A

Major Project Report

on

Product Authentication and Counterfeit Elimination Using Blockchain Technology

Submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology

by

R CHANDRA VIKAS (20EG105438)

Under the guidance of

Dr. B. V. V. Siva Prasad

Associate Professor,

Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ANURAG UNIVERSITY
VENKATAPUR (V), GHATKESAR (M), MEDCHAL (D), T.S - 500088
TELANGANA
(2023-2024)

DECLARATION

I hereby declare that the report entitled **"Product Authentication and Counterfeit Elimination Using Blockchain Technology"** submitted to the **Anurag University** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology (B. Tech)** in **Computer Science and Engineering** is a record of an original work done by me under the guidance of **Dr. B. V. V. Siva Prasad, Associate Professor** and this report has not been submitted to any other university for the award of any other degree or diploma.

Place: Anurag University, Hyderabad

R CHANDRA VIKAS
(20EG105438)

ANURAG UNIVERSITY | SCHOOL OF ENGINEERING

CERTIFICATE

This is to certify that the project report entitled **“Product Authentication and Counterfeit Elimination Using Blockchain Technology”** being submitted by **R Chandra Vikas** bearing the Hall ticket number **20EG105438** in partial fulfillment for the award of the degree of the **Bachelor of Technology** in **Computer Science and Engineering** to the **Anurag University** is a record of Bonafide work carried out by him under my guidance and supervision for the academic year 2023 to 2024.

The results presented in this report have been verified and found to be satisfactory. The results embodied in this report have not been submitted to any other University for the award of any other degree or diploma.

Signature of The Supervisor
Dr. B. V. V. Siva Prasad
Associate Professor

Signature Dean,
Dr. G. Vishnu Murthy
Department of CSE

External Examiner

ACKNOWLEDGMENT

I would like to express my sincere thanks and deep sense of gratitude to project supervisor **Dr. B. V. V. Siva Prasad, Associate Professor, Department of CSE**, **Anurag University** for his constant encouragement and inspiring guidance without which this project could not have been completed. His critical reviews and constructive comments improved my grasp of the subject and steered to the fruitful completion of the work. His patience, guidance and encouragement made this project possible.

I would like to acknowledge my sincere gratitude for the support extended by **Dr. G. Vishnu Murthy**, Dean, Department of CSE, Anurag University. We also express our deep sense of gratitude to **Dr. V V S S S Balaram**, Academic coordinator. **Dr. Pallam Ravi** Project in-Charge Dr.Pallam Ravi Class in-Charge. Project Coordinator and Project review committee members, whose research expertise and commitment to the highest standards continuously motivated me during the crucial stage of my project work.

I would like to express my special thanks to **Dr. V. Vijaya Kumar**, Dean School of Engineering, Anurag University, for his encouragement and timely support in my B.Tech program.

R CHANDRA VIKAS
(20EG105438)

ABSTRACT

Blockchain technologies have gained interest over the last years. While the most explored use case is financial transactions, it has the capability to agitate other markets. Blockchain removes the need for trusted intermediaries, can facilitate faster transactions and add more transparency. This paper explores the possibility to deflate counterfeit using blockchain technology. This paper provides an overview of different solutions in the anti-counterfeit area, different blockchain technologies and what characteristics make blockchain especially interesting for the use case. We have developed three different concepts and the expansion of an existing system concept is pursued further. It is shown that reducing counterfeits cannot be achieved by using technological means only. Increasing awareness, fighting counterfeiters on a legal level, a good alert system, and having tamper-proof packaging are all important aspects. These factors combined with blockchain technology can lead to an efficient and comprehensive approach to reduce counterfeiting.

Keywords – Authentication, Blockchain, Encryption

CONTENTS

List of Figures

List of Tables

1.INTRODUCTION

1.1 Introduction:

Although it may seem like a far-off idea, we are surrounded by a lot of counterfeits. From fashion and retail products to software, digital media, electronics, piracy, and intellectual property, reports put the cost of counterfeiting somewhere around $600bn a year in the US alone. In fact, the International Chamber of Commerce predicts that the negative impacts of counterfeiting and piracy are projected to drain US$4.2 trillion from the global economy and put 5.4 million legitimate jobs at risk by 2022. In Pharmaceuticals, the counterfeit medicine market is now responsible for around 1 million deaths per year, in an industry estimated to be worth $75bn annually. In fact, the counterfeit medicine industry is estimated to be growing at twice the rate of legitimate pharmaceuticals, making it up to 25 times more lucrative than the global narcotics trade. Trust is a central element in all transactions. No matter if sending money or exchanging goods, it becomes difficult if there is no trust between the entities involved. It becomes even more difficult, as with many transactions, third parties are involved, such as banks. Often, not only one third-party is involved in a transaction, but multiple. An international money transfer does not only include the bank of the sender, the bank of the receiver, but also multiple intermediary entities such as clearing houses. The entities involved in the transaction do not only have to trust each other, but also the third parties. Removing these third parties can decrease transaction cost, facilitate faster transactions and add more transparency. Bitcoin has successfully shown that removing such third-parties is possible. The cryptocurrency permits direct sending of coins to a transaction partner, without the need to use banks and clearing houses. The assets are directly transferred from one account to another. There are no intermediaries and thereby no need to trust third parties. In addition, the question if a transaction is valid is not answered by an institution, but by algorithms used. Therefore, it completely removes the need to trust any third party. The technology behind Bitcoin, the blockchain, can however not only be used for financial transactions and crypto currencies in general. The technology has potential to redefine the digital economy because it allows immutable transactions, which can be checked at all times from everyone. This is because the information is publicly available and distributed globally. It is chronologically updated and cryptographically sealed. The full range of applicable use cases for this technology has to be seen, but tracking ownership and history of a product is surely one of them. The possibility to reduce counterfeit using blockchain technology.

Authentication, the act of establishing or conforming something as genuine. Authentication is of utmost importance because the use of counterfeit medicines can be harmful to the health and wellbeing of the patients. Their use may result in treatment failure or even death. Authentication is generally done through the overt or covert features upon the product. We now have more fakes than real drugs in the market. Christophe Zimmermann, the anti-counterfeiting and piracy coordinator of the World Customs Organization. Current anti-counterfeiting supply chains rely on a centralized authority to combat counterfeit products. This architecture results in issues such as single point processing, storage, and failure. Blockchain technology has emerged to provide a promising solution for such issues. In this paper, we propose the block-supply chain, a new decentralized supply chain that detects counterfeiting attacks using blockchain and Near Field Communication (NFC) technologies. Block-supply chain replaces the centralized supply chain design and utilizes a new proposed consensus protocol that is, unlike existing protocols, fully decentralized and balances between efficiency and security. Our simulations show that the proposed protocol offers remarkable performance with a satisfactory level of security compared to the state-of-the-art consensus protocol.

1.2 Purpose of the Project:

Product Authentication and Counterfeit Elimination Using Blockchain Technology is to leverage the unique properties of blockchain to enhance the trustworthiness and security of products in the marketplace. Counterfeit products can pose serious risks to consumers, brand reputation, and economic stability, and blockchain technology offers a robust solution to mitigate these issues, Implement blockchain to create a transparent and immutable ledger of product information, ensuring that consumers can easily verify the authenticity of a product before purchase. This builds trust between consumers and brands. Utilize blockchain to track the entire supply chain process, from manufacturing to distribution. This not only helps in preventing counterfeits but also ensures that consumers are aware of the origins and quality of the products they buy.

Implement unique identifiers such as QR codes linked to blockchain. These identifiers are difficult for counterfeiters to replicate, making it easier to spot counterfeit products. Anti-counterfeiting solutions should protect organizations from financial and reputation losses, and, especially in the case of pharmaceutical products, customer safety and argues that good anti-counterfeiting techniques should generally be simple to apply, but difficult to imitate and have four main features: They should be difficult to duplicate, it should be possible to identify them without special equipment, it should be difficult to re-use them, and it should be visible if they were tampered with. From a product perspective, there are three general technologies to reduce counterfeits.

1.3 Scope of the Project:

Product authentication and counterfeit elimination using blockchain technology have the potential to address a significant challenge in various industries. The scope of this application is broad and extends to sectors where authenticity and traceability of products are critical, such as pharmaceuticals, luxury goods, electronics, food, and more. Features expected to assist the users to confirm the genuineness of a pack. Such features will be significantly visible, and complex or expensive to reproduce.

This includes holograms, color shifting inks, security threads, water marks etc. The advantage of over technologies is that they can be checked by the end consumer. Blockchain can be used to verify the authenticity of electronic components and devices, reducing the risk of counterfeit products infiltrating the market, especially critical in sectors like aerospace and defense.

1.4 Existing System with Disadvantages:

Blockchain technology to authenticate supply chain products as this product may be supplied from multiple third-party distributors and this distributor can make clone/fake/counterfeits of this product BAR CODE and then manufacture fake products and add this counterfeit label to fake product and this fake product can cause huge loss of financial and lives if fake medicine manufacture.

Many businesses rely on third-party vendors. The outsourced supplier has access to all of the original assets, there is a risk that will not only make legitimate products, but also counterfeits. Not only supply chain any other online transaction require third party to complete transaction and peoples has to trust on third parties to complete their transaction and sometime this third parties can make fraud transaction or misuse user data. There are many chances of cloning the product. Even now there are more fakes than real drugs in the market. Features expected to assist the users to confirm the genuineness of a pack. Such features will be significantly visible, and complex or expensive to reproduce.

Disadvantages:

One of the challenging issues is privacy and data protection. In case of any other online transaction to complete the transactions, they must trust these third parties. However, these third parties can sometimes misuse the customer data or commit fraud.

