas City Chie	https://static.nike.com/a/images/t_PDP_1728_v1/f_auto,q_auto:eco/a5e60			
11	Nike Air Force 2	airForce	The first Air Jordan was produced for use by Michael Jor	https://static.nike.com/a/images/t_PDP_1280_v1/f_auto,q_auto:eco/b7d92			
12	Nike Air Jordan I	AirJordan	The first Air Jordan was produced for use by Michael Jor	https://static.nike.com/a/images/t_PDP_1280_v1/f_auto,q_auto:eco/2028cf			
13	Nike Air Max 91	AirMax	Nothing as fly, nothing as comfortable, nothing as proven	https://static.nike.com/a/images/t_PDP_1280_v1/f_auto,q_auto:eco/0afb8e			
14	Nike Men's Invincible R	Invincible Run	The Invincible 3 has high-support technology, with cushic	https://encrypted-tbn2.gstatic.com/images?q=tbn:ANd9GcRNGjtoIDlZUOpr			
15	Nike Men's Pegasus 40	Running	The Nike Air Zoom Pegasus 40 is a daily trainer with a st	https://images.stockx.com/images/Nike-Zoom-Mercurial-Vapor-15-Elite-FG			
16	Nike Free Metcon 6	Men's Training Shoes	the Nike Free Metcon 5 can meet you in the depths, help	https://static.nike.com/a/images/t_PDP_1728_v1/f_auto,q_auto:eco/fbbdd9			
17	Nike City Connect	Men's T-Shirt	The Nike City Connect (MLB Los Angeles Angels) T-Shir	https://static.nike.com/a/images/t_PDP_1728_v1/f_auto,q_auto:eco/673ae			
18	San Diego Wave Esser	Women's Nike Soccer Shor	For day games and warm summer nights, keep cool and	https://static.nike.com/a/images/t_PDP_1728_v1/f_auto,q_auto:eco/89555			
19	Golden State Warriors	Men's Jordan NBA Jacket	Throw on this lightweight Golden State Warriors Jacket w	https://static.nike.com/a/images/t_PDP_1728_v1/f_auto,q_auto:eco/747e1			

Figure 4.1: Company products Data Set

The company data set is collected from real-life products found on companies' sites like Nike and Adidas data set contains 5 columns.
ID, Name, Model, Description, Link Image these fields used to provide more information about the products and help us represent the company product.

1. **ID** column
   the product ID refers to a unique identifier assigned to each product. It is a numerical value that serves as a unique reference for a particular product. The product ID helps to uniquely identify and differentiate one product from another within the system.

2. **Name** Column
   the product name refers to the name or title assigned to a specific product. It is a string value that provides a descriptive or recognizable name for the product. The product name is typically a human-readable label that helps users or participants in the system identify and distinguish different products from one another.

3. **Model** Column
   It is used to differentiate between different variations or versions of a particular product. The product model helps to identify and categorize products based on their specific attributes, features, or specifications. It can be a string or any other suitable data type that uniquely represents a product model.

4. **Description** Column
   the product description refers to a textual description or details about a specific product. It provides additional information about the product, highlighting its features, specifications, characteristics, or any other relevant details that help users understand the product better.

5. **Link image** Column
   URL or file path pointing to an image associated with a specific product. It is used to reference the location of the image file that visually represents the product. The link image can be used to display the product image on a website, application, or any other user interface.

4.2 Blockchain Implementation & Operational Flow:

- **Blockchain Implementation:**

To implement the blockchain for identifying fake products in the supply chain, Ethereum was selected as the blockchain platform due to its robust smart contract capabilities and widespread adoption in various industries. The Ethereum blockchain offers a decentralized and transparent network, making it suitable for enhancing supply chain traceability and trust. Smart contracts were developed using Solidity programming language to automate the verification and tracking processes. The smart contracts were deployed on the Ethereum blockchain network using the Remix IDE and the Truffle development framework. Additionally, to address the scalability challenges associated with Ethereum. Through this blockchain implementation, the aim was to establish an immutable and auditable ledger to securely record product information, transactions, and supply chain events, thereby enhancing transparency, traceability, and trustworthiness within the supply chain ecosystem.

Solidity, specifically designed for smart contract development on the Ethereum blockchain, provided the necessary tools and syntax to define the logic and behavior of the contracts. Truffle, an industry-standard development framework, was employed for the compilation, testing, and deployment of the smart contracts. Truffle's suite of tools, including Ganache for local development, helped in simulating the blockchain environment and conducting thorough testing of the smart contracts before deployment to the main network. By utilizing Solidity and the Truffle framework, the aim was to create robust and reliable smart contracts that enforce the business rules, enhance supply chain transparency, and facilitate the identification of counterfeit products in the supply chain system. The supply chain management paradigm offers promising potential thanks to blockchain technology. On nodes, where each node has a full copy of the blockchain database, blockchain data is kept.

The following are some crucial characteristics of blockchain technology for supply chain management:

1) **Security and Privacy:**
   Public key cryptography is a technique used by blockchain to secure data. Users have a public and private key pair that is used to validate transactions, which are unchangeable and irreversible.
2) **Decentralization:**
   Because blockchain uses a distributed ledger, it is not dependent on a centralized authority or third party.
3) **Transparency:**
   Blockchain data is public, and anyone can check its transactions. A set of guidelines known as a smart contract are used to control the transactions. Ethereum blockchain's Ganache Test Network and the MetaMask cryptocurrency wallet for all transactions. The Manufacturer, the Seller, and the Consumer are the three main stakeholders in the DApp.

- **Operation Flow:**

The operational flow for identifying fake products using blockchain in the supply chain system involves a series of interconnected steps to ensure transparency, traceability, and authenticity. First, the product information, including unique identifiers and manufacturing details is recorded on the blockchain starting from the company. This information is stored in a decentralized and immutable manner, providing an auditable trail of the product's journey. As the product moves along the supply chain, smart contracts embedded within the blockchain network automatically verify and validate its authenticity at key checkpoints. These smart contracts employ predefined rules and algorithms to compare the recorded product information with the expected characteristics of a genuine product. This verification process helps identify any discrepancies that may indicate the presence of counterfeit items. In case a potential counterfeit product is detected, an alert is triggered, notifying the relevant stakeholders, such as manufacturers or distributors. This real-time notification enables swift action to be taken to investigate further and mitigate the risks associated with the presence of fake products within the supply chain. To enhance the identification process, additional mechanisms can be implemented, such as integrating anti-counterfeiting technologies like QR codes are being used. These technologies allow for seamless scanning and verification of product information, further strengthening the accuracy and efficiency of the identification process. Ultimately, the combination of blockchain technology, smart contracts, and anti-counterfeiting technologies forms an integrated system that empowers supply chain participants to identify fake products, minimizing their impact quickly and reliably on consumer safety, brand reputation, and economic losses.

4.3 Description of Software Tools & Programs:

4.3.1 Languages and Frameworks:

1. Front End:

- **React.js:** Allowed us to create a dynamic and responsive user interface for our application. We built reusable components that facilitated the rendering of data, handling user interactions, and displaying real-time product verification results. The component-based architecture of React.js enabled us to easily modularize and organize our code, improving code maintainability and reusability. With React.js, we achieved efficient rendering performance, ensuring a smooth and responsive user interface.

- **Web3.js:** The JS library allowed seamless interaction with the Ethereum blockchain. By integrating Web3.js into our application, we were able to communicate directly with the blockchain network and interact with smart contracts. We utilized Web3.js to connect to the Ethereum blockchain, retrieve data related to product authenticity, and perform transactions securely. The library provided us with convenient functions to handle account management, contract deployment, and contract method invocations. With Web3.js, we could retrieve and display information from the blockchain, such as product ownership records or transaction history, providing users with transparent and verifiable proof of product authenticity.
- **MetaMask:** We integrated MetaMask as a crucial technology. MetaMask is a browser extension that serves as a digital wallet and Ethereum gateway. By incorporating MetaMask into our application, we allowed users to securely manage their Ethereum accounts and interact with the blockchain directly from their web browser. MetaMask provided a seamless and user-friendly interface for users to connect their wallets, authenticate transactions, and sign messages. This integration enabled users to easily verify the authenticity of products by connecting their MetaMask wallets and interacting with the smart contracts deployed on the blockchain. MetaMask's secure key management and transaction signing capabilities ensured that users' private keys and sensitive data were protected. Through the integration of MetaMask, we enhanced the user experience by

providing a convenient and familiar environment for users to engage with the blockchain and perform product verification with confidence.

2. Back End:

- **Truffle:** We employed Truffle as a fundamental technology. Truffle is a development framework that simplifies the process of building, testing, and deploying smart contracts on the Ethereum blockchain. We utilized Truffle to streamline our smart contract development workflow, enabling us to write, compile, and deploy contracts with ease. Truffle provided a suite of tools and libraries that facilitated contract compilation, migration, and deployment to the blockchain network. We leveraged Truffle's testing framework to create comprehensive unit tests for our smart contracts, ensuring their functionality and reliability. Additionally, Truffle's built-in support for automated contract deployments allowed us to deploy and manage our contracts across multiple environments efficiently.
- **Solidity:** Solidity as the programming language for developing smart contracts. Solidity is a high-level, contract-oriented language specifically designed for writing smart contracts on the Ethereum blockchain. We employed Solidity to define the logic and behavior of our product verification smart contract. With Solidity, we were able to implement functions to verify the authenticity of products, manage ownership records, and store relevant data on the blockchain. Solidity's syntax and features allowed us to create secure and efficient smart contracts, ensuring the integrity of the verification process. We leveraged Solidity's support for data types, control structures, and libraries to implement the necessary functionalities for our product verification. Solidity's integration with the Ethereum Virtual Machine (EVM) facilitated the seamless execution of our smart contracts on the blockchain. By utilizing Solidity, we were able to write reliable smart contracts that enable transparent and immutable product verification on the Ethereum network.

- **Ganache:** A we incorporated Ganache as a vital technology. Ganache is a personal blockchain network that allows us to simulate and test our smart contracts locally. We utilized Ganache as a development and testing environment to deploy and interact with our smart contracts without incurring any transaction costs on the actual Ethereum network. Ganache

provided us with a local blockchain instance, complete with accounts, balances, and transaction history, allowing us to mimic real-world scenarios and test the functionality of our product verification system. With Ganache, we could quickly deploy and debug our smart contracts, validate their behavior, and simulate various user interactions. Ganache's transaction logs facilitated the monitoring and analysis of contract interactions, enabling us to identify and resolve any issues or vulnerabilities during the development phase.

4.2.2 Technologies:

1. **Ethereum Blockchain:**

   Ethereum is a blockchain platform with its own cryptocurrency, called Ether (ETH) or Ethereum, and its own programming language, called Solidity. As a blockchain network, Ethereum is a decentralized public ledger for verifying and recording transactions. The network's users can create, publish, monetize, and use applications on the platform, and use its Ether cryptocurrency as payment.

   The proposed system uses Ethereum as the back end Blockchain operating system and uses Ethereum’s programming language Solidity as the high-level programming language for writing smart contracts. Solidity supports inheritance, library importing, etc. Solidity is designed for Ethereum Virtual Machine (EVM)

2. **Smart Contracts:**

   Smart contracts are self-executing agreements with the terms of the agreement directly written into code. They are stored and executed on a blockchain network, such as Ethereum, which ensures transparency, immutability, and decentralized execution. Smart contracts are designed to automatically facilitate, verify, or enforce the performance of contractual obligations. The code within a smart contract contains a set of predefined rules and conditions that govern the interactions between the parties involved. Once deployed to the blockchain, smart contracts can be interacted with by sending transactions to trigger their functions. These functions can update the contract's state, transfer assets, validate conditions, and execute predefined actions. The decentralized nature of smart contracts ensures the immutability of data, providing a trustless and reliable mechanism for executing agreements.

3. GitHub:

For maintaining source code and working together on software development projects, many people utilize the distributed version control system known as Git. Depending on the version being utilized, the particular version number and the year of manufacture may change. What Motivated Us to Use It:

- **Code Management:** Git provides detailed change tracking, recording all modifications, authors, and timestamps. This feature facilitates code review, accountability, and transparency. Git's integration with collaboration platforms enhances code visibility and enables efficient issue tracking and project management. Additionally, Git simplifies deployment and release management by tagging specific commits, making it easier to track and deploy sentiment analysis applications in different environments.

- **Version Control:** Effective version control is made possible by Git by keeping track of changes made to the codebase over time and promoting teamwork. It enables concurrent work on the project and easy collaboration by making branching, merging, and dispute resolution possible. Git's distributed architecture and built-in backup features guarantee code preservation and simple recovery, reducing the likelihood of data loss.

4.4 Hardware Setup Configuration:

4.4.1 PC Specifications:

- Processor: Intel(R) Core (TM) i7-10750H CPU @ 2.60GHZ
- Installed Memory (RAM): 16.0 GB (15.9 GB Usable)
- System Type: 64-Bit Operating System, x64 Based Processor
- Windows Specifications:
    - Edition: Windows 10 Pro
    - Version: 22H2

4.4.2 Minimum System Requirements:

- Processor: A dual-core processor with a clock speed of at least 2.0 GHz or higher.
- RAM: A minimum of 4 GB of RAM for smooth operation.
- Operating System: Compatible with Windows 10 (64-bit), macOS Mojave (10.14) or later, or Ubuntu 18.04 LTS.

Please note that these are minimum requirements, and higher specifications may be beneficial for improved performance, especially when handling large volumes of requests.

4.4.3 Application Usage Instructions:

- Navigate to the Website
- Connect to the Blockchain Network
- Create or Import a Wallet
- Sync with the Blockchain
- Explore the Application
- Use Transactional Gas to Access Functionality

4.4.4 Tools Software Requirements:

1. Truffle:

- Operating System: Windows 7 or later, macOS 10.12 or later, or a Linux distribution (Ubuntu 14.04 LTS or later).
- Processor: A dual-core processor with a clock speed of 2.0 GHz or higher.
- RAM: At least 4GB of RAM.
- Disk Space: Minimum of 200MB of free disk space.
- Node.js: Truffle requires Node.js, so you should have Node.js version 8.9.4 or later installed on your system.

2. Solidity:

- Operating System: Solidity is compatible with Windows, macOS, and Linux operating systems.
- Processor: A dual-core processor with a clock speed of 2.0 GHz or higher is recommended.
- RAM: At least 2GB of RAM is required for Solidity to run smoothly.
- Disk Space: Solidity itself requires 100MB.
- Node.js: Solidity requires Node.js to be installed on your system. The recommended version is Node.js 10.x or later.

3. Web3.js:

- Operating System: Web3.js is compatible with Windows, macOS, and Linux operating systems.
- Web Browser: Web3.js is compatible with Google Chrome, Mozilla Firefox, and Safari. Preferably the latest version
- Node.js: Web3.js will need to have Node.js installed on your system, with a recommended version of 8.x or later.
- RAM: The memory requirements for using Web3.js are minimal and will depend on the overall complexity of your web application.
- Disk Space: Web3.js itself requires very little disk space as it is a lightweight library.

4. Node.js:

- Operating System: Node.js is compatible with various operating systems, including Windows, macOS, and Linux.
- Processor: A dual-core processor with a clock speed of 2.0 GHz or higher is recommended.
- RAM: At least 2GB of RAM is required for Node.js
- Disk Space: Node.js itself requires minimal disk space, typically around 150MB.
- Network Connectivity: Node.js relies on network connectivity to download packages and dependencies.

