RETURN FRAUD PREVENTION SYSTEM USING BLOCKCHAIN

PROJECT REPORT

Submitted by

JOTHILAXMI H (195002055)

NANDINI R (195002075)

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

MAY 2023

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

BONAFIDE CERTIFICATE

Certified that this Report titled **"RETURN FRAUD PREVENTION SYSTEM USING BLOCKCHAIN"** is the bonafide work of **JOTHILAXMI H (195002055), NANDINI R (195002075)** who carried out the work under my supervision.

Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. Chandrabose Aravindan
Head of the department
Head of the Department
Department of
Information Technology
SSN College of Engineering
Kalavakkam – 603110

Dr. A. Sandana Karuppan
Supervisor
Associate Professor
Department of Information
Technology
SSN College of Engineering
Kalavakkam – 603110

Submitted for project viva-voce examination held on ……

EXTERNAL EXAMINER INTERNAL EXAMINER

ABSTRACT

Return fraud has become a significant challenge for retailers worldwide, resulting in substantial financial losses and operational inefficiencies. This project aims to develop a novel return fraud prevention system leveraging blockchain technology to mitigate these issues. The proposed system utilizes a decentralized, transparent, and tamper-proof ledger to record and validate product returns, ensuring data integrity and enhancing the traceability of return transactions. The core components of the system include a unique product identifier (UPI) generated for each sold item, a tamper-evident packaging solution, and a blockchain-based return validation process. The UPI is embedded into the product packaging and registered on the blockchain, which helps in tracing the item's ownership and history throughout its lifecycle. The authenticity of the product can be verified while returning it or while claiming its warranty. The tamper-evident packaging solution prevents unauthorized access and manipulation of the UPI, reducing the possibility of fraudulent returns. The blockchain-based return validation process involves smart contracts that automatically verify the authenticity and eligibility of return requests based on predefined rules, such as return window, condition, and ownership. This approach minimizes human intervention, streamlines the return process, and reduces the likelihood of fraud.

By implementing this blockchain-based return fraud prevention system, retailers can expect increased operational efficiency, reduced fraudulent activities, and improved customer satisfaction. Moreover, this system can

be easily integrated into existing retail management platforms, allowing for seamless adoption and scalability.

ACKNOWLEDGEMENT

We thank God, the Almighty for giving us strength and knowledge to do this project.

We express our deep respect to the founder **PADMA BHUSHAN DR. SHIVNADAR**, Chairman, SSN Institutions. We also express our appreciation to our **DR. V. E. ANNAMALAI**, Principal, for all the help he has rendered during this course of study.

Our sincere thanks to **DR. C. ARAVINDAN**, Professor and Head of the Department of Information Technology, for his words of advice and encouragement and we would like to thank our Project Coordinator **DR. T. SREE SHARMILA**, Associate Professor, Department of Information Technology and all our panel members for their valuable suggestions throughout the project.

We would like to thank and show our deep sense of gratitude to our guide **DR. A SANDANA KARUPPAN**, Associate Professor, Department of Information Technology, for his valuable advice and suggestions as well as his continued guidance, patience and support that helped us to shape and refine our work.

We would like to extend our sincere thanks to all the teaching and non-teaching staffs of our department who have contributed directly and indirectly during the course of our project work. Finally, we would like to thank our parents and friends for their patience, cooperation and moral support throughout our life.

JOTHILAXMI H NANDINI R

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS, ABBREVIATIONS

RFPS	Return Fraud Prevention System
UPI	Unique Product Identifier
Js	JavaScript
CSS	Cascading Style Sheets
HTML	Hyper Text Markup Language
ABI	Application Binary Interface
PHP	Hypertext Pre-processor
QR	Quick Response code
DeFi	Decentralized Finance
ASPA	Authentication Solution Providers' Association
GAO	Government Accountability Office
RFID	Radio-Frequency Identification
dApp	De-centralized Applications
SCQI	Supply Chain Quality Integration

REPORT ORGANISATION

The report is organized as follows:

Chapter 1: Introduction - Describes the problem statement, problem description, objective, explanation about the existing system and introduction to the proposed system.

Chapter 2: Literature Survey - Presents works related to our proposed model.

Chapter 3: System Design - Presents the architecture and working model of the system with detailed description of each module.

Chapter 4: Tools and Technologies used – Presents the software requirements and describes each module used in proposed approach.

Chapter 5: Implementation and Source code– Presents the implementation of the proposed smart contract and the source code of the project.

Chapter 6: Results and screenshots- Presents the output screenshots along with the workflow of the system.

Chapter 7: Conclusion and Future Work- Presents the additional work that can be done to enhance the system in the future.

Chapter 8: References.

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Blockchain is collection blocks that are linked together which stores information. Each block has a timestamp, transaction data and hash of its own and hash of previous block, so it is difficult to tamper with data. Blockchain is a decentralized system. It ensures that every new block added to the blockchain is the one and only true version that is agreed upon by all nodes in the Blockchain. It refers to the collective maintenance of a technical solution that maintains a continuous record file as a reliable database through decentralization.

One benefit of blockchain is that it significantly lowers the likelihood of a data leak. There are numerous shared copies of the same data base in Blockchain, in contrast to traditional methods, which makes it more difficult to carry out a data breach assault or cyber-attack. With its many anti-fraud characteristics, block chain technology has the potential to transform many industries by making business processes smarter, safer, transparent, and more effective than those used in the past.

Return fraud is an online scam that happens when a person buys a product from a business with the intention of replacing it with **used/duplicate/old product** while returning. This results in **loss for the**

sellers who sell their products online. Because of the behaviours of **counterfeiters**, distributors, retailers, and other business partners frequently lose faith in legitimate enterprises. It is **consumer fraud** and commonly defined as deceptive business practices that cause consumers to suffer financial or other losses.

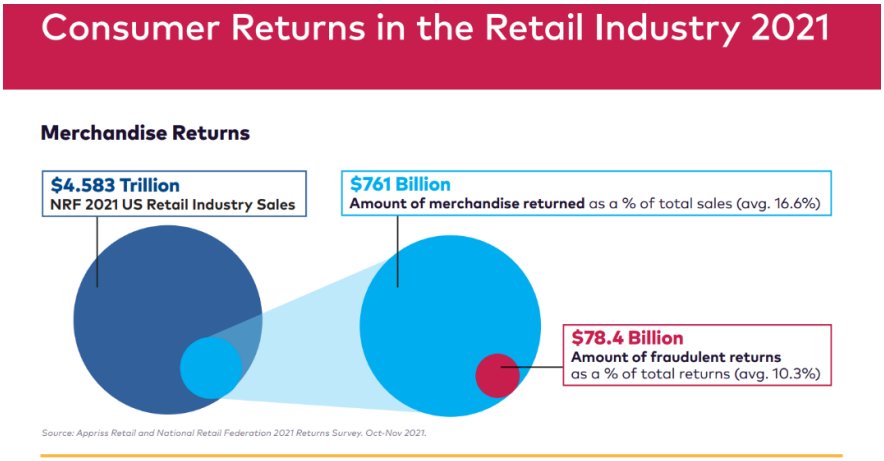

Figure 1.1

Product counterfeiting happens when a product is sold pre-tending to be another product. Trade in counterfeit and pirated goods makes up **3.3% of global trade** as of 2019.According to a report by the Authentication Solution Providers' Association (ASPA), the estimated loss to the Indian economy due to the sale of counterfeit products in 2019 was around **$34.1 billion.** Regarding fake laptops specifically, a study conducted by the United States **Government Accountability Office (GAO) in 2018** found that counterfeit electronic products, including laptops, were prevalent on e-commerce websites. The study purchased **47 electronic products** from third-party sellers on popular consumer portal and found that **20 of them were counterfeit.**

We intend to develop a blockchain based system where counterfeit items can be identified with a QR code scanner, as the product's QR code is linked to Blockchain. Thereby, return fraud is prevented. Blockchain technology is a distributed, decentralized, and digital ledger that keeps track of transactions in a large number of databases and node computers linked via chains. Blockchain is safe because no block can be altered or compromised since the data is **immutable once it has been added to the chain**. Customers or users do not need to rely on third-party users to **authenticate the legitimacy** and safety of a product thanks to Blockchain technology.

1.2 MOTIVATION

The retail industry faces a growing threat from return fraud, which results in billions of dollars in losses annually and undermines consumer trust in the shopping experience. This project is motivated by the need to develop a more robust and transparent return fraud prevention system, capable of effectively addressing this challenge.