- Outsourced suppliers have access to all of the original assets, there is a risk that they can clone the products.
- There will be huge financial and human losses if fake medicine is manufactured in the market.
- Ensuring widespread adoption of the technology. For the system to be effective, all parties involved in the supply chain, including manufacturers, retailers, and consumers, must use the technology. Achieving this level of adoption can be difficult, especially in industries with many different stakeholders and competing interests.

1.5 Proposed System with features:

Blockchain technology which does not require any third party and verification will be done by software algorithm itself without involvement of any third party. In this to avoid forge counterfeit we are converting all products details/barcode into digital signatures and this digital signature will be stored in Blockchain server as this Blockchain server support tamper proof data storage and nobody can hack or alter its data and if by a chance if its data alter then verification get failed at next block storage and user may get intimation about data alter.

In Blockchain technology the same transaction data stored at multiple servers with hash code verification and if data alter at one server, then it will be detected from another server as for the same data hash code will get different. For example, in Blockchain technology data will be stored at multiple servers and if malicious users alter data at one server, then its hash code will get changed in one server and other servers left unchanged and this changed hash code will be detected at verification time and future malicious user changes can be prevented. In the supply chain also all products barcode digital Blockchain signatures will be stored and if any third -party distributor makes a clone of barcode then its signature will be mismatched and counterfeit will be detected. In Blockchain each data will be stored by verifying old hash codes and if old hash codes remain unchanged then data will be considered, as original and unchanged and then new transaction data will be appended to Blockchain as a new block. For each new data storage all block hash codes will be verified.

Advantages:

- In the supply chain also all products barcode digital Blockchain signatures will be stored and if any third-party distributor makes a clone of barcode, then its signature will be mismatched and counterfeit will be detected.
- In blockchain Server once the product is stored cannot be modified.
- Data protection and privacy are maintained. If there is any data alter the user will get intimated and verification fails at the next block.

1.6 Input and Output Design:

Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use while retaining privacy. Input Design considered the following things:

- What data should be given as input?
- How should the data be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when errors occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer- based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volumes of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize instant. Thus, the objective of input design is to create an input layout that is easy to follow

Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other systems through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source of information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can be used easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.

2.LITERATURE SURVEY

Research on product anticounterfeiting traceability systems based on blockchain. International Journal of Advanced Manufacturing Technology

Authors: Zhang, Y., Wang, Y., & Zhang, L.

The author analyzed the performance of their system with a traditional traceability system and found that the blockchain-based system was more effective in terms of anti-counterfeiting and traceability. Traceability has emerged as a prime requirement for a multi-tier and multi-site production. It enables visibility and caters to the consumer requirements of transparency and quality assurance. The proposed system can build a technology-based trust among the supply chain partners, where the distributed ledger can be used to store and authenticate supply chain transactions.

A blockchain-based application system for product

anti-counterfeiting. Authors: J. Ma, S.-Y. Lin, X. Chen, H.-M. Sun, Y.-C.

Chen, and H. Wang

The paper proposed a decentralized Blockchain technology approach to ensure that consumers do not fully rely on the merchants to determine if products are genuine. manufacturers can use this system to provide genuine products without having to manage direct-operated stores, reduces the cost of product quality assurance. Blockchain has received increasing attention and numerous applications have emerged from this technology. A renowned Blockchain application is the cryptocurrency Bitcoin, that has not only been effectively solving the double-spending problem but also it can confirm the legitimacy of transactional records without relying on a centralized system to do so. Therefore, any application using Blockchain technology as the base architecture ensures that the contents of its data are tamper-proof.

Blockchain Beyond Bitcoin, in Communications of the ACM.

Authors: S. Underwood

Blockchain Technology has attracted attention as the basis of cryptocurrencies such as Bitcoin, but its capabilities extend far beyond that, enabling existing technology applications to be vastly improved and new applications never previously practical to be deployed. Also known as distributed ledger

Blockchain technology, blockchain is expected to revolutionize industry and commerce and drive economic change on a global scale because it is immutable, transparent, and redefines trust, enabling secure, fast, trustworthy, and transparent solutions that can be public or private. It could empower people in developing countries with recognized identity, asset ownership, and financial inclusion.

ETHEREUM: A secure decentralized generalized transaction ledger.

Authors: DR. Gavin Wood

Ethereum is a project which attempts to build the generalized technology; technology on which all transaction-based state machine concepts may be built. Moreover, it aims to provide to the end-developer a tightly integrated end-to-end system for building software on a hitherto unexplored compute paradigm in the mainstream: a trustful object messaging compute framework.

Understanding and fighting the medicine counterfeit market‖, in Journal of Pharmaceutical and Biomedical Analysis.

Authors: K. D´egardin, Y. Roggo and P. Margot.

Medicine counterfeiting is a serious worldwide issue, involving networks of manufacture and distribution that are an integral part of industrialized organized crime. Despite the potentially devastating health repercussions involved, legal sanctions are often inappropriate or simply not applied. The difficulty in agreeing on a definition of counterfeiting, the huge profits made by the counterfeiters and the complexity of the market are the other main reasons for the extent of the phenomenon.

Technology designed to combat fakes in the global supply chain‖, in Business Horizons.

Authors: L. Li

This article discusses the growing issue of counterfeit products in the global market due to increased globalization and online shopping. It presents various technologies used in the supply chain to combat counterfeiting, focusing on both product authentication and product tracing and tracking. The article also examines the pros and cons of these technological solutions and highlights success stories in the fight against counterfeits. Additionally, it addresses challenges such as rising anti- counterfeiting costs, collaborative efforts to combat fakes, and the exploration of a comprehensive strategy to tackle the issue.

3. SOFTWARE REQUIREMENT ANALYSIS

3.1 Problem Specification:

Anti-counterfeiting solutions should protect organizations from financial and reputation losses, and, especially in the case of pharmaceutical products, customer safety. argues that good anti-counterfeiting techniques should generally be simple to apply, but difficult to imitate and have four main features: They should be difficult to duplicate, it should be possible to identify them without special equipment, it should be difficult to re-use them, and it should be visible if they were tampered with. From a product perspective, there are three general technologies to reduce counterfeits.

Features expected to assist the users to confirm the genuineness of a pack. Such features will be significantly visible, and complex or expensive to reproduce. This includes holograms, color shifting inks, security threads, water marks etc. The advantage of overt technologies is that they can be checked by the end consumer.

3.2 Modules and their functionalities:

hashlib

This module implements a common interface to many different secure hash and message digest algorithms. Included are the FIPS secure hash algorithms SHA1, SHA224, SHA256, SHA384, SHA512, (defined in the FIPS 180-4 standard), the SHA-3 series (defined in the FIPS 202 standard) as well as RSA's MD5 algorithm (defined in internet **RFC 1321**). The terms "secure hash" and "message digest" are interchangeable. Older algorithms were called message digests. The modern term is secure hash.

Datetime

The datetime module supplies classes for manipulating dates and times. While date and time arithmetic is supported, the focus of the implementation is on efficient attribute extraction for output formatting and manipulation.

JSON

JSON(Javascript Object Notation)specified by **RFC 7159** (which obsoletes **RFC 4627**) and by ECMA-404, is a lightweight data interchange format inspired by JavaScript object literal syntax. JSON exposes an API familiar to users of the standard library marshal and pickle modules. The JSON module always produces str objects, not bytes objects. Therefore, it must support str input.

Pickle

The pickle module implements binary protocols for serializing and de-serializing a Python object structure. *"Pickling"* is the process whereby a Python object hierarchy is converted into a byte stream, and *"unpickling"* is the inverse operation, whereby a byte stream (from a binary file or bytes-like object) is converted back into an object hierarchy. Pickling (and unpickling) is alternatively known as "serialization", "marshaling," 1 or "flattening"; however, to avoid confusion, the terms used here are "pickling" and "unpickling".

Save Product with Blockchain Entry:

In this module the user will enter product details and then upload product bar code images and then digital signatures will be generated on uploaded barcodes and then these transaction details will be stored in Blockchain. Before storing transaction Blockchain will verify all old transaction and upon successful verification new transaction block will be store

Retrieve Product Data:

Using this module, users can search existing product details by entering product id.

Authenticate Scan:

Here in this module, we don't have any scanner so we are uploading original or fake bar code images and then Blockchain will verify the digital signature of uploaded bar code with already stored barcodes and if a match is found then Blockchain will extract all details and display to user else authentication will be failed.

3.3 Functional Requirements:

In software engineering and systems engineering, a **functional requirement** defines a function of a system or its component, where a function is described as a specification of behavior between outputs and inputs. Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describe all the cases where the system uses the functional requirements, these are captured in use cases. Functional requirements are supported by non- functional requirements (also known as "quality requirements"), which impose constraints on the design or implementation (such as performance requirements, security, or reliability). Generally, functional requirements are expressed in the form "system must do <requirement>," while non- functional requirements take the form "system shall be <requirement>." The plan for implementing functional requirements is detailed in the system design, whereas *non-functional* requirements are detailed in the system architecture.

As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements, which specify overall characteristics such as cost and reliability. Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system.

In some cases a requirements analyst generates use cases after gathering and validating a set of functional requirements. The hierarchy of functional requirements collection and change, broadly speaking, is: user/stakeholder request → analyze → use case → incorporate. Stakeholders make a request; systems engineers attempt to discuss, observe, and understand the aspects of the requirement; use cases, entity relationship diagrams, and other models are built to validate the requirement; and, if documented and approved, the requirement is implemented/incorporated. Each use case illustrates behavioral scenarios through one or more functional requirements. Often, though, an analyst will begin by eliciting a set of use cases, from which the analyst can derive the functional requirements that must be implemented to allow a user to perform each use case.

3.4 Non-Functional Requirements:

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, *"how fast does the website load?"* Failing to meet non-functional requirements can result in systems that fail to satisfy user needs.

Non-functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users is > 10000. Description of non-functional requirements is just as critical as a functional requirement.