5. Ganache:

- Operating System: Ganache is compatible with Windows, macOS, and Linux operating systems.
- Processor: A dual-core processor with a clock speed of 1.8 GHz or higher is recommended.
- RAM: At least 2GB of RAM is required for Ganache.
- Disk Space: Ganache itself requires minimal disk space, typically around 200MB.
- Network Connectivity: Ganache creates a local blockchain network for development purposes.
- Graphics Card: Ganache can use graphical user interface (GUI) features or interact with blockchain visualizations,
- Network Ports: Ganache utilizes certain network ports for communication and network connectivity. Ensure that the required ports (such as TCP port 8545).

6. MetaMask:

- Web Browser: MetaMask is primarily a browser extension and works with Google Chrome, Mozilla Firefox, & Brave.
- RAM: overall complexity of the web applications.
- Internet Connectivity: MetaMask requires a stable internet connection to interact with the Ethereum blockchain network and access decentralized applications (DApp).

7. React.js:

- Operating System: React.js is compatible with various operating systems, including Windows, macOS, and Linux.
- Processor: A dual-core processor with a speed of 1.8 GHz.
- RAM: At least 2GB of RAM is required for React.js.
- Disk Space: React.js itself requires minimal disk space.
- Node.js: React.js requires Node.js to be installed on your system. The recommended version is Node.js 10.x or later,
- Code Editor: You will need a code editor or integrated development environment (IDE) to write React.js code.

8. Remix:

- Web Browser: Remix is compatible with web browsers such as Google Chrome, Mozilla Firefox, and Brave.
- RAM: Contracts and the number of open tabs and active plugins, having at least 2GB of RAM.
- Internet Connectivity: This is necessary for loading the Remix interface, interacting with the Ethereum blockchain.
- Solidity Compiler: Node Package Manager (npm)

9. Git:

- Operating System: Windows, macOS, Linux
- Processor: 1.5 GHz or faster processor
- RAM: Minimum 512 MB (1 GB recommended)
- Disk Space: Minimum 50 MB available disk space
- Git Client: Install the Git client specific to your operating system (e.g., Git for Windows, Git for macOS)

10. VS Code:

- Operating System: Windows, macOS, Linux
- Processor: 64-bit processor
- RAM: Minimum 512 MB (1 GB recommended)
- Disk Space: Minimum 200 MB available disk space
- Memory: Minimum 2 GB of RAM.

4.5 Experimental and Results:

4.5.1 Experimental:

- **Experimental Design:**

1- **Data collection:**
The data used in the study was collected from a company's system's database. The dataset included information on products, names, models, descriptions, and the image link.

2- **Data Preprocessing:**
We ensured that the data have no null values, no redundancies or duplicated and the data is consistency. We have data of 5 companies with 100 products of each.

3- **The system testing:**
The system had been tested by the collected data, we have reached the maximum true positive and true negative, as the original codes results that the product is original, if we entered the wrong QR result's this product is fake.

4- **Evaluation:**
The performance of the system was evaluated to get very high accuracy.

- **Hashing functions:**

In our exploration of different hashing functions for our application, we conducted tests to determine their effectiveness in ensuring data integrity and accuracy. After comparing various options, we discovered that the **Keccak-256** hashing function, specifically designed for use in Solidity, exhibited the strongest properties.

We integrated the **Keccak-256** hashing function into our verify function mechanism, which involved hashing the product values and storing them on QR codes. This implementation resulted in achieving a remarkable accuracy rate of 100%. The robust collision resistance of **Keccak-256** played a vital role in ensuring the integrity of our data.

Keccak-256 generates a hash output size of 256 bits, which corresponds to a staggering **2^256 possible hash values**. This extensive range of possibilities significantly reduces the likelihood of collision, making it highly secure for cryptographic applications.

- **Read & Upload QR Code:**

  The product verification process offers users the ability to verify products by uploading a photo of the QR code associated with each product. To achieve this, we integrated an API that assists in the extraction and interpretation of the QR code's information. This API allows us to read the QR code and extract the relevant data for verification.

  To further enhance the user experience, we expanded the functionality by enabling real-time QR code scanning. Users can now visit our website and utilize their device's webcam to scan the QR code directly. As they place the QR code in front of the webcam, the verification function is automatically triggered, swiftly processing the QR code's data and executing the verification process.

  By incorporating the QR code reading API and implementing real-time scanning, we have significantly simplified the verification procedure for users. The option to upload a photo or scan the QR code in real time provides flexibility and convenience, catering to different user preferences. This integration of technology streamlines the verification process and ensures accurate and efficient product authentication.

4.5.2 Results:

Materials:

We used a variety of materials in our project, including Web development using ReactJS, a Blockchain database, Ethereum, Ganache, Web3, and MetaMask. We also used a variety of Programming languages, including Solidity, and ReactJS.

Methods:

o We followed a structured development process, starting with requirements gathering and analysis, followed by system design, implementation, testing, and deployment. We used an iterative development approach, which allowed us to receive feedback while working in the system.

o We conducted a study on the response times of the existing systems, focusing on four key areas: login, search, Scan QR, and Report.

o We measured the response times using a tool that simulated user interactions with the system.

o We repeated the measurements multiple times to ensure accuracy and consistency.

Overall, the Identifying fake products system performed well in terms of response times, with only minor improvements needed in the accuracy.

Conclusion:

o Through our study we noticed that the system performs well in terms of response time, with only minor improvements needed in specific areas.

o Speed and responsive systems are important for providing a great user experience.

o Further studies could explore other factors that affect user experience and system performance and functionality.

User Acceptance Testing:

We conducted user acceptance testing to determine the efficiency of the system. This testing is conducted on the students and parents. We tested more than one criterion through this testing: Usability, User friendly, Efficiency, performance, and learnability.

o Usability: The system is usable, and interfaces are understandable.

o User Friendly: The system is attractive to the user eye.

o Effective: The system helps users to do their tasks easily. Which is improves the user trust.

Hypothesis:

Our hypothesis was that the system would be an effective platform for ensuring the products are original which increases the user trust and the company stability.

Findings:

Our findings showed that the system is indeed an effective platform for identifying fake products. Specifically, we found that:

Convenience: Customers found the system easy to use, with the ability to report fake ones.

Privacy: The decentralized app increases privacy.

Cost-Effectiveness: The system protects the users from buying fake products.

Results to the hypothesis:

The findings we made supported our hypothesis that the system's actually effective platform for ensuring the products are original which increase the user trust and the company stability, The fact that customers found the system easy to use.

Limitations:

We have some limitations in our work, as we study small number of customers. Also, we didn't explore the long-term effects of using the application.

Chapter Five:
Run the Application

(User Functionality)

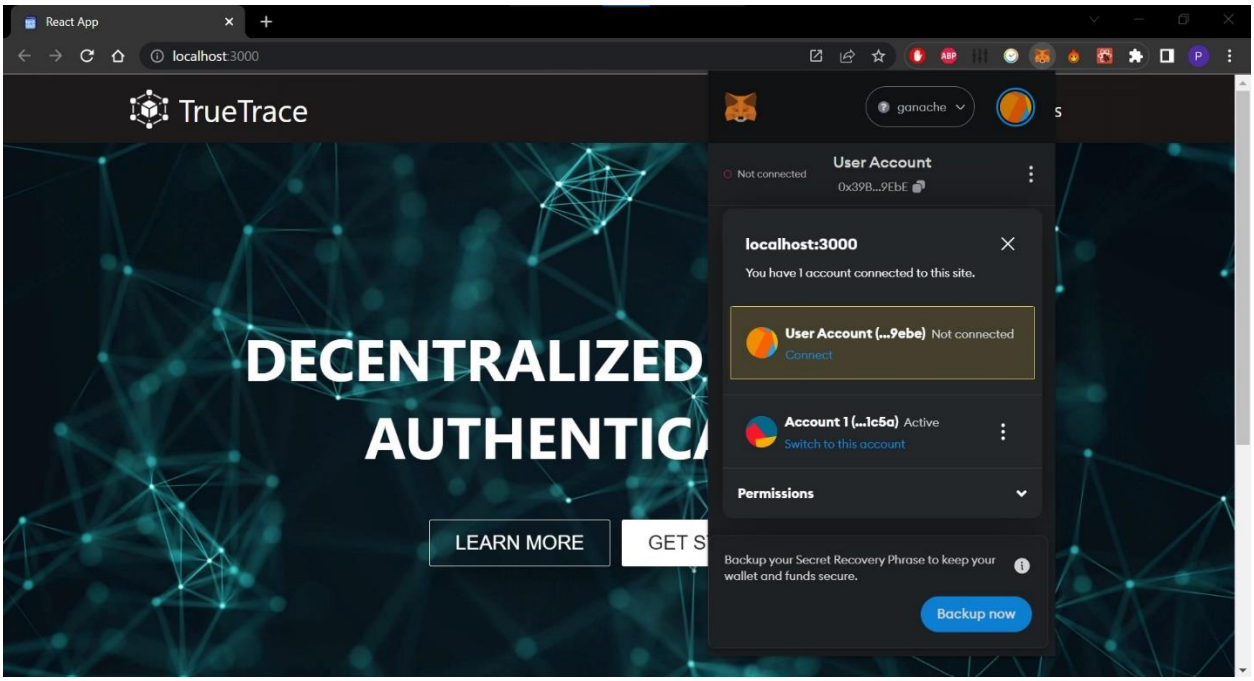

Figure 5.1: Connect MetaMask Virtual Account to Web App

Press the connect button to connect the Web App to a User Virtual Account on the Blockchain using MetaMask.

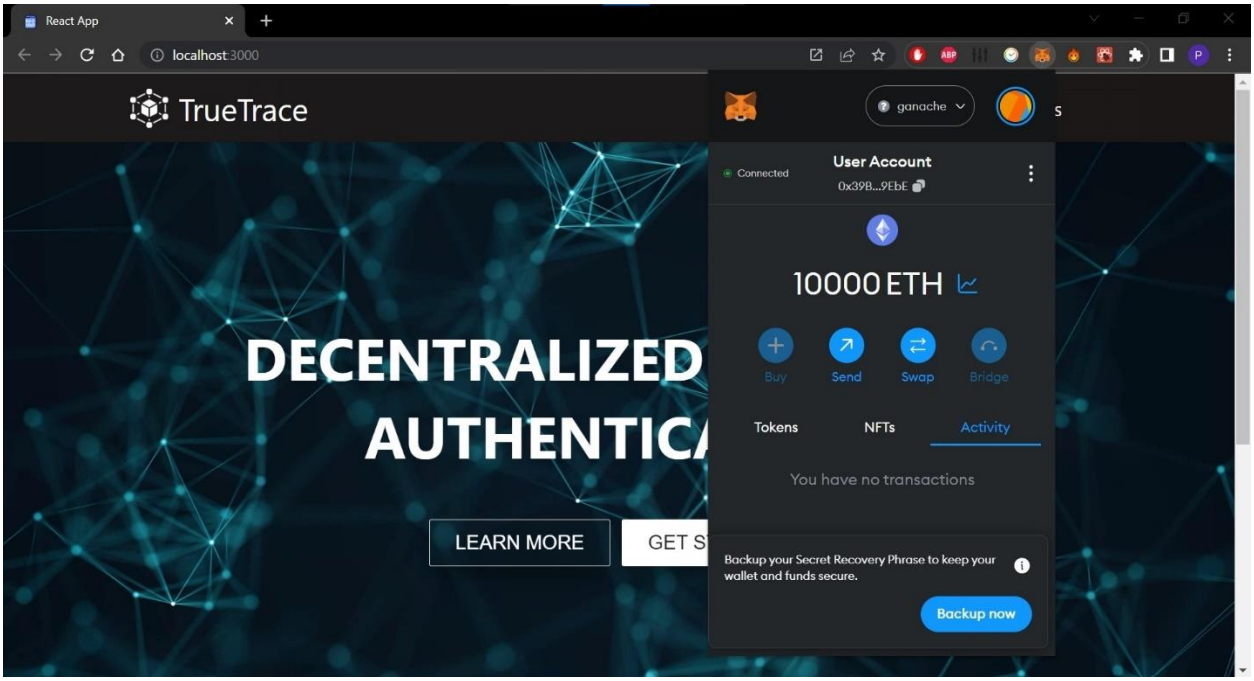

Figure 5.2: Virtual Account successfully connected to the Web App

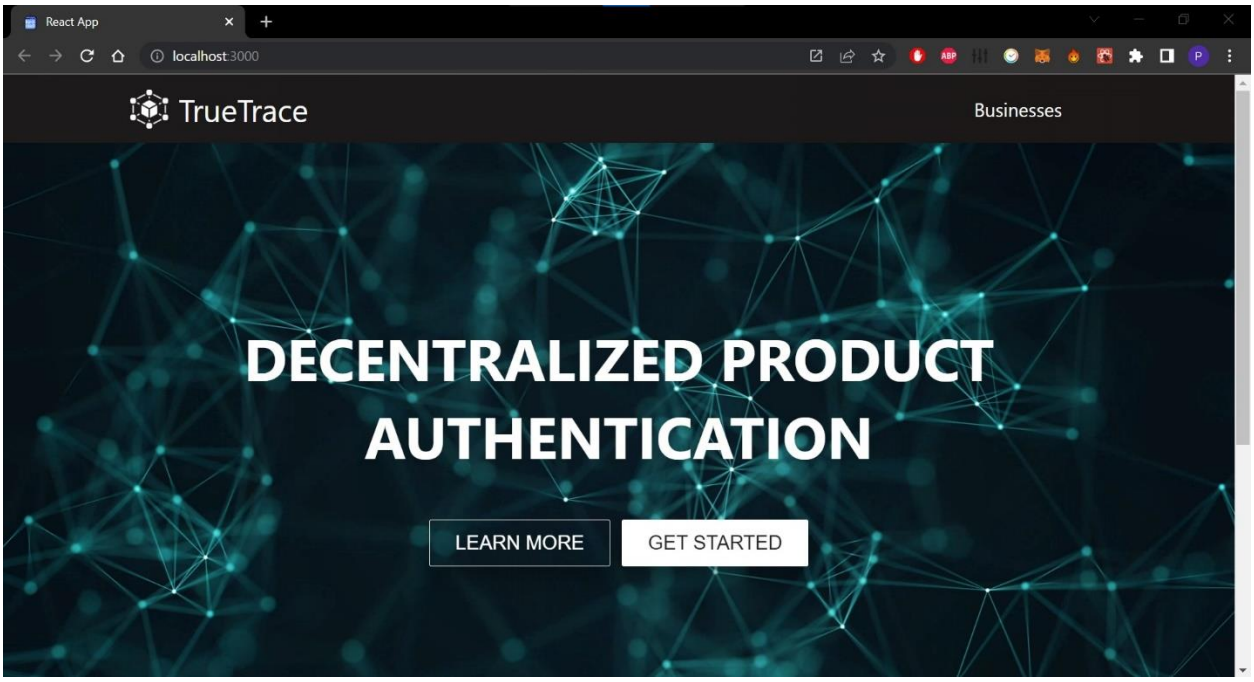

Figure 5.3: Web App Homepage

Choose Get Started.