We aim to build a blockchain-based system that can be used to generate **Ethereum encoded unique and immutable QR code for products** based on the product details given by the manufacturer.

So that,

- Return fraud can be avoided.
- Counterfeit products can be identified with a **QR code scanner**, as the product's QR code is linked to a hash value which is stored in a blockchain.

- Service centres can verify the warranty period details.

1.3 OBJECTIVES

Implement a return fraud prevention mechanism through the use of Blockchain-backed QR codes and secure product data management.

- To Implement login/logout and registration modules for various user roles.
- To enable manufacturers to generate unique QR codes for products, incorporating elements such as Product ID, warranty period, and product name.
- To integrate QR code generation with a Blockchain for enhanced security and traceability.
- To allow manufacturers to specify product location details within the system.
- To facilitate sellers in updating product location and purchase date information.
- To empower customers and service centres to verify product authenticity through QR code scanning.

1.4 EXISTING SYSTEM

There exist various systems of fake product detection, which use Artificial Intelligence, QR codes, Machine Learning and Blockchain. Current systems for authenticating products have various limitations. For instance, brands employ QR codes to verify product authenticity, but these codes can be replicated and used on counterfeit items. Similarly, low-cost RFID tags are used for automatic product identification, but they are vulnerable to cloning, rendering this method unsuitable. In artificial intelligence and machine learning applications, convolutional neural networks (CNNs) require significant time, memory, and training before deployment, making them less efficient. Moreover, AI struggles to detect tag reapplication attacks, where counterfeiters remove legitimate tags from genuine products and apply them to fake or expired ones. As a result, customers, suppliers, and retailers lack the ability to effectively ensure product integrity.

1.5 PROPOSED SYSTEM

The proposed approach is designed for consumer products, utilizing blockchain technology to track items while preserving product and supply chain integrity. This approach grants customers the ability to follow a product's entire history, from its production to its arrival in their hands, through the use of blockchain and QR codes. The primary objective of this system is to combat return fraud by offering a transparent and verifiable tracking method.

By implementing a blockchain-based tracking system, both customers and retailers can access secure and accurate information regarding the origins and movement of consumer goods. This transparency not only fosters trust between parties but also helps to identify and prevent fraudulent returns. The QR codes serve as a user-friendly tool for consumers to quickly and effortlessly authenticate their purchases.

In summary, the proposed return fraud prevention system leverages the power of blockchain technology and QR codes to enhance the tracking and verification process for consumer products. By promoting transparency and trust in the retail ecosystem, this system effectively addresses the widespread issue of return fraud.

1.6 FEATURES OF BLOCKCHAIN

i. Security and Privacy – blockchain uses cryptography to secure its data.

ii. Decentralized – member in the network has a copy of the exact same data in the form of a distributed ledger

iii. Untrace ability – Once the block is entered into the blockchain, it cannot be tampered.

iv. Transparency.

v. Flexibility – Being open source is one of biggest advantages of blockchain.

1.6.1 Ethereum Blockchain

Ethereum is a decentralized blockchain that employs a proof-of-work consensus mechanism. In order to add a block to the blockchain, nodes must solve a mathematical problem, known as a puzzle, using computational resources. This process is called mining, and it verifies that the block has been added and recorded in the blockchain. Mining is a brute force trial and error process, but it is rewarded in Ethereum when a block is successfully added. Therefore, proof-of-work serves as a means of demonstrating that nodes have performed the necessary work to add a block to the blockchain.

CHAPTER 2

LITERATURE SURVEY

2.1 PREVIOUS WORKS

C. Shaik et al. [1] The proposed method is used to prevent counterfeit products using cryptography, QR code and web service. This method requires that every original product manufacturer obtain a cryptographic key pair, securely store their private key and publish their public key on their website as a QR code.

G. Khalil, R. Doss, M. Chowdhury et al. [2] have proposed a method that is about RFID based anti-counterfeiting and anti-theft scheme that can be used to detect counterfeit items at the point of purchase by a consumer.

E. Daoud, D. Vu, H. Nguyen, M. Gaedke et al. [3] have proposed a methodology that develops a ML model to reduce counterfeit products using machine learning-based technology.

M.A. Habib, M.B. Sardar, S. Jabbar, C.N. Faisal, N. Mahmood, M. Ahmad et al.[4] proposed a methodology to analyse the trust issue in supply chain and design a new propose scheme based on blockchain

technology for resolving the problem in the supply chain and automate the whole payment process through a smart contract.

Moin, S. S., & Nguyen, T. T. (2019)et al.[7] : have proposed a Blockchain system for Supply Chain Traceability. The authors propose a framework to meet the business requirements of traceability, highlighting its potential to tackle counterfeit products.

Chen and Shi et al.[8] (2017) : they have proposed a framework for blockchain based SCQI that provides a theoretical basis to intelligent quality management of supply chains based on blockchain technology. RFID technology is used to record quality information, trans- action information. Smart contracts are used to execute quality control and improve the efficiency of the supply chain.

Toyoda, Kentaroh and Mathiopoulos, P Takis et al.[9] (2017) have proposed a system to detect fake product with the help of QR code. End users can scan the QR code assigned to product to get the product details and transaction history, the steps involved Product enrolment, ship product to distributor, and ship product to retailer, end user gets details about the product.

2.2 INFERENCE FROM THIS SURVEY

Through the above literature survey, we conclude that the already proposed systems deliver considerable results in the author's own perspective. But then we are developing this system with enhanced security and immutability, which is not much focused on the previously developed models. This motivates us to develop a model that primarily focuses on Anti-Counterfeiting of products.

CHAPTER 3

SYSTEM DESIGN

The upcoming system aims to prevent return fraud through a decentralized application that will utilize the Ethereum Network as its primary blockchain for recording and managing transactions related to the products of listed companies.

The proposed design intends to enhance the security and transparency of the return process by leveraging the immutable nature of blockchain technology. By doing so, the system can provide an efficient and trustworthy solution for companies to manage their product returns, reducing the instances of fraudulent activity.

The system's working nature has been split into two different tasks,

i. Adding the Product details to the Blockchain.
ii. Updating the product details and scanning the QR to retrieve the details.

3.1 PROPOSED METHODOLOGY

The below Figure 3.1 depicts the pictorial representation of the Project workflow

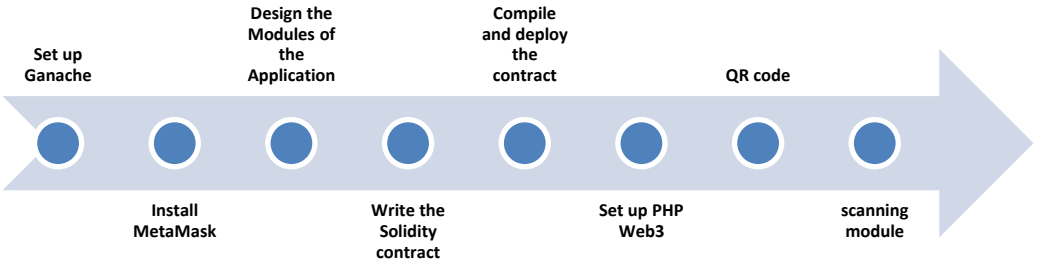

Figure 3.1

i. The first step is to set up a personal blockchain for Ethereum development using Ganache. Ganache provides a safe testing environment for developers to experiment with their code before deploying it to the live network. It is important to create a new workspace with the desired network settings, which will provide a local blockchain to test the Solidity contract.

ii. After setting up Ganache, we need to install the MetaMask browser extension and connect it to the Ganache network. This enables them to manage their Ethereum accounts and connect to the network. Once connected, we can start building the website application.

iii. The website application should include login/logout, registration, and separate modules for manufacturers, sellers, customers, and service providers. HTML, CSS, JS, and PHP can be used to develop the website application.

iv. Next, we need to write Solidity smart contracts to add product details into the blockchain. Solidity is a programming language

used for writing smart contracts on the Ethereum blockchain. These contracts define the rules and logic for adding product details and ensure that the data is stored securely on the blockchain.

v. Once the Solidity contract has been written, it needs to be compiled and deployed to the Ganache network. Remix-Ethereum, a lightweight Solidity compiler, can be used for the compilation process. After compiling the contract, it can be deployed to the network.

vi. To interact with the Solidity contract using PHP, we need to install the PHP Web3 library and configure it to connect to the Ganache network. Then we can write the PHP script that interacts with the Solidity contract. This script will enable users to query the blockchain for product details.

vii. To generate QR codes, we need to include libraries and write functions that are required to generate the QR code. The QR code can be scanned by users, and the details of the product can be displayed.

viii. In order to develop and integrate the scanning module, we can use various technologies such as the Qrious, which is a multi-format 1D/2D barcode image processing library. This library can be used to decode QR codes from image files or a webcam. By developing and integrating the scanning module, users can easily scan the QR code and display the product details.