3.5 Feasibility Study

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are**,**

- ♦ **ECONOMICAL FEASIBILITY**
- ♦ **TECHNICAL FEASIBILITY**
- ♦ **SOCIAL**

FEASIBILITY ECONOMICAL
FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources.

This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal

or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it.

4. SOFTWARE & HARDWARE REQUIREMENTS

4.1 Software Requirements:

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation. The appropriation of requirements and implementation constraints gives the general overview of the project in regard to what the areas of strength and deficit are and how to tackle them.

Operating Systems Supported: Windows based Operating System

Technologies and Languages Used: Python, GUI and Blockchain

4.2 Hardware Requirements:

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation. The appropriation of requirements and implementation constraints gives the general overview of the project in regard to what the areas of strength and deficit are and how to tackle them.

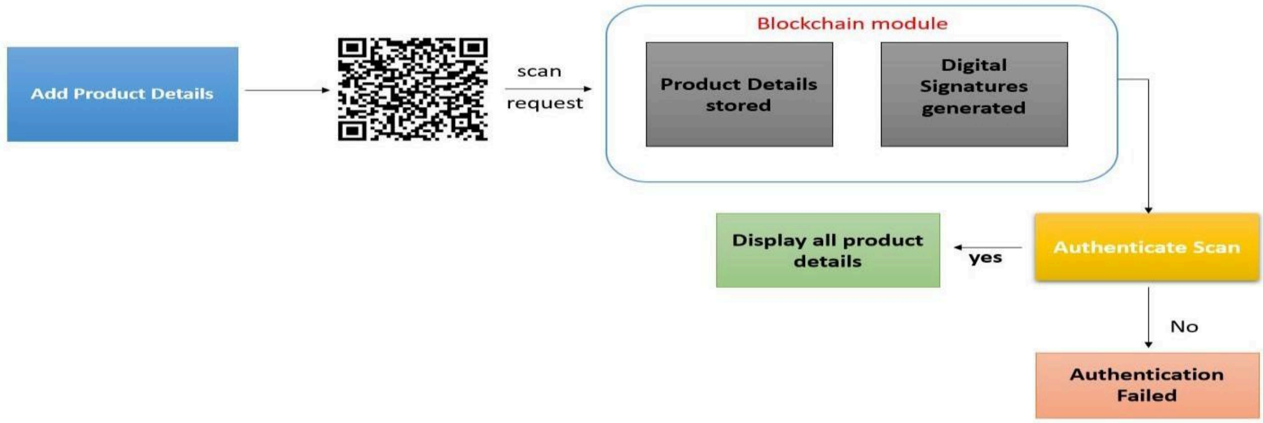
Processor: Pentium IV or

higher RAM: 8GB

Hard Disk: 512GB

5.DESIGN

5.1 System Architecture:



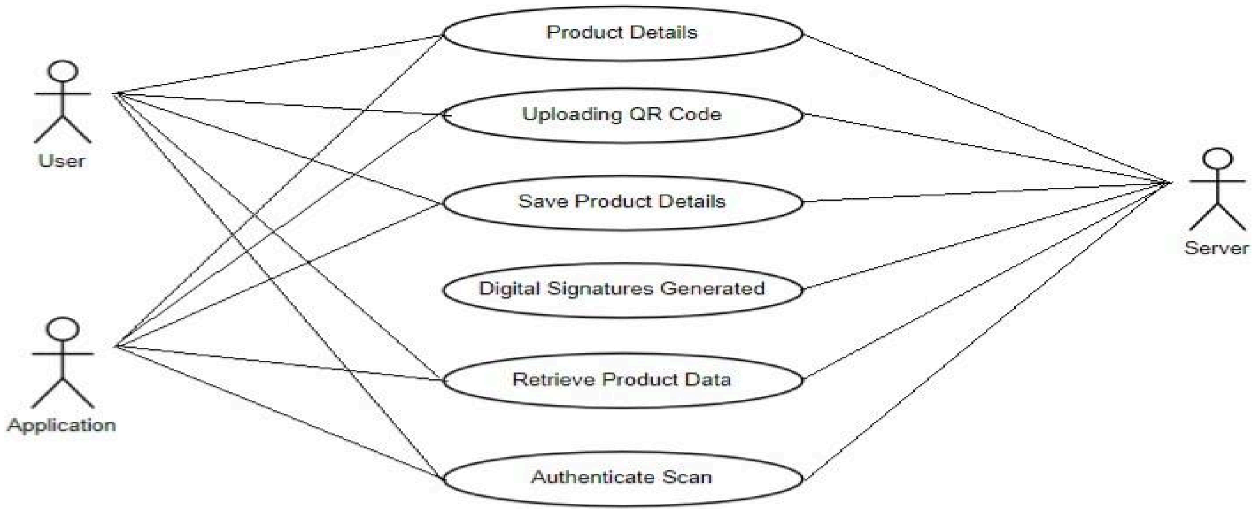
5.2 UML Diagrams:

The Unified Modelling Language (UML) is a standard language for writing software blueprints. The UML is a language for:

- Specifying: It is just like a blueprint created by an architect prior to the construction.
- Visualizing: Visualizing is concerned with deep analysis of systems to be constructed.
- Constructing: Modeling also provides us with mechanisms which are essential while constructing a system.
- Documenting: Finally, modeling justifies its importance by applying all its credentials to be bound in a piece of paper referred to as a document. The UML is a language which provides vocabulary and the rules for combining words in that vocabulary for the purpose of communication.

Use Case Diagram:

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So, when a system is analyzed, together its functionalities use cases are prepared and actors are identified. In this case, the actors could be manufacturers, consumers, and the blockchain network itself. The primary use cases might include product registration, product verification, and counterfeit detection.



Use cases: Horizontally shaped ovals that represent the different uses that a user might have.

Actors: Stick figures that represent the people actually employing the use cases.

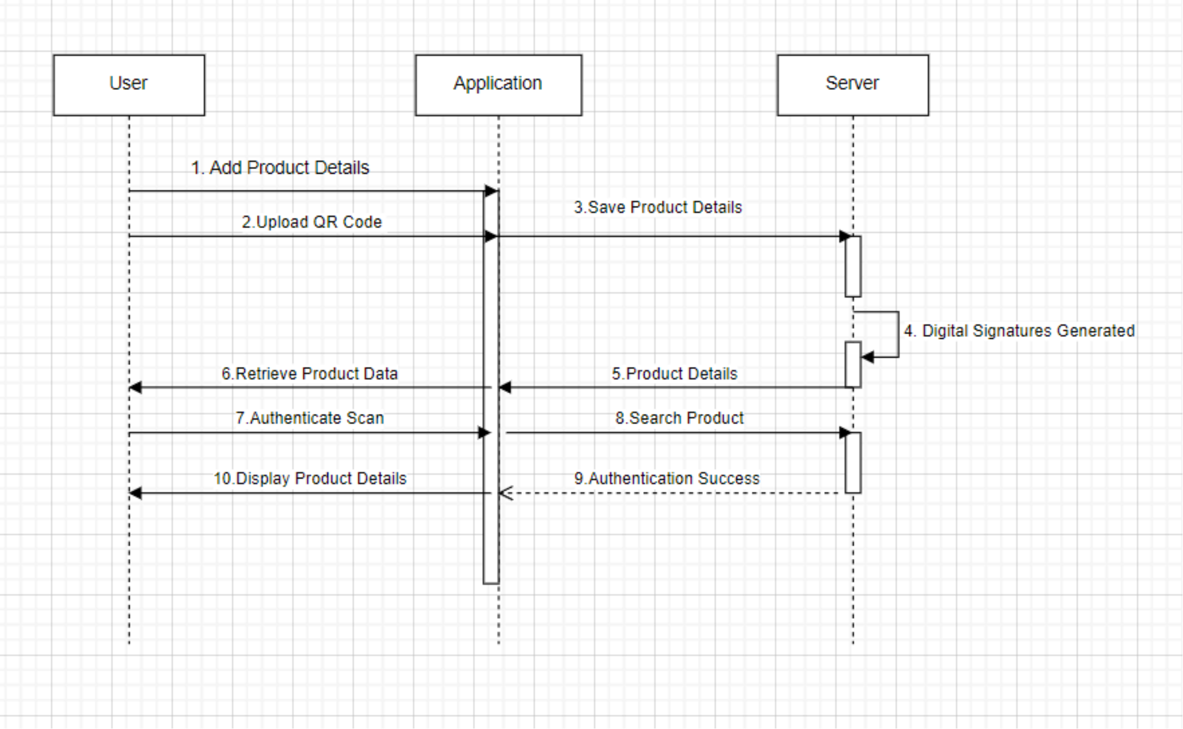
Associations: A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.

System boundary boxes: A box that sets a system scope to use cases. All use cases outside the box would be considered outside the scope of that system. For example, Psycho Killer is outside the scope of occupations in the chainsaw example found below.

Packages: A UML shape that allows you to put different elements into groups. Just as with component diagrams, these groupings are represented as file folders.

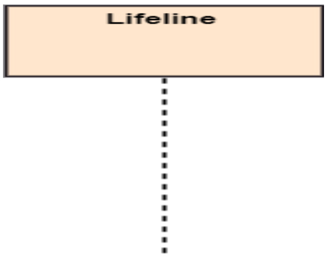
Sequence Diagram:

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.



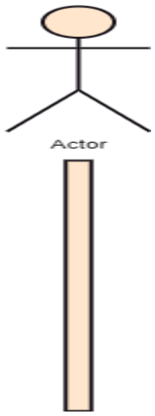
Lifeline

An individual participant in the sequence diagram is represented by a lifeline. It is positioned at the top of the diagram.