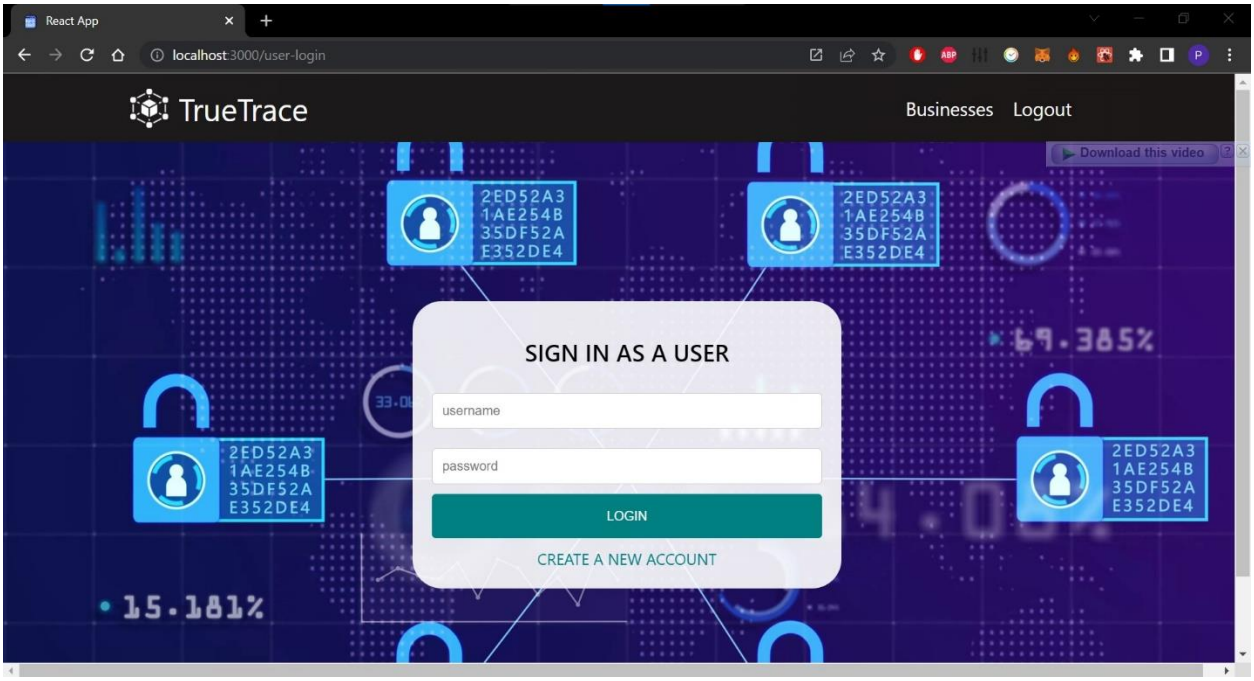

Figure 5.4: User Login Screen

User will choose to create a new account.

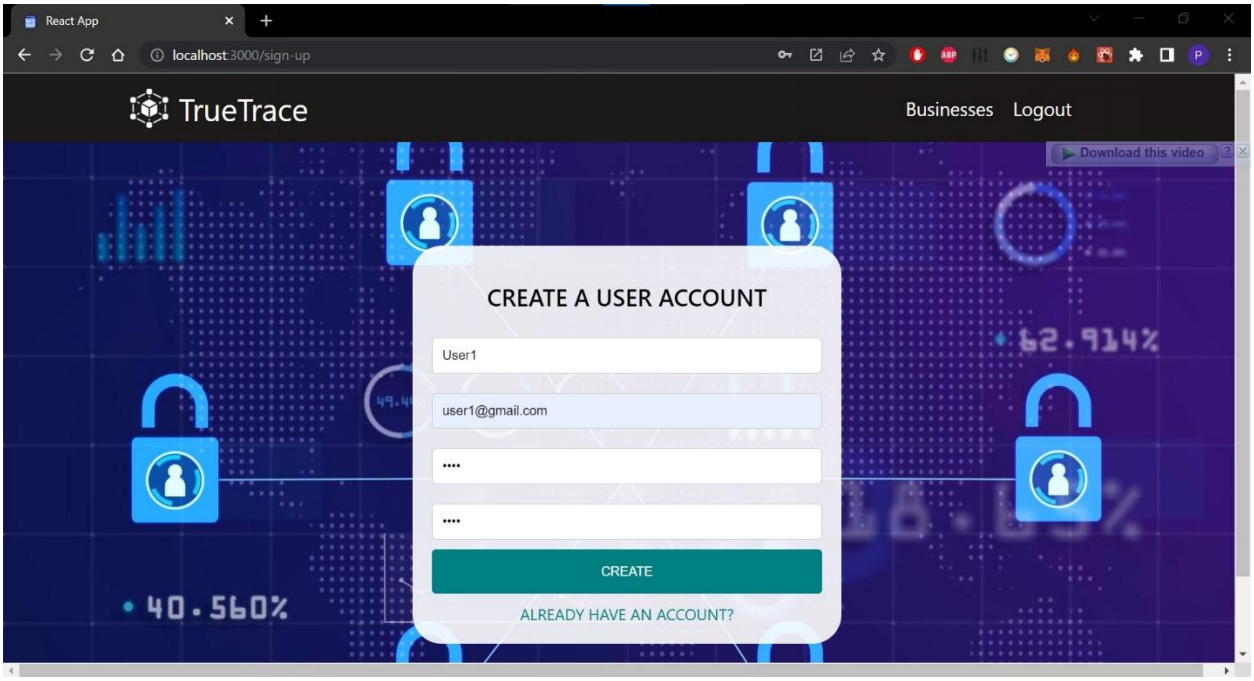

Figure 5.5: User Registration Screen

Users will input the registration data in the form. Name, E-mail and Password.
The create button is pressed.

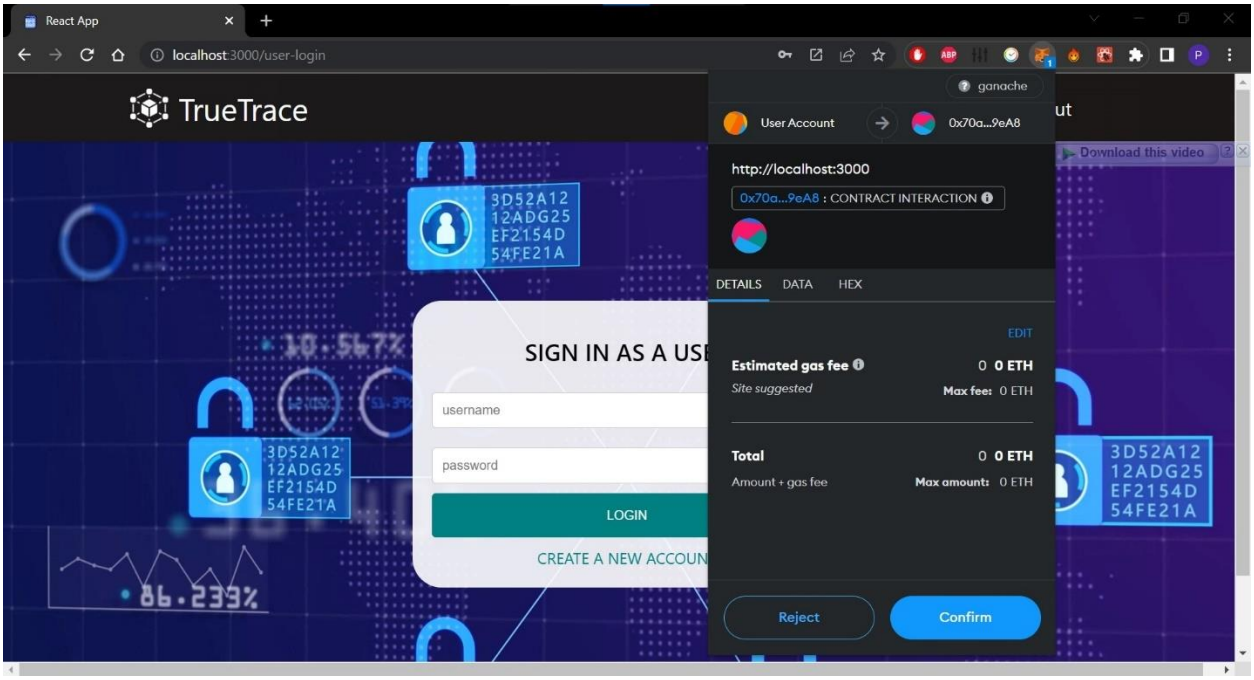

Figure 5.6: Registration Transaction Notification

MetaMask window appears to confirm the transaction saving the User Data on the Blockchain.

Transaction is confirmed.

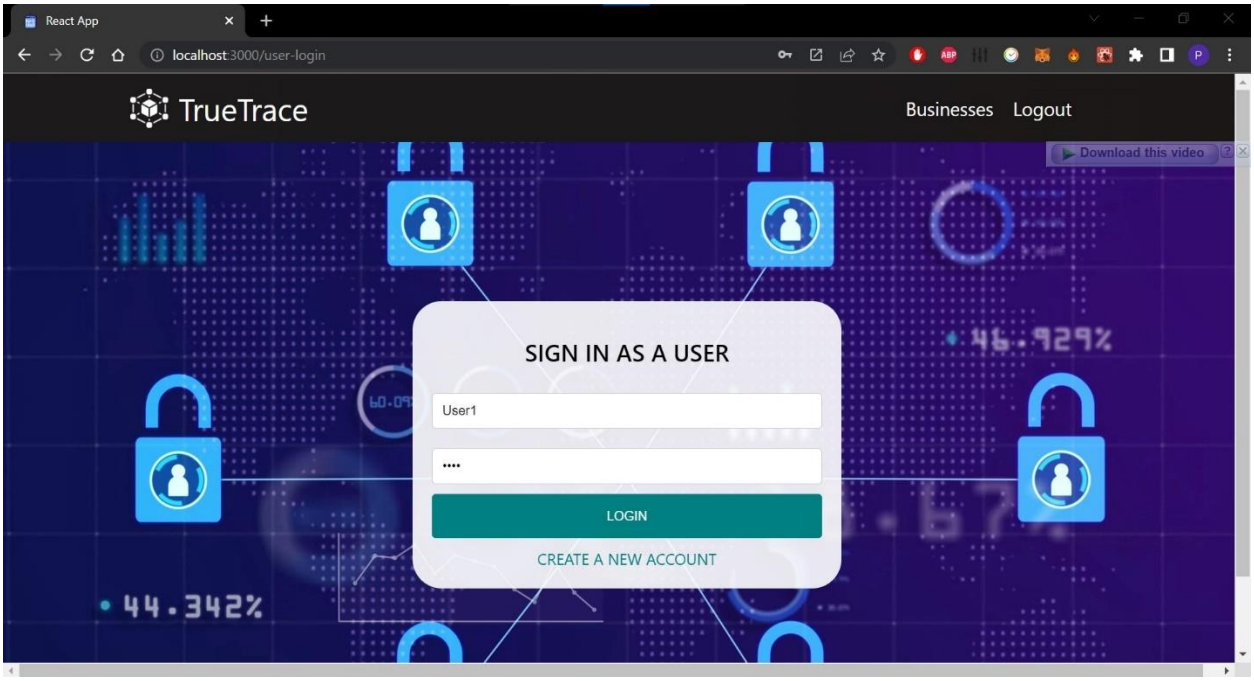

Figure 5.7: Login Screen With User Credentials Entered

Users redirected to the Login screen, they will enter their credentials, Name and Password, and Login. User is Redirected to the landing page when logged

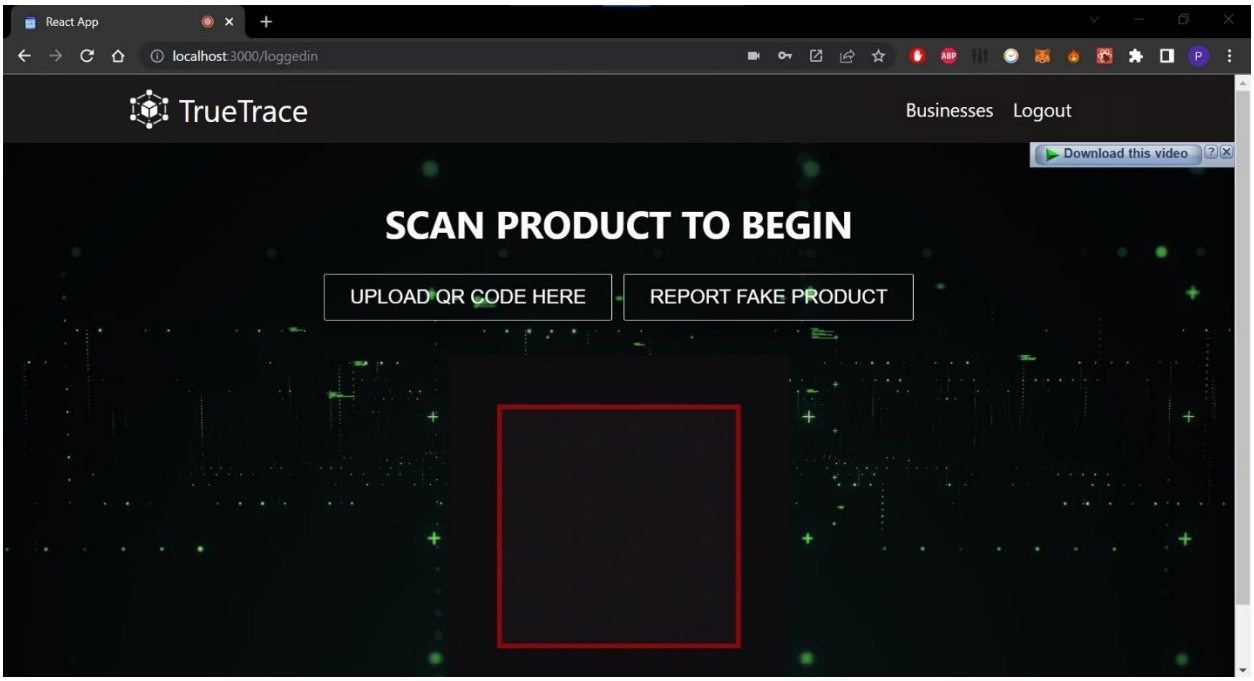

Figure 5.8: User Landing Page

In the Landing page, the user has 3 different functionalities, Check product authenticity via uploading a QR Code, Scan the QR Code for authenticity, or report a fake product.