3.2 SYSTEM ARCHIETECTURE

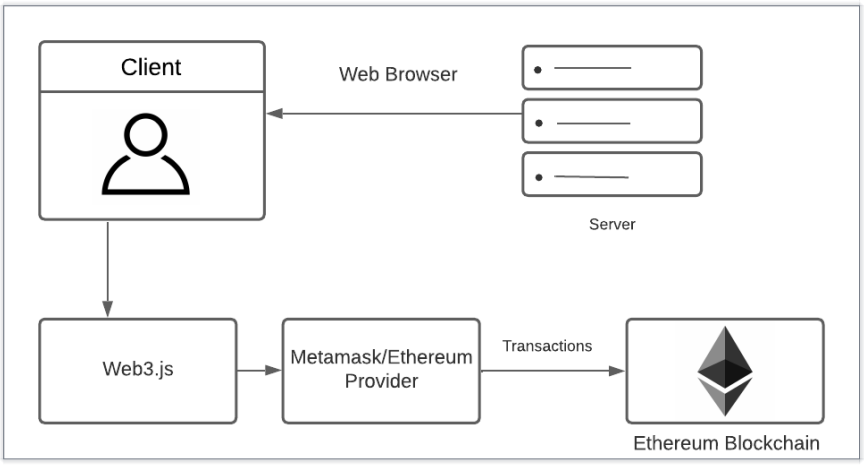

Figure 3.2

The user first logs-in into our application, based on the role of the user the functionalities of our user vary accordingly. As we access local Ethereum Blockchain (via. Ganache) for our project we use Web3.js library to interact with the blockchain. In-order to handle the transactions we need a digital wallet which maintains the simulated Ethers that are needed for carrying out the transactions, this job is done by the MetaMask wallet. According our system's design, as and when we create a new Item and when we update the details of the product transactions takes place. So, whenever the transactions take place blocks are added to the Blockchain.

3.3 SYSTEM WORKFLOW

The below Figure 3.3 is the pictorial representation of the system's workflow.

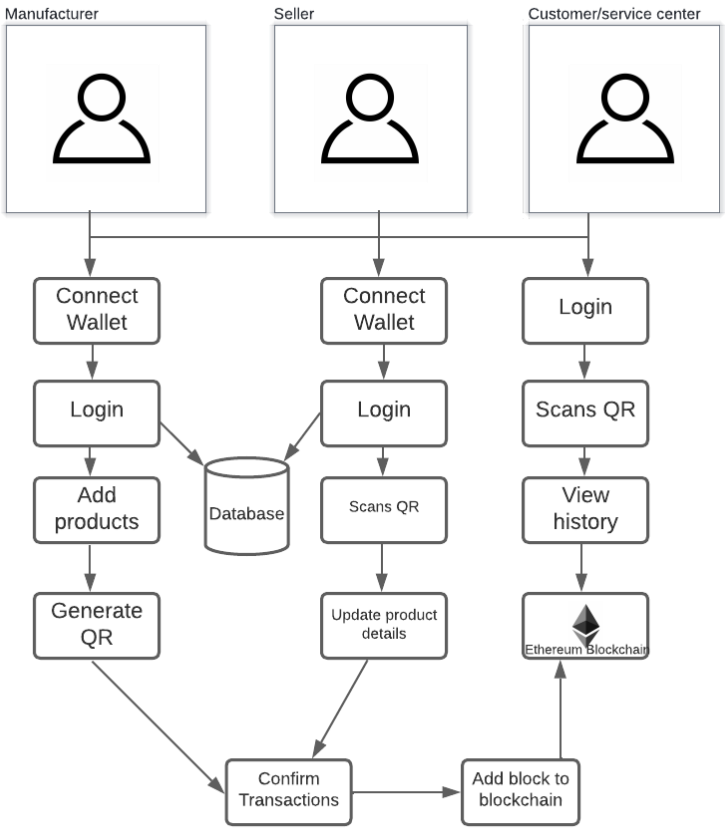

Figure 3.3

The proposed system has a primary goal of maintaining the genuineness of products by allowing customers to track their supply chain history through blockchain technology.

This anti-counterfeiting system is composed of three main roles:

- Manufacturer
- Seller

- Consumer

Manufacturer

The Manufacturer is responsible for logging into their account and generating a unique QR code for the product. They also add necessary product details and use their Ethereum wallet to add a block to the Ethereum blockchain. A mapping is created between the manufacturer's user ID in the local database and their wallet address. The manufacturer can add a block to the digital ledger only if they use their own account and wallet.

Seller

The Seller has access to the product's QR code and can scan it to access the product information entered by the manufacturer. The Seller adds their own information, such as the shop destination, and pushes it into the blockchain. The buyer can then view the product information.

Customers/Service Centers

Customers can verify the authenticity of the product by scanning the QR code, which lists the history of transactions. If the last location in the supply chain history does not match the purchase location, the customer can determine that the product is not genuine. This indicates that the QR code has been copied, and the customer becomes aware of counterfeiting.

CHAPTER 4

TOOLS AND TECHNOLOGIES USED

4.1SOFTWARE REQUIREMENT

Ganache, MetaMask, Remix Ethereum and XAMPP Server

4.2 DEVELOPMENT ENVIRONMENT

4.2.1Software Requirements:

- Ganache: It requires Node.js (v8 or higher) and npm to be installed on your system.
- MetaMask: It requires Google Chrome, Mozilla Firefox, Brave, or Microsoft Edge to be installed on your system.
- XAMPP Server: It requires Apache, MySQL, and PHP to be installed on your system.

4.2.2 Operating System: Ganache, MetaMask, and XAMPP Server can be run on Windows, MacOS, or Linux.

- Processor: A multi-core processor with a clock speed of at least 2 GHz is recommended.
- Memory: A minimum of 8 GB of RAM is recommended for optimal performance.

- Hard Drive Space: At least 20 GB of free hard drive space is recommended.
- Graphics Card: A dedicated graphics card is not required, but it is recommended for better performance.
- Internet Connection: A stable internet connection is required to use MetaMask.

4.3 LANGUAGES & PACKAGES

4.3.1 Php

PHP is a server-side scripting language used for developing dynamic web applications. It is used to interact with the blockchain network, allowing for the retrieval and verification of data stored on the blockchain. Additionally, PHP is used to develop smart contracts that can be executed on the blockchain, allowing for the automation of fraud detection processes.

4.3.2 jQuery

jQuery is a JavaScript library that simplifies the process of HTML document traversal, event handling, and animation. It is used to build a user interface for a fraud detection system, allowing users to interact with the system and view real-time data. It is also used to perform client-side validation of data entered by users, ensuring that only valid data is sent to the blockchain for verification.

4.3.3 JavaScript

JavaScript is a programming language used for developing web-based applications. It is used to build a fraud detection system by providing a client-side interface for interacting with the blockchain network. It is also used to write code that interacts with the blockchain, allowing for the retrieval and verification of data stored on the blockchain. Additionally, it is used to write smart contracts that can be executed on the blockchain, automating fraud detection processes.

4.3.4 Solidity

Solidity is a programming language used to write smart contracts on the Ethereum blockchain. It is used to create smart contracts that can be executed on the blockchain. Solidity is used to write code that verifies data stored on the blockchain and performs fraud detection processes automatically. Function to add product is included in the smart contract which is executed when adding a new product / updating the location details.

4.3.5 Html and CSS

HTML and CSS are markup languages used for creating web pages. They can be used to build a user interface for a fraud detection system, allowing users to interact with the system and view real-time data. HTML and CSS can be used to create forms for collecting data from users, which can be validated using client-side JavaScript before being sent to the blockchain for verification.

4.3.6 Qrious

Qrious is a pure JavaScript library for generating QR codes using HTML5 canvas. It is a simple, dependency-free JavaScript library which uses HTML5 canvas to generate QR codes with variable colors, sizes and error correction levels.