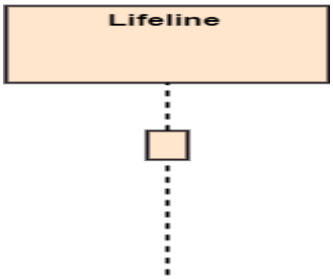
Actor

A role played by an entity that interacts with the subject is called an actor. It is out of the scope of the system. It represents the role, which involves human users and external hardware or subjects. An actor may or may not represent a physical entity, but it purely depicts the role of an entity. Several distinct roles can be played by an actor or vice versa.



Activation

It is represented by a thin rectangle on the lifeline. It describes the time period in which an operation is performed by an element, such that the top and the bottom of the rectangle is associated with the initiation and the completion time, respectively.

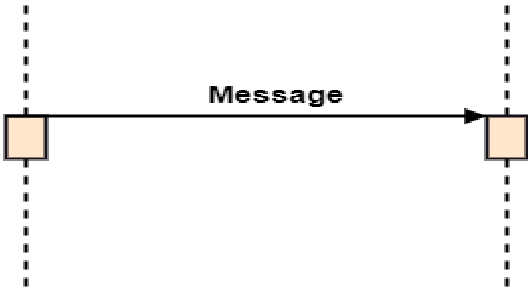


Messages

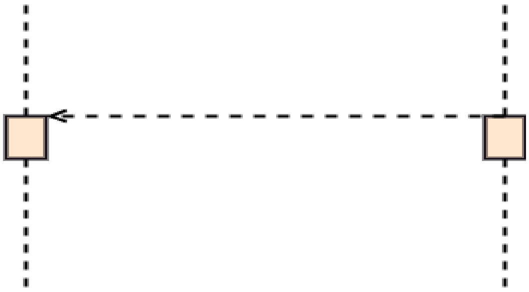
The messages depict the interaction between the objects and are represented by arrows. They are in the sequential order on the lifeline. The core of the sequence diagram is formed by messages and lifelines.

Following are types of messages enlisted below:

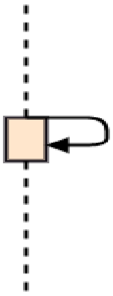
Call Message: It defines a particular communication between the lifelines of an interaction.



Return Message: It defines a particular communication between the lifelines of interaction that represent the flow of information from the receiver of the corresponding caller message.



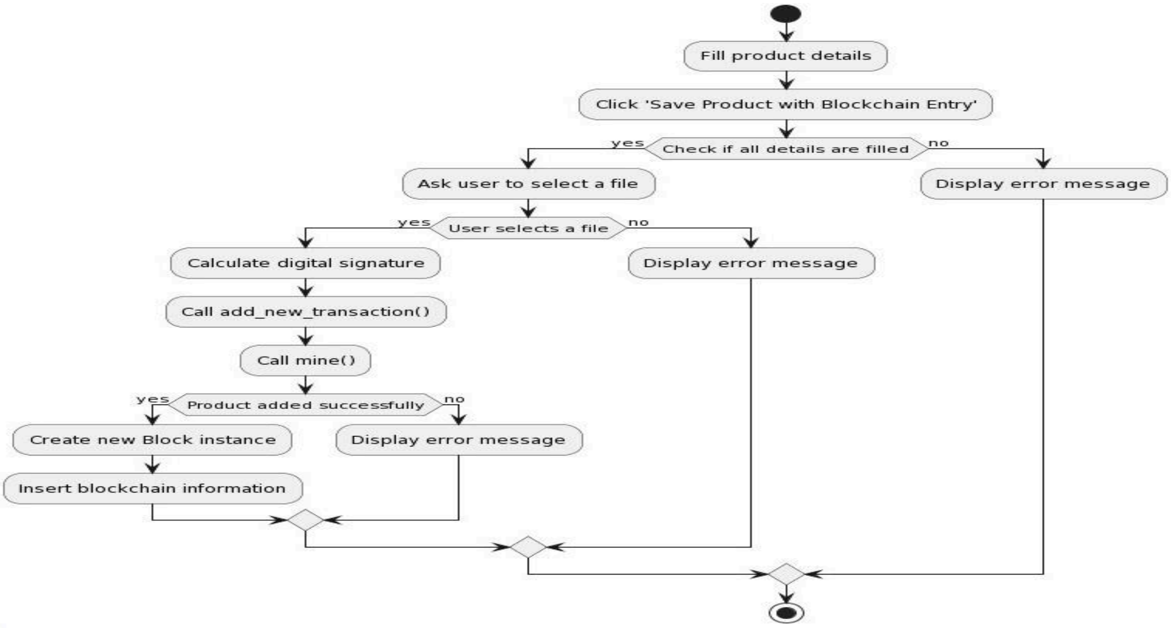
Self-Message: It describes a communication, particularly between the lifelines of an interaction that represents a same lifeline when it is invoked.



Activity Diagram:

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc


Fill product details
Click 'Save Product with Blockchain Entry'
Check if all details are filled
yes
no
Ask user to select a file
Display error message
User selects a file
yes
no
Calculate digital signature
Display error message
Call add_new_transaction()
Call mine()
Product added successfully
yes
no
Create new Block instance
Display error message
Insert blockchain information

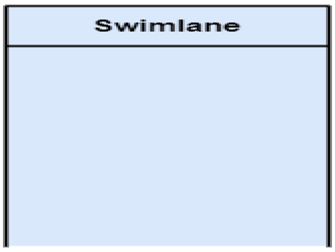
Activities

The categorization of behavior into one or more actions is termed as an activity. In other words, it can be said that an activity is a network of nodes that are connected by edges. The edges depict the flow of execution. It may contain action nodes, control nodes, or object nodes.

The control flow of activity is represented by control nodes and object nodes that illustrate the objects used within an activity. The activities are initiated at the initial node and are terminated at the final node.

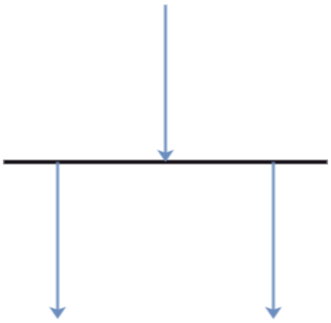
Activity partition

The swimlane is used to cluster all the related activities in one column or one row. It can be either vertical or horizontal. It used to add modularity to the activity diagram. It is not necessary to incorporate swimlanes in the activity diagram. But it is used to add more transparency to the activity diagram.



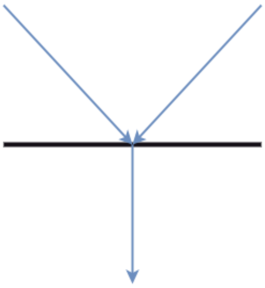
Forks

Forks and join nodes generate the concurrent flow inside the activity. A fork node consists of one inward edge and several outward edges. It is the same as that of various decision parameters. Whenever data is received at an inward edge, it gets copied and split across various outward edges. It split a single inward flow into multiple parallel flows.



Join Nodes

Join nodes are the opposite of fork nodes. A Logical AND operation is performed on all of the inward edges as it synchronizes the flow of input across one single output (outward) edge.



Pins

It is a small rectangle, which is attached to the action rectangle. It clears out all the messy and complicated things to manage the execution flow of activities. It is an object node that precisely represents one input to or output from the action.

6. CODING AND IMPLEMENTATION

6.1 Source Code

Main.py

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
from tkinter.filedialog import askopenfilename



from Block import *
from Blockchain import *
from hashlib import sha256
import os
import datetime


import
webbrowser main
= Tk()
main.title("Authentication of Product & Counterfeits Elimination Using Blockchain")
main.geometry("1300x1200")
global filename
blockchain =
Blockchain()
if os.path.exists('blockchain_contract.txt'):
    with open('blockchain_contract.txt', 'rb') as fileinput:
        blockchain = pickle.load(fileinput)
    fileinput.close()


def addProduct():
    global filename
    text.delete('1.0', END)
    filename = askopenfilename(initialdir = "original_barcodes")
    with open(filename ,"rb") as f:
        bytes = f.read()
    f.close()
    pid = tf1.get()
    name = tf2.get()
    user = tf3.get()
    address = tf4.get()


 if len(pid) > 0 and len(name) > 0 and len(user) > 0 and len(address) > 0:
        current_time = datetime.datetime.now()
        digital_signature = sha256(bytes).hexdigest();
       data = pid+"#"+name+"#"+user+"#"+address+"#"+str(current_time)+"#"+digital_signature
        blockchain.add_new_transaction(data)
        hash = blockchain.mine()
        b = blockchain.chain[len(blockchain.chain)-1]
        text.insert(END,"Blockchain Previous Hash: "+str(b.previous_hash)+"\nBlock No :
"+str(b.index)+"\nCurrent Hash : "+str(b.hash)+"\n")
        text.insert(END,"Barcode Blockchain Digital Signature: "+str(digital_signature)+"\n\n")
        blockchain.save_object(blockchain,'blockchain_contract.txt')
        tf1.delete(0, 'end')
        tf2.delete(0, 'end')
```

```
      tf3.delete(0, 'end')
       tf4.delete(0, 'end')
   else:
       text.insert(END,"Please enter all details")


def authenticateProduct():
   text.delete('1.0', END)


  filename = askopenfilename(initialdir = "original_barcodes")
   with open(filename, “rb”) as f:
       bytes = f.read()
   f.close()
   digital_signature = sha256(bytes).hexdigest();
   flag = True
   for i in range(len(blockchain.chain)):
       if i > 0:
           b = blockchain.chain[i]


          data = b.transactions[0]
           arr = data.split("#")
           if arr[5] == digital_signature:
               output = ''
               text.insert(END,"Uploaded Product Barcode Authentication
               Successfully\n") text.insert(END,"Details extracted from Blockchain after
               Validation\n\n") text.insert(END,"Product ID : "+arr[0]+"\n")
               text.insert(END,"Product Name                : "+arr[1]+"\n")
               text.insert(END,"Company/User Details         : "+arr[2]+"\n")
               text.insert(END,"Address Details             : "+arr[3]+"\n")
```