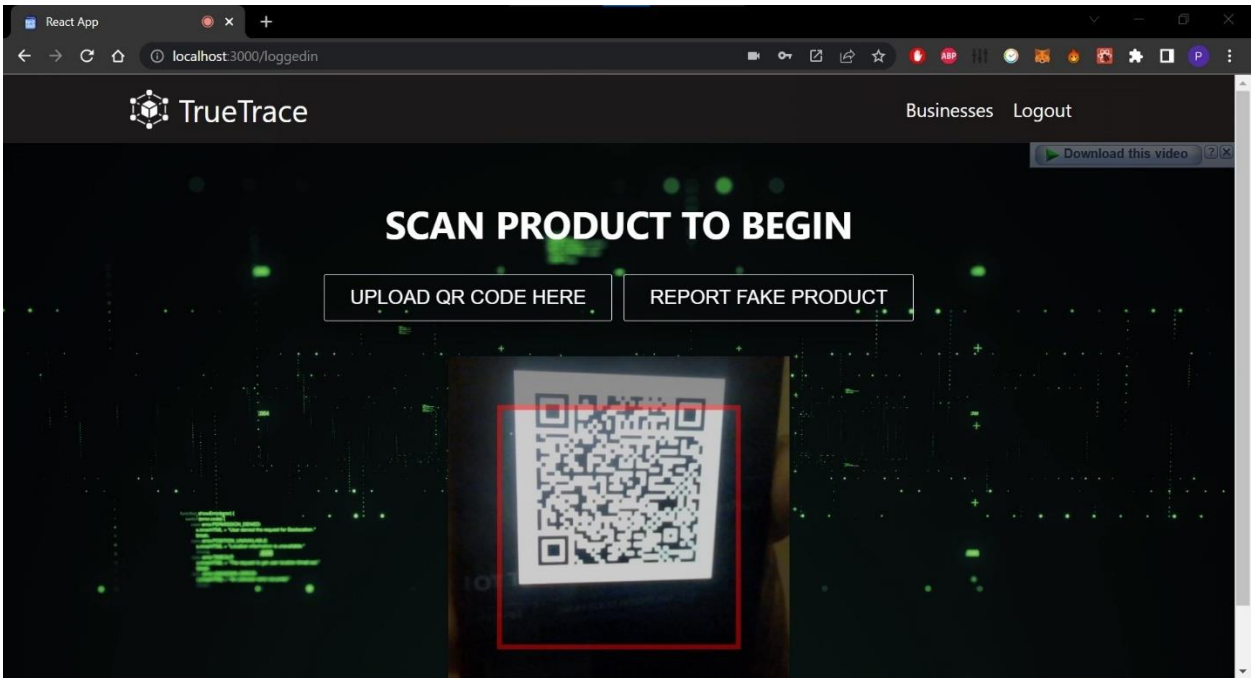

Figure 5.9: User Landing Page With QR Scanner Activated

User Scans a Fake QR Code using the Active QR Scanner

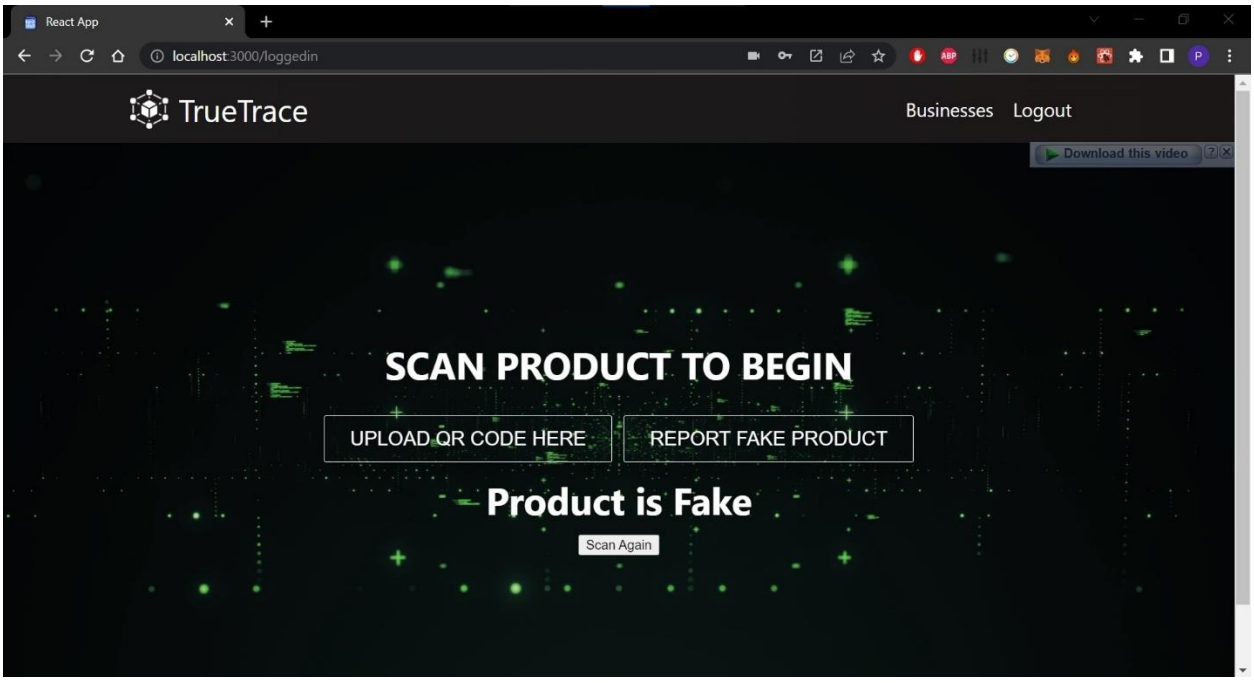

Figure 5.10: QR Code Scanner Output (Fake Product)

User Has Scanned a Fake Product, they will press the Report Fake Product button.

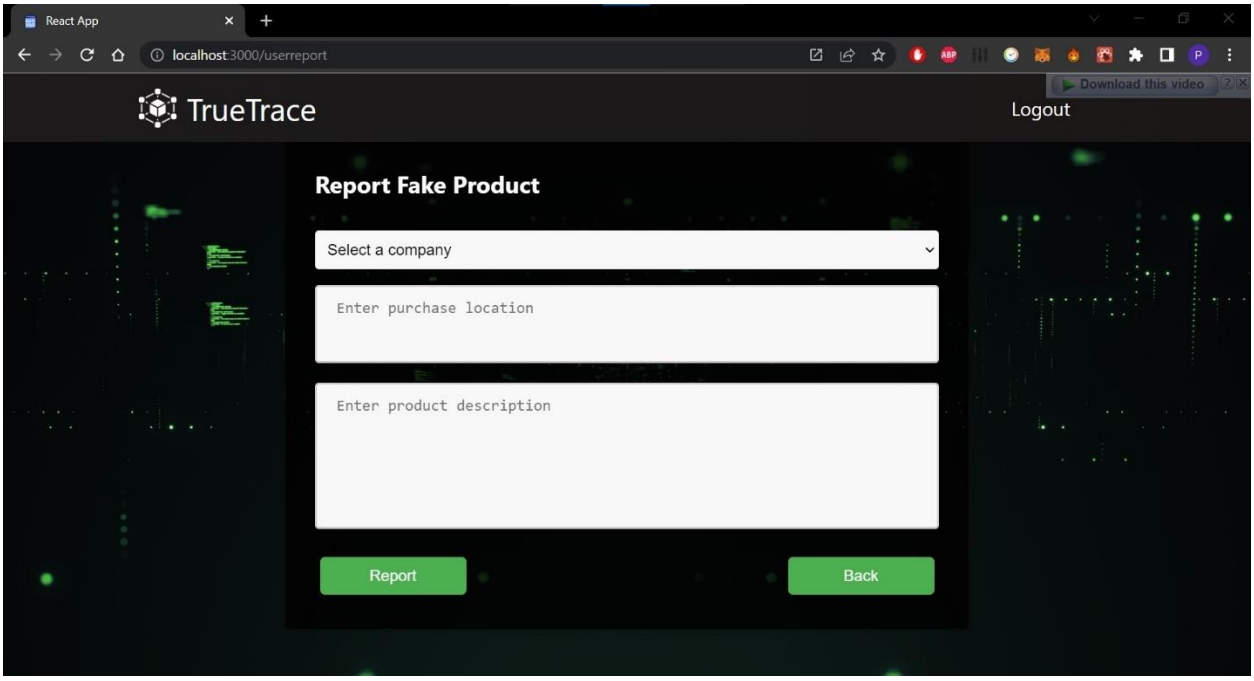

Figure 5.11: Report a Fake Product Page

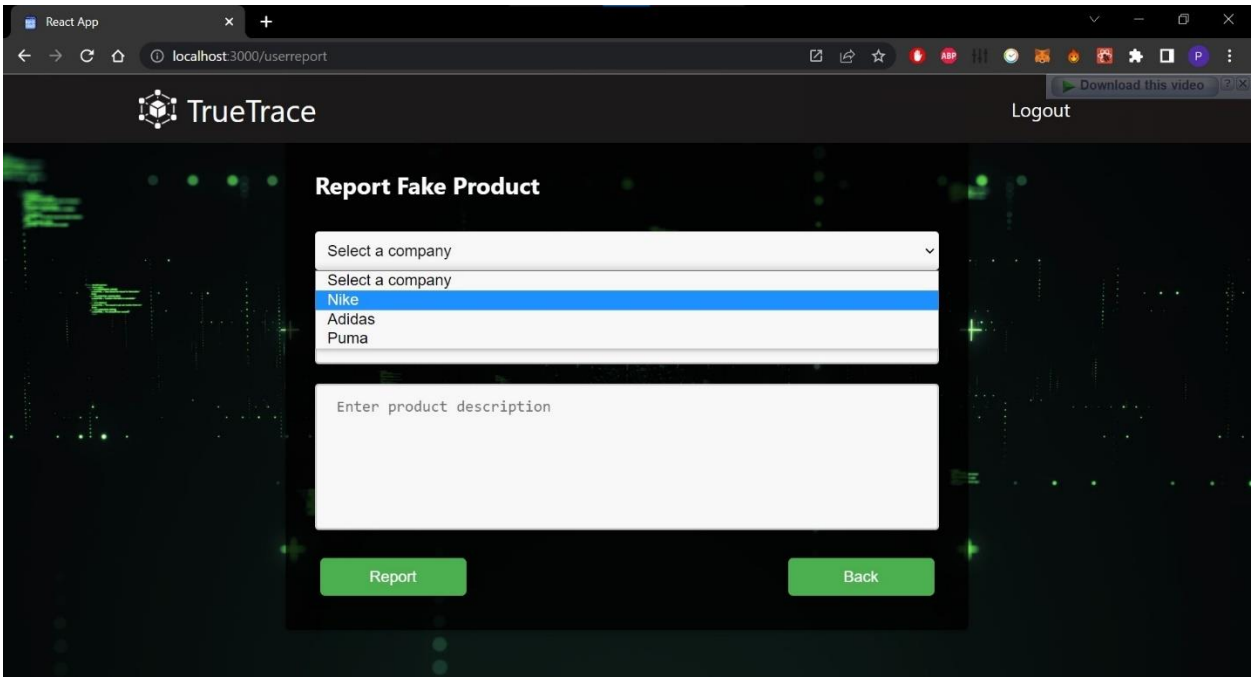

Figure 5.12: Drop Down List to Choose a Company to Report the Fake Product

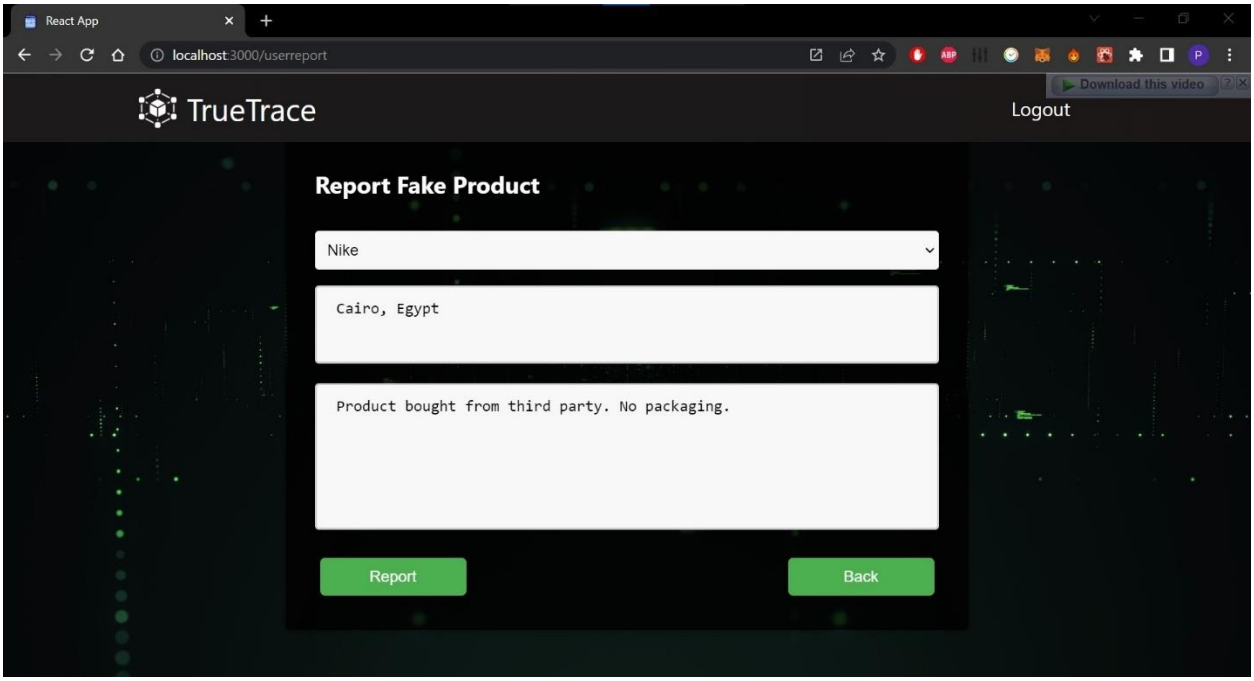

Figure 5.13: Report Data Filled

User will fill the Report Page with the Company name, Report Location, A description of choice and press the Report button.

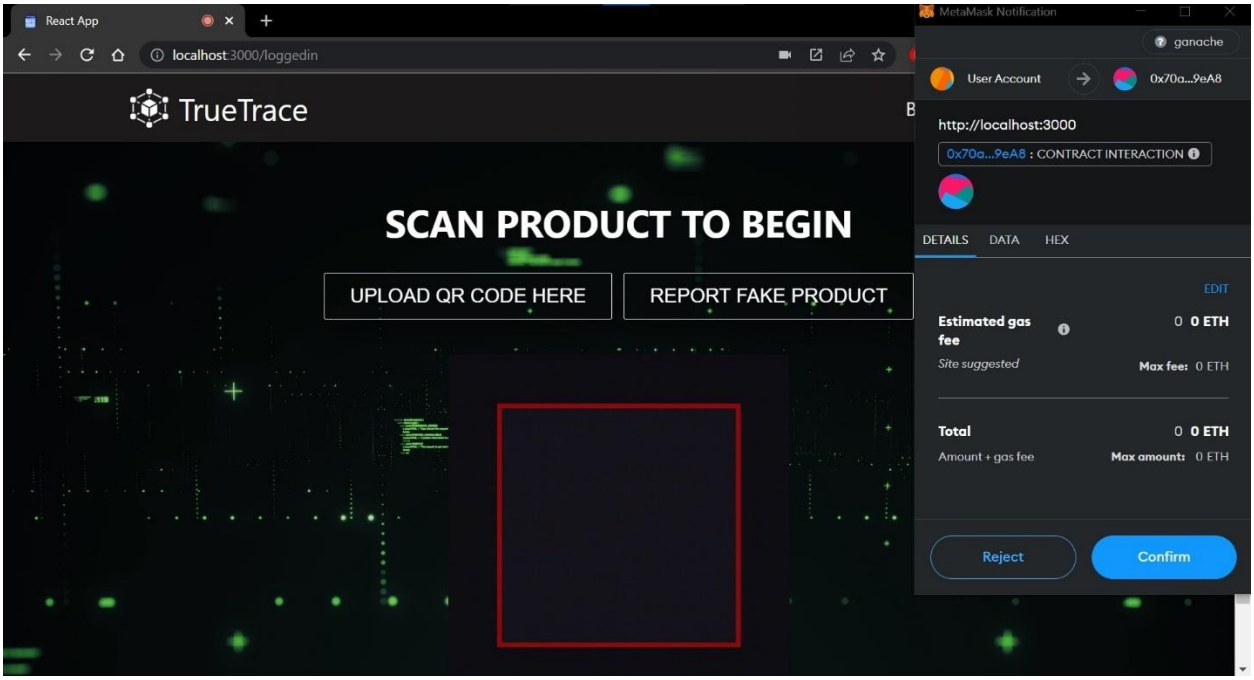

Figure 5.14: MetaMask Notification to Confirm Report Transactions

User Confirms the Report Transaction and is redirected to the Landing page.