4.3.7 Web3.js

Web3.js is a JavaScript library that allows developers to interact with the Ethereum blockchain. It provides a set of APIs that make it easy for developers to build decentralized applications (dApps) on top of Ethereum. With Web3.js, developers can connect to a local or remote Ethereum node and send transactions, query data, and receive events. It supports both synchronous and asynchronous communication, making it easy to build responsive and interactive dApps. Web3.js also provides a set of utility functions that help developers work with Ethereum-specific data types, such as addresses and units of ether. Overall, Web3.js is an essential tool for any developer building applications on the Ethereum network.

4.4 ALGORITHMS

4.4.1 Keccak-256

Keccak-256 is a cryptographic hash function that is used in Ethereum for various purposes, including creating unique addresses and validating transactions. It takes an input message of any length and produces a fixed-size output of 256 bits. The output, known as the hash, is

a unique and deterministic representation of the input message. In Ethereum, Keccak-256 is used to create the addresses of accounts, which are used to send and receive ether and other tokens. When a new account is created, its address is derived from the public key of the account owner using Keccak-256. Keccak-256 is also used to create the hashes of transactions, which are then included in blocks on the Ethereum blockchain. These hashes are used to verify the authenticity and integrity of transactions and ensure that they cannot be tampered with. In summary, Keccak-256 is a critical component of the Ethereum network, ensuring the security and immutability of the blockchain.

4.5 TOOLS AND ENVIRONMENT

4.5.1 Ganache

Ganache is a popular Ethereum blockchain emulator that allows developers to test and debug their smart contracts and decentralized applications (dApps) in a local, private environment. This is especially useful for developers who want to build Ethereum-based solutions, but do not want to use real Ethereum networks, such as the mainnet or testnet, to deploy and test their applications. Instead, Ganache provides a **simulated blockchain environment**, which can be used to execute smart contracts, deploy and test dApps, and simulate transactions with pre-funded accounts, all on the developer's local machine.

One of the main benefits of using Ganache is that it provides a fast and efficient way to test and debug Ethereum smart contracts and dApps. Unlike deploying smart contracts on the live Ethereum network,

which can be slow and costly, Ganache offers a much faster, more **cost-effective testing environment**. This is because Ganache allows developers to simulate transactions and interactions with smart contracts, without the need for real Ether or gas fees.

Another advantage of using Ganache is that it can be easily integrated with various Ethereum development tools and frameworks, such as Truffle and Remix. These tools provide additional functionality and features, such as contract compilation, deployment, and testing, making it even easier for developers to build and test their Ethereum-based solutions. Additionally, Ganache allows developers to configure the blockchain environment to their specific needs, including adjusting gas prices, block times, and chain size, providing a high level of flexibility and customization. Overall, Ganache is an essential tool for Ethereum developers who want to build and test their smart contracts and dApps in a fast, efficient, and cost-effective manner.

4.5.2 MetaMask

MetaMask is a popular browser extension that allows users to interact with Ethereum-based decentralized applications (dApps) from their web browser. It acts as a bridge between the user's web browser and the Ethereum network, allowing users to send and receive Ether and other tokens, as well as interact with smart contracts and dApps.

In return fraud prevention system, MetaMask can be used to verify the identity of users and ensure that only authorized transactions are executed on the blockchain. For example, a user may be required to authenticate themselves using their MetaMask wallet before performing a

transaction, such as transferring funds or signing a smart contract. This can prevent unauthorized access to the system and ensure that only trusted users are able to perform sensitive operations.

4.5.3 Remix-Ethereum

Remix Ethereum is an open-source browser-based integrated development environment (IDE) for writing, testing, and deploying smart contracts on the Ethereum network. It provides a user-friendly interface for developers to write and test their smart contracts, as well as interact with the Ethereum network. Remix is often used in conjunction with MetaMask, which acts as a bridge between Remix and the Ethereum network, allowing developers to test their smart contracts and dApps in a secure and efficient manner.

When using Remix with MetaMask, developers can connect their MetaMask wallet to Remix and use it to test their smart contracts on the Ethereum network. This allows developers to simulate real-world scenarios and interactions with their smart contracts, without the need for a separate testing environment or deploying their contracts to the main Ethereum network. MetaMask also provides a secure way for developers to sign transactions and execute smart contracts, ensuring that their interactions with the network are secure and authenticated.

Remix and Metamask together provide a powerful toolset for Ethereum developers to write, test, and deploy their smart contracts and dApps. With Remix, developers can easily write and test their smart

contracts, and with Metamask, they can securely interact with the Ethereum network and test their contracts in a real-world environment.

Overall, the combination of Remix and Metamask provides a comprehensive solution for Ethereum development, allowing developers to build and test their smart contracts and dApps in a secure and efficient manner.

4.5.4 Xampp server

XAMPP is a popular open-source software package that provides a local web server environment for developers to test and develop their websites and web applications. It is designed to work on multiple platforms, including Windows, macOS, and Linux, and provides a comprehensive set of tools for developers to set up a complete web server environment on their local machine.

The XAMPP server includes the Apache web server, MySQL database server, PHP interpreter, and various other components and modules that are commonly used in web development. These components are pre-configured to work together seamlessly, allowing developers to quickly set up a complete web server environment without the need for extensive configuration or installation.

In addition to providing a local web server environment, XAMPP also includes tools for managing and testing websites and web applications, including a file manager, database administration tools, and various debugging and testing tools. This makes it a powerful toolset for

developers who want to test and debug their websites and web applications in a local environment before deploying them to a live server.

Overall, XAMPP provides a comprehensive and user-friendly solution for developers who want to set up a local web server environment for testing and developing their websites and web applications. Its pre-configured components and tools make it easy to get started, while its comprehensive set of features and tools make it a powerful toolset for web development.

4.6 FEASIBILITY STUDY:

4.6.1 Technical feasibility:

The implementation of blockchain can be done with the following set of tools:

Solidity – for writing smart contracts

MetaMask- is a browser extension that allows us to interact with the Ethereum network

Ganache – is an open-source software, a local blockchain development tool that allows us to create a local test network.

Remix Ethereum - Used to compile smart contracts.

4.6.2 Financial Feasibility:

We use Ganache, a virtual blockchain which sets up 10 default Ethereum addresses with complete private keys, and pre-loads them with 100 simulated Ether each which helps us pay the gas fee required for making transactions.

CHAPTER 5

IMPLEMENTATION & SOURCE CODE

5.1 SMART CONTRACT

A smart contract is a self-executing digital contract that is stored on a blockchain. It is a computer program that automatically enforces the rules and regulations of a contract. Smart contracts are designed to reduce transaction costs, increase transparency, and minimize the need for intermediaries. We've framed the smart contract of our application in such a way that we have written the functions in solidity for adding the product details to the blockchain and to retrieve it from the blockchain.

Smartcontract.sol

```
pragma solidity ^0.6.0;
contract SupplyChain {

  event Added(uint256 index);

  struct State{
    string description;
    address person;
  }
```

```
    struct Product{
        address creator;
        string productName;
        uint256 productId;
        string date;
        uint256 totalStates;
        mapping (uint256 => State) positions;
    }

    mapping(uint => Product) allProducts;
    uint256 items=0;

    function concat(string memory _a, string memory _b) public returns (string memory){
        bytes memory bytes_a = bytes(_a);
        bytes memory bytes_b = bytes(_b);
        string memory length_ab = new string(bytes_a.length + bytes_b.length);
        bytes memory bytes_c = bytes(length_ab);
        uint k = 0;
        for (uint i = 0; i < bytes_a.length; i++) bytes_c[k++] = bytes_a[i];
        for (uint i = 0; i < bytes_b.length; i++) bytes_c[k++] = bytes_b[i];
        return string(bytes_c);
    }

    function newItem(string memory _text, string memory _date) public returns (bool) {
```

```
        Product memory newItem = Product({creator: msg.sender,
totalStates: 0,productName: _text, productId: items, date: _date});
        allProducts[items]=newItem;
        items = items+1;
        emit Added(items-1);
        return true;
    }

    function addState(uint _productId, string memory info) public returns
(string memory) {
        require(_productId<=items);

        State memory newState = State({person: msg.sender, description:
info});

        allProducts[_productId].positions[
allProducts[_productId].totalStates ]=newState;

        allProducts[_productId].totalStates                              =
allProducts[_productId].totalStates +1;
        return info;
    }

    function searchProduct(uint _productId) public returns (string memory)
{

        require(_productId<=items);
        string memory output="Product Name: ";
        output=concat(output, allProducts[_productId].productName);
```

```
        output=concat(output, "<br>Manufacture Date: ");
        output=concat(output, allProducts[_productId].date);

        for (uint256 j=0; j<allProducts[_productId].totalStates; j++){
            output=concat(output,
allProducts[_productId].positions[j].description);
        }
        return output;
    }

     }
```

5.2 PHP FILES

The main functionality of the project i.e Adding the product details, Updating and Retrieving the product details are done by following code.