```
            text.insert(END,"Scan Date & Time            : "+arr[4]+"\n")



 text.insert(END,"Product Barcode Digital Sign: "+arr[5]+"\n")
            output='<html><body><table border=1>'
            output+='<tr><th>Block No</th><th>Product ID</th><th>Product
Name</th><th>Company/User Details</th><th>Address Details</th><th>Scan Date & Time</th>'
            output+='<th>Product Barcode Digital Signature</th></tr>'
output+='<tr><td>'+str(i)+'</td><td>'+arr[0]+'</td><td>'+arr[1]+'</td><td>'+arr[2]+'</td><td>'+arr[
3]+'</td><td>'+arr[4]+'</td><td>'+arr[5]+'</td></tr>'
            f = open("output.html", "w")
            f.write(output)
            f.close()
            webbrowser.open("output.html", new=1)
            flag = False
            break
  if flag:
      text.insert(END,"Uploaded Product Barcode Authentication Failed")

def searchProduct():
  text.delete('1.0', END)
  pid = tf1.get()
  flag = True
  if len(pid) > 0:
      for                    i                    in
        range(len(blockchain.chain)):  if  i
        > 0:
            b = blockchain.chain[i]
            data                    =
            b.transactions[0]  arr  =
            data.split("#")
            if arr[0] == pid:
                output = ''



text.insert(END,"Product Details extracted from Blockchain using Product ID :
                "+pid+"\n\n") text.insert(END,"Product ID     : "+arr[0]+"\n")
                text.insert(END,"Product Name                 : "+arr[1]+"\n")
                text.insert(END,"Company/User Details         : "+arr[2]+"\n")

               text.insert(END,"Address Details              : "+arr[3]+"\n")
                text.insert(END,"Scan Date & Time             : "+arr[4]+"\n")
                text.insert(END,"Product Barcode Digital Sign : "+arr[5]+"\n")
                output='<html><body><table border=1>'
                output+='<tr><th>Block No</th><th>Product ID</th><th>Product
Name</th><th>Company/User Details</th><th>Address Details</th><th>Scan Date & Time</th>'
                output+='<th>Product Barcode Digital Signature</th></tr>'

output+='<tr><td>'+str(i)+'</td><td>'+arr[0]+'</td><td>'+arr[1]+'</td><td>'+arr[2]+'</td><td>'+arr[
3]+'</td><td>'+arr[4]+'</td><td>'+arr[5]+'</td></tr>'


                f = open("output.html", "w")
                f.write(output)


               f.close()
               webbrowser.open("output.html", new=1)
               flag = False
                break
    if flag:
        text.insert(END,"Given product id does not exists")
 font = ('times', 15, 'bold')
```

```
title = Label(main, text='Authentication of Product & Counterfeits Elimination Using Blockchain')
title.config(bg='bisque', fg='purple1')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0, y=5)

font1 = ('times', 13, 'bold')



l1 = Label(main, text='Product ID :')
l1.config(font=font1)
l1.place(x=50, y=100)

tf1 = Entry(main,width=20)
tf1.config(font=font1)
tf1.place(x=240, y=100)

l2 = Label(main, text='Product Name:')
l2.config(font=font1)
l2. place(x=50, y=150)
tf2 = Entry(main, width=20)
tf2.config(font=font1)
tf2.place(x=240, y=150)

l3 = Label(main, text='Company/User Details:')
l3.config(font=font1)
l3.place(x=50, y=200)
tf3 = Entry(main, width=60)
```

```
tf3.config(font=font1)

tf3.place(x=240,

y=200)


l4 = Label(main, text='Address Details:')

l4.config(font=font1)

l4.place(x=50, y=250)


tf4 = Entry(main, width=80)

tf4.config(font=font1)


tf4.place(x=240, y=250)


saveButton = Button(main, text="Save Product with Blockchain Entry", command=addProduct)

saveButton.place(x=50, y=300)

saveButton.config(font=font1)



searchButton = Button(main, text="Retrieve Product Data", command=searchProduct)

searchButton.place(x=370, y=300)

searchButton.config(font=font1)


scanButton = Button(main, text="Authenticate Scan", command=authenticateProduct)

scanButton.place(x=590, y=300)

scanButton.config(font=font1)


font1 = ('times', 13, 'bold')

text=Text(main, height=15, width=120)

scroll=Scrollbar(text)
```

```
text.configure(yscrollcommand=scroll.set)
text. place(x=10, y=350)
text.config(font=font1)