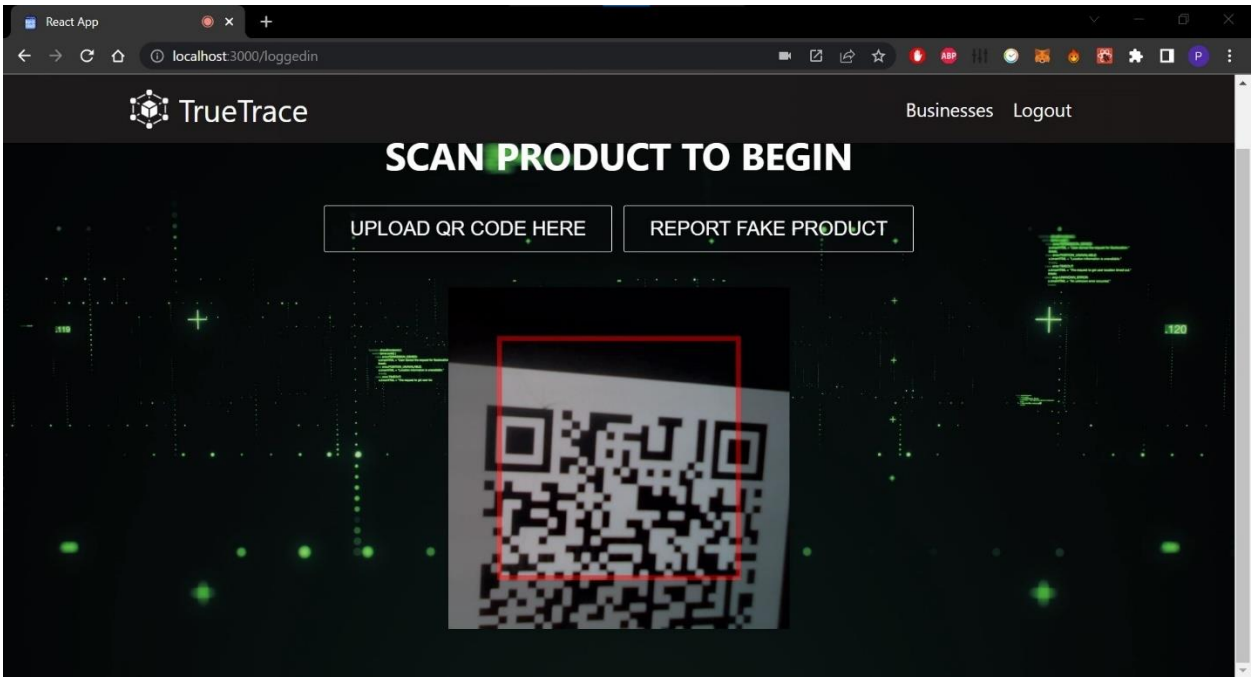

Figure 5.15: User Scanning QR Code

User will now attempt to scan an Authentic product QR Code from the live QR Reader.

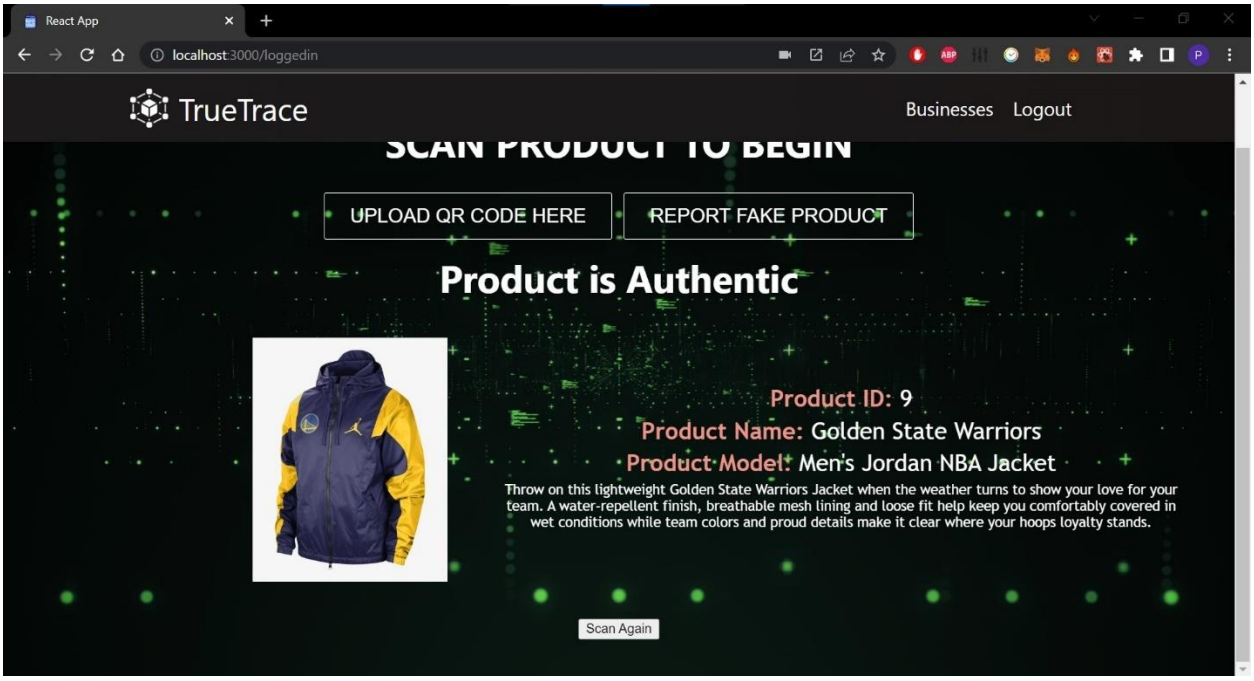

Figure 5.16: Authentic Product Output& Its Info.

After the user scans the Authentic QR Code Product Authenticity Is shown

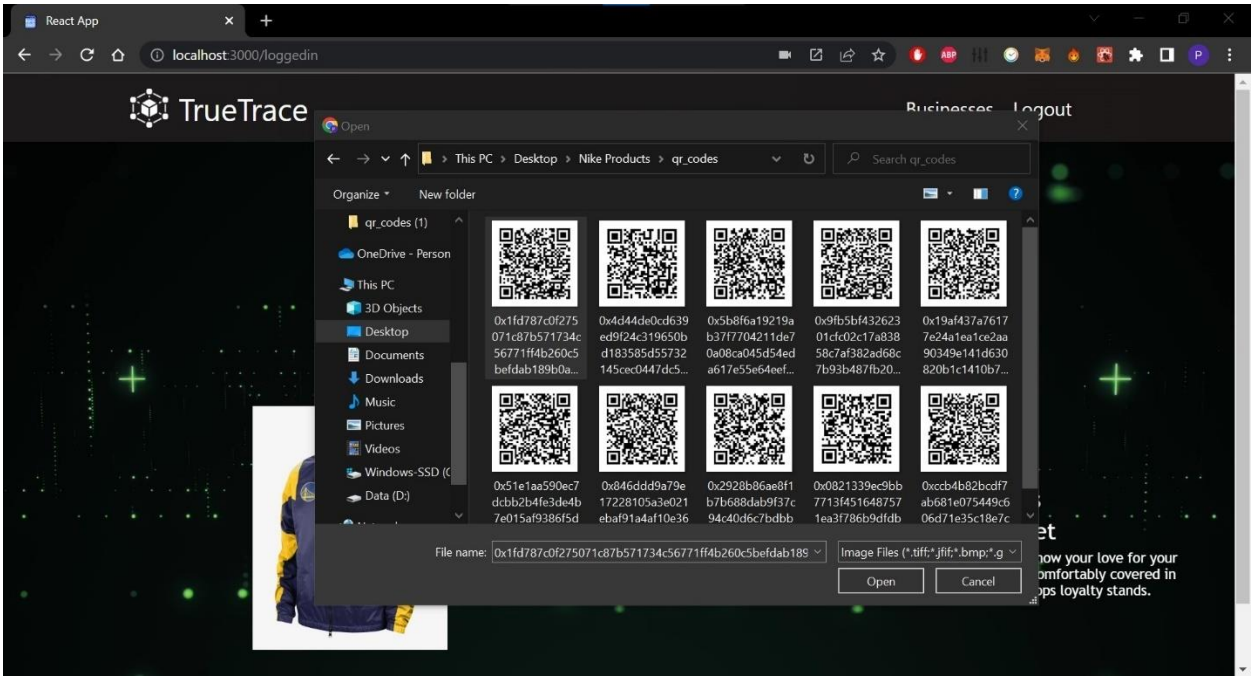

Figure 5.17: Upload QR Code Button Pressed

When a User chooses to upload a QR Code for authenticity, an input window is opened to choose an Image file to verify the QR Code.

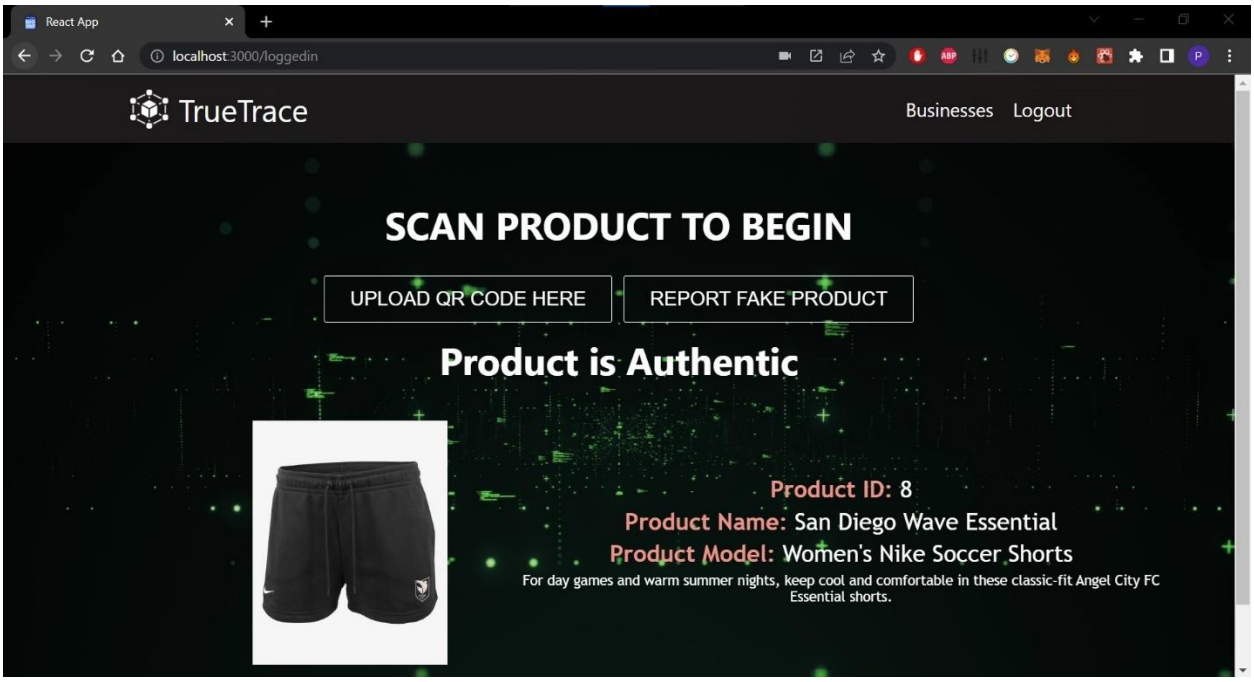

Figure 5.18: Authentic Product Output & Product Info.

(Company Functionality)

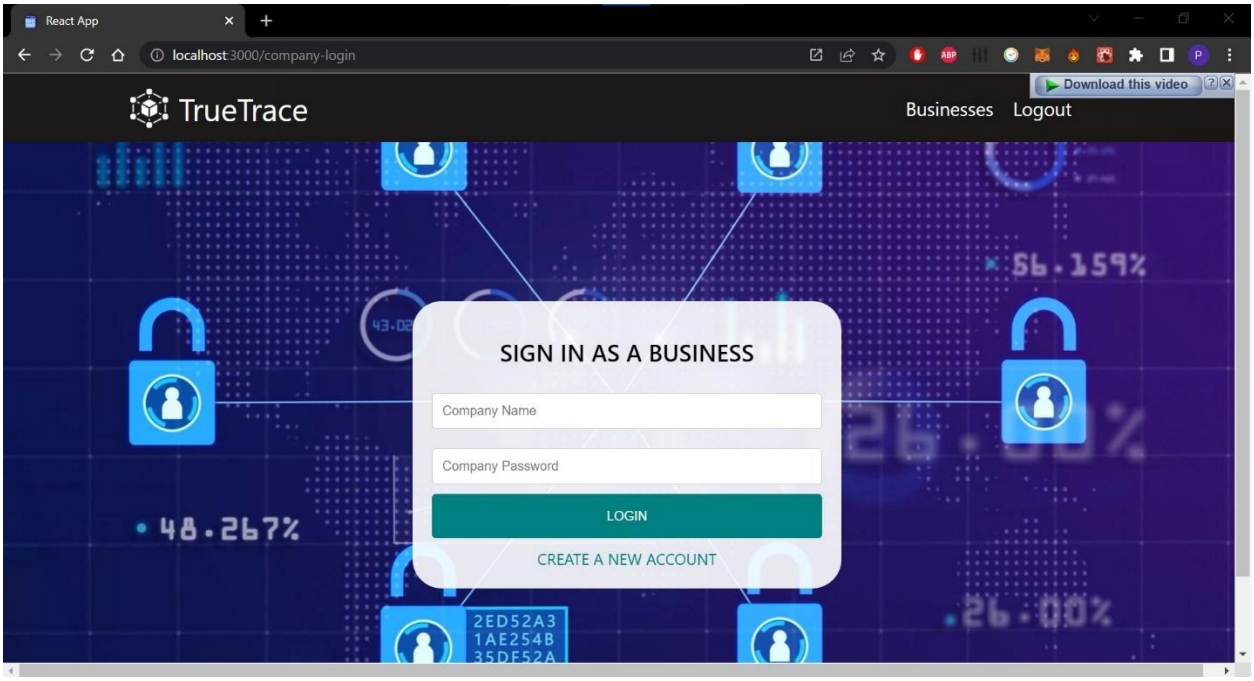

Figure 5.19: Company Login Screen

The company will choose to create a new account.

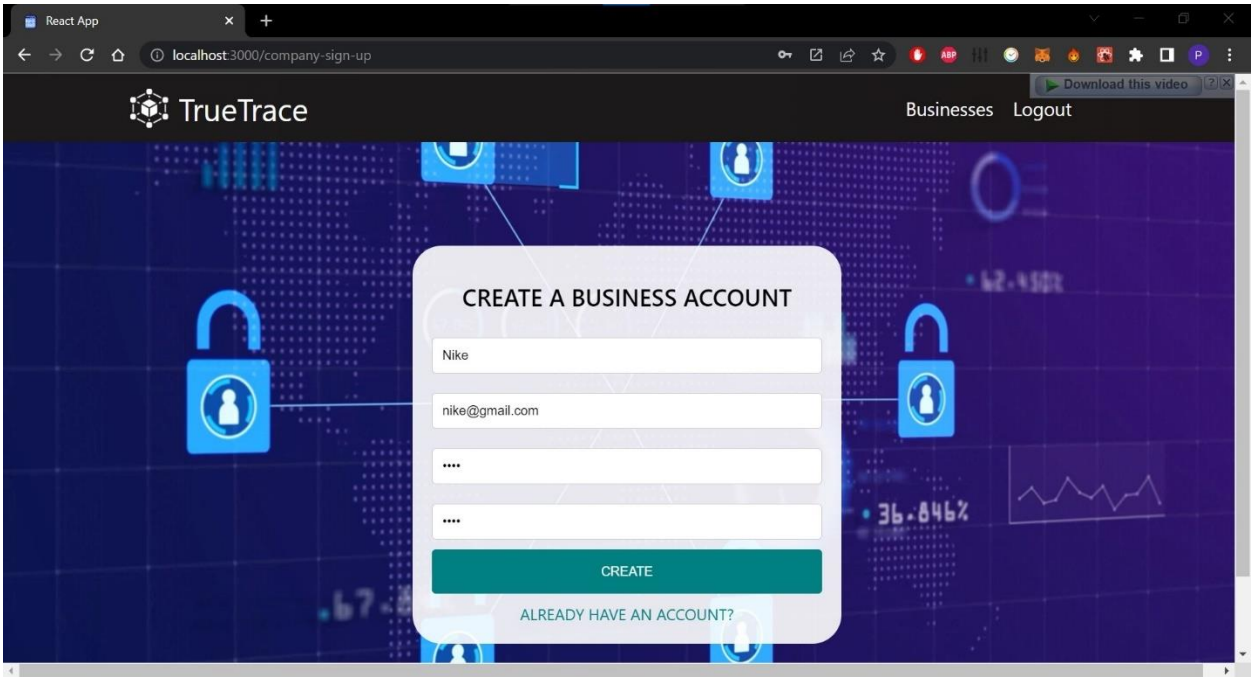

Figure 5.20: Company Registration Screen

The company will input the registration data in the form. Name, E-mail and Password. The create button is pressed.