5.2.1 Addproduct.php

```
<?php
   if( $_SESSION['role']=="Manufacturer" ){
  ?>
   $('#form1').on('submit', function(event) {
       event.preventDefault(); // to prevent page reload when form is
submitted
       prodname = $('#prodname').val();
```

```
        prodtype = $('#prodtype').val();
        prodmodel = $('#prodmodel').val();
        snumber = $('#snumber').val();
        storage = $('#storage').val();
        ram = $('#ram').val();
        Screensize = $('#ss').val();
        Processer= $('#Processer').val();
        wperiod = $('#wperiod').val();
        username = $('#user').val();
prodname=prodname+"<br>Producttype: "+prodtype+"<br>Product
model : "+prodmodel+"<br>Serial number : "+snumber+"<br>Storage :
"+storage+"<br>RAM : "+ram+"<br>Screensize :
"+Screensize+"<br>Processer: "+Processer+"<br>Warranty period :
"+wperiod+"<br>Registered By: "+username;
   console.log(prodname);
 var today = new Date();
 var thisdate = today.getFullYear()+'-'+(today.getMonth()+1)+'-
'+today.getDate();

web3.eth.getAccounts().then(async function(accounts) {
var receipt = await contract.methods.newItem(prodname, thisdate).send({
from: accounts[0], gas: 1000000 }).then(receipt =>
{
var msg="<h5 style='color: #53D769'><b>Item Added
Successfully</b></h5>
<p>Product ID: "+receipt.events.Added.returnValues[0]+"</p>";
qr.value = receipt.events.Added.returnValues[0];
$bottom="<p style='color: #FECB2E'> You may print the QR Code if
required </p>"
```

```
$("#alertText").html(msg);
            $("#qrious").show();
            $("#bottomText").html($bottom);
            $(".customalert").show("fast","linear");
          });
          //console.log(receipt);
        });
        $("#prodname").val('');

    });
```

5.2.2 Checkproduct.php

To check the authenticity of the product by scanning the QR code.

```
function openFile (node) {
var reader = new FileReader();
reader.onload = function() {
node.value = "";
qrcode.callback = function(res) {
if(res instanceof Error) {
alert("No QR code found. Please make sure the QR code is within the camera's frame and try again.");
}
else {
node.parentNode.previousElementSibling.value = res;
```

```
document.getElementById('searchButton').click();
}
};
qrcode.decode(reader.result);
};
reader.readAsDataURL(node.files[0]);
}
```

//Opens Camera to Scan QR code

```
function onScanSuccess(qrCodeMessage) {
   document.getElementById('searchText').value = qrCodeMessage;
}
var html5QrcodeScanner = new Html5QrcodeScanner(
  "reader", { fps: 10, qrbox: 250 });
html5QrcodeScanner.render(onScanSuccess, onScanError);
```

5.2.3 Scanshipment.php

To update the details of the product

```
if (navigator.geolocation) {
navigator.geolocation.getCurrentPosition(showPosition);
}
function showPosition(position) {
var autoLocation = position.coords.latitude +", " +
position.coords.longitude;
$("#prodlocation").val(autoLocation);
```

```
}

$('#form2').on('submit', function(event) {
event.preventDefault(); // to prevent page reload when form is submitted
prodid = $('#prodid').val();
prodlocation = $('#prodlocation').val();
console.log(prodid);
console.log(prodlocation);
username = $('#user').val();
var today = new Date();
var thisdate = today.getFullYear()+'-'+(today.getMonth()+1)+'-
'+today.getDate();
if (username=="0"){
var result= "Manufacturer";
}
else if(username=="1"){
var result="Seller";
}
else if (username=="3"){
var result ="Distributor"
}
var info = "<br><br><b>Date: "+thisdate+"</b><br>Location:
"+prodlocation +"</b><br>Updated by "+ result;

web3.eth.getAccounts().then(async function(accounts) {
var receipt = await contract.methods.addState(prodid, info).send({ from:
accounts[0], gas: 1000000 })
.then(receipt => {
var msg="Item has been updated ";});
```

CHAPTER 6

RESULTS AND OUTPUTS

This section details the outcomes of the developed Return Fraud Prevention system. The output screenshots are presented to show the workflow of the system.

6.1 Setting up of Environment to run the application

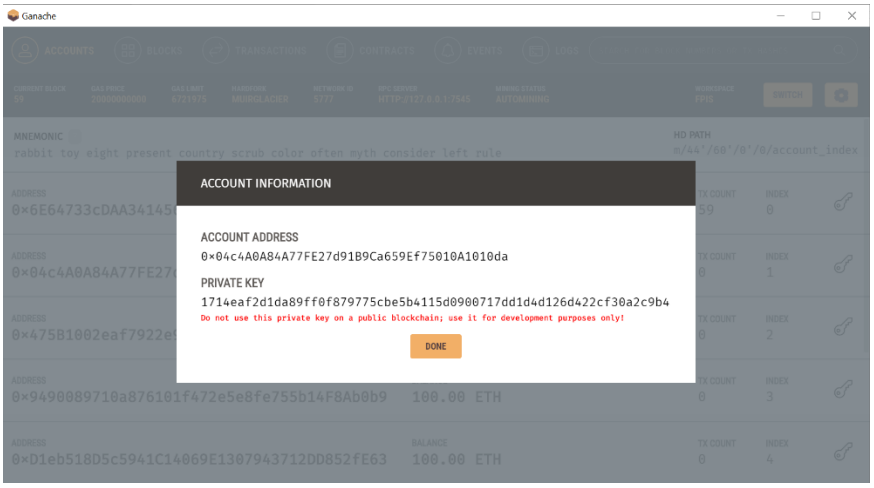

Figure 6.1

Inorder to run the dApp we first need to set up a new workspace in Ganache, then copy the private key of any account from the listed accounts.

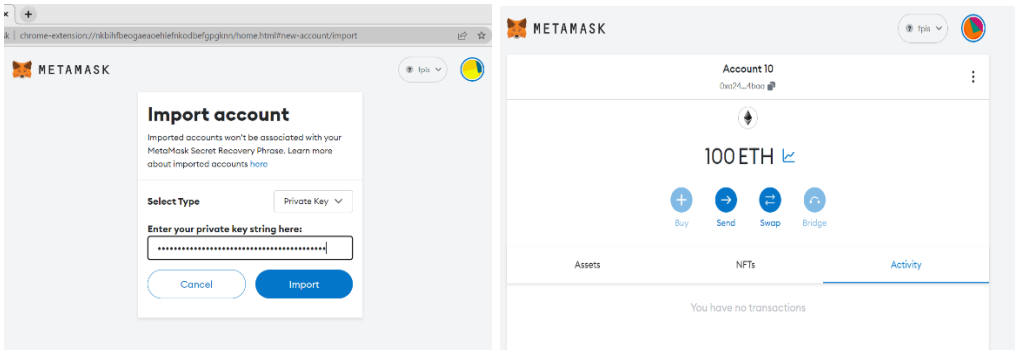

Figure 6.2

To establish the connection with the MetaMask wallet create a test network in the Metamask wallet and import the account by pasting the private key that we extracted from the ganache. If the account is imported successfully then simulated ethers will be added to our account.