main.config(bg='cornflower
blue') main.mainloop()
```

6.2 Implementation:

6.2.1 Python:

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and the indentation requirement of the language makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc. The biggest strength of Python is huge collection of standard libraries which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow
- Test frameworks
- Multimedia

Advantages of Python:

Let’s see how Python dominates over other languages.

1. Extensive Libraries

Python downloads with an extensive library and it contains code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don't have to write the complete code for that manually.

2. Extensible

As we have seen earlier, Python can be **extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities** to our code in the other language.

4. Improved Productivity

The language's simplicity and extensive libraries render programmers **more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

6. Simple and Easy

When working with Java, you may have to create a class to print **'Hello World'**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and **code.** This is why

When people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

8. Object-Oriented

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

9. Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you **download Python** for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to **code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Advantages of Python Over Other Languages

1. Less Coding

Almost all of the tasks done in Python require less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners. Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 GitHub annual survey showed us that Python has overtaken Java in the most popular programming language category.

Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build webapps, perform data analysis and **machine learning**, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

Disadvantages of Python

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

1. Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the client-side. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle.The reason it is not so famous despite the existence of Brython is that it isn't that secure.

3. Design Restrictions

As you know, Python is dynamically-typed. This means that you don't need to declare the type of variable while writing the code. It uses duck-typing. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors.

7.SYSTEM TESTING

7.1 Testing Strategies:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent.

Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified class of application outputs must be exercised.
Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identifying Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional

testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration- oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It has a purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as

specification or requirements document. It is a test in which the software under test is treated as a black box. You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.2 Sample Test Cases:

Step	Test Case	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1.	Add Product	Product ID:1	Product	Product	Pass

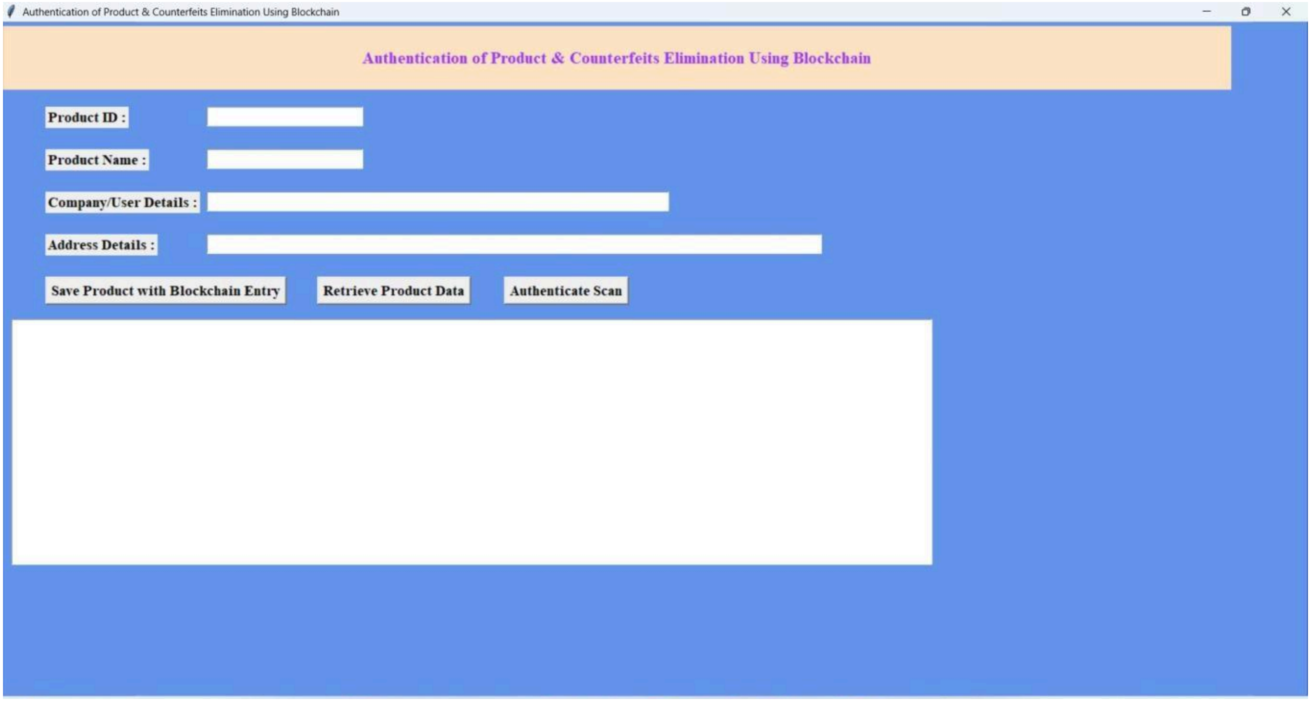
	Details	Product Name: Mobile Company Name: iPhone Location: Hyderabad Upload QR Code	Details stored on the server.	Details Stored in Server.	

Step	Test Case	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
2.	Retrieve Product Data	Product ID:1	Details of the product displayed.	Details of the product displayed**.**	Pass

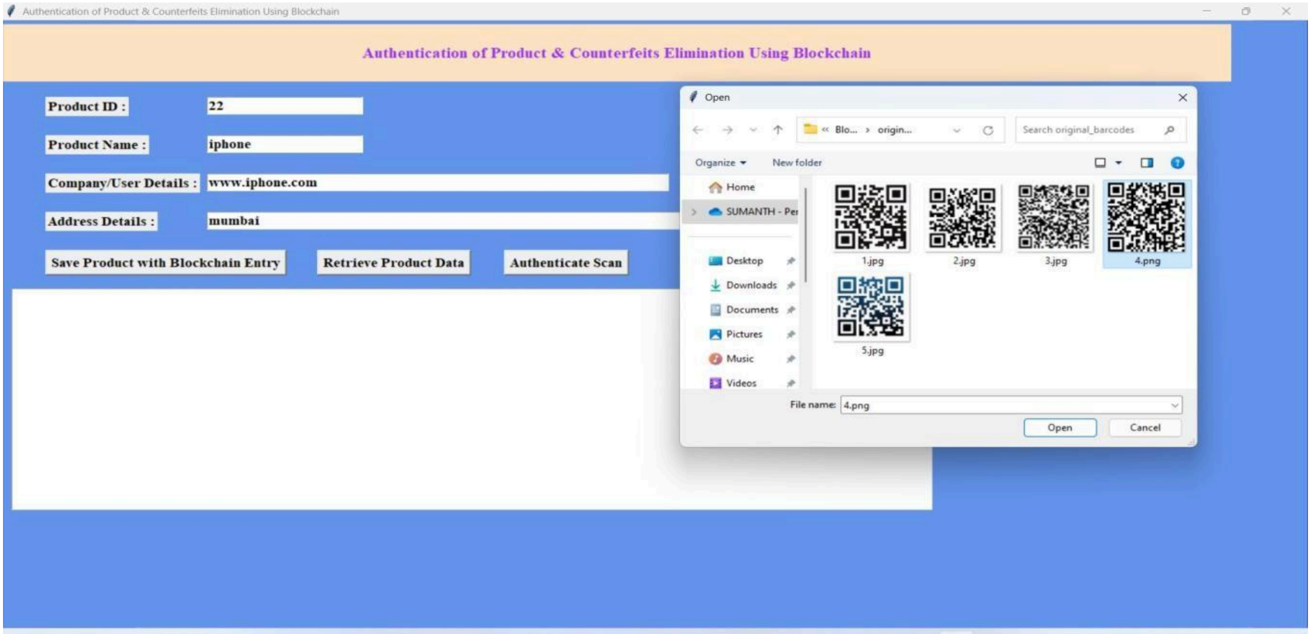
Step	Test Case	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
3.	Authenticate Scan	Fake QR Code	Authentication Failed	Authentication Failed	Pass

8.OUTPUT SCREENS

To run project double click on ‘run.bat’ file to get below screen


Authentication of Product & Counterfeits Elimination Using Blockchain
Authentication of Product & Counterfeits Elimination Using Blockchain
Product ID :
Product Name :
Company/User Details :
Address Details :
Save Product with Blockchain Entry
Retrieve Product Data
Authenticate Scan

In above screen enter product details and then click on ‘Save Products with Blockchain Entry’ button to store product details in Blockchain



Product ID : 22

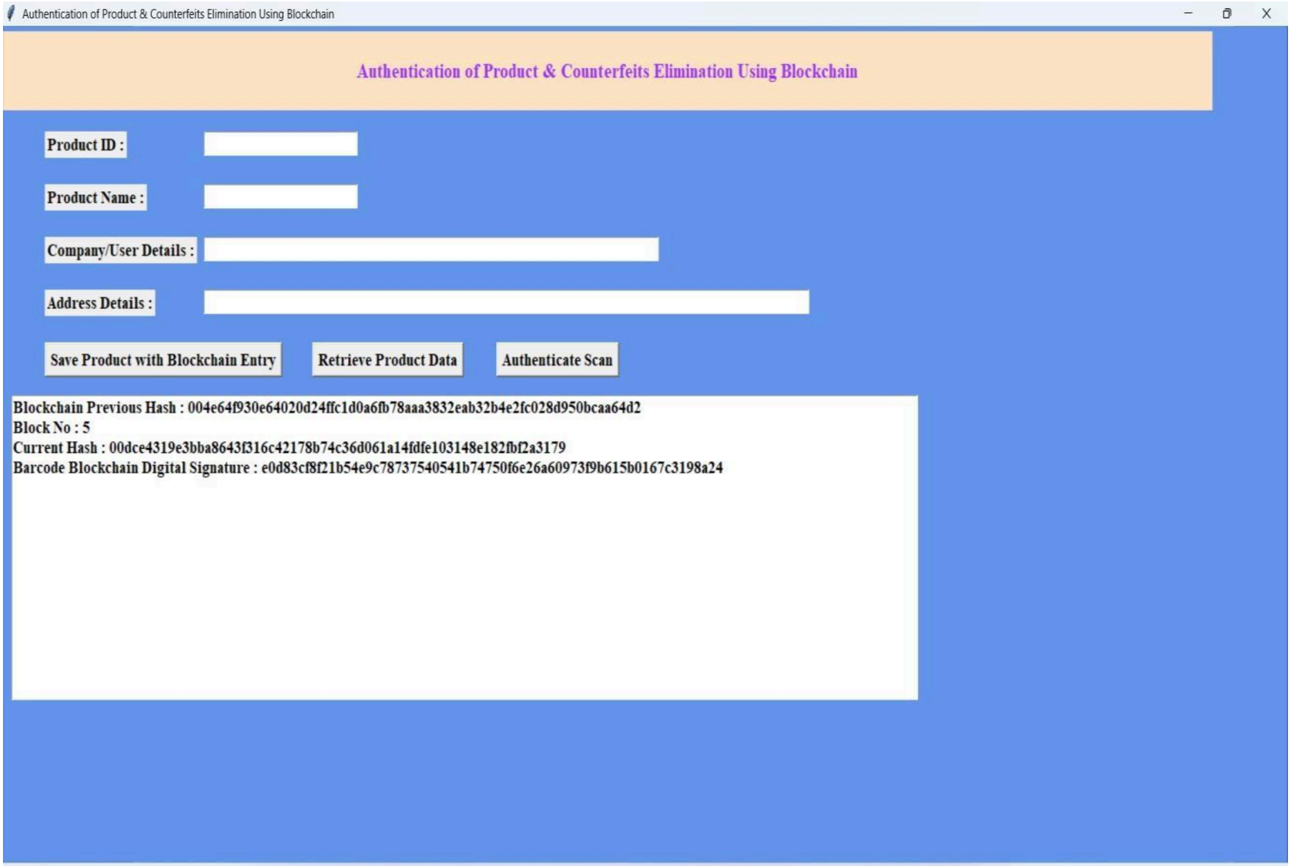
Product Name : iphone

Company/User Details : www.iphone.com

Address Details : mumbai

File name: 4.png

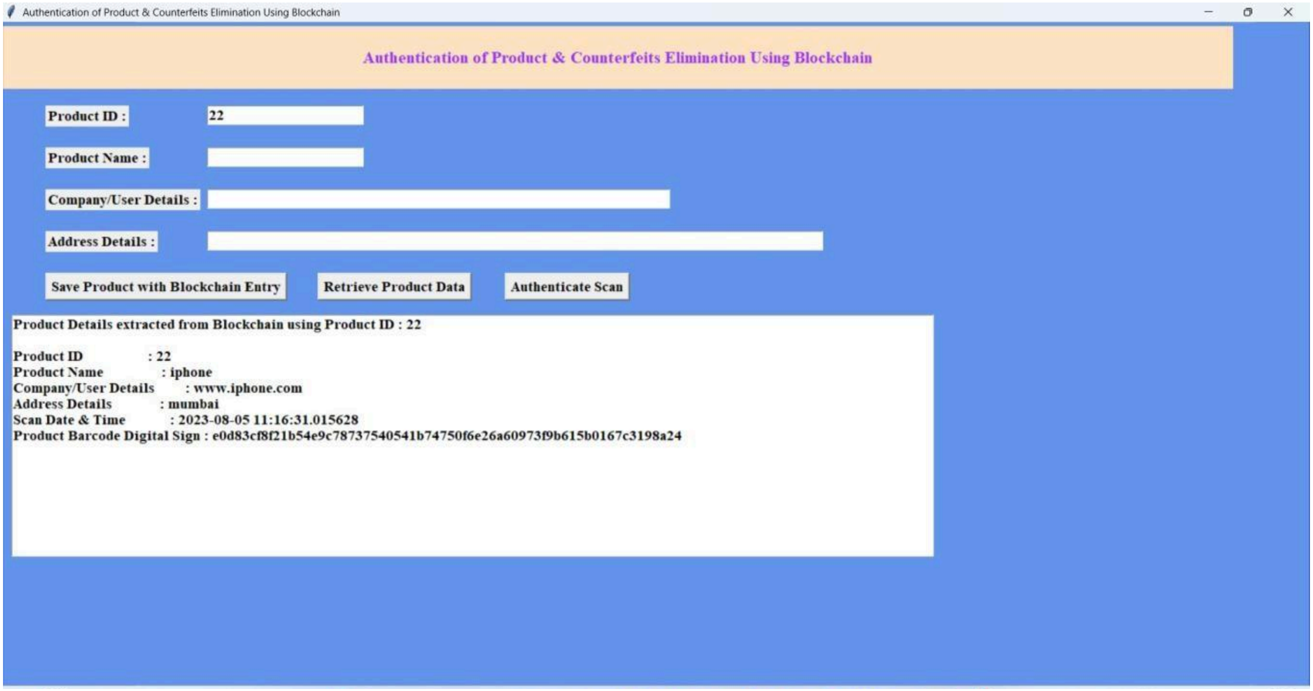
In above screen I entered product details and then selecting and uploading associated BARCODE image and then click on ‘Open’ button to get below result



Authentication of Product & Counterfeits Elimination Using Blockchain
Authentication of Product & Counterfeits Elimination Using Blockchain
Product ID :
Product Name :
Company/User Details :
Address Details :
Save Product with Blockchain Entry
Retrieve Product Data
Authenticate Scan