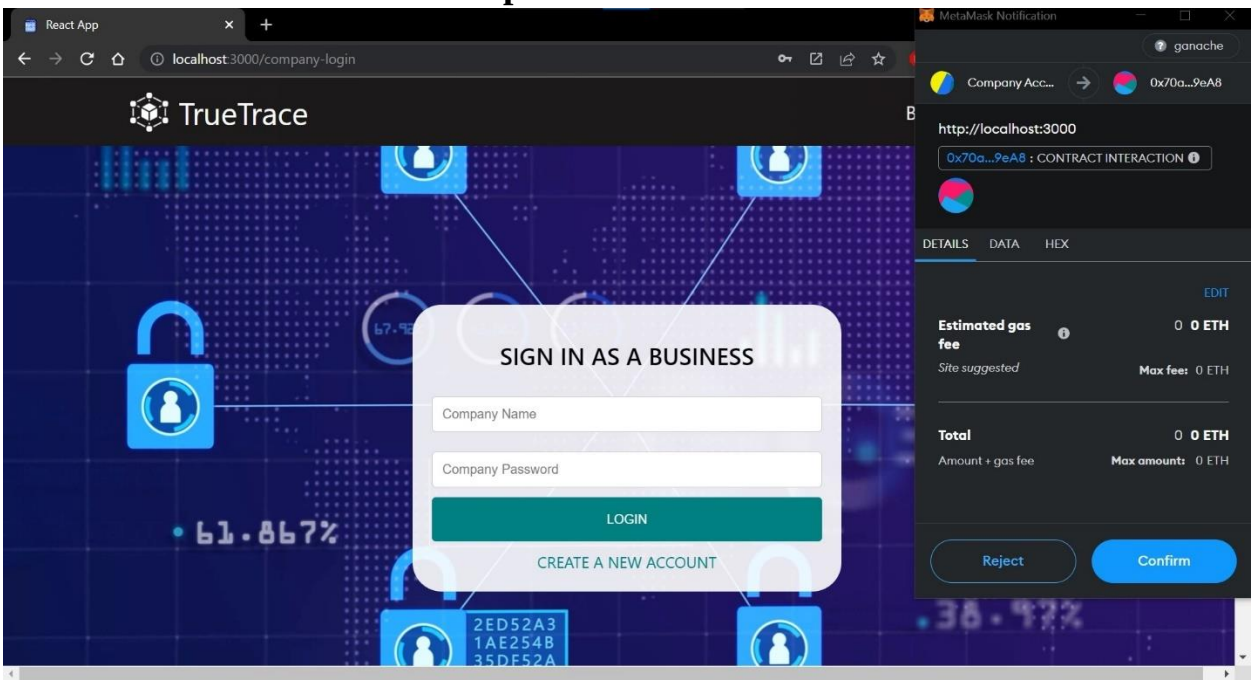

Figure 5.21: Registration Transaction Notification

MetaMask window appears to confirm the transaction saving the Company Data on the Blockchain. Transaction is confirmed.

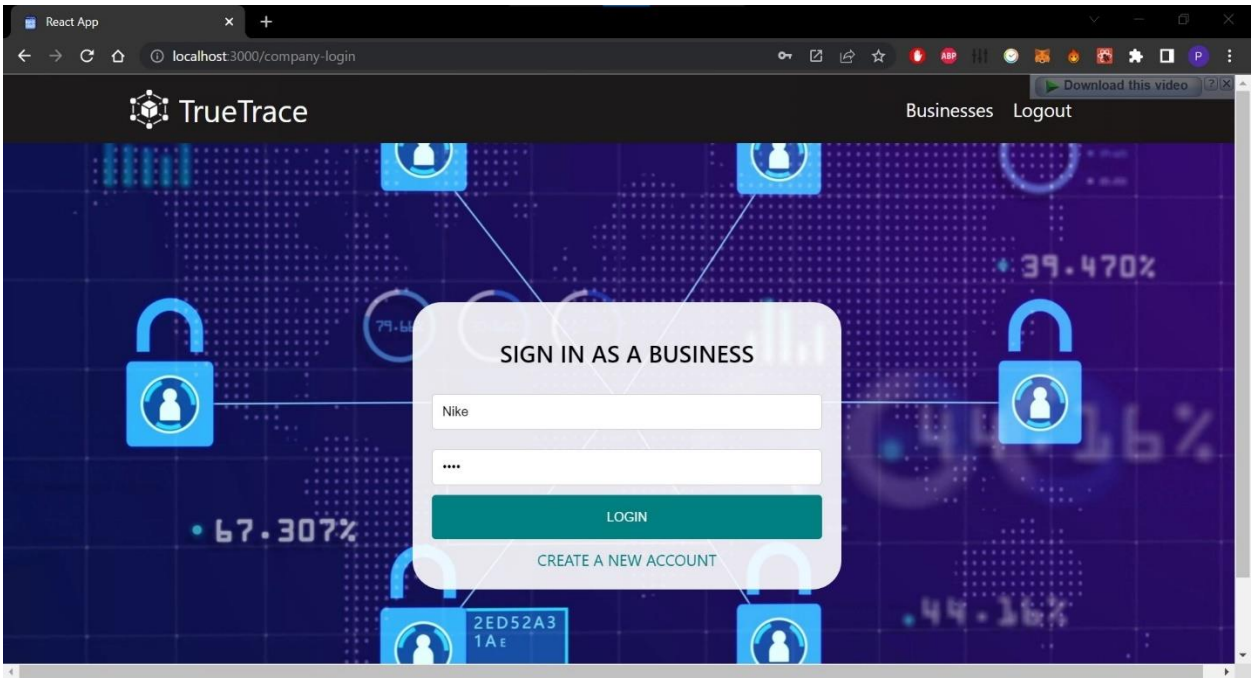

Figure 5.22: Login Screen With Company Credentials Entered

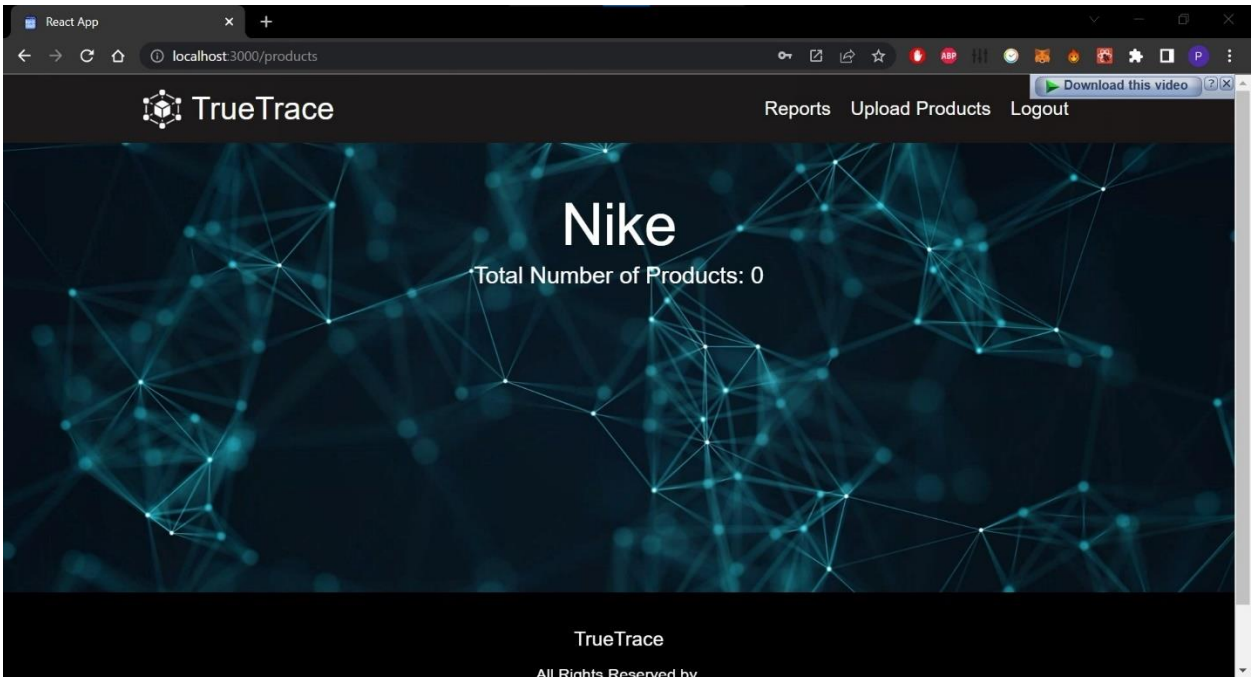

Figure 5.23: Company Landing Page

The Landing page for the company shows the number of products uploaded. In the Navigation Bar, The Company could view its Reports, Upload Products or Log Out.

The company will choose Upload Products.

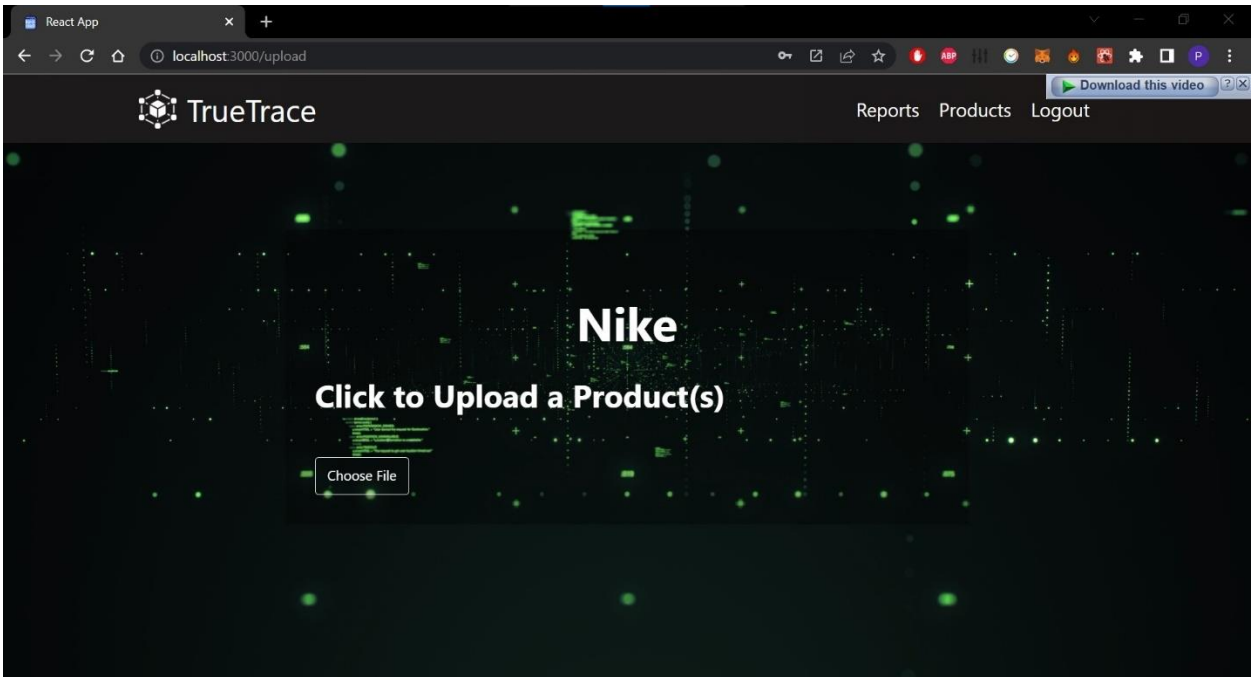

Figure 5.24: Company Upload Products Page

When a Company chooses to Upload Products, It can choose a file to Upload so that the Products are uploaded from a CSV.

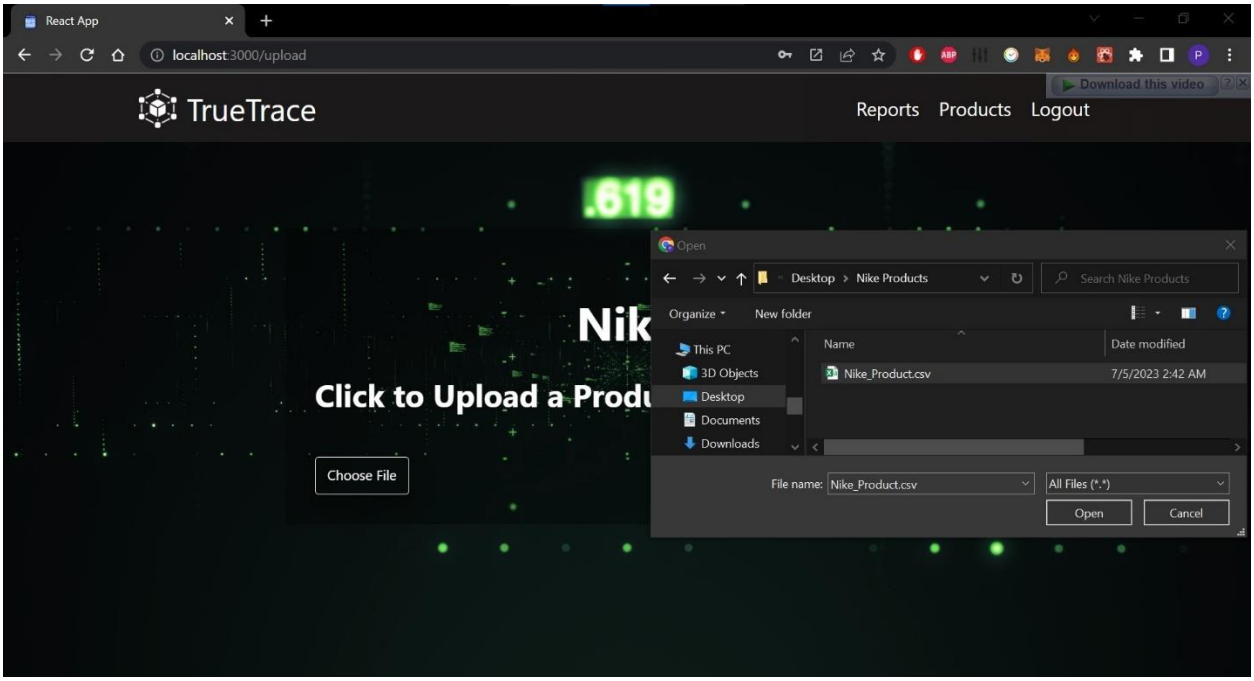

Figure 5.25: Choose CSV File Window

Company will choose a CSV file to upload the Products in it.