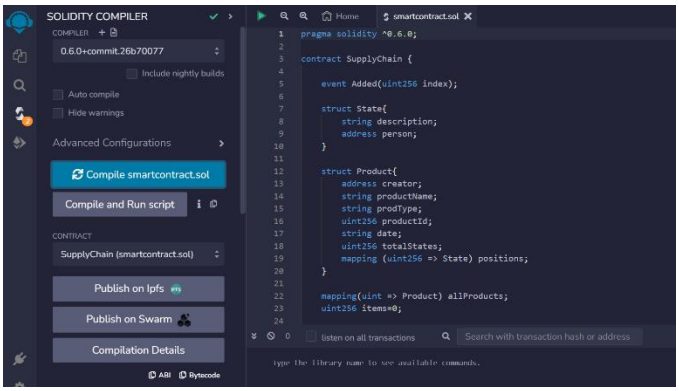

Figure 6.3

Next, we compile the smart contract using the Remix Ethereum environment. We import the smart contract file into the remix Ethereum environment and compile it by selecting the appropriate version of the solidity compiler. (ref figure-6.3)

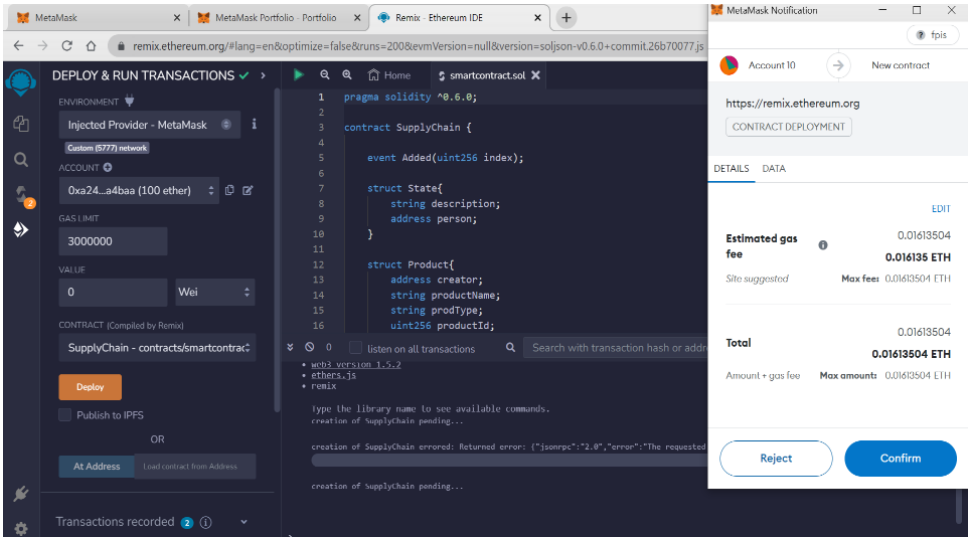

Figure 6.4

Once compiled, we deploy the smart contract by selecting MetaMask environment. When we deploy our contract the MetaMask wallet asks for the confirmation of the transaction as the deployment of the contract consumes the simulated ethers.

Once compiled successfully we get the contract ABI address, which needs to be replaced in the source code for running our project.

To run our application in the local machine we start the Xampp server and open the port which runs the application.

6.2 User Authentication

This application features a basic authentication system that utilizes email and password. The user's login information is stored in the PHP database. The app has two main pages: Login and SignUp. If the user is new to the app, they must go to the SignUp page and register with

their email and password.

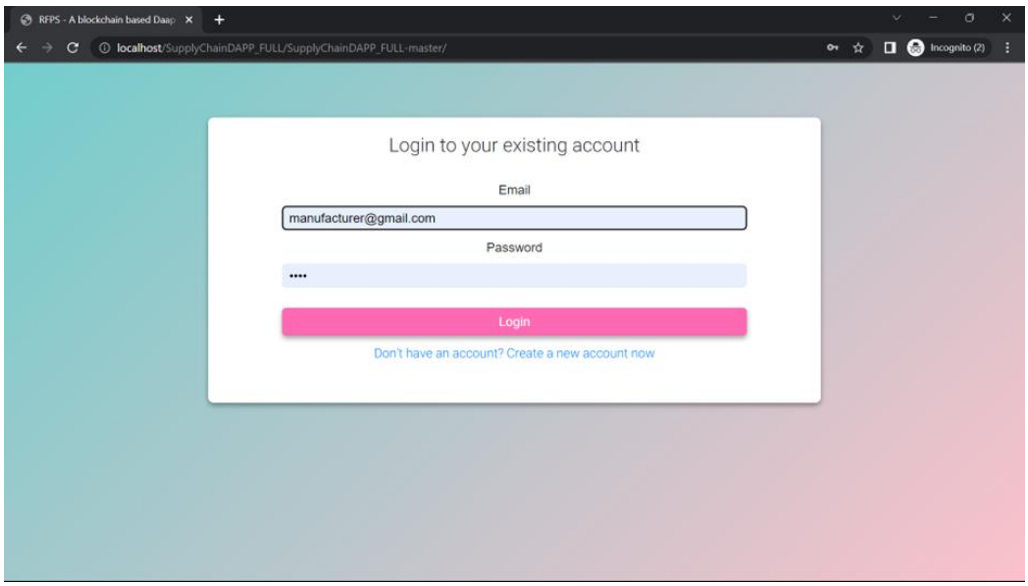

Figure 6.5

They should also specify their role.

- Manufacturer
- Seller
- Customer

However, if they have previously registered, they can use the Login page to access the application. Authentication will only be granted to valid attempts, meaning correct email and corresponding password. If a user enters incorrect login information, they will be locked out of the app and an error message stating "Please check your email and password " will appear. Upon successful login, the user will be directed to the app's homepage depending on the role specified by them.

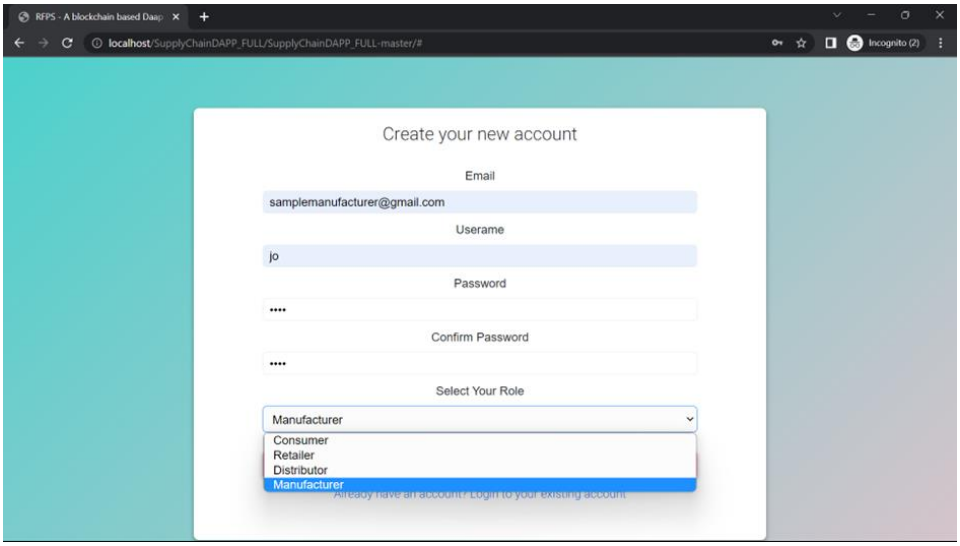

Figure 6.6

6.3 Adding the product details to Generate unique QR code

Manufacturer module's functionality includes Adding the product, Updating the location details of the product, and verifying if the added details are correct or not.(Manufacturer is responsible for entering the correct details because once the block gets added to the blockchain the information cannot be edited).

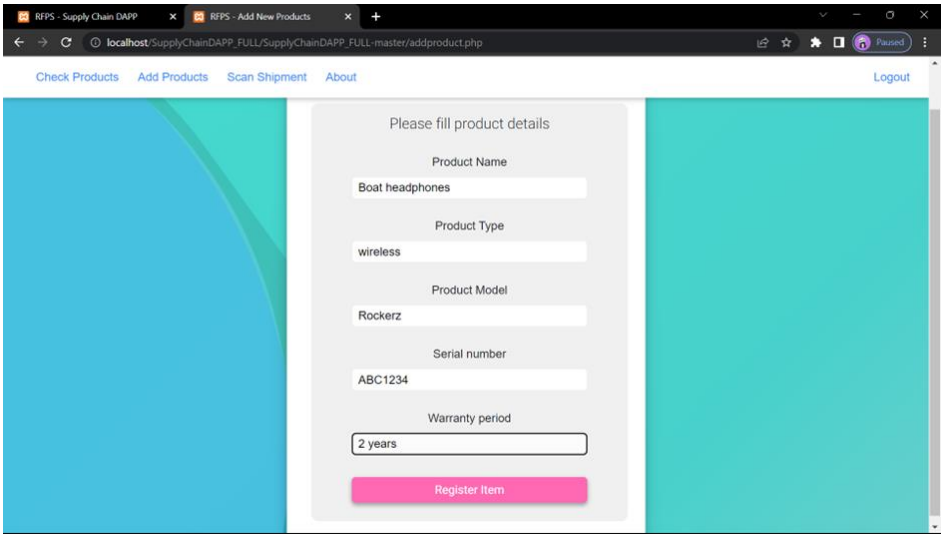

Figure 6.7

Once the Manufacturer adds the product into the blockchain QR code gets generated. This QR code have to be embedded to the laptops which is being manufactured. Also, this QR needs to be embedded as a copy sensitive digital image, which is a concern of the Manufacturer.