```
Blockchain Previous Hash : 004e64f930e64020d24ffc1d0a6fb78aaa3832eab32b4e2fc028d950bcaa64d2
Block No : 5
Current Hash : 00dce4319e3bba8643f316c42178b74c36d061a14fdfe103148e182fbf2a3179
Barcode Blockchain Digital Signature : e0d83cf8f21b54e9c78737540541b74750f6e26a60973f9b615b0167c3198a24
```

In the above screen Blockchain generated a new Block with id 22 and we can see Blockchain hash code of old and new transactions with uploaded bar code digital signature and all these details will be saved click on the “Retrieve Product Data” button to get details.

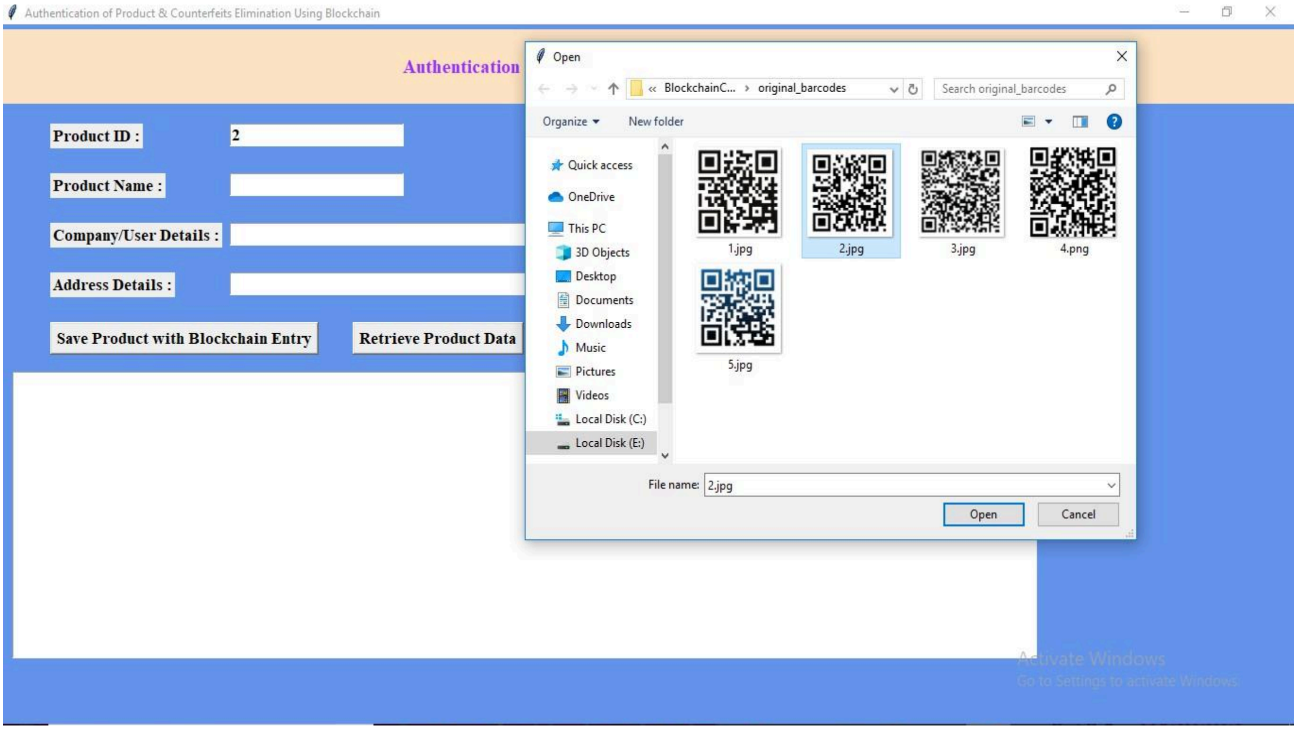

Authentication of Product & Counterfeits Elimination Using Blockchain
Authentication of Product & Counterfeits Elimination Using Blockchain
Save Product with Blockchain Entry
Retrieve Product Data
Authenticate Scan

Product ID : 22
Product Name :
Company/User Details :
Address Details :