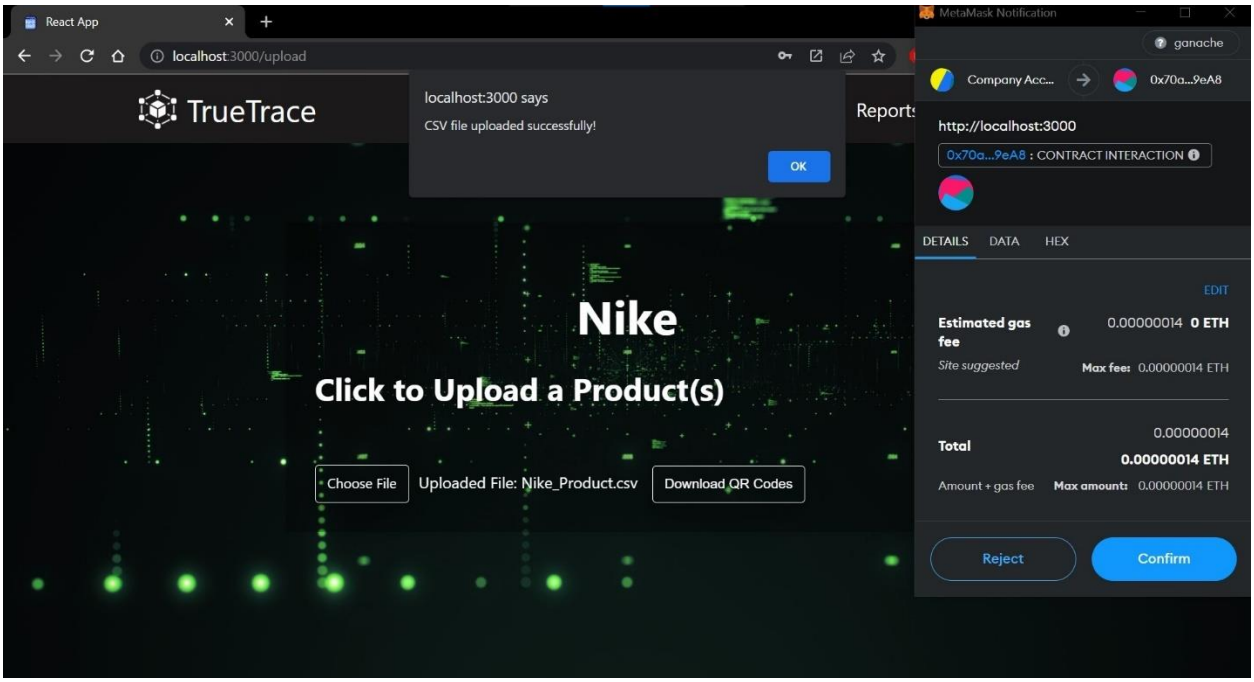

Figure 5.26: CSV File Uploaded Successfully Alert MetaMask Confirmation

The company has now chosen a CSV to upload the products and will confirm the transaction so that the products will be saved to the BlockChain.

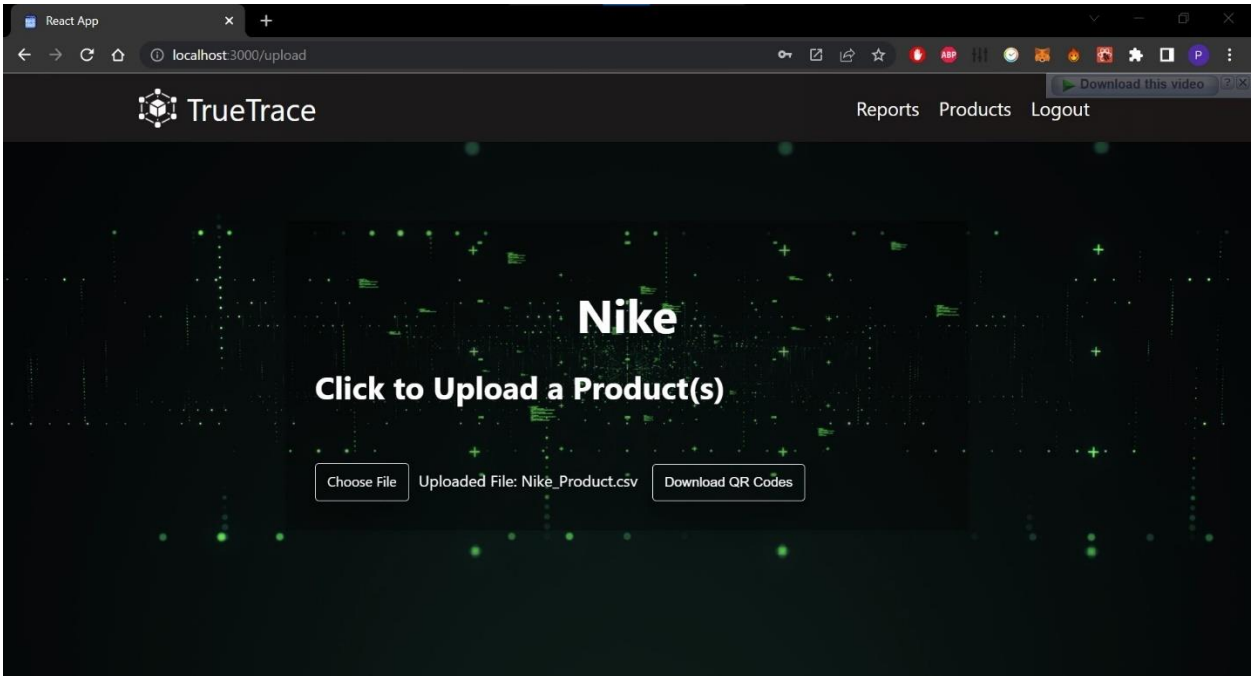

Figure 5.27: Download QR Codes Button Visible

When a CSV is uploaded, the Company can now download the QR Codes for the Products to track their Authenticity.

Figure 5.28: Downloaded QR Codes Folder

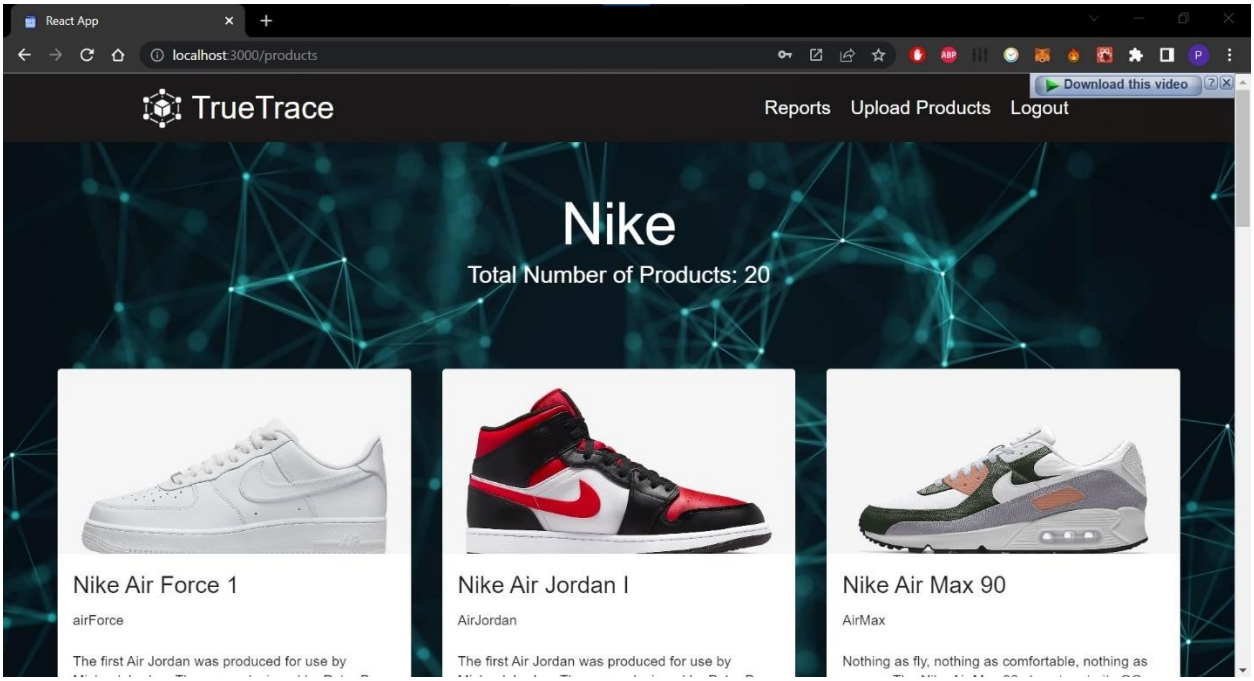

Figure 5.29: Products Page after the Products have been Uploaded

When the Company goes to the Products page, the number of products is shown. In every Product, the Product Image, Product Title, Model and Description are shown.

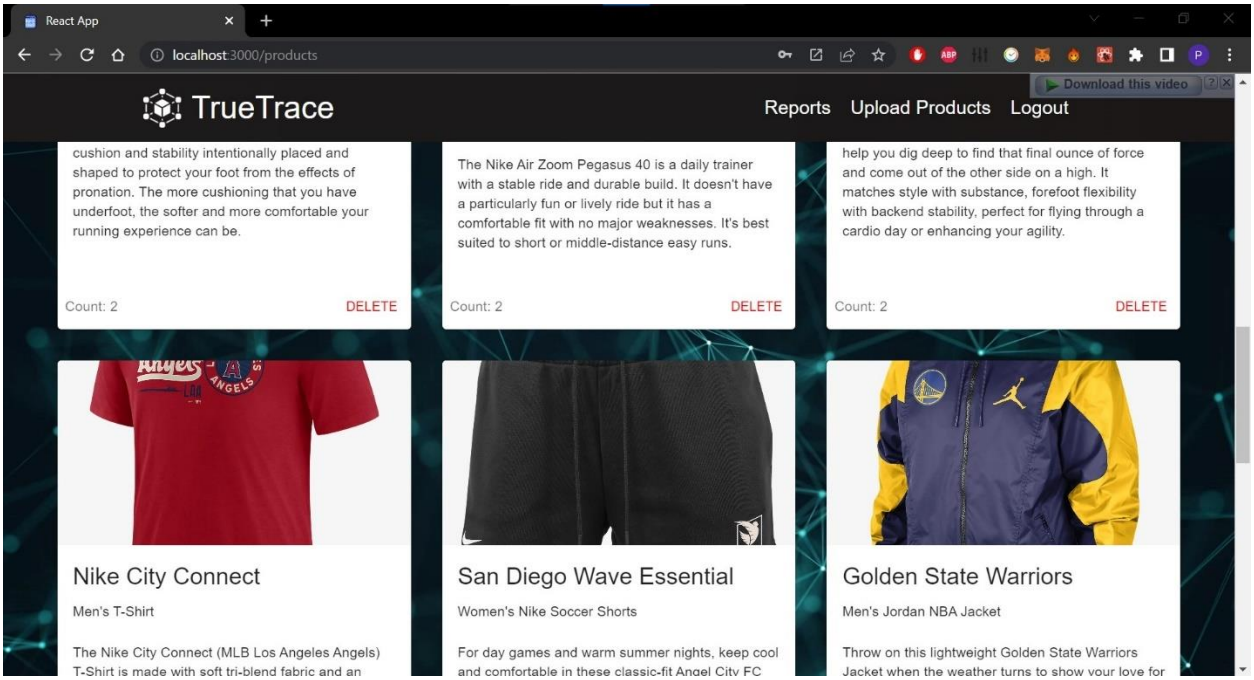

Figure 5.30: Product Shown with count for every model and delete button

Additionally, A company could delete a model for its Products.

The Company attempts to delete the air Force model.

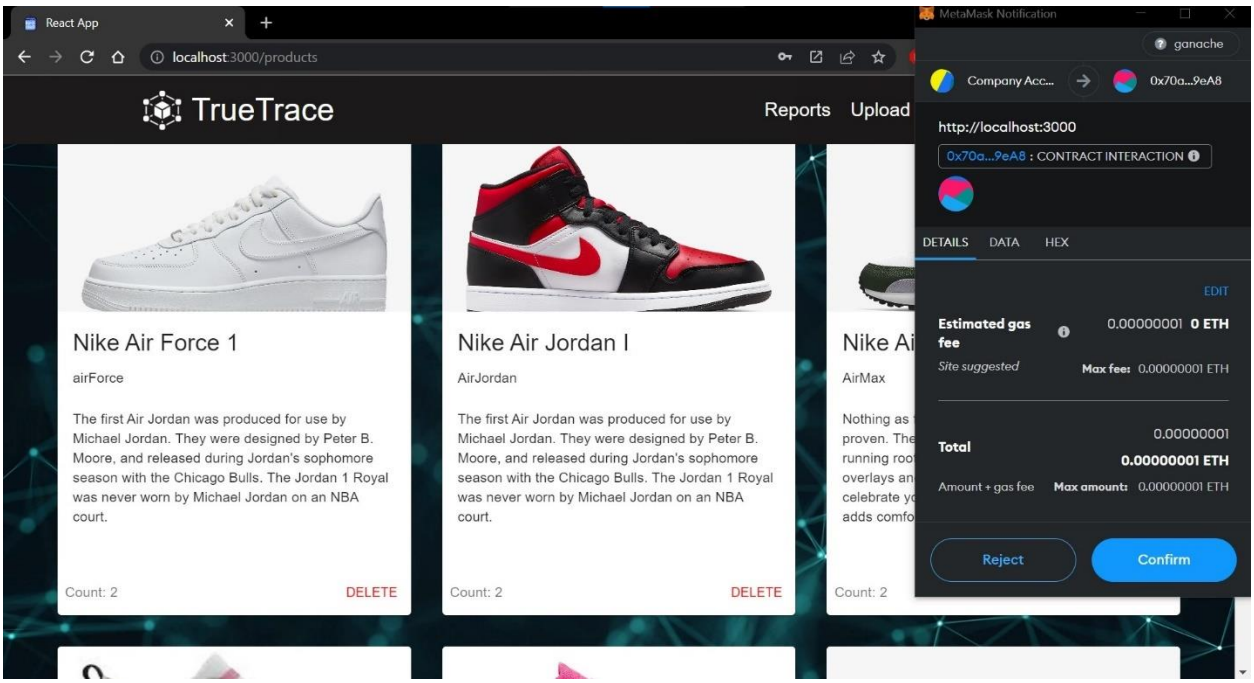

Figure 5.31: MetaMask Notification to Confirm Model Deletion

Company confirms Deletion of model, & removed from the products page.