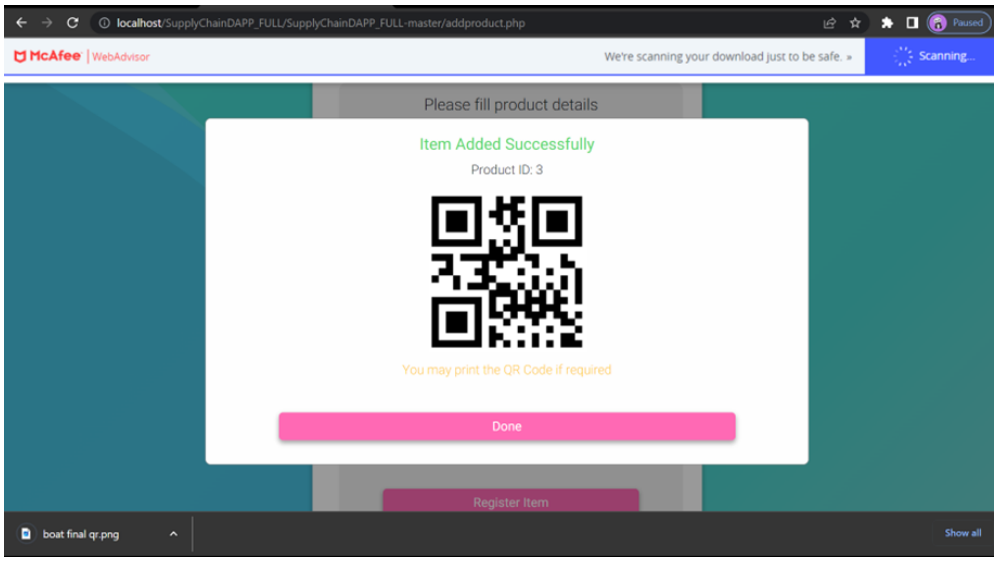

Figure 6.8

6.4 Updating location

Followed by adding the details of a product the Manufacturer can also update the location. This can be done by the Scan shipment functionality.

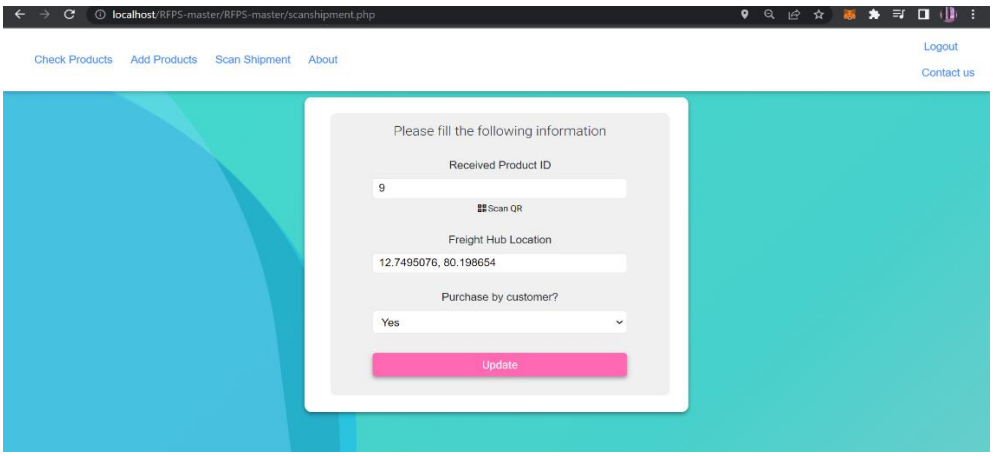

Figure 6.9

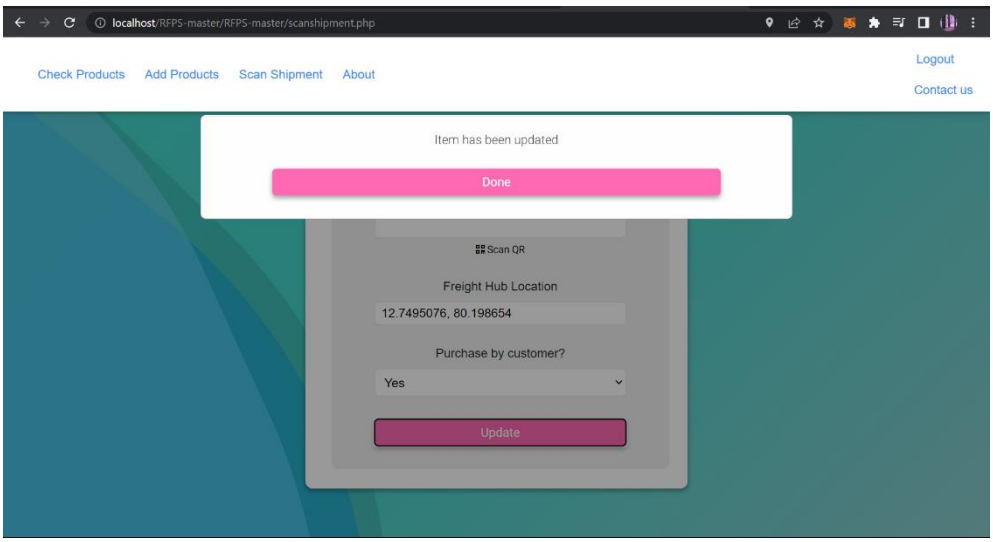

Figure 6.10

Not just manufacturer, even the seller module also has the fun6ctionality of updating the details. So, when the products reach the Seller's market from the Factory, seller scans the QR to update the location detail.

6.5 Retrieving the details

This feature is present is present in all module but its significance is seen in customer's module(figure 6.11). As they're the one who purchase the product and can verify its authenticity. When a customer returns, the person who's in charge of getting the product must scan and check if the Product is fake or not. Even the service centres can check if the product is within the warranty period, just in case if the Customer has lost the bill, they got at the time of purchasing the product.

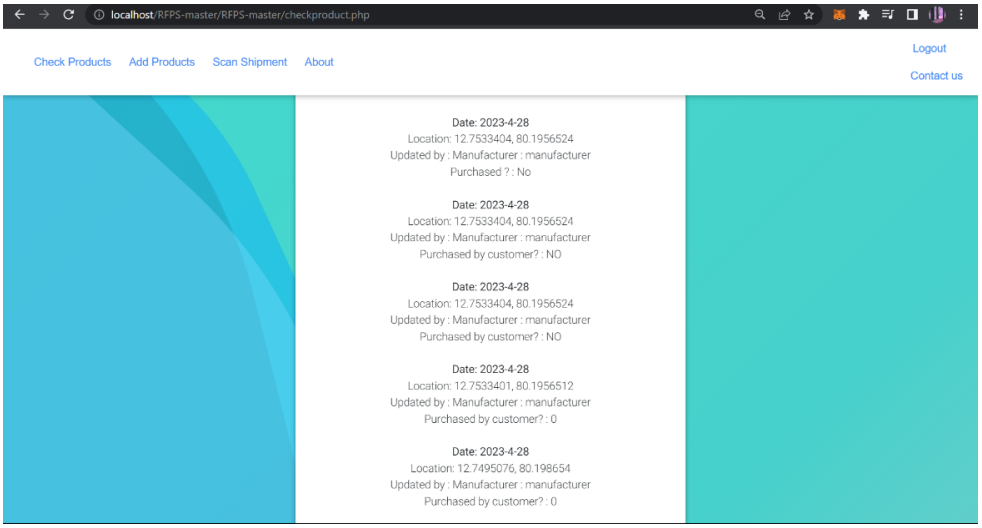

Figure 6.11

For fake products, the following message will be displayed

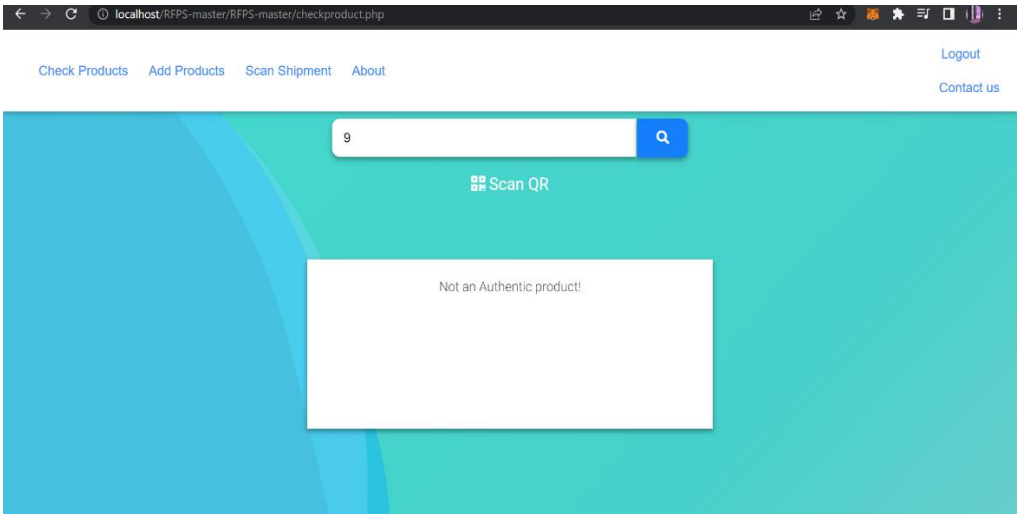

Figure 6.12

We can also scan the QR via integrated camera of the laptop.