```
Product Details extracted from Blockchain using Product ID : 22

Product ID            : 22
Product Name            : iphone
Company/User Details       : www.iphone.com
Address Details           : mumbai
Scan Date & Time          : 2023-08-05 11:16:31.015628
Product Barcode Digital Sign : e0d83cf8f21b54e9c78737540541b74750f6e26a60973f9b615b0167c3198a24
```

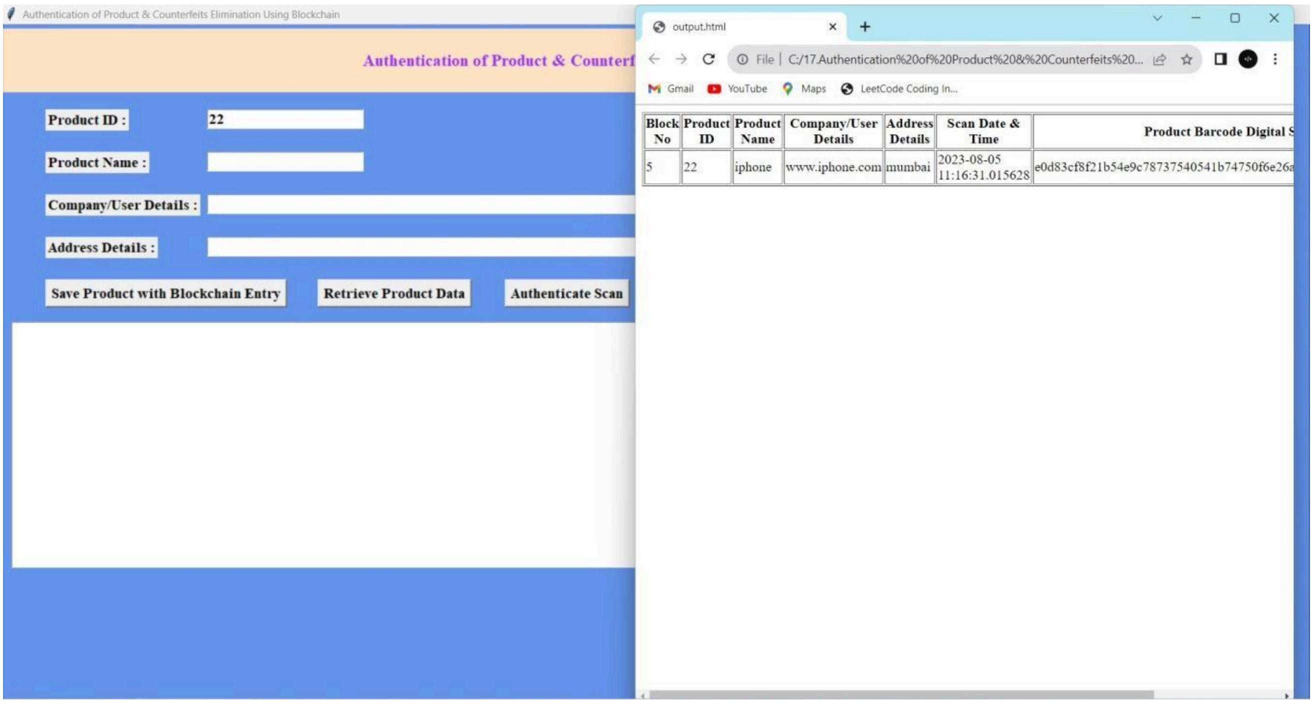
In the above screen I entered product id as 22 and then clicked on ‘Retrieve Product Data’ button to get the above details. Now click on ‘Authenticate Scan’ button to upload product Barcode and then Blockchain will match this uploaded Barcode signature with available stored signatures and if match is found then authentication will be successful else failed.



Product ID : 2
Product Name :
Company/User Details :
Address Details :

File name: 2.jpg

In above screen I am selecting and uploading ‘2.jpg’ file and then click on ‘Open’ button to get below result


Authentication of Product & Counterfeits Elimination Using Blockchain
Authentication of Product & Counterf
Save Product with Blockchain Entry
Retrieve Product Data
Authenticate Scan
output.html
Gmail
YouTube
Maps
LeetCode Coding In...

Product ID : 22

Product Name :

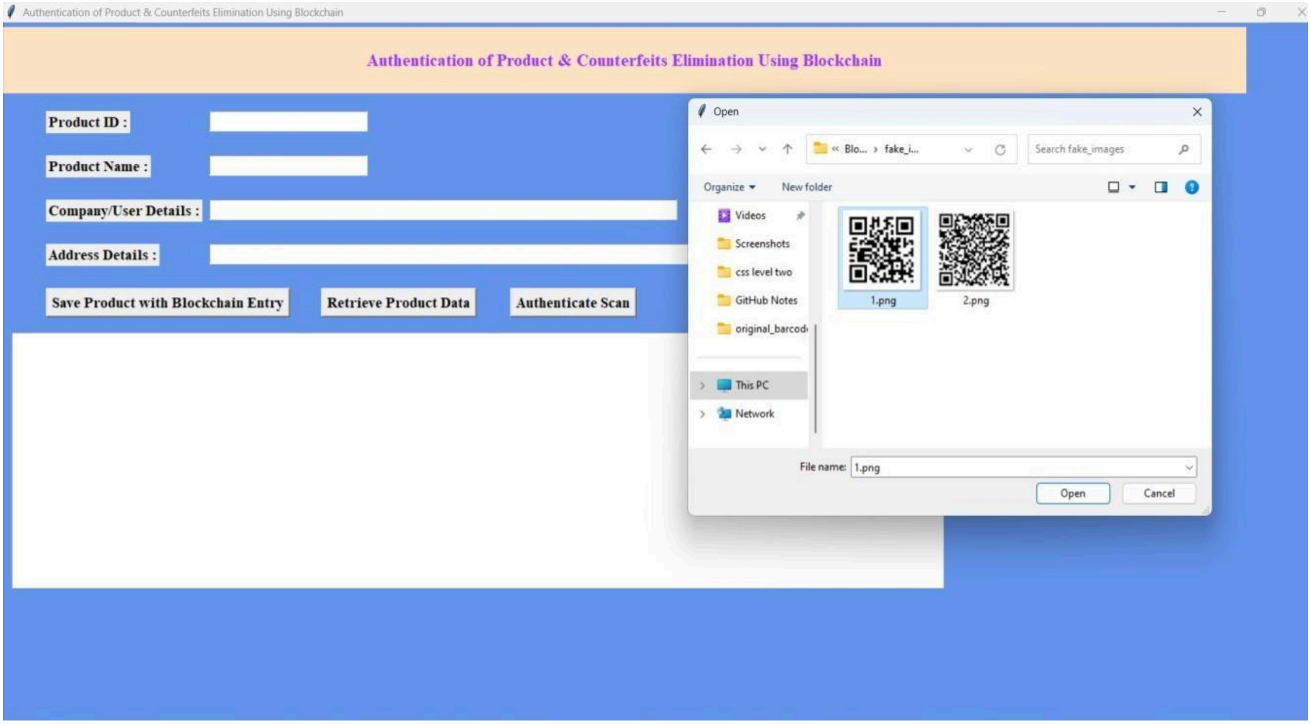
Company/User Details :

Address Details :

File | C:/17.Authentication%20of%20Product%20&%20Counterfeits%20...

Block No	Product ID	Product Name	Company/User Details	Address Details	Scan Date & Time	Product Barcode Digital S
5	22	iphone	www.iphone.com	mumbai	2023-08-05 11:16:31.015628	e0d83cf8f21b54e9c78737540541b74750f6e26a

In above screen in browser author can see all authentication details uploaded product bar code. Now check with fake barcode by uploading from ‘fake bar code’ folder



Product ID :

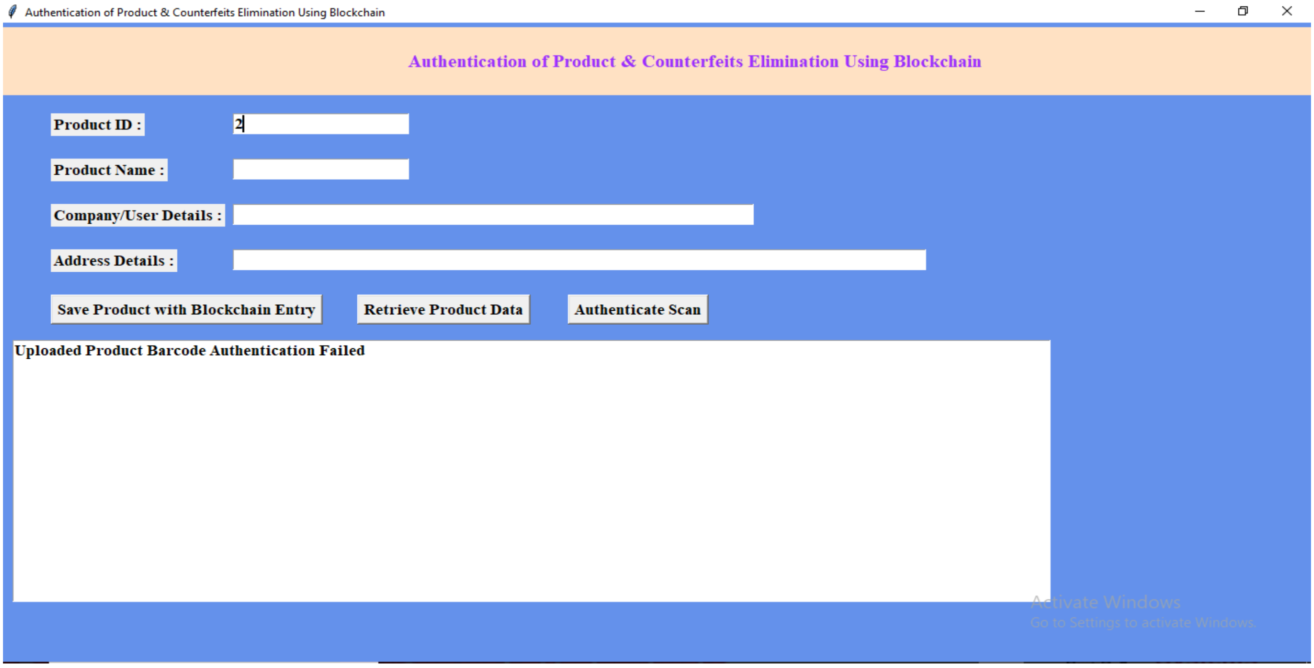
Product Name :

Company/User Details :

Address Details :

File name: 1.png

In above screen uploading barcode from fake folder and below is the result


Authentication of Product & Counterfeits Elimination Using Blockchain
Authentication of Product & Counterfeits Elimination Using Blockchain
Save Product with Blockchain Entry
Retrieve Product Data
Authenticate Scan
Activate Windows
Go to Settings to activate Windows.

Product ID : 2

Product Name :

Company/User Details :

Address Details :

Uploaded Product Barcode Authentication Failed

9.CONCLUSION

The system provides that the product's journey from manufacturing to customer can be recorded, and the customer is assured that the scans were faked. The Manufacturer is able to prove their product is authentic and is also able to track their product pathway. The setup is easy to implement and requires less operation cost. Manufacturers can also adopt RFID or NFC tokens instead of QR codes to further strengthen their system.

10. FUTURE ENHANCEMENT

Multiple techniques to reduce counterfeits were examined in this thesis. These improvements were considered, and their impact on minimizing counterfeits was assessed, in order to be less reliant on external variables. Due to time constraints and the fact that several other system changes were also required, it was not possible to implement all of the suggested changes. The finalization of these implementations for the proposed system, as well as the potential of running pilots, are among the next steps. The concept for reducing counterfeits in the humanitarian supply chain is currently being developed, as is the execution**.**

11. REFERENCES

[1] Satoshi Nakamoto, ―Bitcoin: A Peer-to-Peer Electronic Cash System‖, 2008

[2] Hyperledger, ―Hyperledger Blockchain Performance Metrics‖, V1.01, October 2018

[3] R.C. Merkle, "Protocols for public key cryptosystems," In Proc. 1980 Symposium on Security and Privacy, IEEE Computer Society, pages 122-133, April 1980.

[4] Armin Ronacher, ―Flask Docs‖, http://flask.pocoo.org/docs/‖

[5] G. Wood, ‗Ethereum: A secure decentralized generalized transaction ledger Tech. Rep., 2014.

[6] OECD (2016), Illicit Trade: Converging Criminal Networks, OECD Reviews of Risk Management Policies, OECD Publishing, Paris, https://doi.org/10.1787/9789264251847-en.

[7] M. Castro and B. Liskov, ‗Practical byzantine fault tolerance and proactive recovery."ACM Trans. Comput. Syst., vol. 20, no. 4, pp. 398–461, Nov. 2002.

[8] Clement, E. Wong, L. Alvisi, M. Dahlin, and M. Marchetti, ‗Making byzantine fault tolerant systems tolerate byzantine faults, in Proc. 6th USENIX Symp. Netw. Syst. Design Implement., 2009, pp. 153–168.

[9] Cachin, ‗Architecture of the hyperledger blockchain fabric " Tech. Rep., Jul. 2016..

[10] S. Underwood, ―Blockchain Beyond Bitcoin‖, in Communications of the ACM, vol. 59, no. 11, p. 15-17, 2016.

[11] Deloitte, Israel: A Hotspot for Blockchain Innovation, 2016. [Online]. Available: https://www2.deloitte.com/content/dam/Deloitte/il/Documents/financialservices/israel_a_hotspot_for _blockchain_innovation_ feb2016_1.1.pdf. [Accessed: 2.11.2016].

[12] G. Greenspan and M. Zehavi, Will Provenance Be the Blockchain Break Out Use Case in 2016, 7.1.2016. [Online]. Available: http://www.coindesk.com/ provenance-blockchain-tech-app/. [Accessed: 12.12.2016].

[13] Counterfeit medicines. QA counterfeit. World Health Organization (WHO) 2009. Available from: http://www.who.int/medicines/ services/counterfeit/QACounterfeit-october2009.pdf [last cited on 2010 Jun 12].

[14] An ICC initiative Business Action to Stop Counterfeiting and Piracy (BASCAP). Brand protection directory. The World Business Organization. Available from: http://www.iccwbo.org/bascap [last cited on 2010 Jun 10].

[15] L. Li, ―Technology designed to combat fakes in the global supply chain‖, in Business Horizons, vol. 56, no. 2, p. 167-177, 2013.

[16] White paper. Dhar R. Anti counterfeit packaging technologies. A strategic need for the Indian industry. Confederation of Indian Industry 2009:1-47. Available from: http://www.bilcare.com/pdf/CII_anti_counterfeit_pkg_technologies_report.pdf [last cited on 2010 Oct 29].

[17] Berman, ―Strategies to detect and reduce counterfeiting activity‖, in Business Horizons, vol. 51, no. 3, p. 191-199, 2008.

[18] K. D´egardin, Y. Roggo and P. Margot. ―Understanding and fighting the medicine counterfeit market‖, in Journal of Pharmaceutical and Biomedical Analysis, vol. 87, p. 167-175, 2013

[19] R. C. Merkle, ‗‗A digital signature based on a conventional encryption function,‘‘ in Proc. Conf. Theory Appl. Cryptogr. Techn., 1987, pp. 369–378