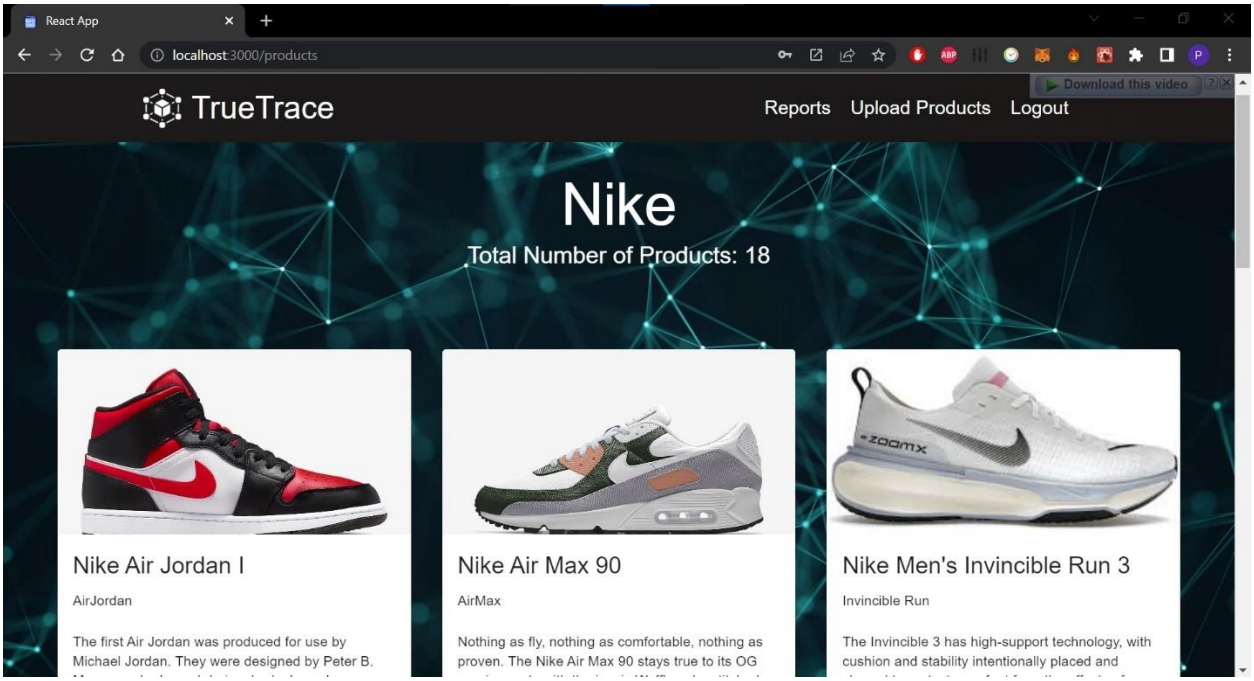

Figure 5.32: Delete Model is removed from the Products Page

The Company will now view its Reports by pressing the button in the Navigation Bar.

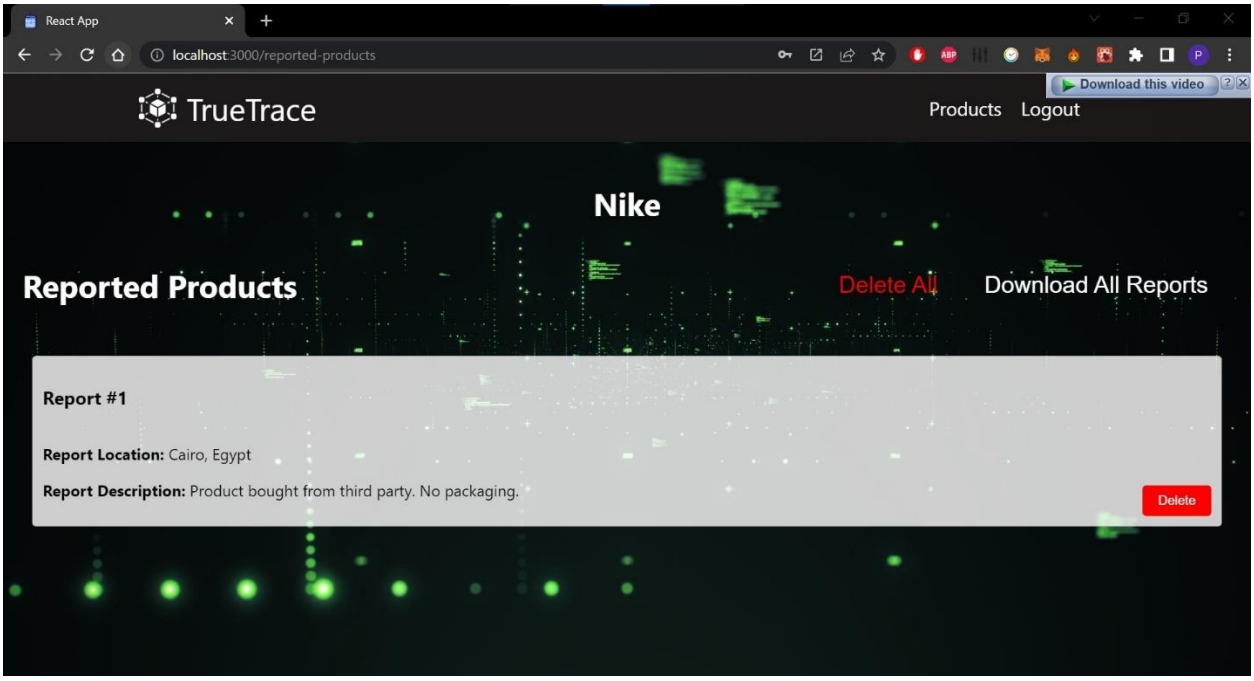

Figure 5.33: Reports Page

Company is now directed to its Reports page where every report submitted

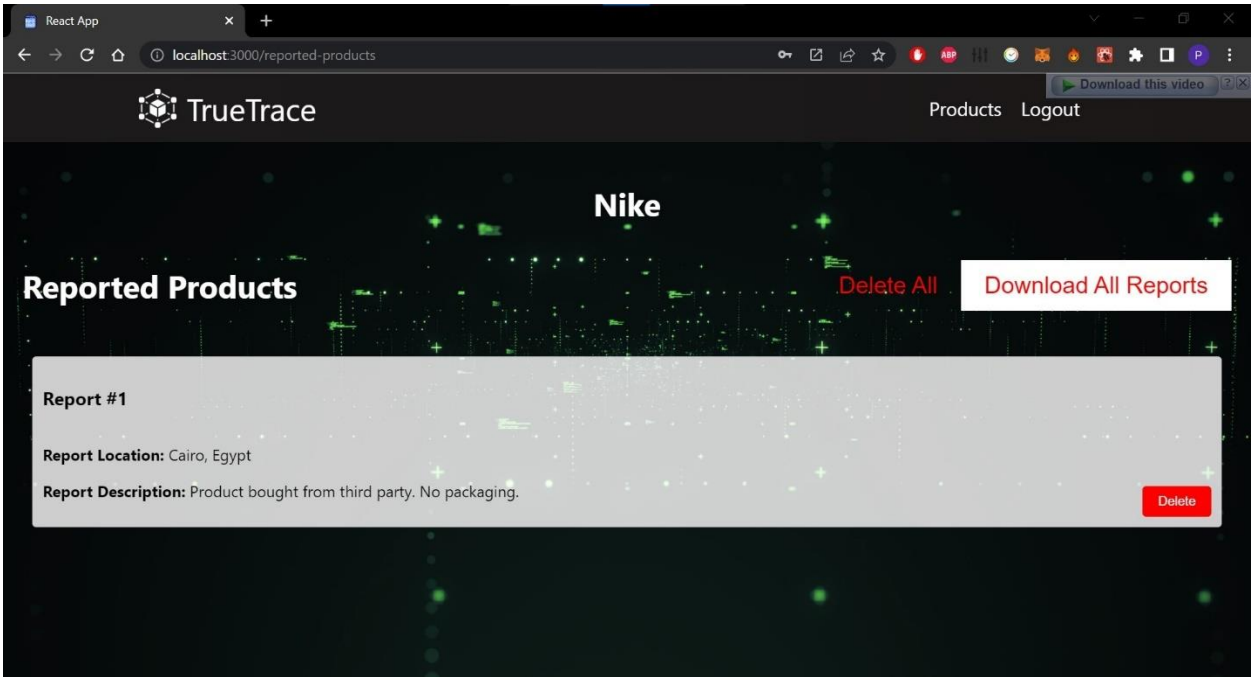

Figure 5.34: Download All Reports Button is Pressed

The Company Now will choose to Delete all the Reports it has received.

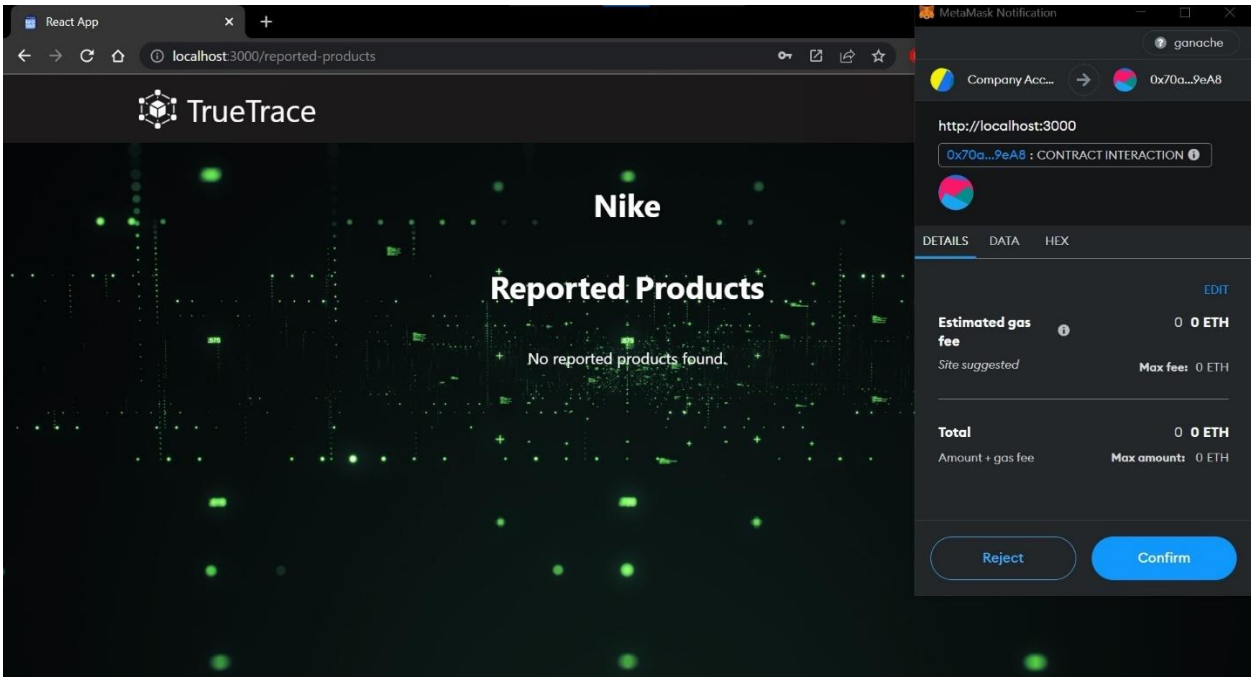

Figure 5.35: All Reports are Deleted

Chapter Six:
Conclusion and Future Work

6.1 Conclusion:

In conclusion, Blockchain Technology could help e-commerce business and customers by preventing product counterfeiting. From the time the product's manufactured until it reaches the customer, the manufacturer, distributor, and the customer will have access to the product's information. A third party or hacker can't alert the product's information between any of the links in the blockchain. Blockchain systems are governed by smart contract codes. When a product's delivered to the consumer and the created QR code is compared, the QR's confirmed. The user may have faith in this blockchain-based application because the code is so straightforward. The code may be made simpler in the next work. Customers won't be familiar with blockchain-based E-commerce websites, thus, to persuade them to purchase a product from this website, they must understand how this website differs from other shopping websites and the benefits this website offers. The website's promotion will be difficult.

6.2 Future Work:

In our plan the future work of our project, there are several areas that can be enhanced:

- **Mobile Application:**
  The system will be operated on different mobile platforms ex: Android, IOS.
- **Notification:**
  The system will be able to send notifications to users when a request comes up and for companies when their request has been answered.
- **Fingerprint:**
  Each new user can create an account using his unique fingerprint that is matched to his data saved at the government's repository.
- **Integrate with e-commerce platforms:**
  We plan to integrate the system with e-commerce platforms like Amazon, eBay, and Shopify. This will enable users to verify the authenticity of products before making a purchase. Also incentivize users to report fake products by implementing a reward system. Users can earn tokens or other rewards for reporting fake products, which can be used to purchase products on the platform or traded on cryptocurrency exchanges.
- **Machine Learning & Artificial Intelligence Integration:**
  To enhance the accuracy and efficiency of fake product detection, consider incorporating machine learning and artificial intelligence techniques into the application. By training models on large datasets of authentic and counterfeit product information, the system can learn patterns, anomalies, and unique characteristics to identify potential fakes more effectively. This integration can also enable continuous learning and improvement over time, keeping up with evolving counterfeiting techniques.

REFERENCES

[1] D. Khazanchi, A. Kumar Vyas, K. Kant Hiran, S. Padmanaban, “Blockchain 3.0 for Sustainable Development 1st Edition”, 2021 ISBN 978-3-11-0702453

[2] B. Adsul and S. Kosbatwar. A novel approach for traceability & detection of counterfeit medicines through blockchain. Technical Report 2539, EasyChair, February 2020.

[3] G. Blossey, J. Eisenhardt, and G. Hahn. Blockchain technology in supply chain management: an application perspective. In Proc. of the 52nd Hawaii International Conference on System Sciences (HICSS’19), Hawaii, USA, January 2019.

[4] S. E. Chang and Y. Chen. When blockchain meets supply chain: A systematic literature review on current development and potential applications. IEEE Access, 8:62478–62494, 2020.

[5] T. Tambe, S. Chitalkar, M. Khurud, M. Varpe, S. Y. Raut, ”Fake Product Detection Using Blockchain Technology,” in International Journal of Advance Research, Ideas and INNOVATIONS in Technology, vol. 7, pp. 314-319, 2021

[6] J. Ma, S. Lin, X. Chen, H. Sun, Y. Chen and H. Wang, ”A Blockchain-Based Application System for Product Anti-Counterfeiting,” in IEEE Access, vol. 8, pp. 77642-77652, 2020

[7] K. Toyoda, P. T. Mathiopoulos, I. Sasase and T. Ohtsuki, ”A Novel Blockchain-Based Product Ownership Management System (POMS) for Anti-Counterfeits in the Post Supply Chain,” in IEEE Access, vol. 5, pp.17465-17477, 2017

[8] S. Rahmadika, B. J. Kweka, C. N. Z. Latt and K. Rhee, ”A Preliminary Approach of Blockchain Technology in Supply Chain System,” 2018 IEEE International Conference on Data Mining Workshops (ICDMW),2018

[9] P. Gonczol, P. Katsikouli, L. Herskind, and N. Dragoni. Blockchain implementations and use cases for supply chains-a survey. IEEE Access, 8:11856–11871, 2020

[10] JETIR2207670 "FAKE PRODUCT IDENTIFICATION SYSTEM USING BLOCKCHAIN", International Journal of Emerging Technologies and

Innovative Research (www.jetir.org), ISSN:2349-5162, Vol.9, Issue 7, page no.g492-g500, July-2022

[11] T. J. Sayyad, ”Fake Product Identification Using Blockchain Technology,” in International Journal of Future Generation Communication and Networking, vol. 14, pp. 780-785, 2021

[12]. Fariha Jahan, Mayel Mostafa, Shahrin Chowdhury “SHA-256 in Parallel Blockchain Technology: Storing Land Related Documents” International Journal of Computer Applications (0975 – 8887) Volume 175 – No. 35, December 2020

[13]. Abhinav Sanghi, Aayush, Ashutosh Katakwar, Anshul Arora, Aditya Kaushik, “Detecting Fake Drugs using Blockchain”, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-10 Issue-1, May 2021.

[14]. G. Vidhya Lakshmi, Subbarao Gogulamudi, Bodapati Nagaeswari, Shaik Reehana, “Blockchain Based Inventory Management by QR Code Using Open CV”, International Conference on Computer Communication and Informatics (ICCCI -2021) Coimbatore, INDIA, Jan. 27 – 29, 2021