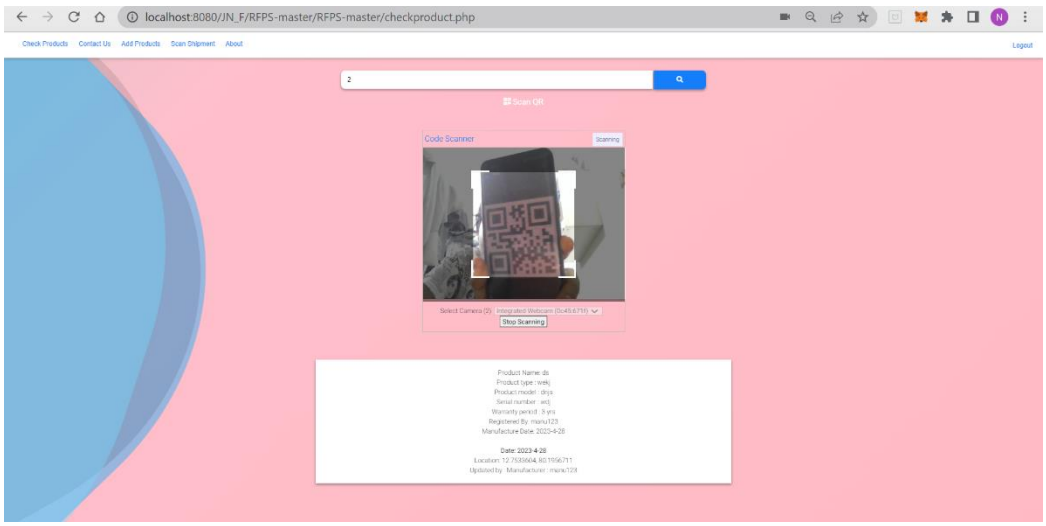

Figure 6.13

CHAPTER 7

CONCLUSION AND FUTURE WORKS

7.1 CONCLUSION

RFPS has demonstrated the potential of blockchain technology to revolutionize fraud detection in the supply chain. The use of QR codes as a means of securely tracking and tracing products, combined with blockchain's tamper-proof and decentralized nature, provides a reliable and efficient solution for detecting fraud and ensuring authenticity in the supply chain.

The project involved the development of a prototype system that utilizes a QR code system to track and trace products throughout the supply chain. Each product is assigned a unique QR code that contains information about its origin, manufacturer, and other relevant details.

The QR code is scanned at each point in the supply chain, and the information is recorded on the blockchain, creating an immutable record of the product's journey from production to distribution.

This system has several benefits over traditional supply chain management systems.

Firstly, it is highly secure and tamper-proof, as the blockchain ensures that no one can alter or delete the recorded information.

Secondly, it is decentralized, meaning that there is no need for a central authority to manage the supply chain, reducing the risk of fraud and corruption.

Furthermore, the use of blockchain-based QR codes also provides a high level of transparency and traceability, allowing consumers to verify the authenticity and origin of the products they purchase. This is particularly relevant in industries such as pharmaceuticals and food, where counterfeit products can have serious health implications for consumers.

However, there are several technical challenges that need to be addressed to ensure the successful adoption of blockchain-based supply chain management systems. One of the main challenges is scalability, as the current blockchain infrastructure may not be able to handle the large volumes of data generated by supply chain transactions. Additionally, there are interoperability challenges, as different blockchain networks may not be able to communicate with each other seamlessly.

In conclusion, the project "Fraud Detection using Blockchain-based QR" represents a significant advancement in the use of blockchain technology for supply chain management. The prototype system developed in this project demonstrates the potential of blockchain-based QR codes to enhance transparency, traceability, and authenticity in the supply chain, providing benefits to businesses, consumers, and regulators alike. However, further research and development are needed to address the technical challenges and ensure the successful adoption of blockchain-based supply chain management systems in the future.

7.2 FUTURE WORKS

1. Integration with decentralized finance (DeFi): DeFi is an emerging field that uses blockchain technology to provide financial services. The integration of blockchain-based fraud detection systems with DeFi could help to prevent fraud and increase security.

2. Collaboration with regulators: Collaboration with regulatory bodies can help to ensure compliance with regulations and legal requirements.

3. Expansion to different industries: Our project can be applied to industries such as healthcare, supply chain management, and real estate.

CHAPTER 8

REFERENCES

[1] C. Shaik, Computer Science & Engineering: An International Journal (CSEIJ) 11 (2021)

[2] G. Khalil, R. Doss, M. Chowdhury, IEEE Access 8, 47952 (2020).

[3] E. Daoud, D. Vu, H. Nguyen, M. Gaedke, Improving Fake Product Detection Using Ai-Based Technology, in 18th International Conference e-Society (2020)

[4] M.A. Habib, M.B. Sardar, S. Jabbar, C.N. Faisal, N. Mahmood, M. Ahmad, Blockchain-based supply chain for the automation of transaction process: Case study-based validation, in 2020 International Conference on Engineering and Emerging Technologies (ICEET) (IEEE, 2020)

[5] Tian, F. An Information System for Food Safety Monitoring in Supply Chains Based on HACCP, Blockchain, and the Internet of Things. Journal of Food Engineering (2018).

[6] Yiannas, F.A new era of food transparency powered by blockchain. Innovations: Technology, Governance, Globalization (2018).

[7] Moin, S. S., & Nguyen, T. T. Blockchain for Supply Chain Traceability: Business Requirements, Consensus, and Cryptography. (2019).

[8] S. Chen, R. Shi, Z. Ren, J. Yan, Y. Shi, J. Zhang, A blockchain-based supply chain quality management framework, in 2017 IEEE 14th International Conference on e-Business Engineering (ICEBE) (IEEE, 2017).

[9] K. Toyoda, P.T. Mathiopoulos, I. Sasase, T. Ohtsuki, IEEE access 5, 17465 (2017).

[10] P. Kumar and R. Tiwari, "A Blockchain-based QR Code System for Fraud Detection in Healthcare Insurance Claims," in Journal of Medical Systems(2020).

[11] M. H. Ahmed and F. Al-Turjman, "Blockchain-based QR Code System for Fraud Detection in Mobile Payments," in Proceedings of the 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Abu Dhabi, UAE, (2018).

[12] J. Gao, Y. Li and X. Li, "Design and Implementation of a Secure Fraud Detection System using Blockchain-based QR Codes," in Proceedings of the International Conference on Computer Science and Education (ICCSE), Colombo, Sri Lanka, (2021).

[13] C. Liu, "Fraud Detection in Banking Transactions using Blockchain-based QR Code System," in Proceedings of the International Conference on Computer Science and Technology (CST), Beijing, China, (2019).

[14] L. Gao, Y. Wu and Y. Chen, "Blockchain-based QR Code System for Fraud Detection in E-Commerce Transactions," in Proceedings of the IEEE International Conference on Smart Computing (SMARTCOMP), New York, USA, (2020).

[15] R. Zhang, J. Zhu and L. Li, "Enhancing Fraud Detection in Supply Chain Management using Blockchain-based QR Code System," in Journal of Industrial Information Integration(2021).

[16] A. Al-Fuqaha, A. Mohammadi, M. Guizani and M. Aledhari, "Blockchain-based QR Code System for Secure and Efficient Transportation," in IEEE Internet of Things Journal, vol. 7, no. 3, pp. 2342-2351, (2020).